Hine, Emmie
2020 Computer Science Thesis

Title: ExperiMan: Automatically Correct and Replicable Online Experiments:
Advisor: Daniel Barowy
Advisor is Co-author/Adviser Restricted Data Used: None of the above
Second Advisor:
Release: release now
Authenticated User Access (does not apply to released theses):
Contains Copyrighted Material: No

ExperiMan: Automatically Correct and Replicable Online Experiments

by

Emmie Hine

Professor Dan Barowy, Advisor

A thesis submitted in partial fulfillment
of the requirements for the
Degree of Bachelor of Arts with Honors
in Computer Science

Williams College
Williamstown, Massachusetts

May 28, 2020

Contents

List of Figures

List of Tables

Abstract

Since 2010, the social sciences have been confronting a replication crisis: previously trusted results have been proven erroneous, despite the application of seemingly rigorous standards and methods. Widely accepted theories, like the concept of ego depletion (the idea that humans have a finite amount of willpower when making decisions) and its cousin, decision fatigue, have come under scrutiny as repeated experiments fail to find a significant effect (38). In the ensuing decade, much has been written about the need to change analysis methods and increase the number of replication studies. However, while online experiments have been suggested as a way to increase transparency and replicability, as of yet no way has been found to embed replicability into the end-to-end experiment process (44).

We introduce ExperiMan, a crowdprogramming language for replicable online behavioral experiments. With a small amount of simple code from the experimenter, it automatically runs their experiment on a crowdsourcing platform. In addition, it permits experimenters to explore the experiment space at low monetary and time cost and with high-quality results when compared to existing techniques.

ExperiMan builds on an existing crowdprogramming system, AutoMan, and leverages the power of the crowd to ensure that robust results are obtained quickly and at low cost. This thesis focuses on Amazon's Mechanical Turk, but the language can be extended to the crowdsourcing platform of the user's choice. We anticipate that this will not only make it more efficient to replicate studies, but also that the ability to quickly explore the experiment space at low cost will improve the quality of studies conducted both on- and offline.

Acknowledgments

A thesis is not written—nor coded—in a vacuum, and I owe an enormous debt of gratitude to many people. First, to my advisor, Dan Barowy, for his guidance through the ups and downs of the thesis and the major. It's been a privilege to work together, both in-person and remotely. His patience and enthusiasm made this a fulfilling and downright fun experience. I'd also like to thank my second reader, Steve Freund, for his help and expertise in bringing this home.

A major is not created in a vacuum, either, and in the Williams Computer Science Department, it truly takes a village. Thank you especially to Duane Bailey, for helping me chart a path that I never expected; and to Andrea Danyluk, Jeannie Albrecht, and Iris Howley, for leading by example every day and showing me that I do belong as a woman in CS. I will always cherish the hours spent working (and snacking) in the CS common room, wandering the third floor of Thompson, and chatting with whoever happened to be around—no matter the hour.

I was not formed in a vacuum, so finally, I want to thank my parents and brothers for their unconditional love and support, even when they don't understand anything anything I'm talking about. None of this would be possible without them.

Chapter 1

Introduction

1.1 Motivation

In 2010, Cornell University professor Daryl Bem published a paper that seemed to prove the impossible: that extrasensory perception, or ESP, is real. Published in the Journal of Personality and Social Psychology, the paper documented a 10-year study that appeared to show with rigorous statistical methods that ESP exists (19). Though he used methods accepted in social sciences at the time, the results were clearly spurious, which provoked a reckoning in the field of psychology. To demonstrate the problem, a paper published in October of 2011 used accepted statistical methods to show that listening to the song "When I'm Sixty-Four" by the Beatles made students, on average, 1.4 years younger than they were before listening to the song—in other words, that "When I'm Sixty-Four" turned back time. The authors of the paper proposed guidelines for researchers and reviewers to try and rectify some of the issues they saw as causing shoddy results, among them the failure to list all variables, the elimination of observations, and the selective reporting of experimental conditions (45). These revelations about "false-positive psychology" prompted a movement to attempt to replicate widely accepted results, only to find that many supposed gold-standard experiments did not produce the same results (39). In 2015, a study found that only 40% of 100 psychology experiments published in top journals replicated (15). The crisis is by no means limited to psychology; an effort looking at experiments across the social sciences, all published in Nature or Science, found that only 62% showed "a significant effect in the same direction as the original study," and that the effects were on average about 50% of the original effect size (14).

Replication studies are time-consuming and costly, and researchers often feel that their funding would be better used on original research. What is necessary is a system that can integrate replication into the research process rather than it being a supplementary effort after the fact; a paper describing ways to address the replication crisis mentions that conducting a large number of small-scale studies can be extremely informative, and that "collaborative crowdsourcing resources for replicating important scientific claims" are becoming more and more important (44).

Thus, online crowdsourcing—or crowdprogramming—is ideally positioned to help address this crisis. Crowdsourcing is a method of task fulfillment that uses small amounts of labor from a large

number of people. Crowdprogramming leverages programmatic techniques to digitize crowdsourcing. With the advent of the Internet, it has become easier to crowdsource problems through services like Amazon's Mechanical Turk (MTurk) (9). The Internet offers access to millions of people across target demographics, and experiments can be conducted rapidly and at low cost (33). However, issues of quality control, access, and usability remain.

Traditional crowdsourcing platforms face issues of population representativeness and answer quality. MTurk workers are primarily located in the United States and India. They tend to be younger and have a lower-than-average household income, so their demographics are not representative of the overall population (16). Furthermore, workers on paid services like MTurk are incentivized to fulfill tasks as quickly as possible, making low-quality and spam responses a problem (46).

A crowdprogramming system will have to address these problems while also ensuring replicability, expressiveness, and efficiency. Experiments need to be replicated to be accepted as sound science, and we need to ensure that social scientists can run experiments without format constraints. Furthermore, the service needs to be more efficient—in terms of cost, time, or both—than existing services in order to be adopted. Most social scientists are not computer programmers, which provides a significant barrier when it comes to leveraging existing crowdsourcing platforms like MTurk (10). Tools have been developed to aid the use of these platforms, but none are optimized towards social science experiments, even as recognition of the utility of conducting research on crowdsourcing platforms grows (33).

1.2 Application

This thesis proposes a platform, ExperiMan, that allows researchers with minimal programming ability to rapidly design and run an experiment on an online crowdsourcing platform of their choice. By building on an existing platform, AutoMan, we gain access to a platform that allows for the automatic posting of questions to MTurk and Google Ads and the statistical analysis of results. Added support for aggregating questions into surveys expands its utility for research. In addition, we enable researchers to provide a grammar of experiments that allows for the automatic posting of multiple variations of questions, which permits the rapid and systematic exploration of a given experiment space. By providing these tools that are at least as fast, cost-effective, and statistically robust as existing tools, we hope to empower researchers to investigate promising areas of a research topic, then rapidly conduct statistically significant, replicable studies on a target population.

1.3 Contributions

The main contributions of this paper are summarized as follows:

- Grammars of experiments: We introduce the idea of grammars of experiments, which can be explored programmatically to identify areas of interest for further investigation.
- Crowdprogramming platform: We introduce ExperiMan, a crowdprogramming platform designed to run behavioral experiments quickly and cheaply on crowdsourcing services.

- Quality control mechanisms: We discuss mechanisms to help ensure valid results from crowdsourced experiments.

1.4 Organization

The remainder of this paper is organized as follows:

Chapter 2 summarizes the historical background of crowdsourcing and its cousin, crowdprogramming, and explores the replication crisis. We also discuss methods of automated statistical analysis with an eye towards how they can be leveraged in crowdsourced experiments.

Chapter 3 discusses the idea of experiment grammars, including what it means to be a robust experiment, and the ranking and unranking techniques used to explore different experiment variations.

Chapter 4 addresses threats to validity of crowdsourced studies. This includes quality control on crowdsourcing platforms and current and future work to ensure that the noise crowdsourcing introduces is filtered out.

Chapter 5 describes the specific implementation of ExperiMan.

Chapter 6 is an evaluation of two pilot experiments run on ExperiMan.

Chapter 7 concludes by summarizing the contributions of this paper and discussing future work.

Chapter 2

Background

This thesis proposes a system designed to allow social science researchers to crowdsource experiments, with the aim of improving experiment replicability and experiment space exploration. As such, this chapter provides an overview of the concept of crowdsourcing and its origins. From there, it addresses modern "crowdprogramming," which involves digital crowdsourcing through programming. From a discussion of the challenges of crowdprogramming and ways to address them, we proceed to investigate methods of automated statistical analysis, then discuss the replication crisis in social sciences and how this thesis could address it.

2.1 Crowdsourcing

The term "crowdsourcing" seems to have been coined in 2005 by contributing *Wired* editor Jeff Howe, who jokingly used it as a portmanteau for "outsourcing to the crowd" (40). Howe went on to write a book about crowdsourcing and maintained a blog called *Crowdsourcing*, where he defines crowdsourcing as "the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers" (24). The "open call format" utilizes a widely accessible platform (such as a website) to disseminate tasks to a large group of people (the "network of potential laborers"). This definition is deliberately broad so as to encompass a wide variety of tasks, but essentially, crowdsourcing is a method of task fulfillment that uses a small amount of labor, whether physical or intellectual, from a large number of people.

2.1.1 Origins

Crowdsourcing has been used to perform tasks and solve problems for hundreds of years. As early as 1598, European countries were offering rewards to anyone who could come up with a simple method of determining longitude at sea, which was crucial for navigating ships (4). This eventually

manifested itself in the British Longitude Act of 1714, which formalized the reward structure (41). With an administrative bureau and reward structure, this was a fairly sophisticated application of crowdsourcing.

In its most basic form, crowdsourcing can be used as a way to pool a large quantity of information from a physical crowd, such as when, in 1906, Francis Galton polled a crowd of 800 people about the weight of an ox. Though individual answers were far off, the median of all the estimates was within 0.8% of the actual weight (21). Though we now rarely have to guess the weight of oxen, crowdsourcing still offers a myriad of uses. In popular culture, this form of knowledge gathering has manifested itself as a "Poll the Mob" help option in the game show *1 vs. 100*, which pits a main contestant agains a "Mob" of 100 people in a trivia competition (1). Crowdsourcing helps millions of people every day; resources like Wikipedia crowdsource encyclopedic knowledge, while navigation app Waze allows users to submit traffic reports and road closures to optimize routes (50). Crowdsourcing has also been gamified to predict the structure of folded proteins, in some cases outperforming traditional algorithms (43).

2.1.2 Digitization

In the years since Dalton's experiment, access to crowdsourcing has been formalized via digital platforms. Crowdsourcing has been digitized to the extent that modern definitions emphasize the online aspect. Enrique Estellés-Arolas and Fernando González-Ladrón-de-Guevara, two crowdsourcing researchers based in Spain, when attempting to come up with an "integrated crowdsourcing definition," state in the definition's first sentence that "crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task" (20). This definition requires crowdsourcing to take place online, a significant shift from the paper calls for participation of the longitude rewards age.

Though this definition excludes physical forms of crowdsourcing, in practice, much formal crowdsourcing today takes place online. Numerous platforms for digital crowdsourcing exist. One of the most popular is Amazon's Mechanical Turk (MTurk), which allows askers to post "microtasks" to be performed by paid workers (3). Because crowds are inexpensive and available on-demand, researchers have begun using services like MTurk to run experiments (33; 44). However, due to issues inherent to the paid worker system that will be discussed further in the next section, researchers have been increasingly moving towards less traditional crowdsourcing platforms to perform research. As part of this, some novel works have begun probing what the crowd can do. Quizz, a "gamified crowdsourcing system," uses publicly available advertising systems (most notably Google Ads) to crowdsource answers to multiple-choice questions (25). However, because Quizz is dedicated to knowledge gathering and verification, it is not usable for behavioral studies.

A distinction should be made between "batch work," the typical use of crowdsourcing, and online behavioral experiments. Batch work typically consists of thousands of tiny tasks—for example, transcribing a shopping receipt—and is often used to train machine learning algorithms (17). Online behavioral experiments, in contrast, leverage the crowdsourcing framework developed for batch work

to conduct social science research. Most crowdsourcing platforms were originally developed for batch work, but some are now being leveraged in new ways for research, and some non-crowdsourcing platforms are being adapted for behavioral studies.

2.1.3 Selected Platforms

Three platforms have emerged as notable in the digital crowdsourcing space: Mechanical Turk, Appen (formerly Figure Eight), and Google Ads (previously Google AdWords). Also of interest to the space but not directly relevant to the digital crowdsourcing sector are services that connect users with workers to perform more specific, often physical, tasks, like oDeskWork and Task Rabbit.

Within the digital space, Mechanical Turk facilitates the presentation of "microtasks," also called "Human Intelligence Tasks" (HITs), to paid human workers. Mechanical Turk is unique in that it is highly programmable. Appen markets itself to large companies as a way to combine human and machine intelligence to label data, focusing exclusively on batch work to facilitate AI projects. Due to demographic, ethical, and quality issues, some researchers have been investigating less traditional platforms, like Google Ads, to use in crowdsourcing.

Mechanical Turk

Mechanical Turk, often abbreviated to MTurk, was originally built for "human computation tasks," small pieces of work that have proven difficult for computers, such as audio transcription or image analysis. Requesters post HITs to the site, optionally specifying qualifications to restrict which workers may work on the task. Workers may log on and choose any task for which they are qualified to work on. Requesters can specify a lifetime for a HIT (the amount of time it remains in the HIT listings) and the duration of the HIT (the maximum amount of time a worker can spend working on a task). Requesters can choose to accept or reject completed work; if a task is rejected, the worker is not paid, and the rejection is registered in the worker's assignment-acceptance rate, which thus signals worker quality. Amazon allows requesters to access MTurk through the MTurk website or the Amazon Web Services (AWS) SDK, which allows for the programmatic access and creation of of requests.

While companies and individuals still use MTurk for its original purpose of performing computational tasks, it has also increasingly been used for behavioral studies. Researchers Winter Mason and Siddharth Suri identify the crucial factors permitting this to be subject pool access, subject pool diversity, low cost and built-in payment mechanism, and a faster theory/experiment cycle (33). Though it is not a perfect platform, researchers are increasingly utilizing it. New automated platforms to facilitate using the service, such as AutoMan, will be discussed further when we explore crowdprogramming (10). Amazon claims that there are 500,000 workers on the platform from 190 countries, but a 2018 study suggests that there are closer to 100,000 available workers, with about 2,000 active at any given time (16). Contrary to what Mason and Suri promote as "subject pool diversity," about 75% of the workers are from the United States, with another 16% from India. While the overall workforce is relatively gender balanced, it is skewed by country; 55% of the U.S. worker population is female, while in India and most other countries, the MTurk workforce is majority male.

It is most unbalanced in Germany, where over over 75% of MTurk workers are male, but Germany comprises just 0.27% of MTurk workers. India's workers, the second largest group after the U.S., are about 65% male (16). Thus, its demographics are not especially representative of the global or online population.

Where MTurk does excel is its ability to marshal workers on demand. By using a retainer model where workers are paid a small amount as they wait for work, then notified when a task is available, a crowd of eight to fifteen workers can be recruited within two seconds to work on a given task (11). Tasks that require more assignments can take more time to complete; a sample survey that requires 500 responses took between seven and twenty-one days to complete, depending on the wage level, although these surveys were run in 2010 and are no longer representative of typical response times (33). However, the retainer model is not a built-in feature; MTurk's primitive interface has forced the development of supplemental tools for maximizing its utility.

The small amount paid to MTurk workers has raised ethical concerns about the use of the platform. A reporter for the New York Times spent eight hours "turking" and made an effective wage of $0.97 an hour; while some workers who use computer scripts and automated tools can make over $12 an hour, over half of "turkers" make less than $5 per hour (34). Services that automate posting are generally optimized to minimize cost to the requester, which thus minimizes wages earned by workers and raises concerns about programming languages exacerbating economic inequalities (12). (We note that AutoMan addresses crowdworker rights by paying the U.S. minimum wage by default, paying workers promptly, and never arbitrarily rejecting work (12).) These ethical issues have prompted some to propose the use of alternative crowdsourcing platforms like Google Ads, which relies on voluntary, unpaid participation.

Appen

Appen (known as Figure Eight before a rebranding in 2020, and CrowdFlower prior to 2018) is a crowdsourcing platform focused on data annotation and collection that combines a high-quality crowd with machine learning models. They claim to have over a million workers on their platform, and users can target workers by language, location, and skillset (5). In a study comparing MTurk and Figure Eight's predecessor, CrowdFlower, CrowdFlower showed a a higher geographical diversity and faster response rate, but low reliability and higher attention-check question failure rates (35). Appen integrates human workers and algorithms, claiming to automatically take care of quality assurance (6). Appen provides an interesting integration of humans and machine learning algorithms, but is targeted towards large companies that need huge quantities of data annotated or images analyzed.

Google Ads

Google Ads advertises itself as a way to "[make] it easy to show the world what's unique about your business, so you can reach customers searching for what you offer," not as a way to harness a crowd to perform small tasks (22). However, as one of the largest advertising platforms in the world, it is easy to get information in front of a large number of people, and it allows for the targeting of specific demographics. Prospective advertisers provide Google with their goal—more calls, store visits, or

website action for their business—as well as the advertising region, message, and budget, then pay only for "results." Results include ad clicks and website visits. Google's proprietary algorithms change ad placements over time to maximize engagement (22).

Looking at the four factors identified by Mason and Suri (subject pool access, subject pool diversity, low cost and built-in payment mechanism, and a faster theory/experiment cycle), Google Ads fulfills all of them: Google Ads grants access to the billions of users who interact with Google; this subject pool is extraordinarily diverse, and crucially can be targeted to "reach users with specialized expertise that is not typically available through existing labor marketplaces;" the cost can be lower than existing services; and unpaid, targeted, intrinsically motivated workers can be faster than paid, non-expert workers (25; 33). Quizz, the aforementioned gamified knowledge-gathering system, has inspired our decision to work to incorporate Google Ads into ExperiMan, but as it is not a traditional crowdprogramming platform, we elected to begin with MTurk.

2.1.4 Challenges

Although crowdsourcing offers great potential for task fulfillment and social science research, the way it is conducted also presents significant challenges. Chief among those is response quality, but cost and time can also be significant factors that must be addressed in order for a system to be widely adopted.

Quality

One of the primary challenges encountered in crowdsourcing is response quality. Paying workers to perform tasks creates an incentive structure that is not always conducive to effective research. Workers are motivated to complete as many tasks as possible, which leads some to answer randomly or use automated bots to complete tasks—poorly—on their behalf (30). Even with monetary bonuses for good performance, the makers of Quizz found that, for their quiz on medical knowledge, MTurk workers perform barely above random, suggesting either that users are totally unknowledgeable about the subject, or that they are answering randomly (25). One survey asking MTurk workers to rate the quality of Wikipedia articles saw a 48.6% invalid response rate. By including a CAPTCHA-like validation question with a verifiable answer, that rate dropped to 2.5% (33). However, adding such a "reverse Turing test" question to catch bad actors does not address the other problem with accessing crowdsourcing services, which is that scheduling tasks, determining wage levels and task lifetimes, and validating responses is tedious, and without some sort of statistical analysis, figuring out how many tasks one needs to run to obtain statistically significant results is difficult and inefficient (33).

For research, one factor contributing to quality challenges is worker demographics. As previously mentioned, the majority of MTurk workers are from the United States and India. Many other services have similar demographics issues, with only CrowdFlower and Prolific Academic (now Prolific, a research-centered crowdsourcing platform further discussed below) identified as having significant numbers of workers outside the United States and India (35). MTurk workers—and workers on crowdsourcing platforms in general—tend to be younger than the U.S. average age and have in-

comes below the U.S. average income (16; 35). Thus, they are not representative of the online or offline population, and the population is small enough that it is difficult to target workers that are experts in any particular subject. This has significant implications for researchers, who may need a demographically representative population or specifically targeted group for an experiment.

Time

When considering the time costs of crowdsourcing, we must consider both experimenter time and experiment runtime. The time an MTurk task takes to run can vary according to the reward, creating a trade-off that programmers must consider. The relationship between compensation and runtime appears to be positive but nonlinear; in 2010 a 500-assignment 3-question survey HIT took about a week to run at a wage of $0.05, nine days to run at a wage of $0.03, and over three weeks to run at $0.01 (33). These times should not be considered representative of completion times now; the platform has grown since 2010, and the results for the experiments we discuss in Chapter 6 were obtained on the order of minutes, not days. Regardless, given that an experimenter does not take the time to recruit a panel of subjects, crowdsourcing can be a way to run tasks faster than recruiting and bringing a group of subjects together.

Cost

Cost can be another significant challenge for crowdsourcing. However, research has shown that the minimum rate at which workers will accept work on MTurk is $1.38 per hour, and the average hourly wage is $4.80. Laboratory subjects typically cost above the U.S. minimum wage, so in comparison, crowdsourcing can be extremely cost-effective, even when factoring in the 20% fee Amazon charges requesters (2; 33). For Google Ads, the quality-adjusted cost is comparable to traditional crowdsourcing platforms (25).

Ethics

Most ethical concerns from MTurk derive from the low wage that workers are paid. As previously mentioned, most MTurk workers make less than $5.00 per hour, well below the U.S. minimum wage of $7.25 per hour, and far below what many consider a living wage (34). Crowdsourced workers are considered independent contractors, not employees, and so are not covered by the Fair Labor Standards Act, which establishes minimum wage and mandatory benefits (42). Some justify the low wages paid by pointing to the fact that crowdsourcing workers are generally not relying on those wages for their primary income and are not obligated to do the work (33). However, some workers work on crowdsourcing sites because they live in economically depressed areas and have no other work options; others have monthly expenses, like insulin, that they cannot cover without supplementary income (34). Thus, to avoid exploiting workers, the system that we are building on top of, AutoMan, by default sets compensation to accord with the U.S. federal minimum wage.

Another ethical issue that has been raised with regards to conducting studies on crowdsourcing sites is informed consent and debriefing. However, on MTurk, HIT preview pages can contain statements on the purpose of the study (and the worker can elect to not continue the HIT after

reading it), and experimenters can show a debriefing statement when the task is complete (33). Thus, crowdsourced studies can fulfill the requirements for human subject research.

2.2 Crowdprogramming

2.2.1 Origins

Crowdprogramming is a form of digital crowdsourcing that "integrates human-based and digital computation," providing a programatic way to interface with a human crowd and crowdsource tasks (10). Much work on crowdprogramming has centered on MTurk, utilizing the low-level API Amazon exposes, and several platforms to automatically manage MTurk tasks have sprung up. These services all attempt to address the problem of accessing crowdsourcing services, and some additionally attempt to automatically address quality issues.

2.2.2 Benefits

Crowdprogramming is an improvement on traditional crowdsourcing because it does not require the requester to manually manage their tasks. Crowdprogramming automatically manages workers and can also ensure response quality. Furthermore, by allowing for programmatic access to different platforms, crowdprogramming can also address the demographics issues that plague many crowdsourcing sites. As an added benefit, it can provide a simplified programming model; by abstracting platform details away, it means researchers can launch studies with minimal programming experience.

2.2.3 Selected Platforms

Several platforms have developed to make crowdprogramming accessible to programmers. We will examine six of them in particular: TurKit, AE, Prolific, AutoMan, VoxPL, and SurveyMan.

TurKit

TurKit provides a web user interface and scripting language to allow requesters to run algorithmic tasks on MTurk. Algorithmic tasks, which depend on the contributions of multiple workers, are distinct from independent tasks, which are generally very similar and can be run in parallel. It relies on a "crash-and-rerun" programming model and local computation to ensure that the algorithm is run in a fault-tolerant way (32). Its algorithmic task approach is expanded on with Turkomatic, which allows users to describe a complex task that is then broken down by MTurk workers into component tasks that are then solved by other MTurk workers (31). One of TurKit's major contributions is abstracting MTurk operations away as a simple function call, which we see continued in other crowdprogramming systems. However, it does not have automatic quality control mechanisms, putting the onus on the programmer to ensure that results are robust.

AE

AE is a "domain-agnostic" machine learning platform for online A/B experiments. It allows users to perform many sequential experiments and automatically creates and deploys experiment batches, while also gathering data and performing analyses. AE is useful for large organizations seeking to optimize specific metrics through A/B tests, but its specificity and requirement of Python knowledge makes it impractical for most social scientists (8).

Prolific

Formerly Prolific Academic, Prolific was founded in 2014 by graduate students from Oxford and Sheffield Universities. It is intended to help academics find research participants from target demographics, something crucial to our platform as well. Prolific workers are less dominated by Americans; a study indicated that 56% are from the U.K. and Europe, with a further 30% from North America. However, the worker pool is not ethnically diverse (35). Furthermore, at 70,000 workers (who on average spend less time on the platform than MTurk workers do), the worker pool is significantly smaller than MTurk's, which is reflected in a response rate in some cases four times slower than on MTurk (36; 35). Overall, response quality tends to be similar to that of MTurk (35). However, Prolific is significantly more expensive than MTurk; a quote for a 5-minute, 500-participant survey is $423.08, a 28.6% increase over the $302.08 it would cost on MTurk using the default U.S. federal minimum wage. The quoted cost increases to over $1100 when a population "nationally representative of UK/US" is requested (37).

AutoMan

The system that we are building on top of, AutoMan, is the first fully automated crowdprogramming system. The programmer writes a simple program specifying the task, budget, and desired confidence level for the response. AutoMan then automatically schedules tasks on MTurk or another crowdsourcing service until the desired confidence level is reached or the budget is exhausted. Based on the estimated time a task will take, AutoMan sets the initial wage using the United States federal minimum wage of $7.25 an hour. This addresses the issues of pay determination and time allocation. By automatically scheduling and re-scheduling tasks (with increased wage and time allowance, if necessary), AutoMan eliminates the need for the human programmer to do so, and because AutoMan continuously calculates the number of responses needed based on the specified confidence level, the programmer can rest easy knowing that, provided their budget is sufficient, they will get a response with the desired level of confidence. AutoMan currently supports multiple-choice questions with one or multiple answers and text entry forms (10).

VoxPL

AutoMan is limited to labeling tasks of the question types described. VoxPL generalizes the platform to allow for crowdsourced estimates of values. It supports both single and multiple estimates, and

uses sophisticated statistical methods to achieve the desired confidence threshold and interval width (i.e., +/- 20 pixels when identifying Waldo in a "Where's Waldo" photo) (9).

SurveyMan

Both AutoMan and VoxPL are limited to single questions (or multiple estimates for an estimation task with multiple parameters). SurveyMan provides a domain-specific language for posting surveys to MTurk. SurveyMan provides tools to ensure that question order is randomized (within constraints provided by the programmer) and performs static analysis to ensure that the survey is well-formed. It also performs path analysis to prove the programmer insight into the maximum number of questions the survey asks and the maximum entropy of the survey. It uses an entropy calculation to identify and discard responses from likely bad actors (47). We aim to take this survey support and expand on it, providing more flexibility and tools for researchers to use, in addition to methods for exploring an experiment space.

2.3 Automated Statistical Analysis

Statistical analysis is crucial for any experiment. Social scientists need to ensure that their results are statistically sound and that they can be confident in their experiment. Experimenters historically have had to do statistical analysis by hand, but now platforms and programming languages like R exist to help statisticians automatically analyze their results. In this section, we will explore existing tools for statistical analysis of crowdsourced programs and their implications for crowdsourcing experiments.

2.3.1 Selected Platforms

We analyze the approaches used by crowdprogramming platforms AutoMan and VoxPL, as well as the analysis tool Tea. These approaches range from confidence level analysis to nonparametric bootstrapping to automatically identifying and running appropriate statistical tests.

AutoMan

AutoMan uses confidence levels to ensure that its results are valid (according to the programmer). When initially creating an AutoMan program, the programmer specifies the desire level of confidence (which defaults to 95% to accord with the standard p-level of 0.05). In other words, the programmer can expect that the answer is a result of random chance 5% of the time. For the first round of HITs, AutoMan spawns the minimum number needed to reach the confidence level if all workers agree on an answer. If agreement has not been reached at the desired confidence level, AutoMan spawns additional questions, applying the Bonferroni correction to increase the necessary confidence level and avoid early termination. For high-entropy questions (like checkbox questions that permit multiple checks or free-text responses) that have many possible answers, a smaller number of agreeing answers are necessary to reach a given confidence level, so AutoMan uses random checkbox filling

to lessen the likelihood of random responses according with each other in case workers submit a question without changing any of the checkboxes. To counter lazy workers with free-text responses, AutoMan only allows the empty string if it is explicitly allowed by the programmer and entered as "N/A" (10).

AutoMan's algorithm relies on workers being independent, but as it is difficult to have multiple accounts, workers can be restricted to a single task in a larger computation. Since collusion is unprofitable, it can generally be assumed that workers will not actively try to game the system. Sybil attacks, where a single entity presents multiple identities, is a theoretical risk, but given the margins of pay for crowdsourcing workers, the time required to set up such an attack would be detrimental to the worker's overall pay (18). Rejecting incorrect answers (and thus impacting workers' acceptance rates) is incentive for workers not answer randomly, preserving the integrity of the statistical analysis (10).

Though most of AutoMan's analysis is automatic, it depends on the programmer to set a confidence level and then decide what to do with the collected data. We will see that some systems attempt to provide a more comprehensive statistical analysis with less input from the programmer.

VoxPL

VoxPL builds on AutoMan's statistical methods, but includes additional sophistication because estimates by their very nature will not agree. VoxPL relies on the basic bootstrap, which is a "nonparametric (i.e., distribution-free) Monte Carlo procedure that estimates error bounds for arbitrary functions of unknown multidimensional distributions" (9). Bootstrapping estimates the desired statistic and calculates the confidence interval. The bootstrap uses a default initial sample size of 12, and doubles the sample size after each round of sampling if it determines that more responses are needed to meet the user's confidence level and has not exceeded the budget. It also uses the Bonferroni correction to avoid terminating too early (9). VoxPL uses a variety of innovative methods to ensure that estimates accord with the programmer's confidence specifications and provides a good springboard for more complex experiments.

Tea

Tea's goal is to "lower [the] barrier to valid, replicable statistical analysis" by allowing users to provide high-level specifications, which are then compiled into a constraint satisfaction problem that determines what statistical tests are appropriate to run and executes them. It can run both parametric and non-parametric tests, and helps avoid false negatives and positives caused by the use of incorrect statistical methods. The user specifies their variables, the study design, any assumptions they are making about the data, and the hypothesis, but not what tests should be performed. Each statistical test is encoded with a set of preconditions (derived from a statistics course, a statistics textbook, and public data science resources) and is only run if the information provided by the programmer satisfies all the preconditions. Currently, Tea only supports tests related to Null Hypothesis Significance Testing (29). Tea eliminates errors resulting from improper use of statistical tests and makes automated statistical analysis more accessible, which resonates with our mission.

Automated statistical testing is a crucial part of social sciences, and ensuring that scientists have the tools they need is central to our goal.

2.4 Replication Crisis

Reproducibility of experiments is crucial for the advancement of science. However, since 2010, scientists have been raising the alarm about social science experiments failing to replicate at alarmingly high rates (39). Much attention has been focused on psychology, which seems to have an especially acute crisis, but many social sciences, including economics, are impacted.

2.4.1 Causes

Three events have contributed to the perception of a crisis in psychology: highly publicized cases of scientific fraud, a set of articles criticizing questionable research practices that have caused inflated false positive error rates, and the Open Science Collaboration's replication project (44). The latter effort saw the Open Science Collaboration attempt to replicate 100 results from three top psychology journals. They found that only 36% of replications had significant results (versus 97% of original studies) and replication effect sizes were on average half the magnitude of those in the original studies. When replication results were combined with the original results, just 70% of the studies had significant results (15). As a result, social science fields have been examining their own research practices, resulting in more replication studies pointing to concerningly low levels of replicability. One study attempting to replicate 18 economics studies found that just 11, or 61%, showed significant effects in the same direction as the original study, and that the replicated effect size was on average 66% as large as the original (13).

2.4.2 Redress

Several statistical methods have been proposed to help address study quality and improve replication, including increasing sample size to increase power, meta-analysis of replication attempts, Bayesian analysis, and resampling or cross-validation (44). However, the field needs more than just new methods; it needs a culture shift. As part of this, researchers also note the need for replications to become more common in psychology. One proposed mechanism is the Registered Replication Report (RRR) mechanism created by the Association for Psychological Science. Under the initiative, collaborative replication projects for specific findings are proposed and approved, the protocol is sent to the scientist behind the original findings for comment, then the RRR is published by the journal regardless of outcome. As of 2018, though, only five RRRs have been published (44). RRRs require a significant organizational effort, scientists from multiple teams, and the money and commitment to conduct replication studies instead of original research.

Having a more decentralized approach to replication that can attempt to replicate studies quickly and cheaply could make replication part of the standard research process, rather than a special effort that takes place years after the original study. In fact, the same paper that describes efforts to redress the replication crisis mentions that many small studies can be informative, and that

"collaborative crowdsourcing resources for replicating important scientific claims" are becoming increasingly important (44). In ExperiMan, it is trivial to re-run an experiment, since experiments are just programs. In addition, ExperiMan allows for a systematic exploration of the experiment space, letting researchers hone in on potentially interesting aspects of a question. Thus, we believe that ExperiMan is ideally positioned to make replication more accessible, changing the culture of research to increase research transparency and quality.

2.5 Summary

Crowdsourcing provides a compelling way to complete tasks requiring many small pieces of human labor relatively quickly and for a low price. When digitized and made programmatic, it becomes crowdprogramming, and a variety of online platforms exist to facilitate crowdsourced work efficiently and ethically. In addition, quality control and statistical analysis can be automated, ensuring quality results and unbiased analysis. Thus, crowdprogramming has huge potential for running social science experiments.

This is critical because many areas of the social sciences are currently experiencing a replication crisis. A low-cost, fast, and documentable way to run experiments could potentially revolutionize fields struggling to integrate replication into the culture of research. ExperiMan offers a way to leverage crowdprogramming to rapidly run and re-run experiments, allowing researchers to create better experiments that are correct by construction and obtain robust results.

Chapter 3

Grammars of Experiments

The approach this work uses to address the requirements of a crowdsourcing platform for behavioral experiments centers around the idea of grammars of experiments. A derivation of a formal grammar is an *instance*; an instance of a grammar of experiments is an *experiment*. Given a grammar, we can derive the entire set of instances in a systematic way. The same is true for a grammar of experiments, allowing ExperiMan to explore the experiment space in a systematic way. By enabling easy exploration of the experiment space, researchers can ensure that their results are not just a product of the specific variables chosen, but true more broadly. This is essential to creating experiments that are correct by construction, meaning they contain no obvious statistical flaws, and are robust and replicable.

We will provide an overview of formal grammars and their utility in this application, then discuss how this grammar-centered approach allows us to effectively explore the experiment space, followed by a brief foray into the idea of grammar robustness.

3.1 Formal Grammars

A formal grammar can be defined as the following:

A context-free grammar G is a four-tuple $G = (N, \sigma, P, S)$ where:

- N is a finite set of nonterminal symbols
- σ is a finite set of terminal symbols disjoint from N
- P is a finite set of production rules of the form $\alpha ::= \beta$ where $\alpha \in N$, all $b \in \beta \in N \cup \sigma$, and α is not the empty string
- $S \in N$ is the designated start symbol

Formal grammars can be used to derive strings in a language using its alphabet that are valid according to the rules of its syntax. ExperiMan uses grammars to define an experiment space. Each derivation of the grammar is thus a single, unique experiment in that space.

Whenever a nonterminal is reached during a derivation, one of the productions that is indicated by its associated production rules must be chosen.

3.1.1 Example Derivation

For example, suppose we have the following grammar, expressed in Backus-Naur form:

$$\begin{aligned} S &::= A \\ A &::= bVcVd \\ V &::= x \mid y \mid z \end{aligned}$$

One derivation could be :

$$\begin{aligned} S &\rightarrow A \\ &\rightarrow bVcVd \\ &\rightarrow bxcVd \\ &\rightarrow bxccd \end{aligned}$$

When we replace the terminals with text to form an experiment grammar, we get the grammar:

$$\begin{aligned} S &::= A \\ A &::= BVCVD \\ B &::= \text{“Consider an ”} \\ V &::= \text{“ox.”} \mid \text{“ocarina.”} \mid \text{“obelisk.”} \\ C &::= \text{“How much does the ”} \\ D &::= \text{“weigh?”} \end{aligned}$$

And derivation:

$$\begin{aligned} S &\rightarrow A \\ &\rightarrow BVCVD \\ &\rightarrow \text{“Consider an ”} \ V \ \text{“How much does the ”} \ V \ \text{“weigh?”} \\ &\rightarrow \text{“Consider an ox. How much does the ”} \ V \ \text{“weigh?”} \\ &\rightarrow \text{“Consider an ox. How much does the ox weigh?”} \end{aligned}$$

So, Dalton's classic experiment can be represented as a grammar with an arbitrary number of variables and thus variants, creating an experiment space of experiment variants.

3.1.2 The Binding Problem

However, suppose we made a different selection for the second instance of V. In the first grammar, we could have a derivation:

$$\begin{aligned} S &\rightarrow A \\ &\rightarrow axbVc \\ &\rightarrow axbyc \end{aligned}$$

Which in the experiment grammar would result in the nonsensical "Consider an ox. How much does the ocarina weigh?"

This illustrates a feature particular to experiment grammars. When one experiment variable is present multiple times in an experiment text, the same value should be propagated throughout the experiment. However, there may be other variables present that should vary. Thus, when generating an experiment text, we must "bind" nonterminals to a specific value depending on which instance we are generating.

Because experiments can be represented as graphs (seen in Figure 3.1), to bind, an algorithm can traverse the graph, associating each nonterminal with a terminal value. When a nonterminal is encountered, we check to see if it has been bound; if it has, the algorithm automatically inserts the value that it has been bound to into the generating experiment text. However, this only works for acyclic graphs, and the challenges posed by cyclic graphs resulting from recursive grammars will be discussed in the next section.

3.1.3 Recursive and Infinite Grammars

The grammars we just looked at are non-recursive, but in general, grammars can be recursive and/or infinite. When trying to explore the experiment space, this causes problems. Search algorithms do not work on an unbounded space. We need to be able to count experiment instances in order to explore them systematically and reason about them statistically, so the experiment space must be finite.

Consider the below grammar and derivation:

$$
\begin{aligned}
S &::= A \\
A &::= BA \\
B &::= a \mid b \mid c
\end{aligned}
$$

$$
\begin{aligned}
S &\rightarrow A \\
&\rightarrow BA \\
&\rightarrow aBA \\
&\rightarrow aaBA \\
&\rightarrow aaaBA \\
&\rightarrow \ldots
\end{aligned}
$$

This grammar is not only recursive, but infinite, and we are faced with a problem. We can bind B to a value the first time it is encountered, but what do we bind A to? Furthermore, if a grammar is infinite, then so is the experiment space, which means it is difficult to envision how it can be systematically explored. We combat this by introducing a depth parameter, k, that expands the grammar to a bounded depth from the start symbol.

In order to ensure binding consistency, we also utilize Variables. A Variable is an object (or Expression) within ExperiMan grammars that has different options available to it, but that always binds to the same value. Other ExperiMan Expressions include:

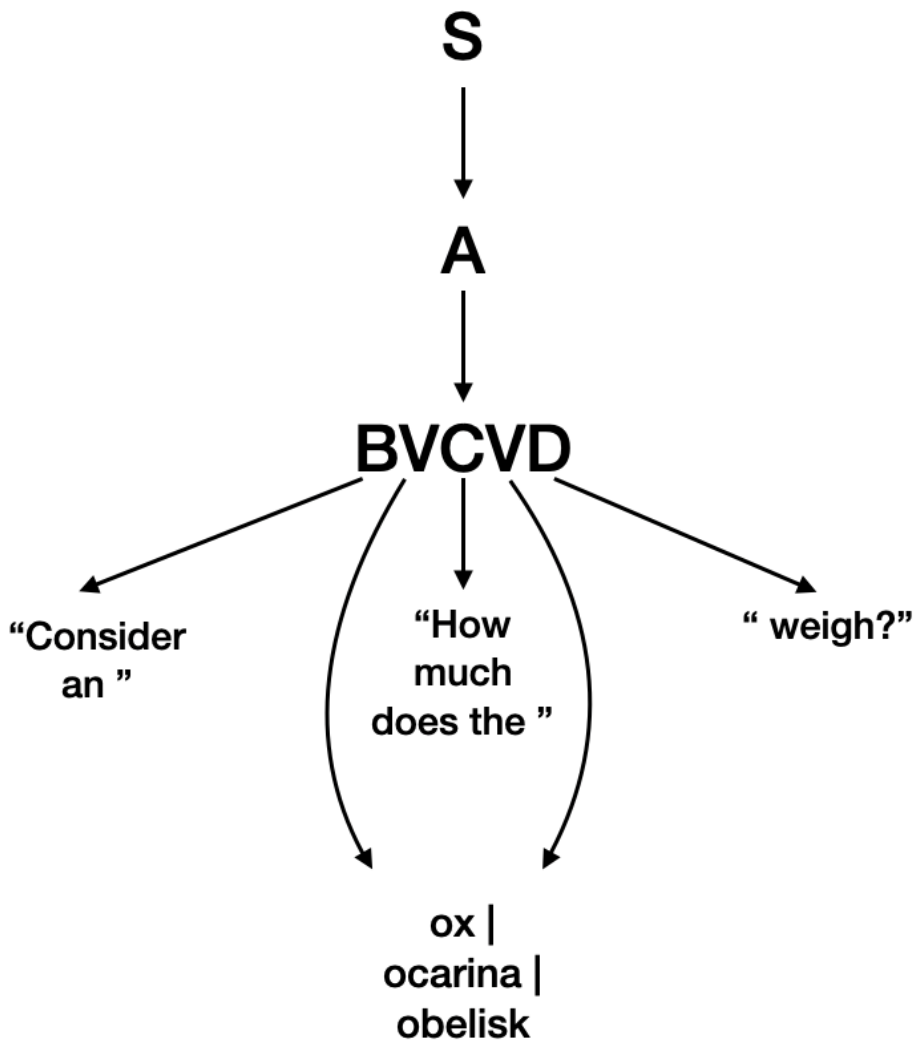

Figure 3.1: A grammar as a graph

- References: References are essentially nonterminals. They contain a Name that is used to map to other Expressions within the grammar. Unlike Variables, they can bind to any option associated with it regardless of previous occurrences.
- Terminals: a Terminal is a piece of text. Together, they form the set σ.
- Choices: a Choice is a list of Expressions that a Reference could map to (for example, $a \mid b \mid c$ in the example above). A Reference that maps to a Choice will map to one of the options.
- Sequences: a Sequence is a series of Expressions joined together as a single clause (for example, BA in the example above).
- OptionProductions: OptionProductions indicate that the Expression contained within is one selection option for a question.

```
type Expression =
    | Ref of nt: Name
    | Terminal of literal: string
    | Choice of Expression list
    | Sequence of Expression list
    | Binding of nt: Name
```

Figure 3.2: The Expression type definition in ML, with OptionProductions excluded

The depth parameter, k, ensures that an infinite grammar can still be well-formed. We take a grammar and expand recursive nonterminals, renaming each new instance. A nonterminal A will be renamed to A_i, where $0 \leq i < k$. Upon reaching depth k, the generation terminates, with A_{k-1} mapping to ϵ.

Note that after expansion, the grammar is strictly a directed acyclic graph. Because we expand all self-referential nonterminals to refer to renamed versions of themselves, which eventually terminate by mapping to the empty string, we know that there cannot be loops. Thus, we have made the grammar—and the experiment space—finite, allowing us to explore it in a systemic way.

3.2 Exploring the Experiment Space

Having established a way to formalize an experiment space, we now discuss ways to explore it. Establishing an experiment space is valuable because in order to determine what aspects of a problem are of interest, an experimenter may want to systematically explore different variations of that problem. For example, a researcher could use binary search to perform a set of experiments that cuts across the experiment space and identify independent variables for investigation. Each variant

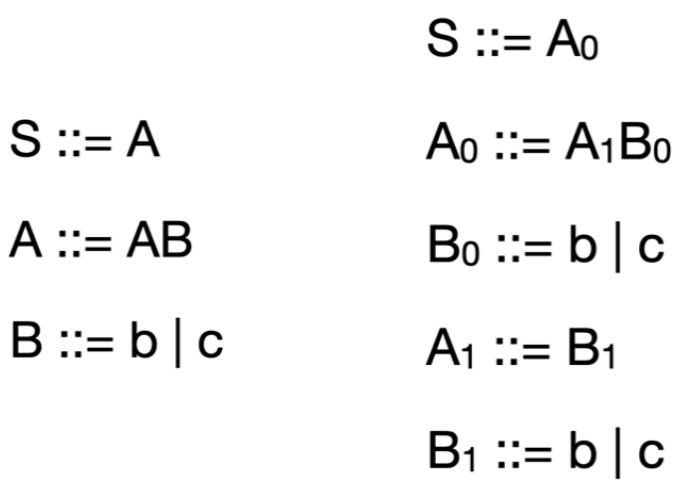

Figure 3.3: A grammar expanded to a depth of $k = 2$

changes what at least one Choice binds to. Thus, Choices correspond to independent variables in the experiment; the responses are the dependent variables.

Describing the experiment as a grammar allows for the generation of every experiment in that space, but in order for such a grammar to be useful, there needs to be a way to systematically explore experiment variations. ExperiMan uses a mixed radix number system to convert experiment instances into vectors of bases and assignments, then assign each instance a unique integer ranking.

In addition, ExperiMan offers a "dryrun" function that prints either all experiment variants or a subset of a size specified by the user to give users a more concrete idea of what the experiment space looks like.

3.2.1 Mixed Radix Number Systems

Mixed radix numeral systems are positional numeral systems where the numeral base can be different for each position. A typical example is that of time; when considering a particular time in hours (say, 2 days, 1 hour, 5 minutes, and 30 seconds), the bases would be [7, 24, 60; 60]. The overall system would be written as:

[2, 1, 5; 30]

[7, 24, 60; 60]

Each base is an integer multiple of the previous unit, the number of the previous unit it takes to make up one of the current unit. The semicolon indicates the radix point, or the point at which the numbers switch from integers to decimals because the units to the right of the unit under consideration (here hours) are smaller than that unit.

In the context of experiment grammars, each Reference to a nonterminal on the right side of the grammar is its own base. This is because each Reference to a nonterminal could be assigned

Algorithm 3.1 The base generation algorithm

```
1: procedure GENERATEBASES(expression, grammar, seenExpressions, soFar)
2:     if expression is a Reference then
3:         soFar ← the result of calling generateBases on what expression maps to in grammar
4:     else if expression is a Sequence or Choice then
5:         soFar ← the combined result of calling generateBases on all expressions in expression
6:     else if expression is a Terminal or Function then
7:         Ignore because there are no bases associated with them
8:     else if expression is a Binding then
9:         if !seenExpressions contains expression then
10:            soFar ← generateBases(expression, grammar, seenExpressions + expression, soFar)
11:        else
12:            Ignore because already bound
13:        end if
14:    end if
15: end procedure
16: return soFar
```

to a different value. However, this is not true of Variables, which are not assigned a base for any occurrence after the first. For a Reference to a nonterminal that maps to a Choice, the base is the number of options in that Choice. For a Reference to a nonterminal that maps to a single Terminal or Sequence, the base is 1.

3.2.2 Ranking and Unranking

Once we can generate a base array for an experiment grammar, we have an easy way to generate variations, as we can pick a number in the range $[0 \dots b_{i-1}]$ for each base b_i. This gives us an assignment array of the same length as the base array. For each assignment a at index i in the assignment array, our algorithm will pick the $a - th$ choice for the nonterminal associated with the base at index i in the base array. Because the choice will not necessarily map entirely to Terminals, the process is recursive in the same depth-first manner as the initial base generation.

Given the base array, we can generate the number of variations in an experiment space by multiplying the bases together. Given a ranking array, we can map it to an integer between 0 and the size of the experiment space (minus 1). Because choosing certain bases may render the assignment of another base irrelevant (for example, if Figure 3.3 allowed A_0 to also map to the empty string and we selected that derivation, the assignment to the base for B_0 would not matter), there may be duplicate variants. That is, the relationship between variants and integers is one-to-one, but not onto. We can calculate the integer rank of a variant with Algorithm 3.2.

This relationship is bijective; given an integer, we can generate the original assignment through Algorithm 3.3.

Once we have a way to easily map experiment instances to integers and back again, we can systematically explore the experiment space. Experimenters can fix a certain variable and change others, conduct a binary search through the space, or simply iterate through variations. When cre-

Algorithm 3.2 The ranking algorithm

1: **procedure** RANK($values, bases$)
2: $accumulator \leftarrow 0$
3: **for** each *value* and *index* in *values* **do**
4: $accumulator \leftarrow accumulator + value \cdot$the product of *bases* from $i + 1$ to the end of the array
5: **end for**
6: **end procedure**
7: **return** *accumulator*

Algorithm 3.3 The unranking algorithm

1: **procedure** UNRANK($variant, bases$)
2: $assignment \leftarrow []$
3: **for** each *base* and *index* in *bases* **do**
4: append $variant \div$the product of *bases* from $i + 1$ to the end of the array $\mod base$ to *assignment*
5: **end for**
6: **end procedure**
7: **return** *assignment*

ating an experiment with ExperiMan, they can specify an instance or allow ExperiMan to randomly choose an instance to post.

3.3 Robustness

ExperiMan's goal is to facilitate the creation of experiments that are correct by construction, meaning they contain no obvious statistical flaws. In doing so, we enable the creation of robust experiments, i.e. experiments that will replicate and show consistent results. By enabling easy exploration of the experiment space, researchers can ensure that their results are not just a product of the specific variables chosen, but true more broadly.

This not only requires that the experiment design be correct, but also that the answers themselves be high quality. As discussed previously, crowdsourcing platforms are vulnerable to bad actors who answer randomly. To counter this, ExperiMan deploys a variety of quality assurance mechanisms to try and filter out random responses. These mechanisms will be discussed in the next chapter.

3.4 Summary

In this section, we have introduced the idea of a grammar of experiments. By combining formal grammar definitions with rules that expand grammars to ensure a directed acyclic graph structure, we can ensure that grammars and the experiment space they represent are finite. This allows us to generate a mixed-radix representation of a grammar, then use simple ranking and unranking algorithms to map base representations to integers and back. This system helps ensure that grammars are correct by construction.

Chapter 4

Combatting Threats to Validity

In this chapter, we discuss factors that present a threat to the validity of ExperiMan experiments, and the existing features and future work intended to combat them.

4.1 Overview

The replicability crisis was precipitated by the revelation that questionable research practices (QRPs) are widespread in the social sciences, indicating the need for a system that is able to address it (44). ExperiMan is designed to automatically address QRPs, including issues of sample size and statistical manipulation, among others. In addition, given that online crowdsourcing systems can harbor bots and lazy workers, it must be able to ensure that results are of high quality.

4.2 Questionable Research Practices

Questionable research practices are widespread in the social sciences. A study asking researchers to anonymously self-admit to various QRPs found extremely high rates of self-admission for certain QRPs, peaking at 78% for "failing to report all dependent measures." As the self-admission rates are believed to be too low (since many researchers likely would not admit to QRPs, even anonymously), the authors derived prevalence estimates from the self-admissions rates. These prevalence estimates approach 100% for QRPs relating to selective reporting of studies and dependent measures and collecting more data or excluding data after examining results (27). ExperiMan is designed to preclude the possibility of engaging in QRPs, ensuring that experiments are correct by construction.

4.2.1 Sample Size

In order to ensure that sample sizes are sufficient, ExperiMan's foundational system, AutoMan, begins by posting the minimum number of tasks to ensure the desired confidence level for a response to the single question if all answers agree, with measures to control for accidental agreement. If initial results have not reached the specified confidence level, it spawns additional tasks. If the user's budget

is reached, AutoMan returns the current answer and confidence level (10). For estimations, AutoMan uses the basic bootstrap to estimate the target statistic, then the percentile method to calculate the confidence interval after a given number of trials (9). For all question types, to correct for multiple comparisons and avoid early termination bias (a common QRP), it applies the Bonferroni correction, which raises the test's confidence threshold every time the confidence is calculated (10). Thus, we are able to achieve statistically valid results with minimal time and monetary cost.

ExperiMan will allow for two types of studies: pilot studies and hypothesis tests. In the pilot study, intended for experiment space probing, researchers specify a sample size, and the study runs until the sample size is reached. The researcher can then analyze their results to determine possible future directions of study. In a future iteration, ExperiMan will also perform automated statistical tests on the results to identify independent variables of interest.

In the hypothesis test study, which we leave as future work, ExperiMan will take a hypothesis (a relationship between two questions in a given survey) specified by the user and apply similar statistical methods to reach a conclusion about whether or not to reject it.

4.2.2 P-Hacking

By automating the process of statistical analysis, ExperiMan discourages the use of p-hacking techniques. While researchers can still perform independent statistical tests on their data, the transparency of ExperiMan data and replicability of its experiment-programs would make questionable behavior nearly impossible, as experiments can just be re-run and the results examined. This follows the practices of top statistical journals (including the Journal of the American Statistical Association), which require authors to submit code and data to maximize reproducibility (28). If desired, journal editors could replicate experiments themselves by just re-running the submitted program.

4.2.3 Selective Reporting

ExperiMan automatically reports all responses it receives, working through its design to ensure that responses are quality; these mechanisms will be discussed in the next section. It would require outright deception at a level not often seen to exclude data points without justification (27). Furthermore, because experiments are easy to run and replicate, it is easy to publish negative results without the stigma associated with "failed" in-person experiments. As with p-hacking, the ease of replicability of ExperiMan experiment-programs would make data exclusion that alters reported results easy to uncover.

4.3 Answer Quality

As discussed in Chapter 2, crowdsourcing faces issues of response quality caused by lazy workers and bots. This section describes implemented measures and future work to ensure response quality.

4.3.1 Randomizing Order

As previously discussed, AutoMan supports random checkbox filling to avoid accidental accord by lazy workers (10). ExperiMan expands on this by leveraging JavaScript and HTML in the MTurk adapter to randomize the order of questions and question options, which helps control for workers randomly checking boxes or systematically selecting the same option for every question. When creating an experiment, Surveys can be defined within Surveys to create question blocks and ensure a macro level of order in an experiment.

4.3.2 Mandatory Responses

To prevent workers from submitting empty surveys, questions are marked as required in the DOM by default. If a worker attempts to submit without answering a question, the submission is blocked and a notification stating that an answer must be inputted points to the first unanswered question.

4.3.3 Targeting Subjects

One major criticism of crowdsourcing websites, but especially of MTurk, is that its demographics are not representative of any given country, and that it is difficult to target a specific demographic (16). Services like Prolific claim to offer demographics targeting services and offer representative samples, but utilizing them increases costs significantly (37). To obtain the most robust experiment results, ExperiMan would ideally offer representative results at low cost.

As discussed in Chapter 2, Google Ads offers the potential for targeting extremely specific demographic groups. To recapitulate, the four factors identified by Mason and Suri as key to a successful crowdsourcing system are subject pool size, subject pool diversity, low cost, and high speed (25). Google Ads fulfills all of these with billions of users from diverse demographics who can be targeted to "reach users with specialized expertise that is not typically available through existing labor marketplaces" (25). Users can be targeted by location, age, gender, and device type. Furthermore, ads can be shown to users based on interest in related topics (23). The cost of running ads that point to surveys can be lower than posting tasks on existing services. Furthermore, targeted, volunteer, intrinsically motivated workers can be faster than paid, non-expert workers (25).

Work done by Karmen Liang and Max Stein implemented a Google Ads adapter for AutoMan. Integrating the adapter into ExperiMan is left for future work. Because of the structure of AutoMan, the only modifications required will be to the new adapter.

4.3.4 Filtering Out Noise

While quality control mechanisms help mitigate the impacts of random responses, the incorporation of surveys introduces an opportunity to actually filter out responses suspected of being the result of lazy or malicious respondents. A survey has a given amount of entropy, which roughly corresponds to the complexity of the survey. SurveyMan (the survey tool described in Chapter 2) describes a calculation of the entropy of a given response to the survey. Using the bootstrap method on identical subsets of survey questions, it defines a one-sided confidence interval (provided by the

user and defaulting to 95%). Responses that are captured within that threshold are classified as "bad actors" (47). Implementing this approach and automatically excluding results that exceed the threshold would allow for the exclusion of bad responses, but not fall into the QRP trap because the exclusion of results is automatic, based on a preset threshold, and does not stem from human examination of the results. We plan to implement this in future work.

4.4 Summary

In this chapter, we have examined current and planned methods to combat threats to crowdsourced experiment validity. ExperiMan can effectively combat QRPs by automating statistical processes and ensuring replicability, and furthermore can apply quality control measures to provide high-quality answers to both individual questions and surveys. It is trivial to re-run an experiment, questions are mandatory by default, and question and option order are randomized by default, automatically precluding the use of QRPs that invalidate results.

Chapter 5

Implementation

In this chapter, we discuss the broad aspects of the ExperiMan implementation. ExperiMan is open-source and available at https://automan-lang.github.io/.

5.1 Overview

ExperiMan is implemented as domain-specific language embedded in Scala. Users create the grammar for their questions (if desired) using either the forthcoming Mad Libs-style interface, which is suitable for simpler grammars, or by manually encoding it, which allows for fuller expressiveness. Grammar objects are then passed into question functions. ExperiMan computes the reward and time-out, schedules the specified number of tasks (defaulting to 30 for a survey if not otherwise specified), and marshals the tasks to the backend. The experiments described in this paper were run on MTurk, but ExperiMan can be extended to different crowdsourcing platforms (10).

5.2 Classes and Functions

AutoMan is implemented with a core and various adapters. The core contains the question superclasses, as well as much of the scheduling and logging functionality. MTurk-specific functionality is delegated to the MTurk adapter, and adapters could be implemented for other crowdsourcing services.

5.2.1 Questions

ExperiMan supports numerous question types, including multiple-choice questions where one or an arbitrary number of answers are correct, free-text entry questions, and questions that ask the worker to estimate a value. It supports question versions with static question definitions and versions where a grammar is provided to allow for question variants. The core contains abstract class definitions for each question type, including the VariantQuestion, which serves as a wrapper that applies a grammar to any other type of question. It also has an abstract definition for the Survey class, which

allows for multiple questions—either VariantQuestions or non-grammar questions—to be asked in a single form.

When a question is posted, MTurk Questions are converted to XML format and passed to the MTurk backend. The HTML within leverages the MTurk HTMLQuestion structure and standard HTML forms. When all the HITs have been completed, answers are aggregated at the survey level and compiled into a CSV document, allowing the user to manually or programmatically analyze the data.

5.2.2 Adapters

Adapters are part of the original AutoMan implementation. AutoMan Adapters can be written for an arbitrary crowdsourcing platform. Adapters provide implementation of the various question types, as well as any methods required by the crowdsourcing platform. For example, the MTurk adapter includes functionality to create worker qualifications, get assignments for HITs, and perform other tasks that require interfacing with the Amazon Web Services SDK. This abstracts away the API calls and enables the requester to construct questions as usual with the DSL without worrying about the specific details of the crowdsourcing platform they are using.

Adapters also implement policies, e.g. for the minimum number of tasks to spawn. For individual questions in MTurk, this defaults to 12, and is increased based on the number of additional answers are necessary to achieve the desired confidence level. (The default of 12 was selected because 12 observations is the threshold at which the width of confidence intervals decreases less rapidly (49).) For surveys without hypothesis testing, the sample size is specified by the user. The aforementioned hypothesis specification that would allow ExperiMan to take advantage of its integrated bootstrap methods to generate more surveys as necessary is left as future work.

5.2.3 Policies

The core defines timeout, price, and aggregation policies for use in the adapters. Timeout policies govern how long workers have to complete tasks and how long tasks are posted for. Price policies determine how much workers are paid for completing a task. Aggregation policies define how answers are selected and returned to the requester. The timeout and price policies also guide how time allotted and worker payment are adjusted if a task times out without being completed. The default policies double time allotted and calculate a new wage based on the likelihood of a task being completed, which is estimated by modeling tasks as independent Bernoulli trials (10).

5.3 DSL

ExperiMan expands on AutoMan's DSL with the addition of GrammarQuestion and Survey constructors. The DSL abstracts away the complications of the crowdsourcing platform and presents crowdsourcing tasks as function calls (10). When using ExperiMan with MTurk, the user creates an implicit instance of an MTurk adapter with their Amazon credentials and whether or not to run in the MTurk sandbox (often used for testing HITs before posting for workers). After constructing their

```
def which_survey(): SurveyOutcome = surveyGrammar(
  budget = 12.00,
  questions = List(
    radioG(
      grammar = lindaGrammar,
      question = lindaQuestionProduction,
      depth = 2,
      variant = 5625
    )
  ),
  initial_worker_timeout_in_s = 30,
  minimum_spawn_policy = UserDefinableSpawnPolicy(200),
  text = "Quick survey"
)
```

Figure 5.1: The DSL definition of the Linda Problem. *lindaGrammar* and *lindaQuestionProduction* are previously defined. *radioG* is a DSL function that creates a radio button question with a Grammar; the depth and variant are also defined for each question. The budget, worker timeout, and number of tasks to spawn are defined at the survey level. *text* becomes the title of the HIT on MTurk.

Grammar(s), the user creates an instance of the appropriate Outcome with the DSL constructors. In addition to the standard budget and policy parameters, these constructors also take Grammar objects, an instance of the corresponding QuestionProduction, and the integer variant to use when constructing the question text. Finally, they call the implicit adapter with the *automan*() method and pattern match on the outcome to obtain the results, which are printed to a CSV file.

ExperiMan allows for the construction of Surveys within Surveys, so users can create "blocks" of questions that can have a random order. Within those blocks, question and question option order are randomized.

5.4 Parser

Future work is planned implement a parser that will allow researchers to create experiments with Mad Libs-style text entry. With this parser, our mini-grammar:

$$
\begin{aligned}
S &::= A \\
A &::= BVCVD \\
B &::= \text{“Consider an ”} \\
V &::= \text{“ox.” | “ocarina.” | “obelisk.”} \\
C &::= \text{“How much does the ”} \\
D &::= \text{“weigh?”}
\end{aligned}
$$

would be encoded as:

“Consider an OBJECT. How much does the OBJECT weigh?”
OBJECT = “ox”, “ocarina”, “obelisk”

Although the current DSL is relatively simple and aimed to be accessible for researchers with no computer science background, adding such an input method will facilitate the creation of simple experiments for researchers in an instantly familiar form.

5.5 Summary

In this chapter, we have explained the implementation of ExperiMan. It extends AutoMan by incorporating Surveys and the ability to encode questions that can have multiple variations based on a Grammar object. ExperiMan's DSL is intended to be easy to use for researchers with little to no programming experience, but future work will implement a Mad Libs-style parser to make ExperiMan even more accessible.

Chapter 6

Evaluation

In this chapter, we outline the criteria for ExperiMan to be useful in running behavioral experiments, detail two pilot studies, and discuss their results.

6.1 Overview

In order for our language to be effective in behavioral experiments, we need to show that is is:

1. Expressive: researchers should be able to encode real experiments
2. Valid: experiments should generate valid data
3. Fast: experiments should be completed quickly
4. Accurate: the system chooses the smallest sample size necessary to disprove the null hypothesis in hypothesis tests
5. Cost-effective: the system spends no more money than necessary

These five factors are all necessary to show that our system can be at least as effective as the current standard, in-person experiments. To evaluate **1**, **2**, and **3**, we created two pilot experiments. One is a replication of the Linda Problem, a classic psychology experiment that has been subject to numerous replication studies, including (26). The other is an adaptation of the Massachusetts Institute of Technology Media Lab's Moral Machine tool. To formally evaluate **4** and **5**, work is ongoing to integrate a bootstrap method that will automatically calculate a minimal sample size to sufficiently reject the null hypothesis with the desired level of confidence, thus also minimizing cost.

6.2 Pilot Studies

As previously mentioned, MTurk currently supports fixed-sample-size studies, and support for adaptive-sample-size hypothesis tests is planned as future work. To show ExperiMan's range and potential for experiment space exploration, we ran two fixed-size surveys as pilot experiments.

6.2.1 The Linda Problem

The Linda Problem was designed by psychologists Amos Tversky and Daniel Kahneman to test the conjunction fallacy. The probability of a conjunction ($P(A \cap B)$) cannot be greater than the probabilities of A or B (as $P(A) \cdot P(B) \leq P(A)$), but the fallacy hypothesizes that people use intuition, not logic, to make judgements about probabilities. Thus, in the initial experiment, the majority of respondents chose the second option despite the fact that the probability of the two events together must be less than the likelihood of the first (48).

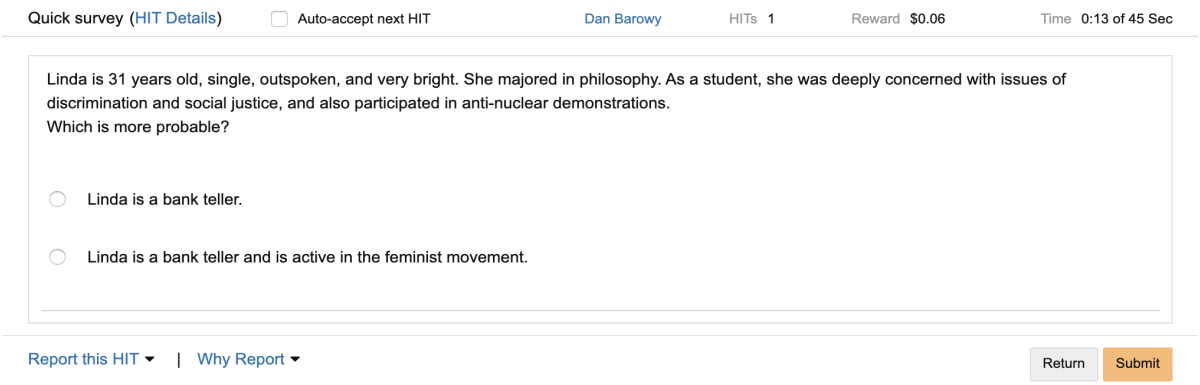

Figure 6.1: The classic Linda Problem posted as a HIT on MTurk

Table 6.1 shows the proportion of respondents who display the conjunction fallacy (indicated as $B \cap F$) in both our pilot study, run on Mechanical Turk ($n = 200$), and Tversky and Kahneman's original 1983 study, run on 142 undergraduates at the University of British Columbia (48).

Group	**n**	**B ∩ F**
MTurk Respondents	200	0.78
T&K Undergrads	142	0.85

Table 6.1: Proportion of respondents exhibiting the conjunction fallacy

This pilot was completed in under 66 minutes. (We believe this to be an outlier, as it is significantly longer than a previous version of the study, which was completed in 21 minutes, and longer than our other pilot studies. Note, however, that it is still significantly faster than the experiments described in (33), and significantly faster than running it in person would be.) Participants were allocated 30 seconds (with a grace period of 15 seconds allocated by default in case workers have slow machines or connectivity issues) and paid $0.06 for the task (determined automatically by MTurk's default minimum wage). Tasks did not timeout and thus ExperiMan did not have to activate its time and wage adjustment policies, so the experiment cost $12.00. Because this was a pilot study, it ran until completion rather than attempting to fulfill a hypothesis with as few assignments (and thus payments) as possible, but even so, compared to the cost and time of recruiting 200 participants for a traditional in-person study, we believe ExperiMan performs favorably on criteria **3** and **5** above. With regards to criteria **1** above, empirically, we can report that ExperiMan is expressive. The

Linda Problem was easy to transform into an ExperiMan grammar, requiring just 79 lines of code (46 of which were to list terminals) to produce a grammar that offered 75,000 variations on name, age, major, issue, demonstration, job, and movement, though it could be expanded to an arbitrary number of variables. To evaluate criteria **2**, we compare ExperiMan's results to an experiment from the period of time called into question by the replication crisis. (Although the study was run before the replication crisis cast doubt on established methods, Tversky and Kahneman have not been accused of QRPs in this experiment.) ExperiMan's results are valid because they are properly controlled and have a large sample size. Using a proportion test with a null hypothesis that the two proportions are equal, we find that we cannot reject the null hypothesis ($p = 0.09396$). Thus, we conclude that the original experiment does replicate. The validity of the conjunction fallacy is backed up by several other replication studies, including (26), supporting the robustness of our results.

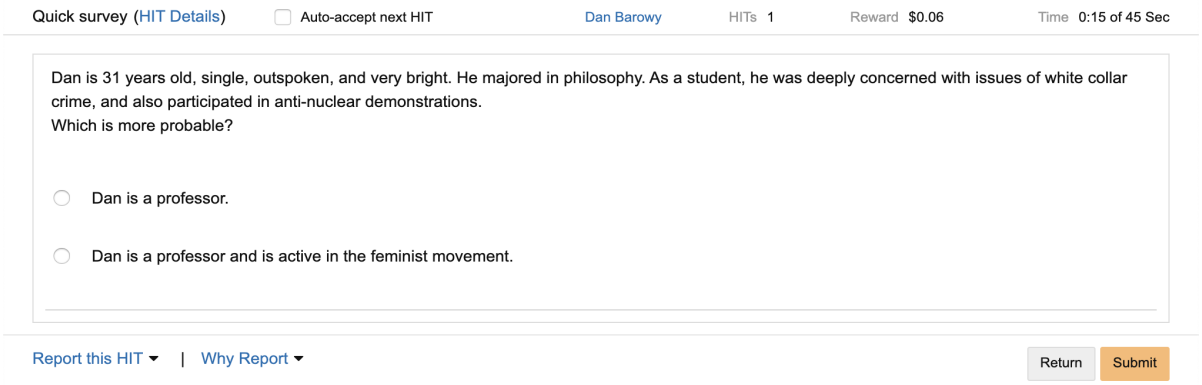

Figure 6.2: A variant of the Linda Problem posted as a HIT on MTurk

We also ran a variant of the Linda Problem featuring Dan the professor, seen in Figure 6.2. Because we encoded the Linda Problem in ExperiMan as a Grammar, changing the variant required changing just one number in a single line of code, and changing the number of tasks spawned involved just changing the integer in the spawn policy.

Both tasks were posted on a weekday at 12:00pm Eastern time. We posted 100 HITs for the Dan variant and observed almost opposite results, as seen in Table 6.2.

Group	**n**	**B ∩ F**
Linda Respondents	200	0.78
Dan Respondents	100	0.15

Table 6.2: Proportion of respondents exhibiting the conjunction fallacy for Linda versus Dan

The Dan variant changed name (and thus perceived gender), issue, and job. We did not randomize options, but we do not believe that bad actor involvement was significant; when comparing a variant of the Linda pilot where we did not randomize options to the results with randomization, we observed that the proportion of respondents displaying the conjunction fallacy was slightly higher, but that the difference was not statistically significant. The Dan experiment was completed in under

16 minutes and cost just $6.00. For that small investment, it points to questions of feminism, gender, activism, and profession as possibly fruitful avenues of further exploration, and further strengthens the conjunction fallacy hypothesis by indicating some of the factors that contribute to the heuristics people use when considering these problems. Running expressive experiments to explore the experiment space is trivial in ExperiMan and allows for the discovery of factors that a researcher may not have otherwise considered exploring.

6.2.2 Moral Machine

The Moral Machine is an "online experimental platform designed to explore the moral dilemmas faced by autonomous vehicles" created by the MIT Media Lab. It presents participants with hypothetical scenarios about what an autonomous vehicle with sudden brake failure should do in a given situation where continuing straight would kill one group of people and swerving would kill another group of people. Participants are forced to make decisions between killing passengers or pedestrians (crossing with the light, against the light, or at a crossing with no light), or sometimes different groups of pedestrians. Human characters include adult women and men, elderly women and men, girls, boys, babies, pregnant women, large women and men, female and male athletes, female and male executives, homeless people, and criminals (the latter two are not distinguished by gender, but appear to be male in the images). Some scenarios also have dogs and cats. Displaying the unique ability of online experiments to reach huge numbers of participants across the world, they obtained 40 million scenario responses in ten languages across 233 countries and territories (7).

From this, they examined global and regional preferences across nine areas, including sparing more versus fewer people, sparing passengers versus pedestrians, and preference for action versus inaction. They found that globally, humans tend to decide to spare pedestrians over passengers, the higher-status over the lower-status, the young over the elderly, and the many over the few. There is also a preference for sparing females over males, though not as strong as the preference for the fit over the large, and a slight preference for inaction (i.e. not swerving) over action. They also divided the globe into "Western," "Eastern," and "Southern" geocultural regions (roughly corresponding to Europe and North America, Asia, and the Southern Hemisphere, respectively) and analyzed regional preferences. Among other things, they found that participants in the "Western" region disproportionately prefer inaction and participants in the "Eastern" region disproportionately spare pedestrians and place less of an emphasis on saving the higher-status. Participants in the "Southern" region hugely emphasize sparing the young, females, and the higher-status (7). Given that the two primary countries of origin of MTurk workers are the United States (categorized as "Western" by the Moral Machine) and India (placed in the "Eastern" group), this offered an opportunity to present several scenarios to MTurk workers via ExperiMan and compare our results to those of the Moral Machine (16). Due to the massive sample size of the Moral Machine, we believe that they offer a solid baseline of comparison.

We posted four batches of HITs. HITs consisted of a three-question survey and took one of two forms. Both types had the same first scenario (seen in Figure 6.3), but half had the scenario in Figure 6.4, and half had the scenario in Figure 6.5. The third question asked the worker to provide

their time zone in a free-text box in order to approximate location. One batch of 50 HITs of each survey type was posted at 4:30am Eastern Time, and one batch of each type at 1:30pm Eastern time on the same weekday. The earlier batches were intended to target workers in India, while the later batches aimed to target workers in the United States, at midday in their respective timezones. Despite the separate batches, concerns about worker overlap were assuaged when we found the HITs of the later batches on the MTurk worker site immediately after posting and were unable to open them, indicating that all had been begun or enqueued by workers. Because of the batches' relatively small size, it is unlikely that the same worker was able to complete one of each batch. Workers were paid $0.15 cents for taking the survey; the entire pilot cost $30.00. Batches took between 17 and 24 minutes to be completed.

Scenario pictures were constructed on the Moral Machine website. Since Moral Machine scenarios follow a set pattern, we constructed a scenario Grammar that was used for each question. We presented workers with both the scenario picture and a textual description of each choice.

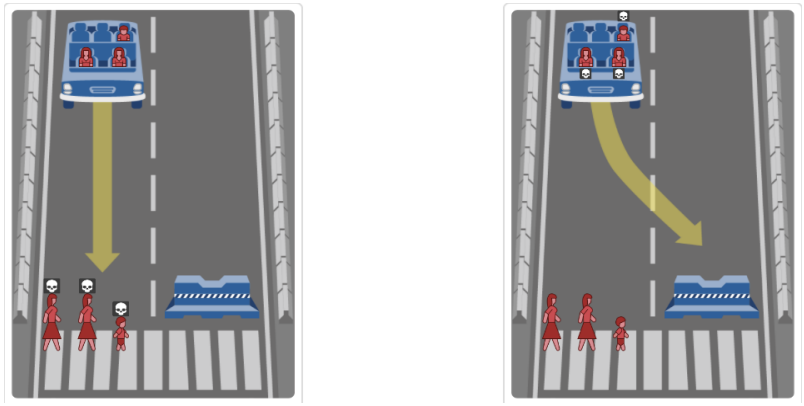

Figure 6.3: Question 1 - Should the car go straight and kill the pedestrians, or swerve and kill the passengers?

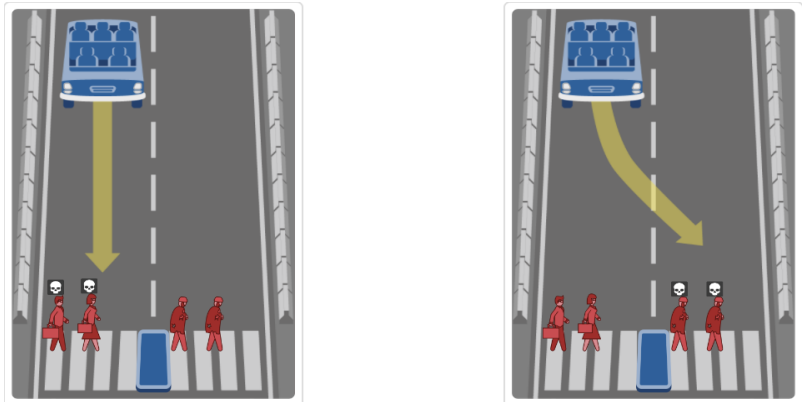

Figure 6.4: Question 2a - Should the car go straight and kill the male and female executives, or swerve and kill the homeless people?

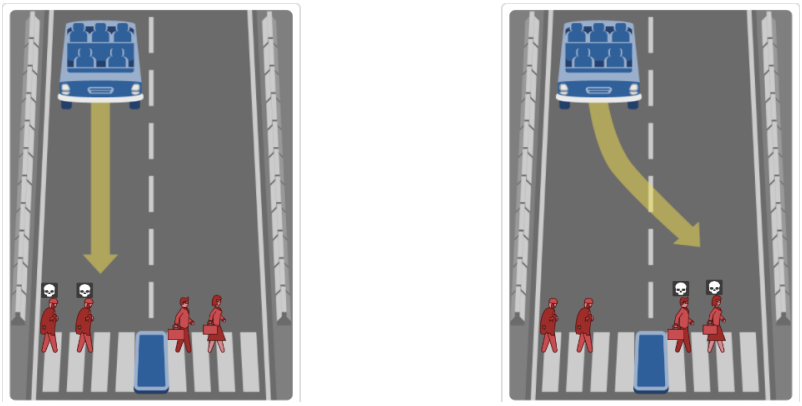

Figure 6.5: Question 2b - Should the car go straight and kill the homeless people, or swerve and kill the male and female executives?

Although we timed the early surveys to target workers in India, we saw a high proportion of European and American workers; 19 of the 100 respondents in the first round of surveys identified their timezone as India Standard Time (IST). Six workers from the second round also indicated they were located in IST. IST was the only timezone identified that belonged to the Moral Machine "Eastern" designation. Workers that left the timezone section blank (as it was not marked as a required question to respect worker privacy) or indicated a nonexistent timezone were excluded from the data (there were 11 such workers across the four batches), as was the one worker that indicated a location in Brazil, which was placed in the "Southern" Moral Machine group (7).

Group	**n**	**Continue Straight**	**Swerve**
Eastern	25	0.44	0.56
Western	163	0.42	0.58

Table 6.3: Proportion table of responses to Question 1 by Moral Machine geocultural location

Interestingly, we observe a preference for action over inaction, contrary to the Moral Machine findings. The difference in proportion between the two geocultural locations is not statistically significant. The Moral Machine paper did not report exact proportions for different factors, so we cannot compare the difference in proportion between the two, but it still indicates areas of further inquiry. Though the Moral Machine results appear to be robust (given the sample size and presence of a Nature Research reporting summary), digging into the demographics of respondents could yield dividends (7). One distinction that may be a factor is education level; most Moral Machine respondents attended college, while MTurk workers tend to have a lower education level (16; 7). Given the flexibility of ExperiMan, it would not be difficult to run additional trials to examine confounding factors. Ultimately, the goal of pilot studies is to point to interesting avenues of inquiry for further experiments, as this has done.

Although we do not observe a significant difference in choice across geocultural groups, we do see a strong preference for sparing the higher-status over the lower-status, which accords with the original findings. We applied a two-sided proportion test on the proportion of aggregated respondents who

Group	n	Continue Straight	Swerve
Eastern	11	0.45	0.55
Western	83	0.47	0.53

Table 6.4: Proportion table of responses to Question 2a by Moral Machine geocultural location

Group	n	Continue Straight	Swerve
Eastern	14	0.64	0.36
Western	80	0.63	0.37

Table 6.5: Proportion table of responses to Question 2b by Moral Machine geocultural location

chose to swerve. The null hypothesis is that the two proportions are equal, and we find that we are able to reject the null hypothesis with p = 0.03337, indicating a statistically significant difference in treatment of higher- versus lower-status figures. The result in the "Eastern" bloc is relatively unexpected, as the "Eastern" group in the Moral Machine group displayed a much lower preference for sparing those of higher status compared to other groups. However, countries in South Asia (India, Nepal, Mauritius, Thailand, and Cambodia) comprised just 20% of the "Eastern" group, so it is possible that other countries' preferences outweighed those of India in the original experiment (7). Regardless, we see that ExperiMan is able to replicate established results, offering promise for further replication studies and for novel experiments.

Overall, we again see evidence that ExperiMan accords with criteria **1**, **2**, **3**, and **5** for pilot experiments. We were able to quickly create a simple grammar to encode Moral Machine experiments and produce results that accord with established, robust results. The experiments were completed quickly, and for a reasonable price. ExperiMan's combination of expressiveness, validity, speed, and cost-effectiveness indicate its value for behavioral studies.

6.3 Summary

In this chapter, we have outlined our evaluation criteria and approach to pilot studies. We show that ExperiMan's results accord with established, replicated literature, and see its potential to obtain robust results in novel experiments. Experiments can be run quickly and at low cost. When considering the resources required to recruit and pay study participants, we believe it compares extremely favorably. Furthermore, since each experiment consists simply of a file of code, they are trivial to re-run.

Encoding studies as grammars of experiments allows for the rapid exploration of the experiment space. As the two variations of the Linda Problem shows, difference in responses can be identified across multiple independent variables. One could envision a future program that identifies such inconsistencies and automatically runs experiments to pin down how each contributes.

Chapter 7

Conclusion and Future Work

In this thesis, we have introduced ExperiMan, a crowdprogramming tool for behavioral experiments that are correct by construction. ExperiMan leverages the idea of grammars of experiments to offer a novel approach to ensuring robust, replicable results.

ExperiMan builds on the existing AutoMan crowdprogramming tool, adding support for surveys and the aforementioned concept of grammars of experiments. With a few simple lines of code, experimenters can build and run an experiment on a crowdsourcing platform. ExperiMan allows for fast, cheap pilot studies, permitting experimenters to rapidly explore the experiment space with high-quality results.

ExperiMan offers great potential to make replicable, crowdsourced experiments more accessible and thus address the replicability crisis in the social sciences. Existing quality control mechanisms are shown to produce robust results, with additional features left as future work. Our pilot experiments show that we can replicate established results. We were able to replicate the conjunction fallacy with statistically significant results, as well as aspects of the Moral Machine experiment; we expect that other observations would replicate given a larger sample size. These results indicate ExperiMan's future utility in running novel behavioral experiments. They also display its usefulness in identifying areas of interest for further study, as study variants can be run by just changing a single variable.

Over the past few years, increased attention has been paid to questionable research practices in the social sciences. These practices have contributed to a replication crisis calling into question studies previously thought to be robust. In a research culture that does not value less-flashy replication studies, work must be done to incorporate replication into the experiment cycle. Programmatically crowdsourcing studies offers a low-cost, rapid way to construct experiments where replication is as simple as re-running a program. ExperiMan is the front line in the movement towards open, replicable science.

Bibliography

[1] 1 vs. 100, December 1 2006.

[2] Amazon. Amazon Mechanical Turk pricing. `https://requester.mturk.com/pricing`, 2019 (accessed December 4, 2019).

[3] Amazon. Amazon Mechanical Turk: Access a global, on-demand, 24x7 workforce. `https://www.mturk.com/`, 2019 (accessed November 24, 2019).

[4] Andrewes, W. J. *The quest for longitude : the proceedings of the Longitude Symposium, Harvard University, Cambridge, Massachusetts, November 4-6, 1993.* Collection of Historical Scientific Instruments, Harvard University, Cambridge, Mass., 1996.

[5] Appen. Crowd management. `https://appen.com/solutions/crowd-management/`, 2020 (accessed May 5, 2020).

[6] Appen. Technology. `https://appen.com/industries/technology/`, 2020 (accessed May 5, 2020).

[7] Awad, E., Dsouza, S., Kim, R., Schulz, J., Henrich, J., Shariff, A., Bonnefon, J.-F., and Rahwan, I. The moral machine experiment. *Nature 563*, 7729 (2018), 59.

[8] Bakshy, E., Dworkin, L., Karrer, B., Kashin, K., Letham, B., Murthy, A., and Singh, S. AE: A domain-agnostic platform for adaptive experimentation. *32nd Conference on Neural Information Processing Systems (NIPS 2018)* (2018).

[9] Barowy, D. W., Berger, E., Goldstein, D., and Suri, S. VoxPL: Programming with the wisdom of the crowd. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems 2017* (2017), 2347–2358.

[10] Barowy, D. W., Curtsinger, C., Berger, E., and McGregor, A. AutoMan: A platform for integrating human-based and digital computation. *Communications Of The ACM 59*, 6 (2016), 102–109.

[11] Bernstein, M., Brandt, J., Miller, R., and Karger, D. Crowds in two seconds: enabling realtime crowd-powered interfaces. *Proceedings of the 24th annual ACM symposium on User interface software and technology 2011* (2011), 33–42.

[12] Brown, B., and Lampinen, A. Acknowledge crowdworkers in crowdwork research. *Communications of the ACM 59*, 11 (2016), 8–9.

[13] Camerer, C. F., Dreber, A., Forsell, E., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Almenberg, J., Altmejd, A., Chan, T., Heikensten, E., Holzmeister, F., Imai, T., Isaksson, S., Nave, G., Pfeiffer, T., Razen, M., and Wu, H. Evaluating replicability of laboratory experiments in economics. *Science 351*, 6280 (2016), 1433.

[14] Camerer, C. F., Dreber, A., Holzmeister, F., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Nave, G., Nosek, B. A., Pfeiffer, T., Altmejd, A., Buttrick, N., Chan, T., Chen, Y., Forsell, E., Gampa, A., Heikensten, E., Hummer, L., Imai, T., Isaksson, S., Manfredi, D., Rose, J., Wagenmakers, E.-J., and Wu, H. Evaluating the replicability of social science experiments in Nature and Science between 2010 and 2015. *Nature Human Behaviour 2*, 9 (2018), 637.

[15] Collaboration, O. S. Estimating the reproducibility of psychological science. *Science 349*, 6251 (2015), aac4716.

[16] Difallah, D., Filatova, E., and Ipeirotis, P. Demographics and dynamics of Mechanical Turk workers. *Proceedings of the Eleventh International Conference on Web Search and Data Mining 2018-* (2018), 135–143.

[17] Difallah, D. E., Catasta, M., Demartini, G., Ipeirotis, P., and Cudr-Mauroux, P. The dynamics of micro-task crowdsourcing: The case of Amazon MTurk. *Proceedings of the 24th International Conference on World Wide Web 2015* (2015), 617–617.

[18] Douceur, J. The sybil attack. *Peer-To-Peer Systems 2429* (2002), 251–260.

[19] Engber, D. Daryl Bem proved ESP is real, 2017.

[20] Estells-Arolas, E., and Gonzlez-Ladrn-de Guevara, F. Towards an integrated crowdsourcing definition. *Journal of Information Science 38*, 2 (2012), 189–200.

[21] Galton, F. Vox populi. *Nature 75*, 1949 (1907), 450.

[22] Google Ads. How it works. `https://ads.google.com/home/how-it-works/`, 2019 (accessed November 30, 2019).

[23] Google Ads. Targeting your ads. `https://support.google.com/google-ads/answer/1704368?hl=en`, 2020 (accessed May 5, 2020).

[24] Howe, J. Crowdsourcing: a definition. `https://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html`, 2006 (accessed December 1, 2019).

[25] Ipeirotis, P., and Gabrilovich, E. Quizz: targeted crowdsourcing with a billion (potential) users. *Proceedings of the 23rd international conference on World Wide Web* (2014), 143–154.

[26] Jarvstad, A., and Hahn, U. Source reliability and the conjunction fallacy. *Cognitive science 35*, 4 (2011), 682.

[27] John, L. K., Loewenstein, G., and Prelec, D. Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science 23*, 5 (2012), 524–532.

[28] Journal of the American Statistical Association. Instructions for authors. `https://www.tandfonline.com/action/authorSubmission?show=instructions&journalCode=uasa20`, 2020 (accessed May 5, 2020).

[29] Jun, E., Daum, M., Roesch, J., Chasins, S., Berger, E., Just, R., and Reinecke, K. Tea: A high-level language and runtime system for automating statistical analysis. *UIST '19: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (2019), 591–603.

[30] Jurca, R., and Radanovic, G. Peer truth serum: Incentives for crowdsourcing measurements and opinions. *arXiv.org* (2017).

[31] Kulkarni, A., Can, M., and Hartmann, B. Collaboratively crowdsourcing workflows with turkomatic. *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (2012), 1003–1012.

[32] Little, G., Chilton, L., Goldman, M., and Miller, R. Turkit: human computation algorithms on Mechanical Turk. *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (2010), 57–66.

[33] Mason, W., and Suri, S. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods 44*, 1 (2012), 1–23.

[34] Newman, A. I found work on an Amazon website. I made 97 cents an hour. `https://www.nytimes.com/interactive/2019/11/15/nyregion/amazon-mechanical-turk.html`, 2019 (accessed November 28, 2019).

[35] Peer, E., Brandimarte, L., Samat, S., and Acquisti, A. Beyond the turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology 70* (2017), 153–163.

[36] Prolific. Participants. `https://www.prolific.co/participants`, 2020 (accessed February 18, 2020).

[37] Prolific. Pricing. `https://www.prolific.co/#pricing`, 2020 (accessed February 18, 2020).

[38] Resnick, B. What psychology's crisis means for the future of science. `https://www.vox.com/2016/3/14/11219446/psychology-replication-crisis`, 2016 (accessed December 4, 2019).

[39] Resnick, B. More social science studies just failed to replicate. Here's why this is good. `https://www.vox.com/science-and-health/2018/8/27/17761466/`

psychology-replication-crisis-nature-social-science, 2018 (accessed December 4, 2019).

[40] Safire, W. Fat tail. https://www.nytimes.com/2009/02/08/magazine/08wwln-safire-t.html?_r=3&ref=magazine&, 2009 (accessed May 7, 2020).

[41] Samms, M. The quest for longitude. *Sea Classics 48*, 12 (2015), 5.

[42] Semuels, A. The internet is enabling a new kind of poorly paid hell. https://www.theatlantic.com/business/archive/2018/01/amazon-mechanical-turk/551192/, 2018 (accessed December 4, 2019).

[43] Seth, C., Firas, K., Adrien, T., Janos, B., Jeehyung, L., Michael, B., Andrew, L.-F., David, B., Zoran, P., and Foldit, P. Predicting protein structures with a multiplayer online game. *Nature 466*, 7307 (2010), 756.

[44] Shrout, P. E., and Rodgers, J. L. Psychology, science, and knowledge construction: Broadening perspectives from the replication crisis. *Annual Review of Psychology 69* (2018), 487–510.

[45] Simmons, J. P., Nelson, L. D., and Simonsohn, U. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science 22*, 11 (2011), 1359–1366.

[46] Steinhardt, J., Valiant, G., and Charikar, M. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (2016).

[47] Tosch, E., and Berger, E. SurveyMan: Programming and automatically debugging surveys. *ACM SIGPLAN Notices* (2014).

[48] Tversky, A., and Kahneman, D. Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review 90*, 4 (1983), 293–315.

[49] Van Belle, G. *Statistical rules of thumb*. Wiley-Interscience, New York, 2002.

[50] Waze. Harnessing real-time, crowdsourced data to improve crisis response. https://www.waze.com/ccp/casestudies/harnessing_real_time_crowdsourced_data_to_improve_crisis_response, 2019 (accessed November 25, 2019).