Behavior Research Methods (2019) 51:1782–1803
https://doi.org/10.3758/s13428-018-1155-z

A thousand studies for the price of one: Accelerating psychological science with Pushkin

Joshua K. Hartshorne[1] · Joshua R. de Leeuw[2] · Noah D. Goodman[3] · Mariela Jennings[1] · Timothy J. O'Donnell[4]

Published online: 11 February 2019

Abstract

Half of the world's population has internet access. In principle, researchers are no longer limited to subjects they can recruit into the laboratory. Any study that can be run on a computer or mobile device can be run with nearly any demographic anywhere in the world, and in large numbers. This has allowed scientists to effectively run hundreds of experiments at once. Despite their transformative power, such studies remain rare for practical reasons: the need for sophisticated software, the difficulty of recruiting so many subjects, and a lack of research paradigms that make effective use of their large amounts of data, due to such realities as that they require sophisticated software in order to run effectively. We present Pushkin: an open-source platform for designing and conducting massive experiments over the internet. Pushkin allows for a wide range of behavioral paradigms, through integration with the intuitive and flexible jsPsych experiment engine. It also addresses the basic technical challenges associated with massive, worldwide studies, including auto-scaling, extensibility, machine-assisted experimental design, multisession studies, and data security.

Keywords Online studies · Robust and reliable research · Massive online experiments · Citizen science

Although some questions psychologists care about involve comparing only two conditions to each other, most require teasing apart the contributions of many intertwined variables. In the past, this has required hundreds, if not thousands, of studies across numerous laboratories, each targeting a specific variable, population, or stimulus set.

In principle, we can now do this work many orders of magnitude more quickly. Given that half the world's population has internet access (ITU Telecommunication Development Sector, 2017), any study that can be run on a computer or mobile device can be run with nearly any demographic anywhere in the world, and in large numbers. This includes not just surveys, but studies involving grammatical judgments, reaction times, decision-making, economics games, eyetracking, priming, sentence completion, skill acquisition, and others—which is to say, most human behavioral experiments (Birnbaum, 2004; Buchanan & Smith, 1999; Germine et al., 2012; Gosling & Mason, 2015; Gosling, Sandy, John, & Potter, 2010; Haworth et al., 2007; Honing & Ladinig, 2008; Krantz, 2001; Meyerson & Tryon, 2003; Papoutsaki et al., 2016; Reips, 2002; Skitka & Sargis, 2006). Extensive research has shown that data from online studies is, if anything, of higher quality than what is typically achieved in the lab (Appendix C).

The feasibility and utility of internet experiments is amply demonstrated by the widespread adoption of Amazon Mechanical Turk (Buhrmester, Kwang, & Gosling, 2011; Mason & Suri, 2012; Paolacci, Chandler, & Ipeirotis, 2010; Stewart, Chandler, & Paolacci, 2017). For example, around one-quarter of recent cognitive psychology articles feature at least one online experiment (Stewart et al., 2017). Fully capitalizing on the promise of the internet, however, requires

✉ Joshua K. Hartshorne
hartshoj@bc.edu

[1] Department of Psychology, Boston College, Newton, MA, USA

[2] Department of Cognitive Science, Vassar College, Poughkeepsie, NY, USA

[3] Department of Psychology, Stanford University, Stanford, CA, USA

[4] Department of Linguistics, McGill University, Montreal, Quebec, Canada

finding a way to go beyond Amazon's subject pool of fewer than 20,000, mostly American and Indian, adults to the full population of three billion internet users (Buhrmester et al., 2011; ITU Telecommunication Development Sector, 2017; Paolacci et al., 2010; Stewart et al., 2015).[1]

In fact, a number of researchers have successfully leveraged the internet to conduct what are effectively hundreds of studies at once: massive online experiments that cover a wider range of demographics, a wide range of stimuli, or both (Blanchard & Lippa, 2007; Bleidorn et al., 2013; Bleidorn et al., 2016; Brysbaert, Stevens, Mandera, & Keuleers, 2016; Condon, Roney, & Revelle, 2017; Fortenbaugh et al., 2015; Gebauer et al., 2014; Germine, Duchaine, & Nakayama, 2011; Halberda, Ly, Wilmer, Naiman, & Germine, 2012; Hartshorne & Germine, 2015; Hartshorne, O'Donnell, & Tenenbaum, 2015; Hartshorne & Snedeker, 2013; Hartshorne, Tenenbaum, & Pinker, 2018a; Hauser, Young, & Cushman, 2008; Johnson, Logie, & Brockmole, 2010; Kajonius & Johnson, 2018; Keuleers, Stevens, Mandera, & Brysbaert, 2015; Killingsworth & Gilbert, 2010; Kumar, Killingsworth, & Gilovich, 2014; Lippa, 2008; Logie & Maylor, 2009; Manning & Fink, 2008; Maylor & Logie, 2010; Nosek, Banaji, & Greenwald, 2002; Peters, Reimers, & Manning, 2006; Reinecke & Gajos, 2014; Riley et al., 2016; Salganik, Dodds, & Watts, 2006; Soto, John, Gosling, & Potter, 2011; Susilo, Germine, & Duchaine, 2013).[2] Many of these studies have often prompted significant revision of theory, including overturning long-standing theoretical accounts of cognitive aging, critical periods, and aesthetic preferences (Fortenbaugh et al., 2015; Germine et al., 2011; Halberda et al., 2012; Hartshorne & Germine, 2015; Hartshorne, Tenenbaum, & Pinker, 2018b; Reinecke & Gajos, 2014).

Researchers have also used a related paradigm to process enormous amounts of data: citizen science (Dickinson, Zuckerberg, & Bonter, 2010; Greene, Kim, Seung, & the EyeWirers, 2016; Hartshorne, Bonial, & Palmer, 2013a, 2014; Kim et al., 2014; Poesio, Chamberlain, Kruschwitz, Robaldo, & Ducceschi, 2013; Simpson, Page, & De Roure, 2014). Citizen science projects recruit large numbers of volunteers to assist in scientific research (Box 1). While citizen science has been much more widely used in other fields (e.g., for categorizing galaxies or tracking bird migrations; Sullivan et al., 2014; Willett et al., 2017), some early successes have shown its potential power for psychology and neuroscience. For instance, mapping the synapses of even a single axon is an extremely time-intensive task. By recruiting over 100,000 volunteers, Kim et al. were able to map 274 retinal axons, finding that different types of bipolar cells vary in how close they synapse to starburst amacrine cell somas. This physical asymmetry in synapse location, coupled with several other properties of these neurons, provides a plausible mechanism for explaining how the mammalian brain detects motion.

Obstacles to broader adoption

Any researcher wishing to engage in massive online experiments or citizen science immediately runs into a significant obstacle: There is no ready-to-use software for implementing them. Indeed, the major online research websites—gameswithwords.org, testmybrain.org, labinthewild.org, projectimplicit.org, and eyewire.org—all use their own custom, in-house software.

The reason for this may not be immediately obvious, given that there are a number of solutions for online studies, including commercial platforms (Qualtrics, SurveyMonkey, LabVanced) and open-source software (jsPsych, PsychoJS, Ibex Farm) (Table 1). However, these systems were designed for relatively small experiments, with a few thousand subjects at most. Internet-scale studies present additional challenges with regard to addressing recruitment, reliability, a range of paradigms, and contingent design.

Recruitment is perhaps the most obvious problem. There may be over three billion people online, but they need some motivation to do your experiment. Many of the existing solutions involve paying subjects through Amazon Mechanical Turk or Qualtrics. However, the number of subjects that can be recruited through these platforms is insufficient for the kinds of studies under discussion here (Buhrmester et al., 2011; Levay, Freese, & Druckman, 2016; Paolacci et al., 2010; Stewart et al., 2015), and paying that many would in any case be prohibitively expensive. Instead, internet-scale research typically relies on gameification, personalized feedback, and other strategies to make participation intrinsically rewarding—strategies that are not generally available through existing platforms (Table 1).

The difficulties associated with *reliability* may not be immediately apparent to individuals who have limited experience running popular websites. In essence, internet-scale research has one inherent vulnerability: Websites are most likely to crash precisely when you need them most. That is, subjects tend to come in large waves (Fig. 1), and this heavy traffic can overwhelm a website and cause it to crash. Because it is during those waves of traffic that we collect almost all of our data, that is the worst possible time for the website to crash. This is one of the most difficult problems in web development. In

[1] For a creative solution for testing children via Amazon Mechanical Turk, see Scott and Schulz (2017).

[2] Here we focus on datasets originating from experiments, rather than on secondary uses of preexisting systems, such as observational studies of online gamers and social networkers or experiments conducted by websites on their users (Bainbridge, 2007; Hardy & Scanlon, 2009; Settles & Meeder, 2016; Streeter, 2015; Wilson, Gosling, & Graham, 2012). Though potentially powerful methods of research, the use of such systems can sometimes raise ethical issues, and in any case is not a realistic option for researchers who do not have their own popular online game or social network.

Table 1 Comparison of prominent software for behavioral experiments on the internet with the proposed software, based on software documentation and discussion with the developers

	Qualtrics	Google Forms	SurveyMonkey	Ibex Farm	Inquisit Web	LabVanced	Dallinger	webDMDX	PsychoJS	jsPsych	psiTurk + jsPsych	Pushkin
Free	Y	Y	N*	Y	N	N*	Y	Y	Y	Y	Y	Y
Graphical user interface	Y	Y	Y	Y	Y	Y	N	Y	Y	D	N	D
Recruitment												
Gameification	N	N	N	N	N	N	N	N	N	N	N	Y
Social media	N	N	Y	N	N	N	D	N	N	N	N	Y
Forum	N	N	N	N	N	N	N	N	N	N	N	Y
Reliability												
Auto-scaling	Y	Y	Y	?	Y	Y	Y	NA	N	N	N	Y
Real-time backups	?	?	?	N	N	N	N	N	N	N	N	Y
Flexibility												
Open-source/customizable	N	N	N	Y	N	N	Y	Y	Y	Y	Y	Y
Display video	Y	Y	N	N	Y	Y	Y	Y	N	Y	Y	Y
Present audio	Y	N	N	N	Y	Y	Y	Y	N	Y	Y	Y
Reaction times	N	N	N	Y	Y	Y	N	Y	Y	Y	Y	Y
Eyetracking	N	N	N	N	N	Y	N	N	N	N	N	Y
Record audio	N	N	N	N	N	Y	N	Y	N	N	N	Y
Multisubject interaction	N	N	N	N	N	Y	Y	N	N	N	N	Y
Multisession	Y	N	Y	N	N	Y	N	N	N	N	N	Y
Contingent design												
Randomization/ counterbalancing	L	N	L	L	L	Y	N	L	L	L	Y	Y
Contingent Qs	Y	Y	Y	N	Y	Y	N	Y	N	Y	Y	Y
OED	N	N	N	N	N	N	Y	N	N	N	N	Y
Active learning	N	N	N	N	N	N	N	N	N	N	N	Y

Although each platform provides some of the functionality needed for some internet-scale studies, only Pushkin fully supports a wide range of studies.

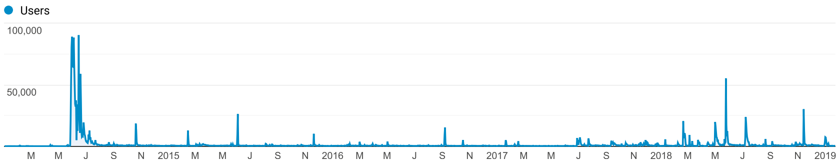


Fig. 1 Daily traffic at GamesWithWords.org, a successful website for massive online experiments and citizen science. Large spikes in traffic are common after the launch of a new experiment, coverage in popular media, or both.

fact, a common method of cyberattack is to overwhelm a website with heavy traffic. Common strategies involve autoscaling (described below) and comprehensive backups. Although large commercial websites such as Google Forms or SurveyMonkey will generally handle these issues, they are fairly limited as platforms for research.

Indeed, our formal and informal surveys of colleagues indicate that access to a *range of experimental paradigms* is a major limiting factor in the adoption of internet-scale research. Most platforms were designed to support a particular paradigm, such as surveys (SurveyMonkey, Qualtrics, Google Forms), psycholinguistics experiments (Ibex Farm), or iterated cultural evolution experiments (Dallinger). Researchers who want to run multiple paradigms may need to learn multiple platforms (Table 1). This provides yet another barrier.

Finally, one of the most powerful uses of massive online experiments and citizen science projects is to gather data on very large numbers of specially chosen experimental stimuli, often with each participant only seeing a small fraction of test items. This raises a number of difficult design questions: Which stimuli should be tested, how many times should each be presented, and to which participants? Traditionally, researchers have answered these questions informally using a combination of intuition, prior experience with experimental paradigms, power analyses, and estimates of the number of participants and time involved in studies. This is possible in the laboratory, where the experimenter has time between subjects to adjust protocols, design new experiments, and so forth. It is not possible for internet-scale studies, for which data collection is continuous and happens at the discretion of the subjects, not the experimenter. A further complication is that with internet-scale studies, we rarely know how many subjects we will have, even to the order of magnitude. Thus, to get full use out of internet-scale research, it is often necessary to have the software response *contingently* as the data comes in.

This can be addressed with optimal experimental design, active learning, or other machine-assisted experimental design techniques (see the Contingent Experiments section). The basic idea behind machine-assisted experimental design is to choose, on the basis of all participants' previous behavior, the stimuli that will provide the most relevant information. This may, for instance, take the form of adapting to one participant's individual characteristics, selecting the next item of interest given responses so far, or choosing an entire study design from a large space provided by the user. Unfortunately, fully supporting machine-assisted experimental design requires a different software architecture from that of existing software (see the How Pushkin Works section). Thus, existing software either does not support machine-assisted experimental design or, in the case of Dallinger, supports only a subset of methods (Table 1).

Pushkin: A platform for internet-scale research

Philosophy and approach

The lab-based experimental paradigm developed over a century ago. A robust approach to internet-scale studies will not appear immediately, nor without a great deal of work. Thus, we have approached the problem in an incremental, scalable way. Although our long-term goal is a robust new paradigm that vastly increases the rate of progress in our science, we do not attempt to do all of this (or even most of this) ourselves. Instead, our approach is to lay a foundation upon which our community can build.

Part of the inspiration comes from Alexander Pushkin (1799–1837), who developed the literary language of Russian. No work of literature exists in a vacuum. Authors lean on established genres (romance, coming of age, fantasy), standard archetypes (vampires, ninjas, jaded private eyes), idioms ("heart on my sleeve," "the center cannot hold," "the sound and the fury"), and direct references (he is a Scrooge/Eeyore/Romeo) in order to quickly evoke characters, scenes, and emotions, rather than building everything from scratch. Prior to Pushkin, little was written in Russian, and this shared cultural vocabulary did not exist. Pushkin promoted what did exist, invented vocabulary, established genres, and coined idioms, building the framework that Doestoevsky, Tolstoy, Chekhov, and others depended on for their masterworks.

Thus, our goal is to establish a shared cultural vocabulary on which an internet-scale research paradigm can be built. This includes producing not just reusable software but also experimental paradigms, analysis methods, and best practices. Just as the Russian literary language continued to develop after Pushkin, our goal is to establish a core set of tools and paradigms upon which others will build.

As such, we stress *interoperability* (our tools should be modular and interface with existing projects), *extensibility* (our tools should be easy to extend), and *broad applicability* (our tools should be useful not just for economics games or psycholinguistics tasks, but for the widest possible range of internet studies). The result, we hope, will be a developer ecosystem that supports rapid building and deployment of not just new experiments, but new tools. This common vocabulary can then be used by developers to build custom applications for specific laboratories or to develop easy-to-use plug-and-play software for specific types of studies (cf. E-Prime). Such ecosystems play vital roles in the technology industry. We believe that they can play a similar role in social science.

At the heart of this is the Pushkin experiment framework itself: A platform for internet-scale research. In particular, Pushkin version 1.0 provides a range of functionality that is needed for massive online experiment and citizen science but that is not addressed by existing software. Importantly, in keeping with our philosophy, Pushkin is not a stand-along piece of software, but rather a highly modular framework that binds together different reusable tools. Although some of these tools are original to Pushkin, many are third-party tools, such as jsPsych, WebGazer.js, RabbitMQ, auth0, Bookshelf.js, WebPPL, and many of the components of Amazon Web Services (see the How Pushkin Works section and Fig. 6). If no existing product meets our needs, we attempt to extend one rather than build something from scratch. For instance, we modified jsPsych to make it easier to integrate with Pushkin. However, the services and libraries used are merely default choices; other researchers could swap them out for others as needed. Similarly, our own original tools can be reused for unrelated projects.

Box 1: Definitions There is no well-established terminology for internet-scale studies. Below, we define some of the terms used in this article.

Broadly multidemographic: a study comparing subjects from a large number of demographic groups. For instance, Hartshorne and Germine (2015) quantified cognitive abilities for every age from 10 to 70 (> 75% of the typical lifespan), and Reinecke and Gajos (2014) quantified visual preferences for subjects from 175 countries (90% of the countries in the world).

Extensively sampled stimuli: a large number of stimuli covering a wide range of the space of potential stimuli. For instance, Brysbaert et al. (2016) collected judgments about 61,800 words; Ferrand et al. (2010) collected lexical decision times for 38,400 words and 38,400 nonwords, and Brady, Konkle, Alvarez, and Oliva (2008) tested memory for 2,500 pictures of objects.

Massive online experiment (MOE): an experiment conducted online that is broadly multidemographic, involves extensively sampled stimuli, or both. Typically involves tens or hundreds of thousands of subjects.

Citizen science: a study in which large numbers of volunteer research assistants help collect data, perform analyses, or otherwise carry out research activities (Bonney et al., 2014; Dickinson et al., 2010; Silvertown, 2009; Simpson et al., 2014). Citizen science projects differ from MOEs in that the volunteers are not research subjects.

Crowdsourcing: a large task is broken down into many small components, each of which is carried out by a different person (Doan, Ramakrishnan, & Halevy, 2011; Howe, 2006). Common examples include spam-filtering, labeling images for search, or checking websites for broken links. Most citizen science projects are examples of crowdsourcing. “Crowdsourcing” is sometimes confusingly used to refer to internet experiments. We avoid that usage here.

Features

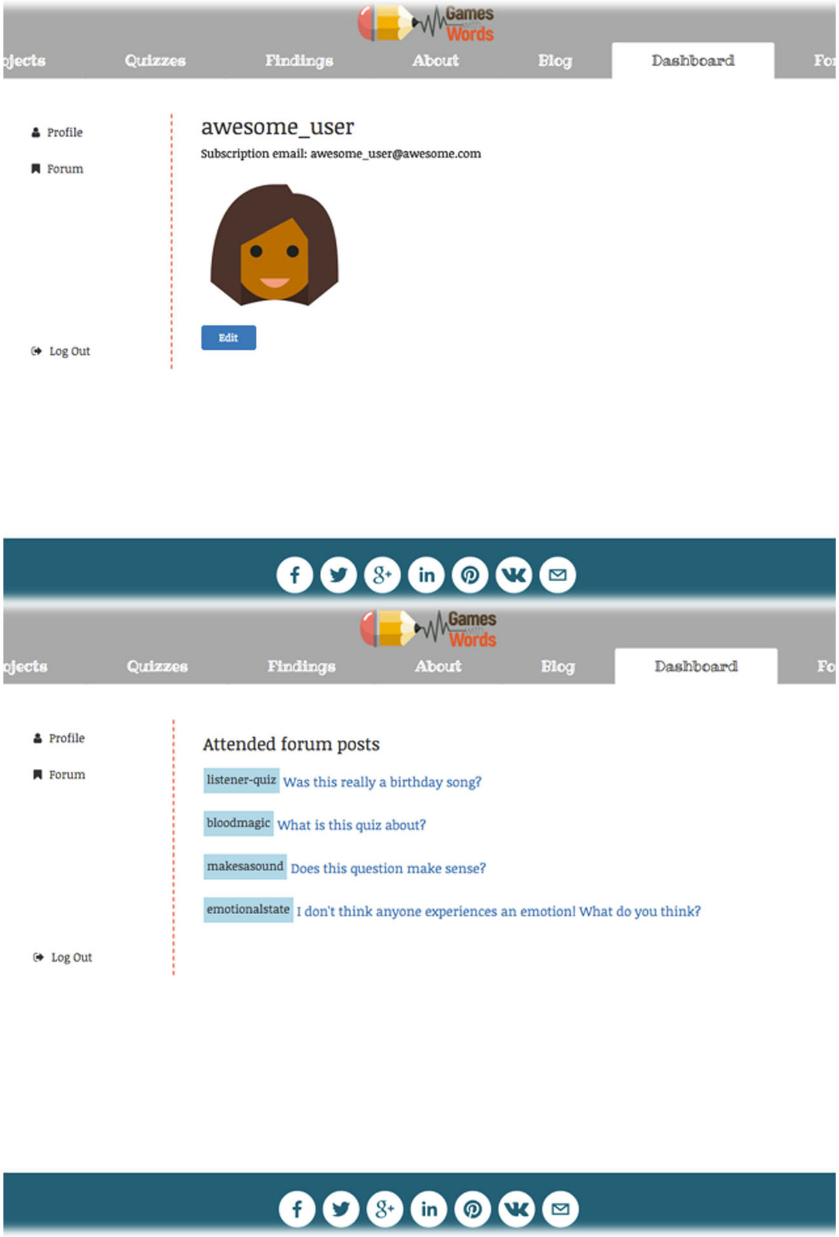
In the Obstacles to Broader Adoption section above, we laid out a number of desiderata for software that are addressed by Pushkin version 1.0. In this section, we describe how these are addressed with the current functionality. In the next section, we provide technical details.

Recruitment and engagement

Pushkin version 1.0 provides a number of mechanisms for recruiting, engaging, and retaining both research subjects and citizen scientists. Note that all these mechanisms are optional, and researchers can use different ones for different studies. Indeed, different kinds of studies will benefit more from different recruitment and engagement mechanisms.

Mailing lists Participants can sign up to receive alerts about new experiments. Using a similar system, gameswithwords.org has built a mailing list several thousand individuals long. Pushkin also allows individuals to sign up to receive information about the results of specific studies and any related publications, which facilitates compliance with common IRB requirements while also providing a mechanism for engagement. Importantly, the mailing list is siloed from data, so there is no way to connect an email address to the subject data.

Sharing To facilitate word-of-mouth recruitment, Pushkin makes it easy for participants to share experiments and other Pushkin webpages via email and social media. (Note that this does *not* give researchers access to subjects’ social media profiles.) Sharing on social media can be quite effective: Since



awesome_user

Subscription email: awesome_user@awesome.com

Attended forum posts

listener-quiz Was this really a birthday song?

bloodmagic What is this quiz about?

makesasound Does this question make sense?

emotionalstate I don't think anyone experiences an emotion! What do you think?

Fig. 2 Example of a citizen science project built with Pushkin, employing such gameification elements as a project progress bar, leaderboard, and shareable badges (see the right-hand side of the screen).

2014, 30% of gameswithwords.org's traffic has come through social media referrals.

Personalized feedback Many research participants appreciate immediate information about the outcome of a study (Huber, Reinecke, & Gajos, 2017; Jun, Hsieh, & Reinecke, 2017; Reinecke & Gajos, 2015). This can consist of a percentile score (*you scored in the 75th percentile on vocabulary/face recognition/working memory*) or a guess about some subject characteristic (*based on our quiz, you are a native speaker of Spanish/elementary school teacher/43 years old*). This can be quite effective. In a study of 5,000 visitors to testmybrain.org, a quarter cited "learning about myself" as the primary motivation for participation (Germine, personal communication, May 18, 2018). Pushkin provides a growing number of templates for such feedback in the form of jsPsych plugins, and developers can easily create their own.

The impact of this feedback can be magnified by allowing subjects to share their results on social media (see the previous section). Although it is somewhat counterintuitive to researchers who have been trained by IRBs to be mindful of subject confidentiality, many subjects are extremely enthusiastic about sharing their results with their friends (cf. the popularity of

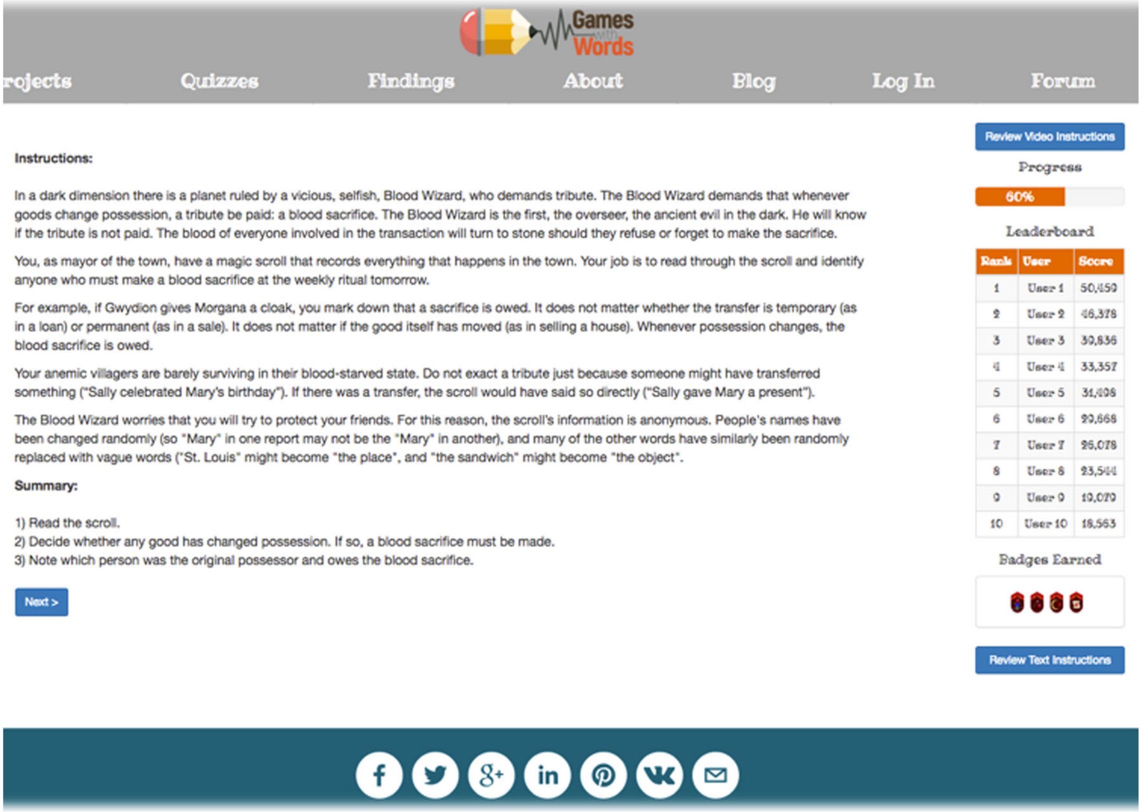
Facebook quizzes). Thus, the ability to share makes research participation more interesting and therefore more valuable to many subjects (Huber et al., 2017; Jun et al., 2017). Again, this is implemented in such a way that researchers do not have access to subjects' social media profiles, and subject data cannot be connected to social media profiles.

Leaderboards and badges

Citizen scientists are motivated by a desire to contribute to science (Reed, Raddick, Lardner, & Carney, 2013). Pushkin provides the option to use leaderboards, badges, and project status bars as a means of visualizing an individual's contribution (Fig. 2). Although primarily intended for citizen science projects, these elements could in principle be used for massive online experiments. For instance, testmybrain.org informs potential subjects of how many subjects have already participated, providing a visualization of how much has been accomplished so far.

Forums Pushkin provides support for an interactive forum in which participants can discuss the research. The forum has optional functionality that is particularly valuable for citizen science projects: the ability to post an item from the study to the forum for discussion and feedback (Fig. 3). Having discussed the item on the forum, anyone can then help code it. These features can be counterintuitive to many researchers, who are used to maintaining research subject naiveté. However, citizen scientists are performing the function of a researcher, and—except for projects that require the *researcher* to be blind to condition—it is often counterproductive for the researcher to be ignorant of the purpose of the project. Moreover, citizen scientists occasionally make important discoveries in their own right, so allowing them to pass along their observations can be very valuable (Becker, 2018).

Participant dashboards For Pushkin projects that allow participants to create persistent identities (see the Range of Experimental Paradigms section), Pushkin provides



Instructions:

In a dark dimension there is a planet ruled by a vicious, selfish, Blood Wizard, who demands tribute. The Blood Wizard demands that whenever goods change possession, a tribute be paid: a blood sacrifice. The Blood Wizard is the first, the overseer, the ancient evil in the dark. He will know if the tribute is not paid. The blood of everyone involved in the transaction will turn to stone should they refuse or forget to make the sacrifice.

You, as mayor of the town, have a magic scroll that records everything that happens in the town. Your job is to read through the scroll and identify anyone who must make a blood sacrifice at the weekly ritual tomorrow.

For example, if Gwydion gives Morgana a cloak, you mark down that a sacrifice is owed. It does not matter whether the transfer is temporary (as in a loan) or permanent (as in a sale). It does not matter if the good itself has moved (as in selling a house). Whenever possession changes, the blood sacrifice is owed.

Your anemic villagers are barely surviving in their blood-starved state. Do not exact a tribute just because someone might have transferred something ("Sally celebrated Mary's birthday"). If there was a transfer, the scroll would have said so directly ("Sally gave Mary a present").

The Blood Wizard worries that you will try to protect your friends. For this reason, the scroll's information is anonymous. People's names have been changed randomly (so "Mary" in one report may not be the "Mary" in another), and many of the other words have similarly been randomly replaced with vague words ("St. Louis" might become "the place", and "the sandwich" might become "the object".

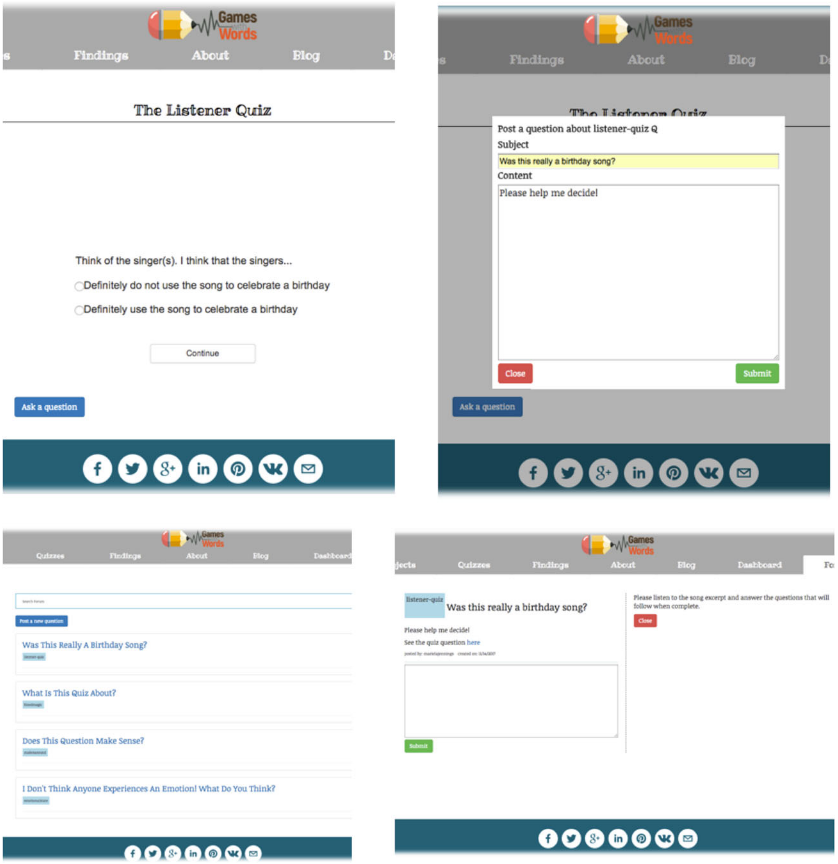
Summary:

1) Read the scroll.
2) Decide whether any good has changed possession. If so, a blood sacrifice must be made.
3) Note which person was the original possessor and owes the blood sacrifice.

Leaderboard

Rank	User	Score
1	User 1	50,459
2	User 2	46,378
3	User 3	39,836
4	User 4	33,357
5	User 5	31,498
6	User 6	29,668
7	User 7	26,078
8	User 8	23,544
9	User 9	19,079
10	User 10	18,563

Fig. 3 The four panels of this figure depict the relationship between citizen science projects and their associated forums in Pushkin. (Top left) A citizen science project in which a volunteer is analyzing a music clip. At the bottom left of the window, there is a button labeled "Ask a question." (Top right) Clicking "Ask a question" brings up a pop-up window, allowing the volunteer to post the item they were working on to the forum, along with a question. (Bottom left) This question is sent to the forum, tagged with the name of the project. (Bottom right) In the forum, users can listen to the clip and discuss the question. Anyone also has the option to respond to the original query (i.e., to code the item in question).



Think of the singer(s). I think that the singers...

○ Definitely do not use the song to celebrate a birthday

○ Definitely use the song to celebrate a birthday

Post a question about listener-quiz Q

Subject: Was this really a birthday song?

Content: Please help me decide!

Was This Really A Birthday Song?

What Is This Quiz About?

Does This Question Make Sense?

I Don't Think Anyone Experiences An Emotion! What Do You Think?

listener-quiz Was this really a birthday song?

Please help me decide!

See the quiz question here

Please listen to the song excerpt and answer the questions that will follow when complete.

Fig. 4 For Pushkin projects that involve persistent identities, participants who are logged in have access to a dashboard. In the dashboard, they can manage their account and also see information about their participation, such as forum posts they are tagged in. In Pushkin version 1.0, access to the dashboard's full functionality requires customization.

dashboards: homepages for registered users that allow them to see information about their participation, such as forum posts they are tagged in, badges they have earned, or their personalized results from massive online experiments they have participated in (Fig. 4). From here, they can also manage their account. In Pushkin version 1.0 the out-of-the-box dashboard functionality is limited, but this is an active area of development, and users can customize the dashboard as needed.

Reliability and stability

Pushkin uses a variety of methods to decrease the probability that the website will crash, and to aid recovery if it does.

The most common reason for a website to crash is for it to be overwhelmed by massive influxes of traffic. This can be addressed by purchasing a very powerful web server. Unfortunately, this is prohibitively expensive. It is also overkill, since most of the time that computing power will go unused (cf. Fig. 1). By default, Pushkin makes use of several powerful methods provided by Amazon Web Services for *auto-scaling*: that is, for flexibly adjusting the amount of computing power available in response to demand (see the Auto-scaling section). This is augmented by specific design features of Pushkin's internal architecture that allow it to "fail gracefully" during periods of heavy traffic (again, see the Auto-scaling section).

Nonetheless, no computer system is immune to crashes. Serious crashes can lead to data corruption or loss. For that reason, Pushkin by default makes use of several redundant mechanisms for backing up data, including real-time backups (see the Backups section).

Range of experimental paradigms

By default, Pushkin uses jsPsych to display stimuli and record responses. In principle, researchers could use any compatible experiment engine, but jsPsych is a particularly robust and flexible option (see Appendix A). It currently allows for the presentation of text, images, video, audio, and any other HTML-formatted content, including animations or interactive displays. Measurements can be made using keyboard responses, mouse clicks, touch input, text input, multiple choice questions and Likert scales, drag-and-drop sorting, visual analog scales, Likert scales, and more. A unique strength of jsPsych is its plugin-based architecture, which allows developers to add new stimulus types and response measures. For example, we created a plugin that allows eyetracking using the Webgazer.js package (Papoutsaki et al., 2016).

Moreover, jsPsych plugins allow for the development of standardized protocols that can be adapted through the adjustment of a set of parameters. For instance, although the implicit association test (IAT; Greenwald, McGhee, & Schwartz, 1998; Nosek et al., 2002) could be implemented as a series of generic stimulus-with-keyboard-response trials, jsPsych provides an IAT plugin that produces the standard layout and feedback of the IAT. Thus, the plugin architecture allows researchers to rapidly develop and disseminate interoperable code for new (and old) experimental paradigms. The growing library of jsPsych plugins means that not only are a wide range of experimental paradigms possible, a growing number of them are quick and easy to implement.

Pushkin augments jsPsych's range of experiments in two important ways. First, it provides a secure subject login system, which enable multi-session and longitudinal designs. It also supports emailing the subjects (with their permission) to remind them about follow-up sessions. (For information on data security, see the Authentication and Logins and Security sections.)

Second, Pushkin provides the infrastructure for a broad range of contingent experiments. We describe this in the next subsection.

Contingent experiments

Pushkin is designed from the ground up to allow dynamic stimulus selection (Fig. 5), and thus is uniquely suited to implementing machine-assisted experimental design algorithms. Most approaches to machine-assisted experimental design rely on mathematically rigorous specifications of (i) the space of the scientific hypotheses of interest, (ii) the space of possible test stimuli, (iii) the space of possible participant responses to the test stimuli, (iv) a measure of the informativity of each response relative to the hypotheses, and (v) algorithms for efficiently searching for good experiments, given these specifications. For instance, in active learning (Settles, 2012), individual experimental stimuli are chosen so as to adaptively minimize uncertainty about the hypotheses in a given hypothesis space using easy-to-calculate local statistical heuristics. In optimal experiment design (Fedorov, 2010; Ouyang, Tessler, Ly, & Goodman, 2018), whole experiments are constructed in order to globally optimize an information gain criterion. We are developing a growing library of templates for specific types of machine-assisted experimental design algorithms in Pushkin.

Because machine-assisted experimental design is not yet common, we conclude this section with a detailed example.

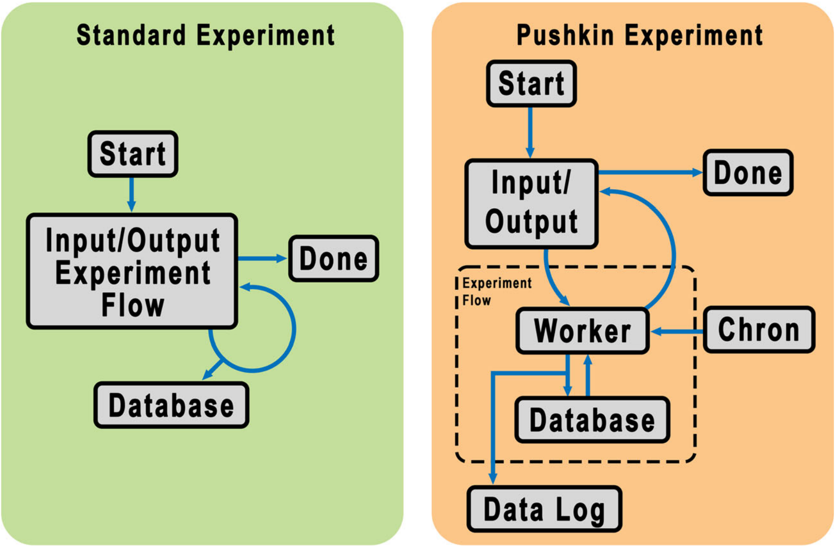


Fig. 5 (Left) Information flow in a standard computerized experiment (e.g., written in jsPsych or PsychoPy). Once the experiment begins, the software loops through each trial, recording the data before going on to the next trial. (Some software packages wait until the end to write the data.) (Right) Information flow in a Pushkin experiment. Pushkin separates input/output procedures (presenting stimuli and collecting the data) from determining what stimulus to display. After each trial, information is sent to a worker, which in addition to recording the results in the database, also decides what to do next. This allows Pushkin applications to dynamically update, choosing which stimuli to display on the basis of both that subject's response and what other subjects have been doing. Two other important features of Pushkin applications are the *data log*, which records a complete history of all writes to the database, enabling version control, and the *Chron* worker, which carries out particular operations at specific times of day. See the main text for a discussion of how these are used.

Readers who are not interested in the details should skip to the next section.

We illustrate using optimal experiment design as formulated by Ouyang et al. (2018). Let's imagine that we are interested in theories explaining reaction times in lexical decision experiments. Lexical decision experiments are a workhorse method in psycholinguistics for studying the processing of words. Subjects must discriminate real words (e.g., "cake," "interrupt," "beige") from nonsense words (e.g., "sleng," "exterrupt," "beigity"). The typical response measures are accuracy and reaction time. It is well known that word frequency affects lexical decision reaction times, with faster responses to more frequent words, though many of the details remain under debate (Adelman, Brown, & Quesada, 2006; Berent, Vaknin, & Marcus, 2007; Ellis, 2002; Morton, 1969; Ratcliff, Gomez, & McKoon, 2004).

Imagine that we wish to compare a set of hypotheses linking words to reaction times. For instance, imagine we wished to compare the hypothesis that reaction time is linearly related to word frequency to the hypothesis that it is logarithmically related. We would formulate each hypothesis as a linear model with frequency or log-frequency as fixed effect and perhaps a variety of random effects. Formally, a hypothesis $m \in M$, is defined by a conditional distribution $P_m(y_x \mid x)$ linking a set of items, x, to measured lexical decision times y_x. Ahead of the experiment we have some prior beliefs about how likely the hypotheses are, $P(M)$, informed by prior results. Our task is to determine what data to collect next—x—with the aim of collecting the data that would be most informative. We can formalize this as maximizing the distance between prior and posterior beliefs:

$$x^* = \arg\max_x D_{\mathrm{KL}}\left[P(M|x, y_x) \parallel P(M)\right]$$

where the Kullback–Leibler divergence $D_{\mathrm{KL}}(\cdot\|\cdot)$ is used as a measure of distance between distributions. A priori, we do not know what the result of any particular experiment will be, so we must marginalize over the possible results y:

$$x^* = \arg\max_x \mathbb{E}_{\tilde{p}(y_x;x)} D_{\mathrm{KL}}\left[P(M|x, y_x) \parallel P(M)\right]$$

Given the specific formulations for p_m, this defines an objective function that can be used to optimally choose what data to collect next. For instance, we could search over possible stimuli in order to choose those stimuli that would best help us distinguish between the hypotheses.

To be clear, machine-assisted experimental design is not different in kind from what researchers normally do: try to design maximally informative experiments. In the same way that inferential statistics help us analyze data, machine-assisted experimental design helps us design experiments. Just as inferential statistics are most useful when the dataset is large and our questions about it are complex, machine-assisted experimental design shines when the hypotheses are many and complex, and when the experimenter has many design choices to make. Machine-assisted experimental design is also particularly helpful when the pace of data collection is too fast for the experimenter to make real-time decisions about what data to collect next—exactly the situation we face in internet-scale studies.

Although mathematical formulations of optimal experiment design, such as the one above, have been available for some time (e.g., Lindley, 1956), the method has not been widely used for two reasons. One is that the design of most experiment software platforms does not permit its use, in particular in the active stimulus selection setting in which machine-assisted experimental design and data collection must be tightly integrated. (The one counterexample being Dallinger, which supports some types of optimal experimental design; Suchow, 2018.) Pushkin's unique architecture allows for a straightforward implementation of a wide range of machine-assisted experimental design protocols. Pushkin users can equip the experiment worker (Fig. 6) with computationally specified competing hypotheses and possible experiments. With these specifications in place, the optimal experiment can often be computed with no further input from the user. Pushkin will continue to make optimal choices about what data to collect next for as long as the experiment runs, thus making efficient use of however many subjects the experimenter manages to recruit.

The second reason is that specifying hypotheses formally can be complex and optimizing experiment design objectives is computationally difficult. Recently, approaches based on probabilistic programming languages (PPLs) have emerged as a viable alternative (Ouyang et al., 2018). PPLs are high-level languages designed for expressing models from artificial intelligence, machine learning, statistics, and computational cognitive science. In PPLs, diverse models are expressed as programs in a common language, and inference algorithms are developed for the language as a whole, rather than for specific models (Goodman, Mansinghka, Roy, Bonawitz, & Tenenbaum, 2008). Probabilistic programming is thus ideal for rapidly specifying and deploying models of each of the components of a machine-assisted experimental design system described above. Although Pushkin users can implement machine-assisted experimental design using any programming language, we are using the probabilistic programming language WebPPL (see Appendix B) to implement a library of reusable tools for machine-assisted experimental design within Pushkin.

Thus, we believe that one of the major contributions of Pushkin will be making machine-assisted experimental design more accessible and easy to implement—and therefore more commonly used.

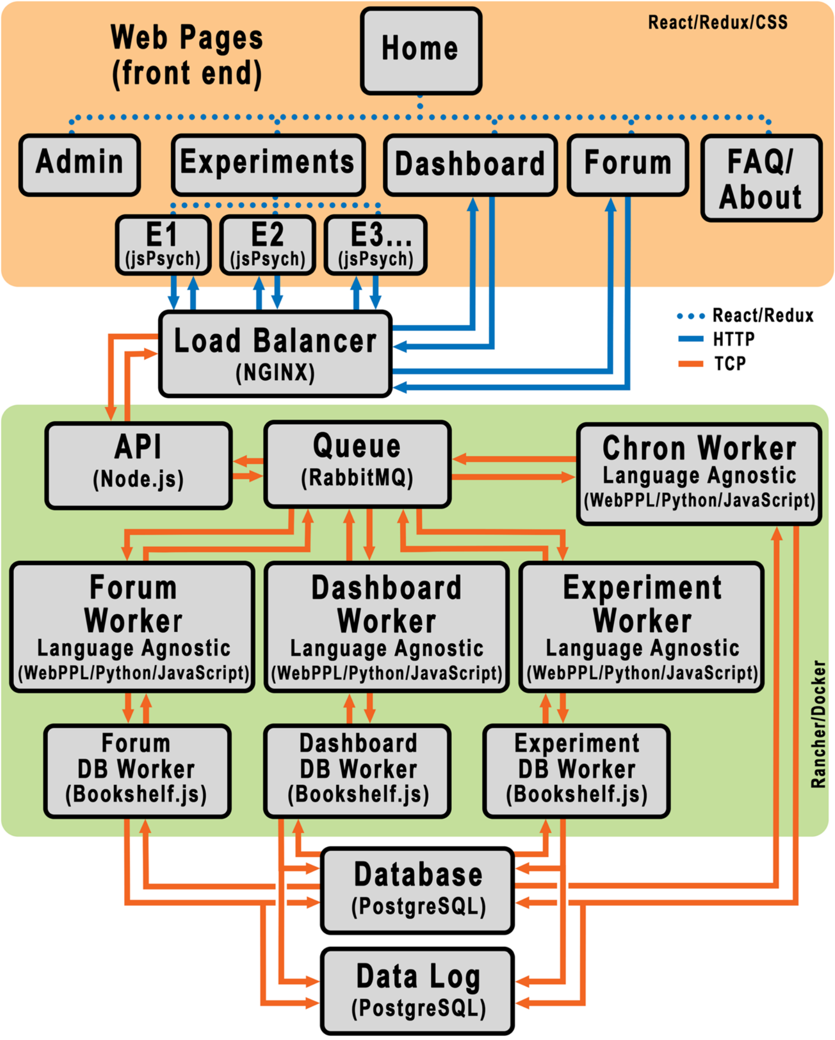


Fig. 6 Schematic of a Pushkin website, which consists of some number of quizzes and some ancillary webpages, such as a forum and a user dashboard. Note that the API, queue, experiment worker, and experiment database (DB) worker are all subsumed under "worker" in Fig. 5. See the main text for a detailed description. Although this is not depicted, each study has its own experiment worker and database worker.

Other features

Webserver setup and management By default, Pushkin employs a number of popular technologies for auto-scaling, version control, and data backups. Detailed instructions on how to set up the webserver and related technology is provided.

Support for reproducibility Because the entire experiment is run via code, reproducing the study merely requires rerunning the code. Similarly, note that Pushkin's data log contains a reasonably comprehensive chronological record of everything that happened during test (see the Backups section).

Stub website Designing a website requires at least basic knowledge of web development. Designing a website that is easy to update, is compatible with different browsers, is optimized for both desktops and mobile devices, and so forth, is troublesome and time-consuming. Pushkin provides a basic website layout that is a sufficient starting point for customization. Researchers who are not familiar with web development can create a website with basic functionality by making minor changes (adding custom images, changing the color scheme and fonts, etc.). More advanced programmers can make major changes to the website layout or create a new website altogether.

How Pushkin works

In this section, we describe many of the technical details as to how Pushhkin version 1.0 supports the functionality described in the previous section. This will primarily be of interest to skilled web developers and/or individuals interested in contributing to the project. Others may wish to skip to the next section.

Figure 6 outlines the structure of a Pushkin website. Pushkin websites consist of three primary parts. At one end are the webpages and associated content, including all jsPsych code and stimulus files. At the other end are the database, which stores lists of stimuli and subject responses, and the data log, which contains a real-time log of every database query, thus providing real-time backup and version control of the database. In the middle is a collection of workers that process participant responses and determine what to do next. The load balancer, which sits between the webpages and the workers, helps facilitate auto-scaling.

Below, we provide additional detail on how this architecture supports the functionality described in the Features section above. In keeping with our philosophy and approach (see the Philosophy and Approach section), we have made extensive use of existing technology and services wherever feasible, including Node.js, React, Redux, PostgreSQL, Rancher, Docker, Auth0, RabbitMQ, and Amazon Web Services. However—again in keeping with the philosophy and approach—the highly modular architecture permits other developers to mix and match. Moreover, given the quickly changing world of web development, it is highly likely that the Pushkin development team will periodically replace some of these technologies as better alternatives emerge.

Auto-scaling Pushkin makes use of several services for auto-scaling. Webpages, images, and videos are hosted in Amazon Web Service's S3 and CloudFront services, which provide rapid, scalable delivery of static content worldwide (Amazon Web Services, 2018).

Processes that require dynamic, server-side computation—such as processing database queries or running machine-assisted experimental design models—are hosted by the Amazon Web Services (AWS) Elastic Cloud Computing (EC2) platform (Amazon Web Services, 2018). An EC2 computer is called an "instance." The same software can be replicated across multiple EC2 "instances," with a load-balancer distributing web traffic to different instances. Thus, if there are three instances, different instances will handle the computations for different subjects. Auto-scaling is accomplished by monitoring usage and dynamically creating or destroying instances as needed. We use Datadog for monitoring usage (Datadog, 2016). For capacity to rapidly respond to demand, creating a new EC2 instance must happen quickly. Although installing Pushkin and its dependencies on a new EC2 instance is automated, it is slow. Thus, ready-to-use copies of the Pushkin software are kept in Docker images that can be rapidly deployed to new EC2 instances by Rancher, a service for managing Docker images.

For database services, we use AWS's AuroraDB, which allows for both horizontal and vertical scaling.

Another feature that helps make Pushkin websites robust to large traffic spikes is the use of a message queue for passing messages between services (Fig. 6). At its heart, a queue is a text file. Services that have a message to pass write the message in the next line of the queue. Other services "listen" for messages addressed to them, immediately deleting them and acting on the instructions. This allows Pushkin applications to fail gracefully: If messages are written faster than they can be read, the queue grows longer and the site slows down proportionally, until auto-scaling provides more capacity and the listeners catch up. Our message queue manager of choice, RabbitMQ, provides a number of other useful features, including the ability to directly influence auto-scaling (Videla & Williams, 2012).

We chose these services for their robustness and cost-effectiveness: gameswithwords.org currently costs $300–$400 per month to support nearly 40,000 visitors per month. However, the modularity of Pushkin allows developers to—with greater or lesser degrees of effort—employ other services instead. Similarly, this modularity will make it easier to upgrade Pushkin in the future as new (versions of) services because available.

Authentication and logins Experiments requiring multiple sessions (longitudinal studies, sleep studies, certain memory paradigms, etc.) necessitate tracking the same individual across multiple visits to the website. For experiments requiring this tracking, researchers can allow subjects to log in. Pushkin uses Auth0, a highly secure and widely trusted service for website logins (Auth0, 2017). Subjects can either create a username and password *or*—if researchers wish and their institutional review board (IRB) allows—log in using an email account or social media profile. The latter option has the advantage of not requiring the subject to remember a username. Note that this does not give the researcher access to the subject's private social media and other data (see also the Security section).

Backups Amazon Web Services' RDS Multi-AZ deployments provide a real-time backup of the PostgreSQL database used by Pushkin. The primary Multi-AZ database instance is backed up by an identical copy hosted in a different geographical location. If the primary instance or its backup loses data—or if, for instance, there is a power outage—the data is recovered and duplicated from the unaffected copy.

In addition, the data log (another PostgreSQL database identical to the main Pushkin database) maintains a record of all queries performed on the primary Pushkin database. The data log serves as the history of a Pushkin project and can be used to restore the primary database in case of failure. The data log has its own real-time backup maintained by AWS. Therefore, instead of having just one main database, Pushkin maintains four databases (a primary database with a copy in a different availability zone, and a data log with a copy in a different availability zone).

In addition to backing up databases by creating identical copies, AWS provides database backup features designed to recover a particular state of a database—database snapshots and automated backups. Automated backups can be turned on for any AWS database, and Amazon RDS automatically takes a snapshot of the data in every database once a day. In addition, the database owner can choose to create additional database snapshots at any point in time (after, e.g., a major spike in website traffic). All of these backup strategies ensure that it is virtually impossible to lose data with a Pushkin project.

Security Data security is a concern for any networked device, whether a webserver or a laptop. Pushkin employs a number of security mechanisms, described below. The overarching approach can be summed up as:

- Default to anonymity rather than confidentiality whenever possible. If even the researchers do not know who the subjects are, security breaches are less problematic.
- Where possible, silo identifiers from data. For example, the recruitment email list—which contains email addresses—is not connected to data.
- Anonymize anything that is not anonymous automatically and as early in the data pipeline as possible.

If login/authentication is not enabled, data collection is anonymous. Identifiers such as IP addresses are not collected.

If login/authentication is enabled, data cannot be made fully anonymous. However, there are a number of layers of protection. First, we use the highly secure Auth0 authentication service to handle user IDs. Logins are handled by the Auth0 webservers, not by Pushkin itself. Participant identifiers (e.g., email and username) are stored in the secure Auth0 database. If the participant authenticates using a social media service (Facebook, Twitter, etc.), their social media username is likewise stored in the secure Auth0 database. Note that all that is accessed is the user's publicly available social media username; private social media data are not accessed or stored.[3]

Crucially, Pushkin applications do not access the participant's external identifiers (e.g., email address), but rather an alphanumeric "token" representing the participant. Thus, the identifiers are stored in Auth0's secure database, and participant data are stored in Pushkin's secure database. For additional protection, Pushkin encrypts the Auth0 tokens as well, providing an additional password-protected firewall. Moreover, the (encrypted) token is stored separately from the data itself; instead, a *different* alphanumeric identifier is used to identify subjects for purposes of analysis. Finally, data are encrypted when traveling between the subject and the website, between the website and Auth0, and between the website and the researcher.

Thus, although it is *possible* to deanonymize data, this requires considerable effort and several passwords. Note that these robust security procedures do *not* mean that Pushkin is unhackable. No security system is unbreakable. Even if the software itself cannot be hacked, humans present a point of weakness (e.g., researchers or participants using easily guessed passwords). Moreover, it is sometimes possible to identify a subject from their data alone, if the questions asked are sufficiently specific (e.g., there may be only one female rabbi in a specific small town; Narayanan & Shmatikov, 2008). However, these considerations apply equally to data collected in the lab, and we encourage researchers to use common sense and robust security procedures for all data. For extremely sensitive data, researchers may be advised to take even more robust security measures than what Pushkin provides out of the box.

Chron The Chron worker is Pushkin's bonus feature. Although it is not an essential component of a Pushkin study, it makes it possible to periodically analyze data and send reports. Since the Chron worker is language-agnostic, it can run scripts written in the researcher's language of choice (Python, JavaScript, WebPPL, R, etc.). The Chron worker can also be used to periodically remove data from subjects who did not complete a study or did not pass screening questions set up by the experimenters. Those are only some of the potential uses for the Chron worker. In large-scale citizen science projects, it can be incorporated to perform tasks such as alerting the researchers when sufficient data have been collected for a set of stimuli, or when other milestones in data collection have been reached. The Chron worker eliminates the need to monitor data collection freeing up the researcher's time and resources.

Using Pushkin

The Pushkin source code is available through GitHub (github.com/pushkin-consortium/pushkin). The source code provides a stub of what is needed for a website similar to gameswithwords.org that hosts multiple massive online experiments and citizen science projects. Users familiar with ReactJS can edit the structure of the website (which pages are available, etc.) as desired.

Currently, to use Pushkin, users download the source code for Pushkin and the source code for jsPsych and (if desired) WebPPL. Users will also need to configure a web server from Amazon Web Services (or, if desired, an appropriate alternative). Users are urged to consult the documentation for the most up-to-date instructions (pushkin-only.readthedocs.io).

Modern website design involves a fairly unintuitive file structure. For instance, the code for a single experiment must be distributed across a variety of folders, with parts of different experiments ending up in the same folder. Likewise, many of

[3] Auth0 does allow websites to request user permission to access private social media data. Pushkin version 1.0 does not use this option.

the best practices that result in efficient, fast websites also result in code that is very hard to read. Thus, for purposes of development, we use a more intuitive file structure. When the user is ready to test or deploy the website, the code is reorganized into a web-appropriate format using webpack (Hlushko et al., 2018) (again, consult the documentation). Importantly, users can work exclusively with the "user-friendly" files and never have to inspect or modify the web-ready version.

Individual experiments can be written in jsPsych (advanced users may choose to use an alternative, but this may require significant extra work). For the most part, development is the same as it would be for any other jsPsych experiment (see documentation at www.jspsych.org). For most projects, the primary difference is in how the results are saved, since Pushkin handles interaction with the database (were data are stored). Setting this up is largely automated (see documentation), and only requires significant customization if the user needs to do a lot of preprocessing of the data before it is stored (for instance, for the purposes of machine-assisted experimental design).

Similarly, there is little extra that the user must do for a standard experiment in which every subject sees all items. The jsPsych library handles experiment flow for simple experiments (i.e., where every subject sees every item). Users who wish to create highly contingent experiments will need to edit the worker for that experiment (see Fig. 6).

As this summary should make clear, although Pushkin provides a powerful template for creating internet-scale projects, using Pushkin using Pushkin currently requires a fair amount of technical expertise, particularly for more complex projects. Our current development priority is making Pushkin more accessible to a wider audience (see the next section).

Future development

Tools to improve ease of use

We are in the process of rolling out command line tools that will greatly simplify the process. (This is one of the reasons to consult the documentation for the latest instructions.) In particular, we are using the popular package manager npm to download, install, update, and manage Pushkin, jsPsych, WebPPL, and their dependencies. Package managers greatly simplify the use of Unix programs and are the gold standard for open-source projects. Using the package manager for Pushkin is currently done through command line but in the future will be available through the graphical user interface (GUI). When complete, the Unix command "npm install pushkin" downloads the latest version of Pushkin along with dependencies. Similar Unix commands download various add-ons, such as additional jsPsych plugins or experiment templates. Importantly, anyone can create add-ons and distribute them through the package manager. Similarly, we are working on command line tools that will simplify webserver configuration. Finally, these tools will also support updating to newer versions of Pushkin. As of this writing, we expect to complete these upgrades by the end of 2018.

More ambitiously, we intend to integrate Pushkin development into the jsPsych GUI), currently in beta. The jsPsych GUI is a web application that allows users to build experiments by creating and organizing a series of trials with a point-and-click interface. The researcher can customize the parameters of each trial through simple menus that require no programming experience. Images, audio, and video can be uploaded for inclusion in the experiment. As users build an experiment, they are shown an immediate live preview, providing critical feedback for novice developers. When the experiment is finished, the GUI automatically assembles jsPsych code, which can then be exported and used.

We are currently extending the GUI to help with deployment of entire Pushkin websites, including facilitating customized subject feedback, social media integration, and machine-assisted experimental design. Note that while the GUI would make Pushkin accessible to researchers who lack programming experience, we intend it to offer advantages even to proficient programmers. To minimize the loss of flexibility that comes from using the GUI, we will extend the GUI to allow editing of the underlying code for each component at all steps of the process and link these changes to the visual state of the GUI. For example, if a researcher has code to parametrically generate stimuli, they will be able to insert this code via a script editor in the GUI and use it when declaring parameters for trials using the visual interface. The GUI will be able to run the code immediately and incorporate it into the live preview.

Complementary to the GUI, we are building a library of experiment templates for common paradigms. These greatly simplify experiment development. For instance, running a self-paced reading experiment may require no more than providing a list of sentences to be included and setting a few parameters in a config file (e.g., how many trials per subject).

We intend to make the experiment template library available through the GUI. A researcher will be able to browse through these examples and select one that closely resembles their desired experiment. This will create a copy of the experiment for the researcher to edit as they see fit. The availability of these prototypes would aid both novices and experienced developers. Researchers who create experiments will have the option to publish them to the package manager, thus not only supporting other researchers but also improving the reproducibility of their own work.

Finally, we are working on providing greater support for using Pushkin. Lack of institutional knowledge appears to be a major impediment to the wider use of internet-scale studies: researchers are simply not familiar with the design constraints and opportunities. One of the missions of the broader Pushkin project is to address that gap through workshops and publications (cf. Hartshorne & Jennings, 2017; Hartshorne, Leeuw, Germine, Reinecke, & Jennings, 2018a). The present article is an example of these activities. We intend to conduct a number of workshops over the next few years and publish a free electronic textbook (for similar examples, see Goodman & Stuhlmüller, 2014; Goodman & Tenenbaum, 2014).

Conclusion

Pushkin provides a suite of tools for conducting massive online experiments and citizen science projects for psychology and the cognitive sciences. It addresses both the design challenges of internet-scale research (recruiting subjects, running longitudinal studies, machine-assisted experimental design, etc.) and the technical challenges (webserver setup and configuration, data security, real-time backups and version control, auto-scaling, etc.). To achieve these ends, Pushkin draws on a wide range of software and hardware technologies. Thus, in addition to being a software framework, Pushkin can be thought of as a collection of best practices.

Other frameworks can provide aspects of this functionality. Most obviously, many of the same experiment designs can be implemented in jsPsych (though it does not by itself support machine-assisted experimental design or longitudinal studies), and some of our recruitment mechanisms are implemented as jsPsych plugins. However, jsPsych is purely experiment software that is meant to be embedded in a larger website. It does not handle data storage and security, backups and version control, auto-scaling, or any of the other parts of running a highly trafficked website. Other platforms, such as Google Forms or SurveyMonkey, provide the website but are very limited in their experiment functionality. LabVanced supports a wide range of experiments, but is not open-source or customizable and does not provide much support for subject recruitment and does not permit machine-assisted experimental design. Zooniverse (Simpson et al., 2014) provides a powerful platform for citizen science projects that involve classification and annotation of images, but does not support linguistic annotation or the collection of psychological data. Thus, although existing platforms provide excellent support for certain paradigms—and indeed, we use many of them—only Pushkin supports a wide range of internet-scale studies.

However, we must acknowledge that Pushkin supports only a subset of the internet-scale studies that are currently possible or will be in the near future. For instance, it does not currently support the sophisticated use of mobile devices (Miller, 2012; Stieger, Lewetz, & Reips, 2017) or wearable sensors or virtual reality. Pushkin focuses on web applications—which are popular among older children and adults—and not mobile apps, which are more appropriate for testing young children. Importantly, the fact that Pushkin is free and open-source, as well as modular and extensible, means that Pushkin should provide an important foundation as the community explores these exciting possibilities.

Author note Work on Pushkin is supported by NSF Grant 1551834 to J.K.H. The authors thank Katharina Reinecke, Laura Germine, Roman Feiman, and two anonymous reviewers for comments and suggestions; Miguel Mejia for graphic design; the attendees at the Fist Annual Pushkin Developers Conference; and Oddball.io.

Appendix A: jsPsych

The jsPsych package is open-source software for developing online experiments using HTML, CSS, and JavaScript. These languages are core to all Web browsers, which means that jsPsych experiments will run on any device that has a web browser, including mobile devices, without the need to install any additional software.

The jsPsych package's core feature is a modular, plugin-based architecture. Different experimental tasks—such as filling out a questionnaire, viewing a stimulus and pressing a key in response, or reading instructions—are implemented as plugins. jsPsych's experiment controller (cf. Fig. 6) controls the flow of the experiment from plugin to plugin.

By design, plugins can implement tasks that vary in their generalizability. For example, one generic plugin is used to display a stimulus and measure the response time for a keyboard press. The researcher can set parameters for the plugin to control what stimulus to display, what keys are valid responses, and how long to wait for a response. Simple modifications of these parameters can generate a very diverse set of experiments. Other plugins are highly tailored for specific types of experiments, such as the scene segmentation task developed by Fiser and Aslin (2001). In this task, objects are displayed on a grid and participants view many grids over the course of the experiment. Experiments typically test whether participants learn the spatial co-occurrence properties of the various objects on the grid. The jsPsych plugin for this paradigm automatically creates these grid scenes based on a few parameters like the list of objects, the number of rows and columns in the grid, and the overall display size of the scene. By creating a plugin tailored to this task, implementing variations of the task by varying plugin parameters becomes an efficient process that can be accomplished by a novice programmer. The jsPsych library already has plugins to support a wide range of studies. The latest version (6.0.2; released April 2018) includes 35 different plugins, and many more have been developed by the community of jsPsych users.

The jsPsych experiment engine provides a framework for assembling collections of plugins into a unified experiment. The engine allows the developer to set up experiments that range in complexity from a linear set of trials to dynamically responsive procedures such as staircasing in psychophysics. jsPsych also provides tools to facilitate reusable code. For example, if an experiment involves a repetitive procedure—for instance, view fixation cross, judge a stimulus, receive feedback, repeat—the procedure can be declared once with an accompanying list of variables that define the set of trials to run using that procedure.

For a complete description of jsPsych features, documentation, and tutorials, see the jsPsych website (http://www.jspsych.org).

Appendix B: WebPPL

WebPPL (pronounced "web people") is a feature-rich probabilistic programming language embedded in JavaScript. It is a universal probabilistic modeling language that makes it pleasant to precisely describe complex models from computational cognitive science, artificial intelligence, natural language processing, machine learning, and related fields. WebPPL provides many different methods for *posterior inference*, including dynamic programming, Markov chain Monte Carlo, sequential Monte Carlo, and variational inference. This combination of powerful representation language and no-fuss inference techniques makes it ideal for describing models of the kind used in the social sciences and for analyzing social science data.

There are exciting synergies between the development of Pushkin and WebPPL with regard to machine-assisted experimental design (see the Contingent Experiments section). After expressing the spaces of models, experiments, and responses as WebPPL programs, it is (surprisingly) straightforward to express active learning or optimal experimental design (OED) as a probabilistic program (see Listing 1). Equation 1 from the main text translates to around 20 lines of WebPPL code, expressing that OED is an inference problem. This gives us a unified basis for both the representation of OED tasks and innovations in the OED system itself.

```
var OED = function(mSample, xSample, ySample) {
 var mPrior = Infer(mSample)             // store prior on models
 Infer(function() {                      // search over x
   var x = xSample()
   var KLDistrib = Infer(function() {    // compute KL for each y
     var y = ySample()                   // $p(y;x)$
     var mPosterior = Infer(function() { // $P(M \mid Y = y)$
       var m = mSample()
       factor(score(m(x), y))
       return m
     })
     return KL(mPosterior, mPrior)      // $D_{\mathrm{KL}}(P(M \mid Y = y) \parallel P(M))$
   })
   var EIG = expectation(KLDistrib)     // $\mathbb{E}_{p(y;x)} D_{\mathrm{KL}}(P(M \mid Y = y) \parallel P(M))$
   factor(Math.log(EIG / maxEIG))  // optional (search by inference)
   return {x: x, EIG: EIG}
 })
}
```

Appendix C: Frequent concerns about online studies

Are the data trustworthy?

Studies have found that internet volunteers comply with instructions and answer truthfully at rates matching or exceeding lab-based subjects, resulting in data with similar psychometric validity (Aust, Diedenhofen, Ullrich, & Musch, 2013; Birnbaum, 2004; Germine, Dunn, McLaughlin, & Smoller, 2015; Germine et al., 2012; Johnson, 2005; Meyerson & Tryon, 2003). Studies with online volunteers generally produce similar results to lab-based studies; any differences that are observed typically are clearly attributable to differences in subject demographics (Birnbaum, 2004; Buchanan & Smith, 1999; Casler, Bickel, & Hackett, 2013; Germine et al., 2012; Gosling, Vazire, Srivastava, & John, 2004; Hartshorne, 2008; Hartshorne & Germine, 2015). The high data quality is not surprising: volunteer participants are highly motivated to participate; if they weren't, they wouldn't.

These findings are further bolstered by the fact that data quality is similarly high for online labor markets such as Amazon Mechanical Turk, despite the potential misalignment of subject incentives (money) and researcher priorities (high-quality data), particularly if proper manipulation checks and attention checks are employed (Behrend, Sharek, Meade, & Wiebe, 2011; Buhrmester et al., 2011; Casler et al., 2013; Goodman, Cryder, & Cheema, 2013; Hauser & Schwarz, 2016; Johnson, 2005; Rand, 2012; Shapiro, Chandler, & Mueller, 2013; Smith, Roster, Golden, & Albaum, 2016).

What if I need precise timing?

Much to the annoyance of psychologists, modern computers were not designed to carefully time stimulus presentation or record reaction times. Experiment software such as E-Prime or Psychophysics Toolbox uses clever software work-arounds to achieve precise timing (Kleiner, Brainard, & Pelli, 2007; Schneider, Eschman, & Zuccolotto, 2002). Experiments that run through browsers face additional hurdles, though most of these, too, can be overcome with clever software workarounds (Adenot & Wilson, 2016; Barnhoorn, Haasnoot, Bocanegra, & van Steenbergen, 2015; Chetverikov & Upravitelev, 2016; de Leeuw & Motz, 2016; Hilbig, 2016; Reimers & Stewart, 2015; Simcox & Fiez, 2014; Slote & Strand, 2016). Indeed, subtle reaction-time studies have been successfully run online for more than a decade (e.g., Crump, McDonnell, & Gureckis, 2013; Keller, Troesch, & Grob, 2015; Nosek et al., 2002; Slote & Strand, 2016).

There is one small but important exception: Online studies are susceptible to a slight lag in reaction time measurement, which varies slightly by computer model, operating system, web browser, and recording method (Barnhoorn et al., 2015; Chetverikov & Upravitelev, 2016; Pinet et al., 2017; Reimers & Stewart, 2015; Semmelmann & Weigelt, 2017; Slote & Strand, 2016). Because this lag is constant, it does not affect differences between within-subjects conditions. Likewise, it causes only a little additional noise in between-subjects studies that use random assignment to condition. However, it does present confounds for studies of demographic effects on absolute reaction time, since different demographics may preferentially use different equipment and thus have different lags. We are currently working on a software fix for this. In the meantime, researchers—whether using Pushkin or not—should be cautious about interpreting demographic effects on absolute reaction time that are smaller than about 20 ms.

Note that extremely precise timing generally requires sophisticated calibration even in the laboratory (cf. Krantz, 2001). Such calibration is challenging to do even in the laboratory, and online subjects may not have the tools or the patience to do that calibration. Thus, studies that need such calibration may not be good candidates for internet-scale research—at least, given current technology. However, for the vast majority of studies, these problems are solved. One of the purposes of Pushkin is to make best-practices for precise online studies more widely available.

What if my research paradigm cannot be conducted online?

It is true that experiments that were designed to be run in the laboratory do not always translate exactly to massive online experiments. Common laboratory paradigms were designed around the constraints and opportunities of the lab, not the internet. Nonetheless, it is frequently possible to redesign a laboratory experiment in a way that addresses the constraints and opportunities of massive online experiments. Even where that is not possible, it is often the case that the research *question* is still addressable through a massive online experiment.

To be clear: some questions cannot (yet) be addressed through massive online experiments for methodological reasons, such as those that currently require magnetic resonance imaging. Likewise, some populations, such as babies, remain difficult to recruit and test online (though see Scott & Schulz, 2017). However, the wide-spread use of online studies gives an indication of just how many research questions can be addressed online (see Stewart et al., 2017). For example, one compendium of online studies for volunteers (i.e., not studies run through Amazon Mechanical Turk, etc.) finds it necessary to subdivide the hundreds of active experiments into 22 categories: cognition, consumer psychology, cyber psychology, developmental psychology, educational psychology, emotions, environmental, forensic psychology, gender, general psychology, health psychology, industrial/organizational psychology, judgment and decision making, mental health, personality, positive psychology, psychology and religion, relationships, sensation and perception, sexuality, social cognition, and social psychology (https://psych.hanover.edu/research/exponnet.html).

One purpose of this article—and of Pushkin itself—is to help more researchers see the ways in which massive online experiments could benefit their own research. To these ends, we note that in the long-term the range of methods available online may be even broader. The rapid growth of consumer electronics is broadening access to increasingly sophisticated research tools at home, such as physiological sensors (FitBit, Sproutling, Empatica), sensors for detecting body posture and even hand gestures (Kinect, Lumo), rudimentary electroencephalography headsets, and virtual reality kits (Cadmus-Bertram, Marcus, Patterson, Parker, & Morey, 2015; Gao, Harari, Tenenbaum, & Ullman, 2014; Harari et al., 2016; Miller, 2012; Montgomery-Downs, Insana, & Bond, 2012; Picard, Fedor, & Ayzenberg, 2015; Poh, Swenson, & Picard, 2010; Ren, Meng, Yuan, & Zhang, 2011; Trull & Ebner-Priemer, 2013). Just as the widespread ownership of personal computers means that subjects no longer needed to come to the lab to do computerized studies, the spread of these technologies is increasing the number of studies subjects can volunteer for without committing to traveling to a specific place at a specific time.

What about the fact that the internet is not a random sample of the population?

Subjects in massive online experiments are not representative of the population, but they are typically much more representative than the subjects in lab-based studies or online labor-market studies (Germine et al., 2012; Gosling et al., 2010; Gosling et al., 2004; Henrich, Heine, & Norenzayan, 2010; Ipeirotis, 2010; Paolacci et al., 2010; Rife, Cate, Kosinski, & Stillwell, 2016). Note, however, that a representative sample ensures that findings will generalize to the population as a whole, but it does not necessarily help determine whether subpopulations vary. For obvious reasons, it is impossible to tell whether an effect varies by ethnicity if only a few members of each ethnic group are included in the sample. Importantly, massive online experiment samples are large and diverse, which allows the researcher to do something better than merely average away demographic variability: *measure* demographic variability. This opens up a vast range of scientific questions (cf. Henrich et al., 2010).

What about ethical concerns and data security?

A number of recent scandals have involved individuals collecting and distributing private social media data on a massive scale without permission, such as the Cambridge Analytica scandal or the OKCupid hack (Grassegger & Krogerus, 2017; Zimmer, 2016). This has led to concerns that about online research with regard to privacy and data security (e.g., Xu, 2018).

Luckily, the same solutions that work in the laboratory can be applied online: good data security procedures and IRB oversight. The fact that Pushkin experiments are fully automated makes both of these easier. Data security is baked into the Pushkin workflow; researchers using Pushkin follow best data security practices by default and without any additional effort (see the Security section). IRB oversight is likewise simplified: because data collection is automated, the IRB can fully vet the subject experience. We believe that, where possible, similar procedures should be incorporated into lab-based experiments

Listing 1 OED implementation. For clarity, we have omitted some book-keeping details.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Adelman, J. S., Brown, G. D., & Quesada, J. F. (2006). Contextual diversity, not word frequency, determines word-naming and lexical decision times. *Psychological Science*, *17*, 814–823. doi:https://doi.org/10.1111/j.1467-9280.2006.01787.x

Adenot, P., & Wilson, C. (2016). Web audio API (W3C editor's draft). Retrieved July 20, 2016, from https://webaudio.github.io/web-audio-api/

Amazon Web Services. (2018). Amazon Web Services: Getting started resource center. Retrieved from https://aws.amazon.com/getting-started

Arnold, J. E. (2001). The effect of thematic roles on pronoun use and frequency of reference continuation. *Discourse Processes*, *31*, 137–162.

Aust, F., Diedenhofen, B., Ullrich, S., & Musch, J. (2013). Seriousness checks are useful to improve data validity in online research. *Behavior Research Methods*, *45*, 527–535. doi:https://doi.org/10.3758/s13428-012-0265-2

Auth0. (2017). Token based authentication made easy—Auth0. Retrieved from https://auth0.com/learn/token-based-authentication-made-easy/

Bainbridge, W. S. (2007). The scientific research potential of virtual worlds. *Science*, *317*, 472–476.

Barnhoorn, J. S., Haasnoot, E., Bocanegra, B. R., & van Steenbergen, H. (2015). QRTEngine: An easy solution for running online reaction time experiments using Qualtrics. *Behavior Research Methods*, *47*, 918–929. doi:https://doi.org/10.3758/s13428-014-0530-7

Becker, K. (2018). How citizen scientists discovered the strangest star in the galaxy. *Nova Next*. Retrieved from http://www.pbs.org/wgbh/nova/next/space/how-citizen-scientists-discovered-the-strangest-star-in-the-galaxy/

Behrend, T. S., Sharek, D. J., Meade, A. W., & Wiebe, E. N. (2011). The viability of crowdsourcing for survey research. *Behavior Research Methods*, *43*, 800–813. doi:https://doi.org/10.3758/s13428-011-0081-0

Berent, I., Vaknin, V., & Marcus, G. F. (2007). Roots, stems, and the universality of lexical representations: evidence from Hebrew.

Cognition, *104*, 254–286. doi:https://doi.org/10.1016/j.cognition.2006.06.002

Birnbaum, M. H. (2004). Human research and data collection via the Internet. *Annual Review of Psychology*, *55*, 803–832. doi:https://doi.org/10.1146/annurev.psych.55.090902.141601

Blanchard, R., & Lippa, R. A. (2007). Birth order, sibling sex ratio, handedness, and sexual orientation of male and female participants in a BBC Internet Research Project. *Archives of Sexual Behavior*, *36*, 163–176.

Bleidorn, W., Klimstra, T. A., Denissen, J. J. A., Rentfrow, P. J., Potter, J., & Gosling, S. D. (2013). Personality maturation around the world a cross-cultural examination of social-investment theory. *Psychological Science*, *24*, 2530–2540. doi:https://doi.org/10.1177/0956797613498396

Bleidorn, W., Schönbrodt, F., Gebauer, J. E., Rentfrow, P. J., Potter, J., & Gosling, S. D. (2016). To live among like-minded others: Exploring the links between person-city personality fit and self-esteem. *Psychological Science*, *27*, 419–427.

Bonney, R., Shirk, J. L., Phillips, T. B., Wiggins, A., Ballard, H. L., Miller-Rushing, A. J., & Parrish, J. K. (2014). Next steps for citizen science. *Science*, *343*, 1436–1437.

Brady, T. F., Konkle, T., Alvarez, G. A., & Oliva, A. (2008). Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences*, *105*, 14325–14329. doi:https://doi.org/10.1073/pnas.0803390105

Brown, R. W., & Fish, D. (1983). The psychological causality implicit in language. *Cognition*, *14*, 237–273.

Brysbaert, M., Stevens, M., Mandera, P., & Keuleers, E. (2016). How many words do we know? Practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age. *Frontiers in Psychology*, *7*, 1116. doi:https://doi.org/10.3389/fpsyg.2016.01116

Buchanan, T., & Smith, J. L. (1999). Using the Internet for psychological research: Personality testing on the World Wide Web. *British Journal of Psychology*, *90*, 125–144.

Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, *6*, 3–5. doi:https://doi.org/10.1177/1745691610393980

Cadmus-Bertram, L. A., Marcus, B. H., Patterson, R. E., Parker, B. A., & Morey, B. L. (2015). Randomized trial of a Fitbit-based physical activity intervention for women. *American Journal of Preventive Medicine*, *49*, 414–418.

Casler, K., Bickel, L., & Hackett, E. (2013). Separate but equal? a comparison of participants and data gathered via Amazon's Mturk, social media, and face-to-face behavioral testing. *Computers in Human Behavior*, *29*, 2156–2160.

Chetverikov, A., & Upravitelev, P. (2016). Online versus offline: The Web as a medium for response time data collection. *Behavior Research Methods*, *48*, 1086–1099. doi:https://doi.org/10.3758/s13428-015-0632-x

Clark, H. H. (1973). The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior*, *12*, 335–359. doi:https://doi.org/10.1016/S0022-5371(73)80014-3

Condon, D. M., Roney, E., & Revelle, W. (2017). A sapa project update: On the structure of phrased self-report personality items. *Journal of Open Psychology Data*, *5*(1), 3.

Crump, M. J. C., McDonnell, J. V., & Gureckis, T. M. (2013). Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research. *PLoS ONE*, *8*, e57410. doi:https://doi.org/10.1371/journal.pone.0057410

Cushman, F., & Greene, J. D. (2012). Finding faults: How moral dilemmas illuminate cognitive structure. *Social Neuroscience*, *7*, 269–279.

Datadog. (2016). Datadog: Real-time performance monitoring. Retrieved from https://www.datadoghq.com

de Leeuw, J. R., & Motz, B. A. (2016). Psychophysics in a Web browser? Comparing response times collected with JavaScript and Psychophysics Toolbox in a visual search task. *Behavior Research Methods*, *48*, 1–12. doi:https://doi.org/10.3758/s13428-015-0567-2

Dickinson, J. L., Zuckerberg, B., & Bonter, D. N. (2010). Citizen science as an ecological research tool: Challenges and benefits. *Annual Review of Ecology, Evolution, and Systematics*, *41*, 149–172. doi:https://doi.org/10.1146/annurev-ecolsys-102209-144636

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, *54*(4), 86–96. doi:https://doi.org/10.1145/1924421.1924442

Ellis, N. C. (2002). Frequency effects in language processing: A review with implications for theories of implicit and explicit language acquisition. *Studies in Second Language Acquisition*, *24*, 143–188.

Fedorov, V. (2010). Optimal experimental design. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*, 581–589.

Ferrand, L., New, B., Brysbaert, M., Keuleers, E., Bonin, P., Méot, A., . . . Pallier, C. (2010). The French Lexicon Project: Lexical decision data for 38,840 French words and 38,840 pseudowords. *Behavior Research Methods*, *42*, 488–496. doi:https://doi.org/10.3758/BRM.42.2.488

Ferstl, E. C., Garnham, A., & Manouilidou, C. (2011). Implicit causality bias in English: A corpus of 300 verbs. *Behavior Research Methods*, *43*, 124–135. doi:https://doi.org/10.3758/s13428-010-0023-2

Fiser, J., & Aslin, R. N. (2001). Unsupervised statistical learning of higher-order spatial structures from visual scenes. *Psychological Science*, *12*, 499–504.

Fontenelle, G. A., Phillips, A. P., & Lane, D. M. (1985). Generalizing across stimuli as well as subjects: A neglected aspect of external validity. *Journal of Applied Psychology*, *70*, 101–107.

Fortenbaugh, F. C., DeGutis, J., Germine, L. T., Wilmer, J. B., Grosso, M., Russo, K., & Esterman, M. (2015). Sustained attention across the life span in a sample of 10,000 dissociating ability and strategy. *Psychological Science*, *26*, 1497–1510.

Gao, T., Harari, D., Tenenbaum, J., & Ullman, S. (2014). *When computer vision gazes at cognition*. arXiv preprint. arXiv:1412.2672

Garvey, C., & Caramazza, A. (1974). Implicit causality in verbs. *Linguistic Inquiry*, *5*, 459–464.

Gebauer, J. E., Bleidorn, W., Gosling, S. D., Rentfrow, P. J., Lamb, M. E., & Potter, J. (2014). Cross-cultural variations in Big Five relationships with religiosity: A sociocultural motives perspective. *Journal of Personality and Social Psychology*, *107*, 1064–1091. doi:https://doi.org/10.1037/a0037683

Germine, L. T., Duchaine, B., & Nakayama, K. (2011). Where cognitive development and aging meet: Face learning ability peaks after age 30. *Cognition*, *118*, 201–210.

Germine, L. T., Dunn, E. C., McLaughlin, K. A., & Smoller, J. W. (2015). Childhood adversity is associated with adult theory of mind and social affiliation, but not face processing. *PLoS ONE*, *10*, e0129612. doi:https://doi.org/10.1371/journal.pone.0129612

Germine, L., Nakayama, K., Duchaine, B. C., Chabris, C. F., Chatterjee, G., & Wilmer, J. B. (2012). Is the Web as good as the lab? Comparable performance from Web and lab in cognitive/perceptual experiments. *Psychonomic Bulletin & Review*, *19*, 847–857. doi:https://doi.org/10.3758/s13423-012-0296-9

Goodman, J. K., Cryder, C. E., & Cheema, A. (2013). Data collection in a flat world: The strengths and weaknesses of Mechanical Turk samples. *Journal of Behavioral Decision Making*, *26*, 213–224.

Goodman, N. D., Mansinghka, V. K., Roy, D. M., Bonawitz, K., & Tenenbaum, J. B. (2008). Church: A language for generative models. In D. A. McAllester & P. Myllymäki (Eds.), UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (pp. 220–229). Corvallis, OR: AUAI Press.

Goodman, N. D., & Stuhlmüller, A. (2014). The design and implementation of probabilistic programming languages. Retrieved February 23, 2017, from http://dippl.org

Goodman, N. D., & Tenenbaum, J. B. (2014). Probabilistic models of cognition. Retrieved from http://probmods.org

Gordon, P. C., Grosz, B. J., & Gilliom, L. A. (1993). Pronouns, names, and the centering of attention in discourse. *Cognitive Science*, *17*, 311–347.

Gosling, S. D., & Mason, W. (2015). Internet research in psychology. *Annual Review of Psychology*, *66*, 877–902. doi:https://doi.org/10.1146/annurev-psych-010814-015321

Gosling, S. D., Sandy, C. J., John, O. P., & Potter, J. (2010). Wired but not WEIRD: The promise of the Internet in reaching more diverse samples. *Behavioral and Brain Sciences*, *33*, 94–95. doi:https://doi.org/10.1017/S0140525X10000300

Gosling, S. D., Vazire, S., Srivastava, S., & John, O. P. (2004). Should we trust Web-based studies? A comparative analysis of six preconceptions about Internet questionnaires. *American Psychologist*, *59*, 93–104.

Grassegger, H., & Krogerus, M. (2017). The data that turned the world upside down. *Vice Magazine*, *30*. Retrieved from https://motherboard.vice.com/en_us/article/mg9vvn/how-our-likes-helped-trump-win

Greene, J. (2014). Moral tribes: Emotion, reason, and the gap between us and them. New York, NY: Penguin.

Greene, M. J., Kim, J. S., Seung, H. S., & the EyeWirers. (2016). Analogous convergence of sustained and transient inputs in parallel on and off pathways for retinal motion computation. *Cell Reports*, *14*, 1892–1900.

Greenwald, A. G., McGhee, D. E., & Schwartz, J. L. (1998). Measuring individual differences in implicit cognition: the implicit association test. *Journal of Personality and Social Psychology*, *74*, 1464–1480. doi:https://doi.org/10.1037/0022-3514.74.6.1464

Halberda, J., Ly, R., Wilmer, J. B., Naiman, D. Q., & Germine, L. T. (2012). Number sense across the lifespan as revealed by a massive Internet-based sample. *Proceedings of the National Academy of Sciences*, *109*, 11116–11120.

Harari, G. M., Lane, N. D., Wang, R., Crosier, B. S., Campbell, A. T., & Gosling, S. D. (2016). Using smartphones to collect behavioral data in psychological science: Opportunities, practical considerations, and challenges. *Perspectives on Psychological Science*, *11*, 838–854. doi:https://doi.org/10.1177/1745691616650285

Hardy, J., & Scanlon, M. (2009). The science behind luminosity. San Francisco, CA: Lumos Labs.

Hartshorne, J. K. (2008). Visual working memory capacity and proactive interference. *PLoS ONE*, *3*, e2716. doi:https://doi.org/10.1371/journal.pone.0002716

Hartshorne, J. K., Bonial, C., & Palmer, M. (2013a). The VerbCorner Project: Toward an empirically-based semantic decomposition of verbs. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1438–1442). Stroudsburg, PA: Association for Computational Linguistics.

Hartshorne, J. K., Bonial, C., & Palmer, M. (2014). The VerbCorner Project: Findings from Phase 1 of crowd-sourcing a semantic decomposition of verbs. *Proceedings of the Association of Computational Linguistics*, *2*, 397–402.

Hartshorne, J. K., & Germine, L. T. (2015). When does cognitive functioning peak? The asynchronous rise and fall of different cognitive abilities across the life span. *Psychological Science*, *26*, 433–443. doi:https://doi.org/10.1177/0956797614567339

Hartshorne, J. K., & Jennings, M. (Organizers). (2017). First annual Pushkin developer's conference, Chestnut Hill, MA.

Hartshorne, J. K., Leeuw, J. R. D., Germine, L., Reinecke, K., & Jennings, M. (2018a). *Massive online experiments in cognitive science*. Workshop at the Annual Meeting of the Cognitive Science Society, Madison, WI.

Hartshorne, J. K., O'Donnell, T. J., & Tenenbaum, J. B. (2015). The causes and consequences explicit in verbs. *Language, Cognition, and Neuroscience*, *30*, 716–734.

Hartshorne, J. K., & Snedeker, J. (2013). Verb argument structure predicts implicit causality: The advantages of finer-grained semantics. *Language and Cognitive Processes*, *28*, 1474–1508.

Hartshorne, J. K., Sudo, Y., & Uruwashi, M. (2013b). Are implicit causality pronoun resolution biases consistent across languages and cultures? *Experimental Psychology*, *60*, 179–196.

Hartshorne, J. K., Tenenbaum, J. B., & Pinker, S. (2018b). A critical period for second language acquisition: Evidence from 2/3 million English speakers. *Cognition*, *177*, 263–277. doi:https://doi.org/10.1016/j.cognition.2018.04.007

Hauser, D. J., & Schwarz, N. (2016). Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants. *Behavior Research Methods*, *48*, 400–407. doi:https://doi.org/10.3758/s13428-015-0578-z

Hauser, M. (2006). Moral minds: How nature designed our universal sense of right and wrong. New York, NY: Ecco/HarperCollins.

Hauser, M. D., Young, L., & Cushman, F. (2008). Reviving Rawls's linguistic analogy: Operative principles and the causal structure of moral actions. In W. Sinnott-Armstrong (Ed.), Moral psychology: Vol. 2. The cognitive science of morality: Intuition and diversity (pp. 107–143). Cambridge, MA: MIT Press.

Haworth, C. M. A., Harlaar, N., Kovas, Y., Davis, O. S. P., Oliver, B. R., Hayiou-Thomas, M. E., . . . Plomin, R. (2007). Internet cognitive testing of large samples needed in genetic research. *Twin Research and Human Genetics*, *10*, 554–563. doi:https://doi.org/10.1375/twin.10.4.554

Henrich, J., Heine, S. J., & Norenzayan, A. (2010). The weirdest people in the world? *Behavioral and Brain Sciences*, *33*, 61–83. doi:https://doi.org/10.1017/S0140525X0999152X

Hilbig, B. E. (2016). Reaction time effects in lab- versus Web-based research: Experimental evidence. *Behavior Research Methods*, *48*, 1718–1724. doi:https://doi.org/10.3758/s13428-015-0678-9

Hlushko, E., Kaper, R., Larkin, S., Braimbridge, A., Grisogono, G., Menichelli, J., . . . Stewart, J. (2018). webpack (Software). Retrieved from https://webpack.js.org/

Honing, H., & Ladinig, O. (2008). The potential of the Internet for music perception research: A comment on lab-based versus web-based studies. *Empirical Musicology Review*, *3*, 4–7.

Howe, J. (2006). The rise of crowdsourcing. *Wired Magazine*, *14*(6), 1–4.

Huber, B., Reinecke, K., & Gajos, K. Z. (2017). The effect of performance feedback on social media sharing at volunteer-based online experiment platforms. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 1882–1886).

Ipeirotis, P. G. (2010). Demographics of Mechanical Turk (NYU Working Paper CEDER-10-01). New York, NY: New York University, Leonard N. Stern School of Business.

ITU Telecommunication Development Sector. (2017). ICT facts and figures. Retrieved from https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2017.pdf

Johnson, J. A. (2005). Ascertaining the validity of individual protocols from web-based personality inventories. *Journal of Research in Personality*, *39*, 103–129.

Johnson, W., Logie, R. H., & Brockmole, J. R. (2010). Working memory tasks differ in factor structure across age cohorts: Implications for dedifferentiation. *Intelligence*, *38*, 513–528.

Judd, C. M., Westfall, J., & Kenny, D. A. (2012). Treating stimuli as a random factor in social psychology: A new and comprehensive solution to a pervasive but largely ignored problem. *Journal of Personality and Social Psychology*, *103*, 54–69.

Jun, E., Hsieh, G., & Reinecke, K. (2017). Types of motivation affect study selection, attention, and dropouts in online experiments. In C. Lampe, J. Nichols, K. Karahalios, G. Fitzpatrick, U. Lee, A. Monroy-Hernandez, & W. Stuerzlinger (Eds.), Proceedings of

ACM Human–Computer Interaction (Vol. 1, Article 56). New York, NY: ACM Press.

Kajonius, P. J., & Johnson, J. (2018). Sex differences in 30 facets of the five factor model of personality in the large public (n = 320,128). *Personality and Individual Differences*, *129*, 126–130.

Kaufman, A. S. (2001). WAIS-III IQs, Horn's theory, and generational changes from young adulthood to old age. *Intelligence*, *29*, 131–167.

Kehler, A., & Rohde, H. (2013). A probabilistic reconciliation of coherence-driven and centering-driven theories of pronoun interpretation. *Theoretical Linguistics*, *39*, 1–37.

Keller, K., Troesch, L. M., & Grob, A. (2015). First-born siblings show better second language skills than later born siblings. *Frontiers in Psychology*, *6*, 705. doi:https://doi.org/10.3389/fpsyg.2015.00705

Keuleers, E., Stevens, M., Mandera, P., & Brysbaert, M. (2015). Word knowledge in the crowd: Measuring vocabulary size and word prevalence in a massive online experiment. *Quarterly Journal of Experimental Psychology*, *68*, 1665–1692. doi:https://doi.org/10.1080/17470218.2015.1022560

Killingsworth, M. A., & Gilbert, D. T. (2010). A wandering mind is an unhappy mind. *Science*, *330*, 932–932.

Kim, J. S., Greene, M. J., Zlateski, A., Lee, K., Richardson, M., Turaga, S. C., . . . the EyeWirers. (2014). Space–time wiring specificity supports direction selectivity in the retina. *Nature*, *509*, 331–336.

Kleiner, M., Brainard, D., & Pelli, D. (2007). What's new in Psychtoolbox-3? *Perception*, *36*(ECVP Abstract Suppl), 14.

Krantz, J. H. (2001). Stimulus delivery on the web: What can be presented when calibration isn't possible. *Dimensions of Internet Science*, 113–130.

Kumar, A., Killingsworth, M. A., & Gilovich, T. (2014). Waiting for merlot: Anticipatory consumption of experiential and material purchases. *Psychological Science*, *25*, 1924–1931.

Levay, K. E., Freese, J., & Druckman, J. N. (2016). The demographic and political composition of Mechanical Turk samples. *Sage Open*, *6*(1). doi:https://doi.org/10.1177/2158244016636433

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, *27*, 986–1005.

Lippa, R. A. (2008). Sex differences and sexual orientation differences in personality: Findings from the BBC Internet survey. *Archives of Sexual Behavior*, *37*, 173–187.

Logie, R. H., & Maylor, E. A. (2009). An Internet study of prospective memory across adulthood. *Psychology and Aging*, *24*, 767–774.

Manning, J. T., & Fink, B. (2008). Digit ratio (2d:4d), dominance, reproductive success, asymmetry, and sociosexuality in the BBC Internet study. *American Journal of Human Biology*, *20*, 451–461.

Mason, W., & Suri, S. (2012). Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, *44*, 1–23. doi:https://doi.org/10.3758/s13428-011-0124-6

Maylor, E. A., & Logie, R. H. (2010). A large-scale comparison of prospective and retrospective memory development from childhood to middle age. *Quarterly Journal of Experimental Psychology*, *63*, 442–451.

Meyerson, P., & Tryon, W. W. (2003). Validating Internet research: A test of the psychometric equivalence of Internet and in-person samples. *Behavior Research Methods, Instruments, & Computers*, *35*, 614–620.

Miller, G. (2012). The smartphone psychology manifesto. *Perspectives on Psychological Science*, *7*, 221–237.

Montgomery-Downs, H. E., Insana, S. P., & Bond, J. A. (2012). Movement toward a novel activity monitoring device. *Sleep and Breathing*, *16*, 913–917.

Morton, J. (1969). Interaction of information in word recognition. *Psychological Review*, *76*, 165–178. doi:https://doi.org/10.1037/h0027366

Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy, 2008 (SP 2008)* (pp. 111–125). Piscataway, NJ: IEEE Press.

Nosek, B. A., Banaji, M., & Greenwald, A. G. (2002). Harvesting implicit group attitudes and beliefs from a demonstration web site. *Group Dynamics: Theory, Research, and Practice*, *6*, 101–115. doi:https://doi.org/10.1037/1089-2699.6.1.101

Ouyang, L., Tessler, M. H., Ly, D., & Goodman, N. D. (2018). webppl-oed: A practical optimal experiment design system. In C. Kalish, M. Rau, J. Zhu, & T. T. Rogers (Eds.), Proceedings of the 40th Annual Meeting of the Cognitive Science Society (pp. 2192–2197). Austin, TX: Cognitive Science Society.

Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, *5*, 411–419.

Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). *WebGazer: Scalable webcam eye tracking using user interactions*. Article presented at the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016), New York, NY.

Peters, M., Reimers, S., & Manning, J. T. (2006). Hand preference for writing and associations with selected demographic and behavioral variables in 255,100 subjects: The BBC Internet study. *Brain and Cognition*, *62*, 177–189.

Picard, R. W., Fedor, S., & Ayzenberg, Y. (2015). Multiple arousal theory and daily-life electrodermal activity asymmetry. *Emotion Review*, *8*, 62–75. doi:https://doi.org/10.1177/1754073914565517

Pinet, S., Zielinski, C., Mathot, S., Dufau, S., Alario, F.-X., & Longcamp, M. (2017). Measuring sequences of keystrokes with jspsych: Reliability of response times and interkeystroke intervals. *Behavior Research Methods*, *49*, 1163–1176.

Poesio, M., Chamberlain, J., Kruschwitz, U., Robaldo, L., & Ducceschi, L. (2013). Phrase detectives: Utilizing collective intelligence for Internet-scale language resource creation. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, *3*, 1–44. doi:https://doi.org/10.1145/2448116.2448119

Poh, M.-Z., Swenson, N. C., & Picard, R. W. (2010). A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *IEEE Transactions on Biomedical Engineering*, *57*, 1243–1252.

Rand, D. G. (2012). The promise of Mechanical Turk: How online labor markets can help theorists run behavioral experiments. *Journal of Theoretical Biology*, *299*, 172–179.

Ratcliff, R., Gomez, P., & McKoon, G. (2004). A diffusion model account of the lexical decision task. *Psychological Review*, *111*, 159–182. doi:https://doi.org/10.1037/0033-295X.111.1.159

Reed, J., Raddick, M. J., Lardner, A., & Carney, K. (2013). An exploratory factor analysis of motivations for participating in zooniverse, a collection of virtual citizen science projects. In *46th Hawaii International Conference on System Sciences (HICSS) 2013* (pp. 610–619). Piscataway, NJ: IEEE Press.

Reimers, S., & Stewart, N. (2015). Presentation and response timing accuracy in Adobe Flash and HTML5/JavaScript Web experiments. *Behavior Research Methods*, *47*, 309–327. doi:https://doi.org/10.3758/s13428-014-0471-1

Reinecke, K., & Gajos, K. Z. (2014). Quantifying visual preferences around the world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 11–20). New York, NY: ACM Press.

Reinecke, K., & Gajos, K. Z. (2015). LabintheWild: Conducting large-scale online experiments with uncompensated samples. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing* (pp. 1364–1378). New York, NY: ACM Press.

Reips, U.-D. (2002). Standards for Internet-based experimenting. *Experimental Psychology*, *49*, 243–256. doi:https://doi.org/10.1027/1618-3169.49.4.243

Ren, Z., Meng, J., Yuan, J., & Zhang, Z. (2011). Robust hand gesture recognition with Kinect sensor. In *Proceedings of the 19th ACM International Conference on Multimedia* (pp. 759–760). New York, NY: ACM Press.

Rife, S. C., Cate, K. L., Kosinski, M., & Stillwell, D. (2016). Participant recruitment and data collection through Facebook: The role of personality factors. *International Journal of Social Research Methodology*, *19*, 69–83.

Riley, E., Okabe, H., Germine, L., Wilmer, J., Esterman, M., & DeGutis, J. (2016). Gender differences in sustained attentional control relate to gender inequality across countries. *PLoS ONE*, *11*, e165100. doi:https://doi.org/10.1371/journal.pone.0165100

Rudolph, U., & Forsterling, F. (1997). The psychological causality implicit in verbs: A review. *Psychological Bulletin*, *121*, 192–218.

Salganik, M. J., Dodds, P. S., & Watts, D. J. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, *311*, 854–856.

Salthouse, T. A. (2004). What and when of cognitive aging. *Current Directions in Psychological Science*, *13*, 140–144.

Salthouse, T. A. (2009). When does age-related cognitive decline begin? *Neurobiology of Aging*, *30*, 507–514.

Schneider, W., Eschman, A., & Zuccolotto, A. (2002). E-Prime user's guide. Pittsburgh, PA: Psychology Software Incorporated.

Scott, K., & Schulz, L. (2017). Lookit (part 1): A new online platform for developmental research. *Open Mind*, *1*(1), 4–14. doi:https://doi.org/10.1162/opmi_a_00002

Semmelmann, K., & Weigelt, S. (2017). Online psychophysics: Reaction time effects in cognitive experiments. *Behavior Research Methods*, *49*, 1241–1260.

Settles, B. (2012). Active learning. San Rafael, CA: Morgan & Claypool.

Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistic* (Vol. 1, pp. 1848–1858). Stroudsburg, PA: Association for Computational Linguistics.

Shapiro, D. N., Chandler, J., & Mueller, P. A. (2013). Using Mechanical Turk to study clinical populations. *Clinical Psychological Science*, *1*, 213–220. doi:https://doi.org/10.1177/2167702612469015

Silvertown, J. (2009). A new dawn for citizen science. *Trends in Ecology and Evolution*, *24*, 467–471.

Simcox, T., & Fiez, J. A. (2014). Collecting response times using Amazon Mechanical Turk and Adobe Flash. *Behavior Research Methods*, *46*, 95–111. doi:https://doi.org/10.3758/s13428-013-0345-y

Simpson, R., Page, K. R., & De Roure, D. (2014). Zooniverse: Observing the world's largest citizen science platform. In *Proceedings of the 23rd International Conference on World Wide Web Companion* (pp. 1049–1054). New York, NY: ACM Press.

Skitka, L. J., & Sargis, E. G. (2006). The Internet as psychological laboratory. *Annual Review of Psychology*, *57*, 529–555.

Slote, J., & Strand, J. F. (2016). Conducting spoken word recognition research online: Validation and a new timing method. *Behavior Research Methods*, *48*, 553–566. doi:https://doi.org/10.3758/s13428-015-0599-7

Smith, S. M., Roster, C. A., Golden, L. L., & Albaum, G. S. (2016). A multi-group analysis of online survey respondent data quality: Comparing a regular USA consumer panel to MTurk samples. *Journal of Business Research*, *69*, 3139–3148.

Soto, C. J., John, O. P., Gosling, S. D., & Potter, J. (2011). Age differences in personality traits from 10 to 65: Big Five domains and facets in a large cross-sectional sample. *Journal of Personality and Social Psychology*, *100*, 330–348. doi:https://doi.org/10.1037/a0021717

Stewart, N., Chandler, J., & Paolacci, G. (2017). Crowdsourcing samples in cognitive science. *Trends in Cognitive Sciences*, *21*, 736–748. doi:https://doi.org/10.1016/j.tics.2017.06.007

Stewart, N., Ungemach, C., Harris, A. J., Bartels, D. M., Newell, B. R., Paolacci, G., & Chandler, J. (2015). The average laboratory samples a population of 7,300 Amazon Mechanical Turk workers. *Judgment and Decision Making*, *10*, 479–491.

Stieger, S., Lewetz, D., & Reips, U.-D. (2017). Can smartphones be used to bring computer-based tasks from the lab to the field? A mobile experience-sampling method study about the pace of life. *Behavior Research Methods*. Advance online publication. doi:https://doi.org/10.3758/s13428-017-0991-6

Streeter, M. (2015). *Mixture modeling of individual learning curves*. Article presented at the International Conference on Educational Data Mining Society, Madrid, Spain.

Suchow, J. (2018). Dallinger (Software). Retrieved from https://github.com/Dallinger

Sullivan, B. L., Aycrigg, J. L., Barry, J. H., Bonney, R. E., Bruns, N., Cooper, C. B., . . . Kelling, S. (2014). The eBird enterprise: An integrated approach to development and application of citizen science. *Biological Conservation*, *169*, 31–40.

Susilo, T., Germine, L. T., & Duchaine, B. (2013). Face recognition ability matures late: Evidence from individual differences in young adults. *Journal of Experimental Psychology: Human Perception and Performance*, *39*, 1212–1217. doi:https://doi.org/10.1037/a0033469

Trull, T. J., & Ebner-Priemer, U. (2013). Ambulatory assessment. *Annual Review of Clinical Psychology*, *9*, 151–176. doi:https://doi.org/10.1146/annurev-clinpsy-050212-185510

Tucker-Drob, E. M. (2011). Global and domain-specific changes in cognition throughout adulthood. *Developmental Psychology*, *47*, 331–343.

Videla, A., & Williams, J. J. W. (2012). RabbitMQ in action: Distributed messaging for everyone. Shelter Island, NY: Manning.

Willett, K. W., Galloway, M. A., Bamford, S. P., Lintott, C. J., Masters, K. L., Scarlata, C., . . . Smith, A. M. (2017). Galaxy Zoo: Morphological classifications for 120,000 galaxies in HST legacy imaging. *Monthly Notices of the Royal Astronomical Society*, *464*, 4176–4203.

Wilson, R. E., Gosling, S. D., & Graham, L. T. (2012). A review of Facebook research in the social sciences. *Perspectives on Psychological Science*, *7*, 203–220. doi:https://doi.org/10.1177/1745691612442904

Xu, A. R. (2018). Scholars have data on millions of Facebook users: Who's guarding it? *New York Times*.

Zimmer, M. (2016). OkCupid study reveals the perils of big-data science. *Wired Magazine*. Retrieved from https://www.wired.com/2016/05/okcupid-study-reveals-perils-big-data-science/