

Running Head: CROWDSOURCING IN PSYCHOLOGICAL RESEARCH

Methodological Concerns and Advanced Uses of Crowdsourcing in Psychological Research.

Jesse Chandler

Princeton University – Woodrow Wilson School of Public Affairs

Pam Mueller

Princeton University – Department of Psychology

Gabriele Paolacci

Ca' Foscari University of Venice - Department of Management

## Author Note

The authors wish to thank John Myles White for help developing and testing the API syntax. Jesse Chandler, Postdoctoral Research Associate, Woodrow Wilson School of Public Policy, Princeton University (jjchndl@princeton.edu), Pam Mueller, Graduate Student, Department of Psychology, Princeton University (pmueller@princeton.edu); Gabriele Paolacci; Graduate Student; Department of Management, Ca' Foscari University of Venice (paolacci@unive.it).

### Abstract

Crowdsourcing has become an increasingly popular research technology across the sciences. Behavioral scientists have noticed the ease with which workers can be recruited and paid using crowdsourcing services and have begun using them to recruit research participants. We review early research observing that crowdsourced participants are similar to other convenience samples of research participants. Extending this work, we note that researchers may overlook crucial differences between crowdsourced and traditionally recruited participants that provide unique opportunities and challenges. We describe how to implement advanced data collection features such as prescreening, longitudinal data collection, variable cash incentives, and the possibility of employing crowdsourcing to fulfill the role of traditional research assistants. We also identify and examine previously ignored drawbacks and suggest solutions to minimize their effects.

*Keywords:* crowdsourcing, internet research, longitudinal research

## Methodological Concerns and Advanced Uses of Crowdsourcing in Psychological Research

Crowdsourcing is an increasingly popular method of allocating and managing labor. Just as businesses have used the web to outsource labor, a number of websites have been developed to aid specific academic projects (Gaggioli & Riva, 2008). For example, reCaptcha verifies that website users are human by asking them to transcribe distorted images of words, and also helps digitize illegible portions of books (von Ahn et al., 2008). Galaxy Zoo ([www.galaxyzoo.org](http://www.galaxyzoo.org)) solicits “citizen-scientists” to view and classify astronomical images. For businesses with smaller or shorter-term projects, a number of companies (e.g., Crowd Cloud, Mechanical Turk) offer various ways to access large pools of workers to complete more modest tasks. These sites are primarily used by businesses seeking to outsource menial and repetitive tasks, but social scientists have increasingly become interested in crowdsourcing as a viable alternative to traditional methods of participant recruitment.

In this paper we provide an overview of research crowdsourcing, highlighting ways to capitalize on the strengths and avoid the limitations of Mechanical Turk, which is currently the dominant crowdsourcing application in the social sciences. On Mechanical Turk, participants (“workers”) browse Human Intelligence Tasks (“HITs”) by title, keyword, reward, availability, etc. and complete HITs of interest. They are paid by “requesters” upon successful completion of the accepted tasks. Requesters can discretionarily reject submissions or assign bonuses to workers, ensuring that work is of relatively high quality. Further, although workers are anonymous, their work is linked to a unique alphanumeric string, making it possible to seek out or avoid workers that meet specific criteria (e.g., nationality or worker reputation).

Although reviewers and editors are sometimes skeptical of data obtained from crowdsourcing websites, initial cross-sample investigations of crowdsourcing have demonstrated that data obtained from crowdsourcing websites is similar to data collected from more traditional subject pools, leading to the conclusion that crowdsourced workers are a useful source of high quality data. While the similarities between crowdsourcing and traditional methods of participant recruitment have allayed concerns about whether crowdsourcing is a viable method of data collection, other differences remain a cause for concern: participants remain members of crowdsourcing websites for longer than participants remain members of traditional subject pools and there is no limit to the number of studies in which people may participate, increasing the likelihood that they will complete many similar studies. Workers may also share information about available studies. Consequently, project selection or responses may be influenced by prior knowledge about the survey contents or identity of the experimenter.

In contrast to these challenges, crowdsourcing has several benefits that make it more appealing than traditional Internet sampling. Crowdsourcing allows for research methods such as unobtrusive prescreens, longitudinal data collection, and performance-based payment bonuses that were previously difficult with Internet convenience samples. These benefits have been ignored for several reasons: the tools that make them possible are not commonly known, documentation is poor and the logic underlying how to apply these tools within a research context is not always obvious.

To address these issues, this paper is divided into several sections. First, we provide a brief overview of the advantages of and concerns about crowdsourcing research participants as compared to traditional methods of recruiting convenience samples. Second, we provide a brief introduction to the Mechanical Turk interface for novice users. Third, we describe how to use the

advanced features in Mechanical Turk that make it possible to prescreen participants, collect longitudinal data, pay task-dependent bonuses, use workers to code qualitative data and (most importantly) prevent duplicate respondents.

To achieve these aims, it was necessary to collect survey data to address some unanswered questions about Mechanical Turk workers. These data were collected from a sample ( $N = 300$ ) typical of those recruited for empirical research: participation was restricted to Americans who have successfully completed  $> 95\%$  of all previously submitted tasks. This sample was supplemented with an additional 20 high-productivity workers who were known to be among the most productive 1% of workers in a sample of previously completed research HITs. These workers were included to ensure a more accurate description of this sample. These high productivity workers were recruited passively by posting the survey with visibility restricted (using methods described in this paper) so that only they could see it. Unless stated otherwise, all claims reported in this paper about the Mechanical Turk population in general that are based on these data use only the original sample of 300 workers, while all findings that compare workers by productivity level include the supplementary sample of especially productive workers.

### **Advantages of Using Mechanical Turk to Recruit Research Participants**

Mechanical Turk has quickly attracted a great number of social scientists that use experimental and survey methods. It offers a number of strengths that can make it a useful supplement to traditional subject pools.

#### **Mechanical Turk is Fast, Cheap and Convenient.**

Perhaps the most appealing features of Mechanical Turk are the speed and low cost of data collection. For example, the survey we posted collected 300 respondents over three days at an average compensation rate of \$1.49/hr. Most workers accept compensation of less than \$2 per

hour (Chilton, Horton, Miller, & Azenkot, 2009), with the average HIT paying about \$5 per hour (Ipeirotis, 2010). Wages will likely increase as researchers and businesses compete for high quality workers.

Speed of data collection is affected by the pay rate (Mason & Watts, 2009; Buhrmeister et al., 2011); however, since the median reservation wage is only \$1.38/hour (Horton & Chilton, 2010), experimenters can typically obtain a critical number of participants very quickly and at a low cost. Further, payments are easily distributed to workers en masse and there is no need to maintain payment receipts or other income tax records (provided that you do not pay an individual worker more than \$600 over a single year).

### **Mechanical Turk Workers can Produce High Quality Data.**

Despite the low wage, the quality of data that workers produce is relatively high. If anything, workers are more attentive than traditional subject pools (Paolacci, Chandler, & Ipeirotis, 2010). Paralleling earlier research on participants recruited from the Internet (Gosling, 2004; Krantz & Dalal, 2000), several studies have found that samples recruited from Mechanical Turk exhibit similar behaviors and effect sizes as traditional subject pools within a variety of experimental paradigms (Horton, Rand, & Zeckhauser, in press; Paolacci et al., 2010; Suri & Watts, 2010; Sprouse, 2011). Data quality seems to be virtually independent from pay rates (Buhrmester, Kwang, & Gosling, 2011; Marge, Banerjee & Rudnicky, 2010; Mason & Watts, 2009). Further, attempts to verify participants' response accuracy through comparing location of residence to logged IP addresses and consistency in self-reported demographics have revealed that workers are generally honest about these details (Rand, 2011; see also Ipeirotis, 2011).

This is not to suggest that crowdsourced participants are without their limitations. Although experimenters can exclude poor-quality workers (as measured by low approval rates

for submitted HITs), even high-quality workers are motivated to complete HITs efficiently and may thus miss important instructions or carelessly respond to subjective measures. As a further complication, the environments in which workers complete HITs are heterogeneous and there may be distractions that may lead them to miss experimental manipulations. Our survey revealed that although most workers completed the HIT from home (86%) and alone (73%), they are often doing other activities simultaneously: 18% of them were also watching TV, 14% of them were listening to music and 6% of them were instant messaging with at least one other person. Thus, although psychological findings are generally replicable among crowdsourced workers, it should not be assumed that this is equally true for all experimental paradigms, especially those that rely on subtle manipulations or require undivided attention.

Working within these constraints, there are a few easy ways to monitor data quality. Passive techniques can be used to identify problematic workers. Catch trials can be included to identify workers who agree with unlikely or even impossible statements (Downs, Holbrook, Sheng, & Cranor, 2010; Paolacci et al., 2010). Likewise, scale responses can be examined for excessive use of the same response category (Johnson, 2005).

Poor quality responses are often as much a result of the instrument as the respondent and can be improved through careful design (for a review see Couper, 2008). Participants are also often unaware of how carefully researchers expect them to attend to questions; drawing attention to this can increase data quality (see Oppenheimer, Meyvis, & Davidenko, 2009). However, there are a number of specific techniques that capitalize on the incentive structure in Mechanical Turk to ensure that workers pay attention. Within Mechanical Turk, money is only earned for correctly completed tasks. As a result, factual questions may focus workers' attention and lead them to answer with greater care. Supporting this, one study recruited workers and experts to

evaluate the quality of Wikipedia pages. The ratings were uncorrelated, except when workers were also required to include answers to objectively verifiable questions (Kittur, Chi, & Suh, 2008).

Attention can also be directly incentivized in Mechanical Turk by dividing payment between money for simply completing a task and a bonus payment for correctly answering factual questions about the task they completed. In a pair of nearly identical studies (conducted by the second author of the current paper) participants were paid either an initial sum for participating or a smaller initial sum with the remainder paid as a bonus for successfully factual details about the experimental manipulation. In both cases, the total payment for a participant who completed the experiment and successfully answered the manipulation check was equal, but the success of the manipulation check was significantly higher in the bonus contingent experiment (98.2%) relative to the lump sum experiment (87.0%),  $\chi^2(1, N = 494) = 23.03, p < .001$ .

### **Mechanical Turk Workers are Diverse.**

The available workforce is composed of hundreds of thousands of individuals in a wide distribution of ages, ethnicities, levels of education and income (Paolacci et al., 2010). Most workers are currently living in the United States or India and are generally young. Reflecting the more general demographics of younger cohorts, they also tend to be more educated but earn less than the average person in their country of origin (Ipeirotis, 2010). Despite these overall patterns, there is considerable heterogeneity. Workers range in age from 18 to more than 70 and the income distribution parallels that of their country of origin. Likewise, political affiliation and support for various public policies often parallel the distribution observed in high quality panel

data, sometimes more closely than other convenience samples and always more so than student samples (Berinsky, Huber, & Lenz, 2010).

Our survey of American workers corroborated these findings. Our sample was mostly female (62%), young ( $M_{age} = 34.31$ ,  $SD = 12.3$ ) and, for the most part, educated: 18.7% had a postgraduate degree, 32.7% had a completed college degree, 13.0% had a high school diploma or less. Further, we found that the population was disproportionately likely to identify as white (80%) and Asian (8%), relative to the U.S. population as a whole (75% and 3.6% respectively). Although underrepresented, a significant number of participants identified as Black (8.0% versus 12.3% of the population as a whole); 5.4% also identified as of Hispanic ethnicity (versus 12.5% of the population as a whole).

In short, although not a perfectly representative population, workers on Mechanical Turk are clearly *more* representative than many traditional subject pools on potentially important demographic characteristics. More importantly, the workforce is orders of magnitude larger than the typical college subject pool and requesters can screen workers before hiring them, allowing experimenters full flexibility in the recruitment process. Thus, it is potentially easy to obtain nearly any kind of sample, including those that are not Western, educated, industrialized, rich, and Democratic (“WEIRD”; Heinrich, Heine, & Norenzayan, 2010).

### **Mechanical Turk Reduces Important Threats to Experimental Validity.**

Mechanical Turk has a number of features that allow researchers to avoid certain threats to experimental validity that are difficult to avoid in other Internet recruiting techniques (Horton et al., in press). Relative to Internet recruiting strategies, selection bias and attrition are less of a concern in samples recruited from Mechanical Turk. The factors that lead people to join the Mechanical Turk workforce are varied but frequently center on earning money or “killing time”

(Paolacci et al., 2010). In contrast, traditional convenience sampling procedures tend to rely on initial interest in the topic of study to attract participants.

More importantly, Mechanical Turk allows researchers to reduce this problem by converting unmeasurable selection bias into measurable attrition. In our survey, workers indicated that pay, task length and difficulty were the most relevant attributes when selecting a HIT (corroborating earlier findings about workers motives for participating). This is the only information they need to be presented with when initially accepting a HIT, providing that it is clearly indicated that workers will receive additional information on a subsequent consent page. Additional relevant details can then be provided on a consent form page embedded within the survey itself (workers are not penalized for “returning” a HIT when they see its actual content).

This approach reduces potential sources of selection bias to i.) Mechanical Turk membership, ii.) interest in research studies in general, and iii.) willingness to accept a particular level of compensation (which has not yet shown to be relevant: Buhrmester et al., 2011; Marge et al., 2010; Mason & Watts, 2009). Many survey websites can be configured to record participants who do not complete a response. Thus, if workers drop out and return the HIT selection bias has become attrition. Finally, in our experience, the attrition rate for Mechanical Turk HITs is extremely low (e.g., 4% in the survey reported in this paper).

Another benefit of crowdsourcing participants is that demand effects are reduced. As with other Internet-based experiments, experimenter demand effects are completely eliminated (see Gosling, 2004 for a detailed discussion of pros and cons of web-based studies).

Additionally, Mechanical Turk can attenuate demand effects built in to survey materials by splitting variables across prescreening measures or different HITs, obscuring the relationship between different components of the survey or experiment. By doing this researchers can present

measures that look unrelated to participants (for a detailed treatment of demand characteristics in research materials see Rosenthal & Rosnow, 2009).

Finally, and perhaps most importantly, Mechanical Turk makes it possible to prevent the same participant from completing a survey more than once. Traditional web-based recruiting typically relies on identifying duplicate participants by filtering duplicate IP addresses, but this excludes participants from the same household and misses duplicate respondents who have different IP addresses at different times (e.g., their ISP assigns dynamic IP addresses or they use a public WiFi connection). In contrast, Mechanical Turk can prevent workers from completing a HIT twice based on their WorkerID, which is unique to individual workers and linked to a credit card. Although this is not an issue with a single, small experiment, as we discuss below, this may be a particularly useful feature when conducting programmatic research.

### **Mechanical Turk makes it Easy to go Beyond the Traditional Survey or Experiment.**

Mechanical Turk has a number of features that make it possible to run less conventional experiments with ease. For example, the large number of workers makes it easy to recruit participants into experiments in group dynamics (e.g., Suri & Watts, 2010). Likewise, the bonus payment system makes it easy to implement designs with real monetary incentives. Additionally, the unique identification number assigned to each worker makes it possible to prescreen participants and maintain a pool of workers who fulfill specific criteria or to recontact individual workers participating in longitudinal designs. We discuss how to use these features in the “Advanced Data Collection Techniques” section below.

### **Some Lingering Concerns about Mechanical Turk**

#### **Participation in Conceptually Related Experiments**

Although Mechanical Turk by default prevents the same worker from completing a single HIT multiple times, some care is required if one wants to prevent workers from being included in conceptually or methodologically similar studies. While undergraduate subject pools are continually replenished with naïve participants, Mechanical Turk is not. Duplicate responders are of concern not only because they violate assumptions of statistical independence but also because prior knowledge about the purpose of an experiment, familiarity with an experimental manipulation, or reason to suspect deception influence participant responses (Brock & Beckker, 1966; Edlund et al., 2009; Glinski, Glinski, & Slatin, 1970; Silverman, Schulman, & Wiesenthal, 1970), although precisely how they differ depends on participants' construal of the task and perceived relationship with the experimenter (Rosnow & Aiken, 1973; Sawyer, 1975).

Pooling the data from the authors and several collaborators resulted in a sample of 16 408 HITs distributed across 132 batches. Within this sample we found substantial reason to be concerned about duplicate responses. These HITs had been completed by a total of 7498 workers. The average worker completed 2.24 HITs ( $SD = 3.19$ ), but a very small minority of workers were responsible for submitting most of the HITs. The most prolific 1% of workers from this sample was responsible for completing 11% of the submitted HITs (the highly productive workers we described earlier were recruited from this group), and the rest of most prolific 10% were responsible for completing 41% of the submitted HITs (see Figure 1). A similar distribution was observed by Berinsky and colleagues (2010) across six experiments conducted by the authors over a period of several months—in their sample 24% of the workers participated in two or more experiments and 1% completed five or more of these experiments.

There is nothing particularly special about the demographics of more productive workers although they tend to be somewhat older, more educated and more likely to be White than the

general worker population. They also tend to be somewhat more focused than the general pool of workers: when completing the demographic survey they were more likely to be alone and less likely to be engaged in other tasks like listening to music, watching TV, or chatting online (see Table 1), suggesting that they may be particularly suitable for experiments that are sensitive to participant attention (e.g., those that rely on reaction time).

The most productive workers in the original sample of HITs who also responded to our survey did not report spending more time using Mechanical Turk than less productive workers during the previous week. However, they were unusually likely to complete the survey. The most prolific 1% of workers comprised 4% of the sample and remainder of the most prolific 10% comprised a further 20% of the sample, significantly more than expected by chance,  $\chi^2(1, N = 300) = 290.69, p < .001$ . These figures corroborate earlier worker self-reports that suggested that Mechanical Turk has a small population of very active workers (Ipeirotis, 2010). Further, it is important to note that this survey was posted using a new RequesterID and was only available for 3 days, so reputation effects do not explain the overrepresentation of productive workers. One possible explanation is that this survey had the keywords “survey”, “research”, and “experiment” associated with it (as recommended to increase response rates; Chilton et al., 2009) and that the productive workers in our sample were those who seek out social science research HITs.

For established researchers, the problem of repeat workers across several experiments probably increases: more than half of all workers (55%) reported having a list of favorite requesters that they monitor for available HITs, and 58% of those who followed favorite requesters (about a third of the entire sample) reported that this list included academic researchers. The most productive workers are also especially likely to read blogs about Mechanical Turk and follow specific requesters (see Table 1). Thus, researchers should be aware

that they may have a loyal following who have completed their experiments in the past, read debriefing materials, and deliberately seek out their experiments. We discuss how to avoid duplicate responses in the “Advanced Data Collection Techniques” section below.

### **The Mechanical Turk Subject Pool is Contaminated by Other Research Studies**

In addition to the possibility that a worker completed previous experiments for a researcher, there is the risk that they have completed similar experiments for *other* researchers. This is especially true for researchers working with memorable, commonly used experimental paradigms. In our survey, a substantial proportion of workers reported participating in some of the more common and easily describable experimental paradigms, such as the prisoner’s dilemma or the “trolley problems” commonly used to illustrate moral reasoning. As would be expected, the most productive workers are also the ones who are most likely to have participated in common experimental paradigms (see Table 2). This stands in contrast to earlier claims that Mechanical Turk offered a subject pool of naïve participants (Chilton et al., in press). Based on these findings, it seems that, without taking steps to filter or measure non-naïve participants, Mechanical Turk may not be appropriate for commonly used paradigms. For less used paradigms, researchers can minimize this problem by sharing lists of who has completed specific experimental manipulations. We explain how to do this in the “Advanced Data Collection Techniques” section.

### **Workers may Communicate with Each Other about Experiments**

A third potential problem is worker crosstalk. Mechanical Turk workers maintain online forums where they share information and opinions about Mechanical Turk, which could potentially lead to foreknowledge in experimental participants. Empirical research on college undergraduate populations has demonstrated that participants do share information with each

other, at least when sufficiently motivated (e.g., when incentives are offered for a correct response; Edlund, et al., 2009). However, our survey revealed that crosstalk is less of a problem on Mechanical Turk.

Only 26% of participants reported knowing someone else who used Mechanical Turk personally, and only 28% reported reading forums and blogs about Mechanical Turk. Further, when asked to rank the frequency with which they discuss or read about various aspects of Mechanical Turk, the actual purpose or contents of Mechanical Turk HITs is far less important than pragmatic considerations such as the amount requesters pay (see Table 3). Only half of the respondents who actually read blogs (about 13% of the overall population) reported ever seeing a discussion about the contents of a social science research study online. However, workers do on occasion discuss experiments on discussion boards (accompanied by links to the HIT) and they have been known to inadvertently share details that are a part of the experimental manipulation. Thus, researchers should probably ask workers how they found the HIT at the end of their survey and monitor discussion boards that refer a lot of respondents.

In sum, researchers can use Mechanical Turk to quickly and cheaply recruit diverse research participants. Below we briefly review how to construct a basic HIT in Mechanical Turk for those who are unfamiliar with the service. Following this, we address some of the concerns we raised about data collected from Mechanical Turk empirically.

### **Using Mechanical Turk – A Brief Overview**

In the following section we provide a brief introduction to Mechanical Turk, explaining how to create and manage experimental HITs.

#### **Creating and Publishing a HIT**

It is relatively easy to create a batch of HITs on Mechanical Turk. First, you need an account. To create one, go to the Mechanical Turk website ([mturk.com](http://mturk.com)). Select the “Get Started” button on the right side of the page. On the next page select the “Create an Account” button. You can use a preexisting Amazon account or create a new account specifically for Mechanical Turk. A dedicated account may be preferable because when you contact workers through Mechanical Turk the email address associated with your account will become visible to them. The account name will be visible to workers before they select the HIT and should be chosen accordingly. You will also need to provide a U.S. billing address to fund the account (with the corresponding credit card, debit card, Amazon Payments account or U.S. bank account). Access to Mechanical Turk from other countries is possible through companies that have obtained local rights to do so (e.g., Crowdfunder in Canada).

Once you have created an account you are ready to post a batch of HITs by clicking on the “Design” tab in the Requester homepage. HITs can either be modified from existing templates or created from scratch. In both cases, there are three steps that the requester must complete.

1. “Enter Properties”: In the “Describe your HIT” subsection you can specify the title and a brief description of the HIT, and provide keywords that will help workers search for your HIT (e.g., survey, research, study). For academic research, including the keyword “survey” is especially important for maximizing the number of workers who find the HIT (Chilton et al., 2009).

The “Working on your HIT” subsection allows you to specify how long the HIT will be available (if not fulfilled more quickly) and how long a worker may take to complete a HIT. Most importantly, in this subsection you can define the characteristics of the workforce that will

complete or even view your HIT, including the minimum quality of workers (“HIT approval rate (%)” qualification) and their provenience (“Location” qualification), allowing requesters to constrain their workers to a single country or compare people in different countries using two different country-constraints, e.g., Eriksson & Simpson, 2010). Note that a requester can create custom qualifications that can prescreen workers on whatever dimension the requester needs (see “Advanced Data Collection Techniques” below).

Finally, the “Paying Workers” section contains parameters relevant to the number of workers needed and the amount of compensation that each will be paid.

2. “Design Layout”: Here you can craft what the HIT actually looks like to workers. There are two approaches to setting the HIT’s contents. First, you can construct a survey entirely within Mechanical Turk using HTML. We do not recommend this method because it is limited to a single page HTML form, making random assignment difficult and preventing the use of advanced survey features. An alternative approach is to provide a link to an external survey website in the HIT. A unique verification number can be provided (or generated by the user) at the end of the survey and submitted by the worker on the HIT page to complete the HIT. If you only provide a link, be sure that the HTML code directs workers’ browsers to open in a new window. If a worker clicks on an url and leaves the page the HIT is on, they sometimes have trouble getting back to that page to complete the HIT and get paid. The external website can also be integrated within the HIT window (for a description of how to do this see Mason & Watts, 2010).

3. “Preview & Finish”: After determining the properties of your HIT, click on the “Finish” button in the lower right hand corner. This will add the HIT to a list of HIT templates ready to be made available to workers. Click on the “Publish” tab at the top of the screen, select

the HIT you wish to publish, review the contents and properties of the HIT, and finally select the “Publish HITs” button on the bottom right corner of the page.

### **Managing Your HITS and Workforce**

Under the “Manage” tab you can view and modify previously published HITs by selecting the “Batches” option. Each batch is displayed in a separate box that includes information about how quickly the average worker completes the HIT and the effective hourly wage. Note that the wage is estimated by dividing the payment by the average time between when a HIT is accepted and submitted. This timing should not be relied upon as an empirical measure: the estimated time will generally be longer than the actual completion time, but it can be dramatically shorter if workers complete the survey or experiment before accepting the HIT from Mechanical Turk.

Selecting the “Results” button within a given HIT will let you view a summary of the work submitted so far, and allow you to download, modify and upload spreadsheet files to approve/reject submissions. The “Workers” option under the Manage tab allows requesters to assign/revoke qualifications to workers (see “Creating a Qualification” below) and block/unblock them by downloading, modifying and uploading the spreadsheet file containing the list of workers who have previously worked for the requester. The block function should not be used as a shortcut to prevent workers from completing future studies, as it may result in them being banned from Mechanical Turk: Blocks should be reserved for individuals who are doing consistently poor work.

A requester can also perform these actions on individual workers without downloading and modifying spreadsheet files by clicking on the “Manage HITs individually” link on the upper right hand corner of the “Manage Workers” page.

Note that Amazon reserves the right to delete information stored on the Mechanical Turk website (HITs, HIT templates, and qualifications) after 120 days if it has not been accessed. Therefore, we recommend downloading the .csv files for each HIT after data collection is complete even if the information requested in the HIT is only a worker-provided ID code so that you have a record of all the workers who have participated in your HITs, as well as information about speed of data collection, etc. This information will be especially relevant if you decide to use some of the advanced data collection techniques described below.

### **Advanced Data Collection Techniques**

In this section we describe advanced functions of Mechanical Turk that can be used to improve data quality or conduct studies involving several separate components (e.g., prescreening and longitudinal studies). These methods have advantages over current common practice. Typically, studies that use prescreening and longitudinal data collection make the HIT available to everyone with instructions for people who are not desired to ignore the HIT. This requires researchers to police workers to ensure that they follow these instructions and (in the case of prescreening) can lead workers to lie in order to meet study inclusion criteria or alert them to the relevance of the prescreening measure to the experiment or survey. Likewise, many studies try to avoid duplicate respondents by asking workers to avoid a HIT if they have completed similar HITs from a particular lab (again creating policing issues).

Concerns about worker honesty, the suspicion such instructions may induce, and unnecessary administrative work can all be avoided through the use of qualifications to prescreen participants. Qualifications can also be used to distribute related measures across different HITs (reducing demand effects by concealing the relationship between different variables), conduct longitudinal research, and screen out workers who have completed similar experiments for you

or even for other researchers. Finally, we explain how to grant bonuses (allowing researchers to incentivize workers) and use workers to code qualitative data.

### **Creating Qualifications**

Qualifications are values assigned to workers by requesters. HITs can be restricted to workers with specific values assigned to a qualification, allowing requesters to control who completes a specific HIT. There are three ways to create and assign qualifications. The most common is the web interface, which although relatively simple lacks much of the functionality of the other two. The web interface is best for simple qualifications that are manually granted to small numbers of workers who have worked for the requester before. The second interface, the Command Line Tools (CLT), is a downloadable program that balances functionality and simplicity. The third option is the Amazon Web Services API, which requires the use of an outside interface. Although we will not discuss this approach further, should those without significant programming experience want to explore the API further, we have found that boto (<http://code.google.com/p/boto/>) provides the most straightforward access to Mechanical Turk.

**The Web Interface.** Basic qualifications can be created using Mechanical Turk's web interface without any coding knowledge. To do so, select "Manage" > "Qualification Types", then "Create New Qualification Type." Name and describe the qualification to distinguish it from other qualifications you might create. Note that workers can see the name of the qualification. Once created, the qualification can then be assigned by selecting "Manage" > "Workers" and downloading a .csv file containing all workers who have completed previous HITs for the requester (if the qualification is not visible, click on "Customize View" and add it).

The .csv file contains columns labeled *WorkerID*, *Link to Individual Worker Page*, *Number of HITs approved or rejected (for you)*, *Number of HITs approved (for you)*, *Your*

*approval rate*, *CURRENT Blocked Status*, and *UPDATE Blocked Status*. If a requester were to create a qualification named “Gender,” two columns labeled *CURRENT-Gender* and *UPDATE-Gender* would also be visible. *UPDATE-Gender* is where the requester would assign new values to your workers. For instance, all women could be assigned the value “1.” To change the value associated with a worker for a particular qualification, place this value in the update column (or type “revoke” to assign no value), then save the file and upload it to Mechanical Turk using the “Upload CSV” button on the same page. Qualifications can be assigned to individual workers under “Manage” > “Worker”s and then selecting individual WorkerID numbers.

Once this qualification is created, subsequent HITs could be restricted to women only by requiring that all eligible workers have a value of 1 assigned for the Gender qualification. Note that using this method the population of women who can see the HIT are restricted to those who have worked for the Requester.

**The Command Line Tools.** The Command Line Tools (CLT) offers additional flexibility. In particular, the CLT allows you to create qualification tests that can be assigned automatically by Mechanical Turk or manually to any worker, regardless of whether they have completed HITs for the requester in the past.

***Installing the Command Line Tools.*** The CLT can be downloaded from Amazon (<http://aws.amazon.com/developertools/694>) by following the installation instructions in the software documentation

([http://mturk.s3.amazonaws.com/CLT\\_Tutorial/UserGuide.html#installation](http://mturk.s3.amazonaws.com/CLT_Tutorial/UserGuide.html#installation)).

On a Mac, make sure the Java and Command Line Tools settings are as indicated in the installation guide. Then, within the terminal, navigate to the “bin” folder located in the same directory as the CLT (e.g., by typing `cd /Applications/aws-mturk-clt-1.3.0/bin/`). This folder

contains files that correspond with many commands you might wish to use (i.e., *grantBonus*, *createQualificationType*, etc.). You can view the files in the terminal by typing the *ls* command in a Mac Finder window.

In subsequent sections, all CLT commands will be presented for the Mac/Unix framework for clarity. Mac/Unix commands take the form: *./grantBonus.sh*, while the identical commands for Windows would merely include the text: *grantBonus*.

***Creating a prescreening qualification that is automatically scored.*** The CLT allows a requester to create a prescreening questionnaire that can be automatically scored. Workers are assigned a value based on their responses and importantly, cannot retake the questionnaire to override their initial responses (to see if this makes better surveys visible). Thus, for example, a qualification could be created and awarded to everyone who reports that they are a parent.

A qualification requires three text files to be created and saved in the bin folder. These will specify the question that will be asked to participants, the values that will be assigned for specific answers to the question and the properties of the qualification. Assume that a researcher wants to create a qualification to identify parents. The question could be placed in an XML file that we will call “Parent.question.” The values associated with different answers that Mechanical Turk will reference will be saved in an XML file called “Parent.answer.” The properties of the qualification will be saved in a text file called “Parent.properties.” There are many potential question types (multiple choice, text entry, etc.) one can use for qualifications, so we will not include them all here. Samples of question and answer files that can be used as templates can be found in the “samples” folder installed with the CLT. Additional resources can be found in the Amazon Mechanical Turk Developer Guide at:

<http://docs.amazonwebservices.com/AWSMechTurk/2007-06->

[21/AWSMechanicalTurkRequester/ApiReference\\_QuestionFormDataStructureArticle.html](http://docs.aws.amazon.com/AWSMechTurk/2007-06-21/AWSMechanicalTurkRequester/ApiReference_QuestionFormDataStructureArticle.html)

(Questions) and [http://docs.aws.amazon.com/AWSMechTurk/2007-06-](http://docs.aws.amazon.com/AWSMechTurk/2007-06-21/AWSMechanicalTurkRequester/ApiReference_AnswerKeyDataStructureArticle.html)

[21/AWSMechanicalTurkRequester/ApiReference\\_AnswerKeyDataStructureArticle.html](http://docs.aws.amazon.com/AWSMechTurk/2007-06-21/AWSMechanicalTurkRequester/ApiReference_AnswerKeyDataStructureArticle.html)

(Answers)

The properties file specifies additional features of the qualification. Note that throughout the examples that follow commands listed within parentheses are optional and the list of properties discussed here is not exhaustive. Parentheses should not be included in the text files that will be used by the CLT.

The properties file for our example of the parent qualification would contain the following syntax:

*name=Parent*

*description=workers with children*

*(keywords=parent, children)* – words workers might search to find this qualification

*(autogrant=false)* – if true, awards the qualification to anyone who requests it without having to answer questions

*(autograntvalue=5)* – the qualification value autogrant to workers

*(sendnotification=true)* – if true, alerts the worker that they have successfully been awarded the qualification

Once these files are created, the qualification is uploaded using the command `/createQualificationType.sh -properties Parent.properties (-question Parent.question -answer Parent.answer -noretry)`; the last argument prevents a worker from taking a qualification test more than once. This command will output a file named “Parent.properties.success.”

Within this file, and also printed onscreen, is the qualification type ID number, signifying that the qualification has been created and is ready for use. A HIT can then be created that requires a specific value for this qualification and workers will be informed that they need to complete it prior to accepting this HIT.

Workers who complete the qualification successfully, or who are assigned the qualification by a requester will automatically gain access to the HIT. The QualificationID number is all that is needed to grant this qualification to workers. Thus, other researchers interested in recruiting parents could specify that their HITs require the same qualification.

***Assigning a Qualification to Workers Based on Survey Response.*** As with the web interface, the CLT allows requesters to limit the availability of a HIT to a subset of workers for follow-up or longitudinal studies.

Continuing with our previous example, a requester may wish to follow-up with parents who reported particularly high levels of parenting stress in an initial survey. Since these data were not collected in a qualification, qualifying workers will need to be identified based on their survey responses and assigned a new qualification, “stressedparents,” by the requester. To do so, the requester needs the WorkerID numbers [*workerid*], which can be found in the .csv file associated with the HIT. The syntax for assigning this qualification to an individual worker is then:

```
./assignQualification.sh -qualtypeid stressedparents -workerid [workerid] (-score 2 -  
donotnotify)
```

If a score is not included in the command, workers will be assigned the default value 1.

The *-donotnotify* option allows qualifications to be assigned to a worker without their awareness. This is useful for researchers who want to disguise the relationship between a

qualification and a HIT to prevent workers from knowing why they were eligible to participate in a particular HIT.

The qualification can be assigned to eligible parents en masse by creating a tab-delimited file [*workers.txt*] containing a column for the WorkerIDs [*workerid*] and a column for the scores you wish to assign to them [*score*]. The syntax is then:

```
./assignQualification.sh -input participatedparents.properties.success -scorefile  
workers.txt.
```

After assigning these qualifications, two strategies can be used to recruit workers. The HIT can be made visible only to workers with the correct qualification value and the researcher can wait for them to see and complete the HIT. This method is relatively slow and has a somewhat low response rate: it took a week for us to recruit 20 out of a possible 62 highly productive Mechanical Turk workers using this technique. The advantage of this method is that it prevents workers from realizing why a HIT is visible to them, or even that it might not be visible to others, minimizing demand characteristics.

A researcher can also directly contact eligible workers by selecting “Manage” > “Manage HITs individually” and selecting the appropriate HIT > “Download Results.” Assignments are categorized on this page as “Approved,” “Rejected,” and “Pending Review,” with individual WorkerIDs listed under each tab. Clicking on a WorkerID presents the options to send an email or grant an individual bonus. Researchers who have conducted longitudinal research on Mechanical Turk by directly contacting participants and asking them to complete a follow-up study have typically obtained response rates greater than 60% (Berinsky et al., 2010; Buhrmester et al., 2010).

***Collecting Responses in Sequence for Longitudinal Research.*** Qualifications earned by workers can be overwritten with a new value by the creator of the qualification. This feature allows a requester to present several different surveys that workers complete in sequence. For example, a requester who wants workers to complete ostensibly unrelated measures in a particular order (e.g., a parenting stress questionnaire and a questionnaire about child educational outcomes) can use a single qualification to manage access to all measures and assign different values to workers who are at different stages in the sequence of questionnaire completion (e.g., “1” for workers eligible to complete the first questionnaire and “2” for workers who have completed the first questionnaire and are thus eligible to complete the second questionnaire). Each questionnaire in the sequence could then be associated with a HIT that was only visible to workers with a specific value for this qualification, ensuring that they are completed in order.

***Preventing Duplicate Workers.*** As discussed earlier, duplicate workers are a serious problem on Mechanical Turk. Researchers have developed a number of strategies to avoid duplicate workers that are sometimes adequate but also have serious drawbacks. We review these methods briefly before extending the logic of qualifications to offer a more comprehensive approach.

One method used to eliminate duplicate workers is to keep a list of all workers who have participated in similar experiments and to exclude them from subsequent experiments post hoc, which allows researchers to simply delete duplicate respondents. However this approach is time consuming, financially inefficient, and may encourage data mining – it seems unnecessarily tempting to analyze data before duplicate participants are excluded and if this analysis “works” researchers may be less motivated to check whether analyses are equally supportive of hypotheses after duplicate workers are excluded.

An alternative is to create a single Mechanical Turk HIT that links to an external webpage that in turn links to a second webpage containing the experiment. This linking page can be edited, allowing several different experiments to be run from the same HIT batch, with additional assignments added to the original HIT whenever a new experiment is released. Although this allows a requester to easily avoid duplicate workers across many experiments, over the long run this approach sacrifices speed because workers are more likely to complete recently posted HITs (Chilton et al., 2009). Further, once posted, you cannot change properties of the HIT (e.g., compensation) to reflect the requirements of individual experiments.

Qualifications are far more efficient at preventing duplicate workers while allowing a requester to vary features of HITs (pay rate, instructions, etc.). A system to prevent duplicate workers can be created by extending the logic of conducting longitudinal research described in the section above. First, a qualification must be created that automatically grants a numerical value (e.g., “0”) to anyone who requests it. Workers who want to complete the HIT will see that the qualification is required and request it. An automatically granted qualification is created with syntax similar to that described in “Creating a Prescreening Question (above) except *autogrant* is set to *true* and *autogrant value* is set to *0*. Second, all HITs of a given type must require that a worker have a value of 0 for this qualification in order to complete them. Third, when workers complete the experiment, they need to be assigned a new qualification value (e.g., “1”) using the updating methods described in “Collecting Responses in Sequence for Longitudinal Research” above. Since HITs require a qualification value of “0”, workers who have completed a study (earning a value of 1) will be unable to participate in others. A value assigned by the researcher cannot be overridden so the worker cannot simply retake the qualification to gain access.

The list of excluded workers can be supplemented with workers who have not yet completed the qualification, as long as the requester has their WorkerIDs, making it possible to proactively assign a value to workers who have participated in related experiments before the qualification existed or even by other experimenters.

### **Awarding Bonuses**

As discussed earlier, bonuses are an effective way to incentivize workers to attend to experimental details. They also have other uses (e.g., paying workers for completing experimental economics studies with variable payouts; Horton et al., in press). Bonuses can be granted to individual workers through the web interface. However, this is not practical when awarding bonuses to large numbers of workers. The CLT offers a relatively simple way to grant bonuses *en masse*.

We will present the “no coding required” strategy for the Windows version of the CLT, and the “minimal coding required” strategy for the Unix version used by Macs. Both strategies can be used on either system, although as discussed earlier, the syntax used will vary depending on the coding platform used. To assign bonuses (e.g., for getting the manipulation check correct), each worker must have a unique ID entered in both the HIT and the survey data file so that individual worker’s responses are identifiable.

***No coding required.*** Bonuses can be assigned directly from the terminal by typing commands after the C:\ prompt.

- 1) Download your survey data, and determine who deserves bonuses based on your criteria.
- 2) Copy the unique IDs of those who deserve bonuses into a column on a new sheet, and sort alphabetically.

- 3) Download your Mechanical Turk data (the .csv file) and open it using Excel. (The unique IDs for all participants should appear here along with their Mechanical Turk WorkerIDs.) Merge the two files with the unique ID variable as the key variable.
- 4) Paste the WorkerIDs and AssignmentIDs into a .csv file (you can make it as long as you want). Bonus amounts do not need to be identical, e.g., for game theory studies.
- 5) Copy all the rows you have, and Paste Special - Unformatted Text into a blank Word document.
- 6) Replace all the tab characters (^t, or under Special in the Find box) with a blank space.
- 7) Copy all this data, and paste it onto the command line (use the mouse; ctrl+V doesn't work).
- 8) The CLT will display a confirmation as it awards each bonus, and will inform you if you run out of money or if something else goes wrong.

***Minimal coding.*** If you are planning to use the CLT regularly, the minimal coding strategy is quicker, more straightforward, and avoids potential errors in cutting, pasting, and replacing text. To grant bonuses first create a tab-delimited file (Excel and SPSS files can be saved in this format) with columns titled *AssignmentID*, *WorkerID*, *amount*, and *reason*. Save the tab-delimited file, e.g., “bonusfile.txt,” to the directory in which the CLT is installed.

You will also need to save the bonus script (see the appendix for the Mac version) to the same directory. Using a text editor program, save the text of this script in the directory where the tools are installed, using the extension .py to denote a Python script, e.g., “bonusscript.py.”

When you are ready to grant bonuses, navigate (using Terminal on a Mac) to the directory containing the tools and files (e.g., `cd /Applications/aws-mturk-clt-1.3.0/bin/`). The script provided in the appendix will create a second script of all the commands to award each bonus. The syntax to run the script against the tab-delimited file of participants is

`./bonusscript.py bonusfile.txt outputfile.sh`. Run the output script on its own to grant all the bonuses: `./outputfile.sh`.

### **Mechanical Turk Workers as Research Assistants**

Although Mechanical Turk has primarily interested researchers as a source of research participants, workers can also be useful research assistants. Workers can search for stimuli on behalf of researchers and norm them much faster than the combined effort of research assistants and traditional pilot participants (for an example of how to collect speech corpora see Lane, Waibel, Eck, & Rottmann, 2010). Other research has demonstrated that Mechanical Turk workers can in aggregate complete transcription as accurately as professional transcribers and for a fraction of the cost (Marge et al., 2010).

Workers can also code data collected either by researchers (e.g., to pilot or validate stimuli) or by research participants (e.g., to code responses). Although some coding tasks likely require carefully trained researchers to make professional judgments, many coding tasks (e.g., valence of a statement) require no special training.

There are two methods of coding data on Mechanical Turk. First, trusted workers can be recruited to code an entire data set, much as an individual researcher or research assistant would. However, long, high stakes tasks are not ideally suited to Mechanical Turk as the stakes for both the requester (of delays or poor quality work) and the worker (of having the HIT rejected) are high and can result in fatigue effects or other changes within rater across time (Wolfe, Moulder & Myford, 2001).

Second, the task could be distributed across many judges as a batch of much simpler HITs, each requiring that the rater codes a single target of interest. Just as an experiment conducted on Mechanical Turk is effectively a single HIT conducted by many workers,

Mechanical Turk can allow you to create a batch of HITs containing a number of different targets that can each be rated by a set number of workers.

**Designing a Coding HIT.** Unlike data from experimental HITs, which is usually easiest to collect in an external website, coding is best collected within Mechanical Turk because Mechanical Turk will automatically track how many workers have coded each statement and ensure that a worker does not code the same statement twice.

Setting up this kind of HIT is relatively straightforward. First you will need to put the information you wish to have coded into a format in which workers can access it. Text can be placed directly into a .csv file. Images and other media can be uploaded to the web and url links placed in the .csv file. The first row of the .csv file should contain labels for each column. At a minimum this must include a variable name with which Mechanical Turk will identify potential targets to be coded (e.g., “description”). Additional columns can also be included, such as reference numbers that link the statements to be coded with the correct rows in a master data set. Each target to be coded should be placed in its own row.

Once the .csv file is created, the process of designing this kind of HIT is similar to creating a HIT to recruit research participants. However, the number of assignments per HIT indicates the number of workers you want to rate *each item*, not the total number of raters required.

When designing the visual appearance of the HIT itself (in the “design layout” tab), place the variable name inside curly brackets after a dollar sign (e.g.,  $\{description\}$ ) where you wish it to appear. Also include a response box or other method for the worker to input data using HTML. You will then be prompted to upload the .csv file in the “Publish” tab.

**Analyzing Crowdsourced Coding.** A HIT constructed in the manner described above will produce a data set with one row per target and one column per worker requested. Many different workers will appear in each column and the column in which a response appears is dictated by the order in which it was submitted. If the workers provide categorical responses, then Kappa remains an appropriate statistic with which to measure reliability because it measures rates of absolute agreement among workers, which is not sensitive to which responses are in a given column.

Conventional reliability tests for continuous variables assume that each column is provided by a single rater because the relative evaluations of each target (as opposed to absolute agreement) is of importance. Since crowdsourced data does not meet this assumption, a one-way, random effects intraclass correlation should be used instead (McGraw & Wong, 1996). This particular intraclass correlation assumes that the ordering of observations across columns is irrelevant. The tradeoff is that any variance that is not attributable to the target is considered error, which can result in relatively lower correlations. However, this can be offset by the sheer number and speed of workers on Mechanical Turk.

### **Concluding Comments**

In this paper we review the effectiveness of crowdsourcing websites to augment traditional empirical research, with specific reference to Mechanical Turk. There are many practical and some theoretical reasons to use crowdsourcing websites to collect data. We extend previous studies that have emphasized similarities between crowdsourcing and traditional means of data collection by identifying and describing how to implement the unique features of the platform that allow researchers to prescreen participants, spread measures across multiple HITs

that are temporally separate, and pay (variable) bonuses to workers. Further, workers can be used to generate stimuli, clean data or perform other tasks beyond experimental participants.

We also examine potential drawbacks of collecting crowdsourced data and address them. Inattentive participants are a common problem for most surveys and low-impact experiments, but attention can be increased through careful experimental design and by using (or implying the use of) performance-contingent incentives. A survey revealed that cross-talk between participants, either in person or online, appeared to be less of a problem than we expected. Although it is not difficult to find examples of discussion threads about research on Mechanical Turk forums, our findings suggest that only a small proportion of workers read them.

Non-naïve respondents appear to be a more serious issue that will increase over time and that is currently not appreciated by the field. This is understandable as the vast size of the pool of available workers can lead researchers to infer that workers recruited from Mechanical Turk are “less savvy” than traditional subject pools (e.g. Horton et al., in press). However, the pool of available workers is large but not infinite and many workers have completed dozens, if not hundreds of experiments and surveys. Previous research has found that workers report actively seeking out empirical research (Chilton et al., 2009) and in this paper we demonstrate the consequences: There exists a sub-population of extremely productive workers which is disproportionately likely to appear in research studies. As a result, knowledge of some popular experimental designs has saturated the population of those who quickly respond to research HITs; further, workers who read discussion blogs pay attention to requester reputation and follow the HITs of favored requesters, leading individual researchers to collect fans who will undoubtedly become familiar with their specific research topics.

The ease and low effort of data collection Mechanical Turk may make it tempting for researchers to quickly collect data with little thought about whether the design is original or of high impact or to use it as an easy outlet for student projects that have little potential to contribute to the field. Although data collection is fast and easy on Mechanical Turk the pool is far from unlimited and should be treated with care. For unique programs of research this is relatively easy when providing that HITs are offered sparingly and access is restricted using the methods described in this paper to prevent duplicate workers. For more commonly used methods and measures the pool of Mechanical Turk workers presents a commons dilemma for researchers: individual researchers should not assume that their respondents are naïve and groups of researchers would be better off if they can coordinate their recruitment efforts.

In the long run, an effective use of crowdsourcing websites will require not only efforts to avoid duplicate participants, but also greater creativity in experimental design. The ability to incentivize performance afforded by Mechanical Turk, along with the increasing ease with which media can be created and streamed through off the shelf survey packages and the ability to easily study group dynamics using customized research platforms (Horton et al., in press) make it possible to develop studies that are engaging and of high psychological impact (Ellsworth, 2010). Moreover, social scientists should investigate advanced hypotheses about person-situation interaction (using prescreening) or dynamic processes (using longitudinal data) that are less vulnerable to demand effects and currently not being studied in this population. Given that coordinating groups of participants, tracking participants over time, managing the payment of incentive and disguising “ostensibly unrelated studies” are among the most logistically difficult tasks in traditional subject pools, Mechanical Turk may be especially suited for these purposes.

## References

- von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, *321*, 1465-1468
- Berinsky, A. J., Huber, G. A., & Lenz, G. S. Using Mechanical Turk as a Subject Recruitment Tool for Experimental Research. Unpublished manuscript.
- Brock, T. C., Becker, L. A. (1966). 'Debriefing' and susceptibility to subsequent experimental manipulations, *Journal of Experimental Social Psychology*, *2*, 3-5.
- Buhrmester, M. D., Kwang, T., Gosling, S. D. (2011). Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, *6*, 3-5.
- Chilton, L. B., Horton, J. J., Miller, R. C., & Azenkot, S. (2009). Task search in a human computation market. In Proceedings of the ACM SIGKDD workshop on human computation (1-9). In P. Bennett, R. Chandrasekar, M. Chickering, P. Ipeirotis, E. Law, A. Mityagin, F. Provost & L. von Ahn (Eds.) *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation* (77–85). New York: ACM.
- Couper, M. (2008). *Designing effective web surveys*. New York; Cambridge University Press.
- Downs, J. S., Holbrook, M. B., Sheng, S., & Cranor, L. F. (2010). Are Your Participants Gaming the System? Screening Mechanical Turk workers. In *Proceedings of the 28th international conference on Human factors in computing systems* (2399-2402). New York: ACM.
- Edlund, J. E., Sagarin, B. J., Skowronski, J. J., Johnson, S. J., Kutter, J. (2009). Whatever happens in the laboratory stays in the laboratory: The prevalence and prevention of participant crosstalk. *Personality and Social Psychology Bulletin*, *35*, 635–642.

- Ellsworth, P. C. (2010). The rise and fall of the high-impact experiment. In M. H. Gonzales, C. Tavis, & J. Aronson (Eds.), *The scientist and the humanist: A festschrift in honor of Elliot Aronson*. New York: Psychology Press.
- Gaggioli & Riva (2008). Working the Crowd. *Science*, *12*, 1443.
- Glinski, R.J., Glinski, B.C. and Slatin, G.T., (1970). Nonnaivety contamination in conformity experiments: sources, effects, and implications for control. *Journal of Personality and Social Psychology*, *16*, 478–485.
- Gosling, S., Vazire, S., Srivastava, S., & John, O. (2004). Should We Trust Web-Based Studies? A Comparative Analysis of Six Preconceptions About Internet Questionnaires. *American Psychologist*, *59*, 93–104.
- Henrich, J., Heine, S., & Norenzayan, A. (2010b). Most people are not WEIRD. *Nature*, *466*, 29.
- Horton, J., & Chilton, L. (2010). The labor economics of paid crowdsourcing. *Proceedings of the 11th ACM Conference on Electronic Commerce*. New York: ACM. Available at SSRN: <http://ssrn.com/abstract=1596874>
- Horton, J.J., Rand, D.G., and Zeckhauser, R.J., In press. The Online Laboratory: Conducting Experiments in a Real Labor Market. *Experimental Economics*
- Johnson, J. A. (2005). Ascertaining the validity of Web-based personality inventories. *Journal of Research in Personality*, *39*, 103-129
- Ipeirotis, P. 2011. *Do Mechanical Turk workers lie about their location?* Retrieved from <http://behind-the-enemy-lines.blogspot.com/2011/03/do-mechanical-turk-workers-lie-about.html>).

- Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the ACM conference on human factors in computing systems* (pp. 453–456). New York: ACM.
- Krantz, J. H., & Dalal, R. (2000). Validity of web-based psychological research. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 35–60). New York: Academic Press.
- Lane, I., Weibel, A., Eck, M., & Rottman, K. (2010) Tools for collecting speech corpora via Mechanical-Turk. *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk* (184–187). Stroudsburg, PA: Association for Computational Linguistics.
- Marge, M.; Banerjee, S.; Rudnicky, A.I. (2010). Using the Amazon Mechanical Turk for transcription of spoken language. *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on* (5270-5273). Washington: Institute of Electronics and Electrical Engineers.
- McGraw, K. O., Wong, S. P. (1996). Forming Inferences about some intraclass correlation coefficients. *Psychological Methods, 1*, 30-46.
- Mason, W., & Watts, D. (2009). Financial incentives and the “performance of crowds. In P. Bennett, R. Chandrasekar, M. Chickering, P. Ipeirotis, E. Law, A. Mityagin, F. Provost & L. von Ahn (Eds.) *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation* (77–85). New York: ACM.
- Oppenheimer, D. M., Meyvis, T., & Davidenko, N. (2009). Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology, 45*, 867–872.

Paolacci, G., Chandler, J., & Ipeirotis, P. (2010). Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5.

Rand, D. G. (2011). The promise of Mechanical Turk: How online labor markets can help theorists run behavioral experiments. *Journal of Theoretical Biology*, forthcoming

Rosenthal, R., & Rosnow, R. L. (2009). *Artifacts in Behavioral Research*. New York: Oxford.

Suri, S., & Watts, D. J. (2010). Cooperation and contagion in networked public goods experiments. Retrived from <http://arxiv.org/abs/1008.1276>.

Wolfe, E. W., Moulder, B. C., Myford C. M. (2001). Detecting differential rater functioning over time (DRIFT) using a Rasch multi-faceted rating scale model. *Journal of Applied Measurement*, 2, 256-280.

Table 1

*Distractedness and Involvement Among Mechanical Turk Workers*

	Overall	No	0-90 <sup>th</sup>	90-98 <sup>th</sup>	99 <sup>th</sup>	M-H $\chi^2$ <sup>b</sup>
		productivity	percentile	percentile	percentile <sup>a</sup>	
		information				
With other people	27%	32%	20%	23%	15%	4.95*
Listening to Music	14%	18%	10%	10%	0%	8.85**
Watching TV	18%	24%	12%	14%	15%	3.53†
Chatting online	6%	9%	3%	0%	3%	5.45*
Read Mturk blogs	28%	26%	26%	36%	40%	3.64†
Follow Requesters	55%	43%	68%	71%	72%	19.55***
Follow Academic	33%	27%	39%	32%	48%	4.93*
Requesters						

*Note.* Percentages are the proportion of respondents who affirmed that they engaged in this particular behavior. Productivity percentiles were assigned based on the number of HITs completed in a 132 previous samples.

<sup>a</sup>Includes high productivity workers who completed the initial questionnaire ( $N = 13$ ) and a targeted supplemental sample ( $N = 20$ ) recruited immediately after collection of the initial sample.

<sup>b</sup>Chi-Square and significance tests for Mantel-Haenszel linear-by-linear association test.

† $p < .06$ , \* $p < .05$ , \*\* $p < .01$ , \*\*\* $p < .001$

Table 2

*Previous Exposure to Common Experimental Paradigms*

	Overall	No productivity information	0-90 <sup>th</sup> percentile	90-98 <sup>th</sup> percentile	99 <sup>th</sup> percentile <sup>a</sup>	M-H $\chi^2$ <sup>b</sup>
Prisoner's Dilemma	56%	36%	71%	85%	88%	68.71***
Ultimatum Game	52%	32%	65%	78%	94%	69.12***
Dictator Game	0%	22%	51%	64%	76%	64.79***
Trolley Problem	30%	10%	33%	68%	85%	107.95***
p-Beauty contest	7%	5%	10%	10%	9%	6.68**

*Note.* Percentages are the proportion of respondents who affirmed that they engaged in this particular behavior. Productivity percentiles were assigned based on the number of HITs completed in a 132 previous samples.

<sup>a</sup>Includes high productivity workers who completed the initial questionnaire ( $N = 13$ ) and a targeted supplemental sample ( $N = 20$ ) recruited immediately after collection of the initial sample.

<sup>b</sup>Chi-Square and significance tests for Mantel-Haenszel linear-by-linear association test.

\*\* $p < .01$ , \*\*\* $p < .001$

Table 3

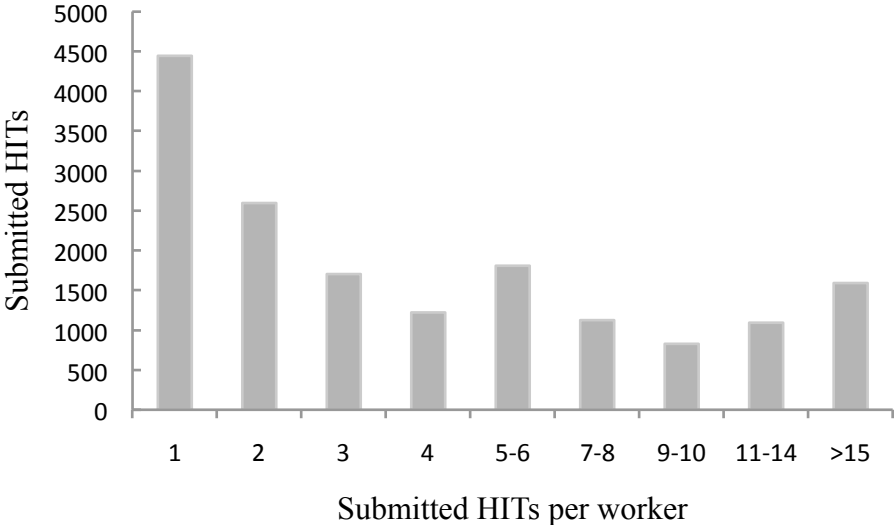
*Topics of Conversation about Mechanical Turk*

	<u>In Person</u>		<u>Online</u>	
	Mean Rank	Overall Rank	Mean Rank	Overall Rank
How much the HIT pays	5.64 (2.22)	1	4.41 (2.51)	1
How long the HIT takes	4.67 (1.80)	2	4.06 (1.78)	3
How fun the HIT was	4.08 (2.00)	3	3.77 (2.28)	6
How difficult it is to complete	3.94 (1.65)	4	3.90 (1.49)	4
How to successfully complete the HIT	3.53 (1.58)	5	3.82 (1.70)	5
Purpose of the HIT	2.98 (1.92)	6	3.64 (2.39)	7
Requester reputation	2.45 (2.02)	7	4.15 (2.15)	2

*Note.* Participants ranked all discussion topics by frequency. Mean rank scores are reversed so that a larger number denotes greater frequency. Overall rank is the rank order of aggregated means.

Figure Caption

*Figure 1.* Number of HITs completed by workers of different levels of productivity.



## Appendix A: Bonus Script

```
#!/usr/bin/python

import os, sys

input_file = sys.argv[1]

records = open(input_file, 'r').read().replace("\r", "\n").split("\n")

output_file = sys.argv[2]
f = open(output_file, 'w')

f.write("#!/bin/bash\n")

f.write("pushd $MTURK_CMD_HOME/bin/\n")
for r in records[1:]:
    (assignmentID,workerID,bonus,comment) = r.strip().split('\t')
    cmd="./grantBonus.sh -workerid %s -assignment %s -amount %s -reason %s"
    %(workerID,assignmentID,bonus,comment)
    f.write(cmd + "\n")

f.write("popd\n")
f.close()

os.system("chmod +x " + output_file)
```

