Behav Res (2016) 48:1086–1099
DOI 10.3758/s13428-015-0632-x

Online versus offline: The Web as a medium for response time data collection

Andrey Chetverikov[1,2] · **Philipp Upravitelev**[3]

Published online: 14 July 2015

Abstract The Internet provides a convenient environment for data collection in psychology. Modern Web programming languages, such as JavaScript or Flash (ActionScript), facilitate complex experiments without the necessity of experimenter presence. Yet there is always a question of how much noise is added due to the differences between the setups used by participants and whether it is compensated for by increased ecological validity and larger sample sizes. This is especially a problem for experiments that measure response times (RTs), because they are more sensitive (and hence more susceptible to noise) than, for example, choices per se. We used a simple visual search task with different set sizes to compare laboratory performance with Web performance. The results suggest that although the locations (means) of RT distributions are different, other distribution parameters are not. Furthermore, the effect of experiment setting does not depend on set size, suggesting that task difficulty is not important in the choice of a data collection method. We also collected an additional online sample to investigate the effects of hardware and software diversity on the accuracy of RT data. We found that the high diversity of browsers, operating systems, and CPU performance may have a detrimental effect, though it can partly be compensated for by increased sample sizes and trial numbers. In sum, the findings show that Web-based experiments are an acceptable source of RT data, comparable to a common keyboard-based setup in the laboratory.

Keywords Response times · Web · Online · JavaScript · Task difficulty · Visual search · Reaction times

Online studies provide an inexpensive method for behavioral data collection from large samples of participants. An experimenter sets up a study using either a custom script or a prearranged setting, such as PsiTurk (McDonnell et al., 2014) or jsPsych (de Leeuw, 2015), advertises the experiment online, and enjoys a steady flow of data (or at least hopes to enjoy one). The question that has bothered scientists since the dawn of the Web era is "How reliable are such studies?" Especially troubling are experiments that include response time (RT) data collection. A number of recent studies have tackled the issue by trying to see whether RTs can be reliably measured online.

The first line of studies was aimed at testing the reliability of hardware by using precise key-pressing equipment instead of human participants. Neath and colleagues (Neath, Earle, Hallett, & Surprenant, 2011) used a photodiode connected to a solenoid to test different Apple systems with different presentation software (Psychophysics Toolbox, Java, JavaScript, or Flash). Upon stimulus onset, the photodiode detected changes in display luminosity and triggered a solenoid that pressed a key. They concluded that Psychophysics Toolbox provided the highest accuracy in measuring RTs, whereas JavaScript, Java, and Flash showed higher standard deviations (5–10 ms) and positive skewness. On the other hand, Simcox and Fiez (2014) used a macro that generated keypresses without a keyboard and demonstrated that RT measurement errors

Electronic supplementary material The online version of this article (doi:10.3758/s13428-015-0632-x) contains supplementary material, which is available to authorized users.

✉ Andrey Chetverikov
a.chetverikov@psy.spbu.ru

[1] Department of Psychology, Saint Petersburg State University, nab. Makarova 6, Saint Petersburg, Russia 191002

[2] Cognitive Research Lab, Russian Academy of National Economy and Public Administration, Moscow, Russia

[3] ConsultantPlus, Moscow, Russia

due to Flash were actually negligible. Reimers and Stewart (2015) compared RTs generated with the Black Box Toolkit (www.blackboxtoolkit.com) and collected with Flash and JavaScript. They showed that both methods were quite comparable and had delays of about 30 ms and less than 10-ms standard deviations (*SD*s). Schubert et al. used their own Flash-based scripting library, ScriptingRT (Schubert, Murteira, Collins, & Lopes, 2013). ScriptingRT was slightly slower and more deviant than most of the offline software, but in absolute terms the *SD*s were less than 5 ms. Keller et al. used repetitive keypress events generated by the operating system when a key was held down to test their Java-based Webexp software (Keller, Gunasekharan, Mayo, & Corley, 2009). They found that although there was a delay at shorter time intervals, it was not big (about 12 ms on 300 ms of repetitive keypresses), and the *SD*s were less than 5 ms in most of the conditions. The obtained results look promising: Whereas most studies have shown that online RT measurements have delays, the *SD*s are only slightly larger than would be expected from offline measurements. However, even though the software used for programming the experiment might not be a problem for online studies by itself, it is possible that large variability in the participants' hardware, operating systems, browsers, and environment might add excessive noise to the collected data.

Similarly promising findings were reported by Damian (2010), who estimated the effect of response device polling rate on the deviation of the obtained RTs from the true mean. The typical USB keyboard is polled with a frequency of 125 Hz. Thus, the measurement error due to the infrequent polling would range from 0 to 8 ms. Other factors, such as operating system settings, could further decrease or increase the polling rate. Assuming that the errors due to infrequent polling are uniformly distributed, Damian (2010) found that such errors are negligible when compared to the variation between and within participants. These results could be generalized to any kind of uniformly distributed noise. However, it is unlikely that all sources of noise would have uniform distributions. For example, a common caution against online studies is that participants may be distracted by some external factors (e.g., a pet suddenly in need for attention, or a phone call). Such noise would create a few relatively long responses that would not be described by a uniform distribution. Moreover, sources of errors may not be independent, further cautioning against the generalization of the results obtained by Damian (2010).

A second line of studies has compared participants online and offline to see whether such concerns were warranted. There have been several successful attempts to replicate effects previously observed offline using online samples: Reimers and Maylor (2005) replicated the effects of age differences on task-switching performance measured with RTs; Keller et al. (2009) replicated RT findings from a self-paced reading study; Simcox and Fiez (2014) reproduced flanker and lexical decision effects; and Crump et al. (2013) replicated Stroop, task-switching, flanker, Simon, visual-cuing, and masked-priming effects. Although these findings show the usefulness of online studies, they do not allow for a direct comparison between offline and online results, because they were obtained in different studies.

To our best knowledge, only two studies have made a direct attempt to collect data from participants both online and offline. Reimers and Stewart (2007) were the first to show that although the RTs from online Flash, offline Flash, and an offline C script were different, with C being the fastest and online Flash the slowest, the *SD*s were the same. This finding is important, because usually it is the differences between conditions that matter, and not the absolute times. The probability of detecting such differences does not depend on the absolute values of the RTs, but on the difference between them and their *SD*s. Given that the RT errors are systematic and that there is no time drift, delays in RTs are not really important, except if they are an object of research in their own right. Schubert et al. (2013) compared congruent and incongruent conditions in a Stroop task between Flash and offline software. Although they found a main effect of software, with Flash producing slower RTs than the offline software, there was no interaction between the software and congruency conditions. The confidence intervals for RTs in the online and offline conditions were quite similar, as well, though they were not compared directly. The present article continues the line of studies aimed at testing the reliability of online RT measurement by comparing online and offline samples. With the exception of Reimers and Stewart (2007), previous studies have not tried to separate the effects of hardware, software, and environment on RTs. To fill this gap, the experiment reported here was based on a within-subjects design. All participants took part in the same study in four settings: "Response box"—offline with PsychoPy (Peirce, 2007) and a Cedrus RT-830 button box; "Keyboard"—offline with PsychoPy and a simple keyboard; "Web from lab"—online with JavaScript within the lab; and "Web from home"—online with JavaScript outside the lab.

Comparison of the RTs obtained with the Cedrus RT-830 response box and the RTs obtained with a keyboard allowed us to estimate the effect of hardware, whereas other settings allowed us to see the impact of changing the software (PsychoPy vs. JavaScript in the lab) and environment (in vs. out of the lab). In addition to simply measuring RTs and analyzing their distribution, we also introduced a difficulty factor, to see whether the differences between settings depended on the difficulty of the task itself. We reasoned that increasing the task difficulty would increase the impact of individual differences, and the effects of hardware and software would become less noticeable.

Following the main part of the experiment, we collected an additional large sample of participants online to estimate the variability of RT data provided by online studies. Previous studies had tackled this question by comparing a limited number of work stations (e.g., Reimers & Stewart, 2015; Simcox & Fiez, 2014). In contrast, we used a real online sample to assess the impacts of operating system, browsers, CPU, GPU, and amount of RAM.

Study 1

Method

Participants Twenty participants were recruited from the staff and/or students at the Faculty of Psychology, St. Petersburg State University (ten male, ten female; 19 to 37 years old, M = 23.7). They were not paid for their participation.

Procedure The order of settings was balanced between participants using a Latin square design. We ran each setting on a separate day, and the time between settings varied from one to seven days, depending on the availability of participants. A very simple visual search task was used. For each trial, two, four, or six colored squares (1° of visual angle) appeared at the horizontal midline (equally spaced, 3° distance between the centers of adjacent squares) centered on the screen. We did not use a head- and chin-rest, although the observer distance was controlled to some extent by positioning a chair at a specific distance (40 cm). Observers had to locate a square of a specific color and press the "a," "s," "d," "h," "j," or "k" key to indicate the target position. The keys were explained to observers before the experiment. Observers were further asked to press "a," "s," and "d" with the ring, middle, or index finger of their left hand, and "h," "j," or "k" with the index, middle, or ring finger of their right hand. The same labels were put on the Cedrus response box keys, so that there were three keys for each hand.

The experiment consisted of three blocks of trials. The number of squares varied between the blocks: two squares in the first block ("d" and "h" were used), four squares in the second ("s," "d," "h," and "j"), and six squares in the third (all keys were used).

The target colors varied within the blocks. A total of six colors were used, with equidistant hues (HSV color space: 0-, 60-, 120-, 180-, 240-, and 360-deg hues, with saturation and value set to 1.0). Each target color was repeated 24 times within each block, and its position was counterbalanced, resulting in a total of 432 trials in the whole experiment. Distractor colors were chosen randomly. For each color, trials were grouped in series, and the target color was presented before each series for 5 s, then for 0.5 s before each trial. Thus, for each set size, participants first completed a series of trials with one color, then a series of trials with another color, and so on.

Materials and apparatus For the lab version of the experiment we used a PC with Intel Pentium 4, 3.00-GHz CPU, 2 GB RAM, and NVIDIA GeForce 6200 video card running the Lubuntu 12.10 operating system. It was connected to a 19-in. Acer V193 display with a 1,280 × 1,024 pixel resolution and 60-Hz refresh rate. Response times were collected with a Genius SlimStar 100 USB keyboard ("Keyboard" and "Web from lab" conditions) or with a Cedrus RB-830 response box ("Response box" condition). The experiment script was written with PsychoPy (Peirce, 2007). In the PsychoPy implementation, the stimulus presentation was synchronized with the display refresh rate and followed the simple routine: drawing of stimuli to the back buffer of the video card, flipping the back and forward visual buffers so that the stimuli were drawn on the screen, resetting the timer, and waiting for the response. The reliability of visual presentation with PsychoPy has been extensively tested, and no problems were found under standard conditions (Garaizar & Vadillo, 2014). The keyboard polling rate was not adjusted, so a usual 125-Hz polling rate was used. The Cedrus RB-830 response box was connected through a USB port, as well. Given that the response box has an internal timer, it is not affected by the polling rate of USB port. However, it may be affected by variations in the time needed to send the timer reset command from the computer to the box.

The online version of the experiment was written in JavaScript using the jQuery library (http://jquery.com, version 1.11.0) to control the presentation of stimuli and RT collection. The stimulus sizes were set in pixels so that in the lab they would be the same as the stimuli created with PsychoPy. Although in principle it is possible to ask observers in online studies to report their viewing distance and display size to make it possible to vary the stimulus sizes accordingly, it is unlikely that many will agree to measure their display, and there is no way to control the viewing distance. Thus, we did not aim to do so.

For the online setting, within the laboratory the participants used the Chromium browser (version 22.0.1299). Outside the laboratory they used different browsers (14 Chromium-based, 4 Firefox, 1 Internet Explorer, and 1 Safari) and different operating systems (15 Windows 7, 2 Windows 8, 2 OS X, and 1 Linux).

Table 1 summarizes the major sources of errors in RT measurements and the differences between the conditions we used. The low-level factors, such as operating system and hardware drivers, influence all RT measurements (Plant & Quinlan, 2013). Even in the case of the response box, delays

Table 1 Major sources of noise in RT measurements in different conditions

	Response Box	Keyboard	Web From Lab	Web From Home
OS and drivers	+	+	+	+
Response polling rate	−	+	+	+
Synchronization with display refresh rate	−	−	+	+
Browser and JavaScript	−	−	+	+
Heterogeneity of hardware and environment	−	−	−	+
Within- and between-observer variation	+	+	+	+

Plus and minus signs indicate that the source of error was present or absent in the corresponding condition

in communication between the response box and the computer could add variation to RTs. The second major factor, response device polling rate, is negligibly small for specialized response boxes. A comparison between the "Response box" and "Keyboard" conditions allowed us to test the effect of that factor. Three more factors—synchronization with display refresh rate, browser, and JavaScript Issues—could be tested by comparing the "Web from lab" and "Keyboard" conditions. JavaScript and Web browsers lack capabilities for reliable synchronization with the display refresh rate and could have additional unknown variability in code execution times or timer responses. The "Web from lab" and "Keyboard" conditions were run with exactly the same software; thus, the difference between them would allow us to estimate the effect of JavaScript implementation as opposed to PsychoPy implementation. The next major source of noise, arguably raising the most concerns regarding online studies, is variation in the hardware and software used by the participants in online experiments. A comparison between the "Web from lab" and "Web from home" groups allowed us to estimate the effect of this factor.

Results

Accuracy Data with RTs above 10 s were discarded as outliers (there was one such answer in each condition). The descriptive statistics as a function of set size and condition are shown in Table 2. Accuracy was analyzed with repeated measures analyses of variance (ANOVAs) using data aggregated by participant, setting, and set size. A main effect of set size, $F(2, 38) = 17.43$, $p < .001$, $\eta^2_G = .063$, was accompanied by a main effect of setting, $F(3, 57) = 4.20$, $p = .009$, $\eta^2_G = .036$, but not by their interaction, $F(6, 114) = 0.47$, $p = .833$, $\eta^2_G = .003$.

Figure 1 shows that the accuracy decreased as expected with the increasing set size. Pairwise *t*tests with Benjamini and Hochberg (1995) corrections indicated that the accuracy was higher for the "Keyboard" setting than for any other condition: "Keyboard" provided higher accuracy than "Response box," $t(59) = 4.36$, $p < .001$, "Web from lab," $t(59) = 2.27$, $p = .041$, and "Web from home," $t(59) = 3.87$, $p < .001$. Participants with the "Web from lab" setting were more accurate than those with the "Response box" setting, $t(59) = 2.45$, $p = .035$.

Response times: Means and standard deviations The means and distributions of RTs are shown in Fig. 2. Several aspects of the data are visible at this point. First, for all settings there is an expected increase in RTs with the increase of set size. Its magnitude is close in all settings, although the difference between settings seems to decrease with increasing set size. Second, the shapes of the distributions in different settings are also quite similar. Increasing the set size, on the other hand, clearly makes the distribution wider and shifts its location. Lastly, there seems to be a large and unexpected increase of the variance in the "Web from lab" condition for set size 4. Further analysis demonstrated that it stemmed from a large number of slow RTs from one participant and, after the exclusion of that participant, *SD*s became similar to those in the other conditions ($M = 417$, $SD = 79$, for set size 2; $M = 536$, $SD = 116$, for set size 4; and $M = 604$, $SD = 98$, for set size 6).

Table 2 Descriptive statistics for RTs and accuracy, aggregated by participants

	Response Box			Keyboard			Web From Lab			Web From Home		
Set Size	2	4	6	2	4	6	2	4	6	2	4	6
RT mean	319	473	535	378	524	565	443	577	614	429	545	608
RT *SD*	66	114	78	60	94	46	133	215	105	72	94	87
Share of errors	4.3	6.0	6.4	2.5	3.8	5.1	3.7	4.5	5.4	4.0	5.2	6.2

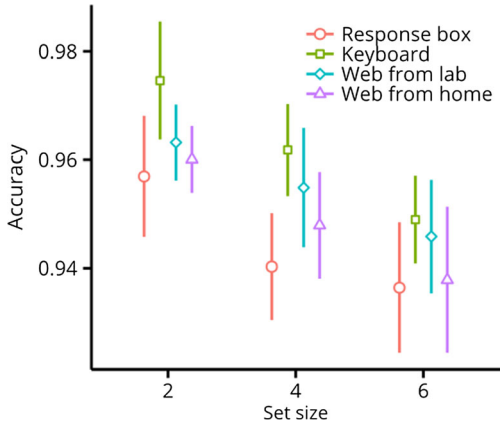

Fig.1 Accuracy by condition and set size. Lines show 95 % confidence intervals

However, whether these data were included or not did not affect the results reported later, and thus we report the analyses without exclusion of that participant.

We used a simple repeated measures ANOVA to look for the effects of setting and set size with RTs aggregated for each participant. We analyzed RTs without any transformation (such as logarithmic transformation), because we were interested not only in the means but in the *SD*s as well. Transformed data may also have reduced *SD*s, and the effects of transformation may be differ between settings, thus obscuring the potentially existing differences. The ANOVA indicated significant main effects of set size, $F(2, 38) = 142.44$, $p < .001$, $\eta^2_G = .373$, and setting, $F(3, 57) = 8.30$, $p < .001$, $\eta^2_G = .126$, but not of their interaction, $F(6, 114) = 1.18$, $p = .324$, $\eta^2_G = .006$. We then analyzed the *SD*s of RTs in a similar fashion and found only an effect of set size, $F(2, 38) = 5.70$, $p = .007$, $\eta^2_G = .026$, but not setting, $F(3, 57) = 1.29$, $p = .286$, $\eta^2_G = .021$, or their interaction, $F(6, 114) = 1.64$, $p = .143$, $\eta^2_G = .016$. A paired comparison of the theoretically most different settings, "Response box" and "Web from home," did not yield significant differences when each set size was treated separately or when the data were aggregated (*p*s > .110).

We then applied linear mixed-effects regression with the lme4 package in R (Bates, Maechler, Bolker, & Walker, 2013), treating set size as a continuous predictor and participant as a random effect.[1] If the influence of setting decreased with increasing task difficulty, then the slopes of the set size effect should be different between the settings. Sequential difference contrasts were used so that each next setting was compared to the previous one, with the settings sorted by mean RT. That is, we compared "Response box" to "Keyboard," "Keyboard" to "Web from lab," and "Web from lab" to "Web from home." lme4 does not provide *p* values for linear mixed-effects regression, since it is not clear how to estimate the degrees of freedom, but *t* values > 2 roughly correspond to "significant" differences (see, e.g., Baayen, Davidson, & Bates, 2008).

The results demonstrated differences on a base level (set size = 2), with RTs being smaller for "Response box" than for "Keyboard," $B = 76.43$ (9.96), $t = 7.67$; for "Keyboard" than for "Web from lab," $B = 72.00$ (9.94), $t = 7.24$; and for "Web from home" than for "Web from lab," $B = 26.79$ (9.98), $t = 2.68$. We observed a significant decrease in the set size slope from the "Response box" to the "Keyboard" setting, $B = 7.51$ (2.31), $t = 3.24$. Thus, the difference in means between these settings became smaller with increased set size. Comparisons of the set size slopes for "Keyboard" versus "Web from lab" and for "Web from lab" versus "Web from home" did not yield significant effects.

Response times: Analyses of distributions We analyzed the distribution parameters of RTs with the gamlss package in R (Rigby & Stasinopoulos, 2005). The distribution parameters were estimated separately for each participant in each combination of set size and setting. Response times were fitted best by a skew-*t* distribution (AIC = 1,627), followed by an exponential Gaussian distribution (AIC = 1,643), which in turn fit better than either a log-normal (AIC = 1,676) or a normal (AIC = 1,775) distribution. The ex-Gaussian is a mix of a Gaussian and an exponential distribution, defined by three parameters: μ_{EG} and σ_{EG} correspond to the mean and standard deviation of the Gaussian part, respectively, and τ_{EG} corresponds to the rate of exponential decay.

Figure 3 demonstrates that the difference between the skew-*t* and ex-Gaussian distributions was quite small, and both provided a reasonably good fit to the data. A normal (Gaussian) distribution provided a notably worse approximation, underscoring the inappropriateness of using means and standard deviations in RT analysis. Although skew-*t* provided a slightly better fit than ex-Gaussian, its parameters turned out to be harder to interpret in terms of decision-related processes. For example, the skew-*t* distribution had a lower μ for "Response box" than for "Keyboard" when six squares were shown, whereas for lower set sizes the situation was the reverse. Thus, we decided to use the ex-Gaussian distribution in our analysis, especially given the fact that this distribution has previously been shown to be useful in analyses of RTs in visual search tasks (Kristjánsson & Jóhannesson, 2014; Palmer, Horowitz, Torralba, & Wolfe, 2011).

[1] Here and later, we report the results for a random-intercept model that did not include random effects for regression slopes. Random-intercept models are computationally simpler and less conservative than random-slope models (Barr, Levy, Scheepers, & Tily, 2013). In our data, this general rule was also confirmed: There were even fewer significant differences between experimental settings when random-slope models were used. Thus, we decided to use random-intercept models to show that even using less conservative methods produced no evidence in favor of the effect of the setting on the shape of the RT distribution.

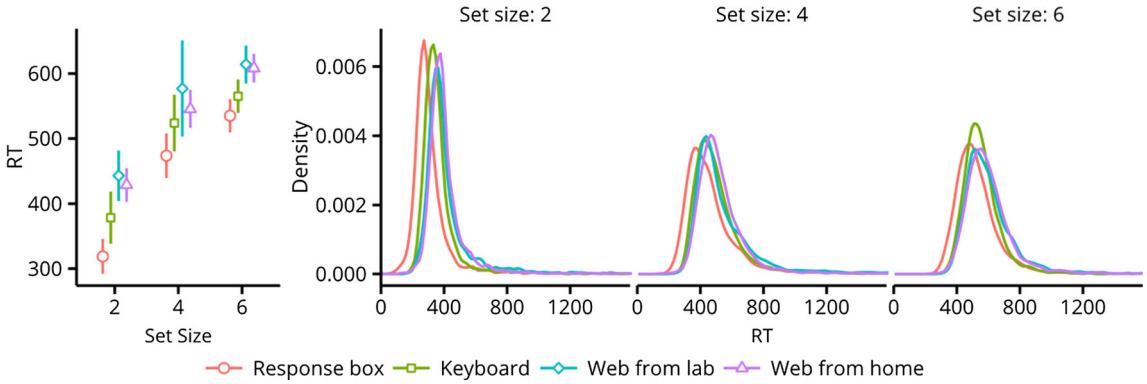

Fig.2 Means (left panel) and distributions (right panel) of response times (RTs, in ms). Lines in the left panel show 95 % confidence intervals, controlling for between-subjects variability (Cousineau, 2005; Morey, 2008)

Table 3 shows the mean values for μ_{EG}, σ_{EG}, and τ_{EG}. Notably, the large differences in mean values and high standard deviations for the "Web from lab" condition (Table 2) correspond to differences in τ_{EG}. We first ran the linear mixed-effects regression including only the setting as predictor and using treatment contrasts with "Response box" as a baseline, to see whether there were any differences in the distribution parameters in general (random intercepts for participants were included). The results indicated an effect of setting on μ_{EG} (*t*s = 2.76, 4.64, 4.79 for the "Keyboard," "Web from home," and "Web from lab" conditions, respectively), and a difference between "Response box" and "Web from lab" in τ_{EG}, B = 27.12 (11.08), t= 2.45. Importantly, "Web from home" did not differ from "Response box" in τ_{EG}, B = 12.51 (11.08), t = 1.13.

Then, we used linear mixed-effects regression with sequential difference contrasts to assess the effects of set size and setting and their interaction on distribution parameters. The regression model included random intercepts for each participant. Besides the main effects of set size, which are irrelevant to the present discussion, μ_{EG} was lower for "Response box" than for "Keyboard," B = 51.46 (16.78), t = 3.07, which in turn gave lower RTs than "Web from lab," B = 38.09 (16.78), t = 2.27. In addition, σ_{EG} was lower for "Keyboard" than for "Response box," B = 12.67 (6.12), t = 2.07. The interaction effects were not significant.

Separate comparisons at each level of set size demonstrated similar results. As is indicated in Table 4, differences in μ_{EG} emerged between all settings except for the "Web from home" and "Web from lab" comparison, and a difference in τ_{EG} between "Keyboard" and "Web from lab" for set size 6.

Since the "Web from home" condition was the most interesting for us, we repeated the analysis described above contrasting "Web from home" with all other levels (Table 5). The results indicated that μ_{EG} in the "Web from home" setting was higher than in the "Keyboard" and "Response box" setttings. No differences in σ_{EG} or τ_{EG} were found.

Finally, we applied a Friedman test with setting as an independent variable and RT as the dependent variable to be sure that we did not miss any of the effects, since the distribution of obtained parameters estimates may itself be far from normal. The Friedman test indicated differences in μ_{EG} ($p < .001$), but no other effects.

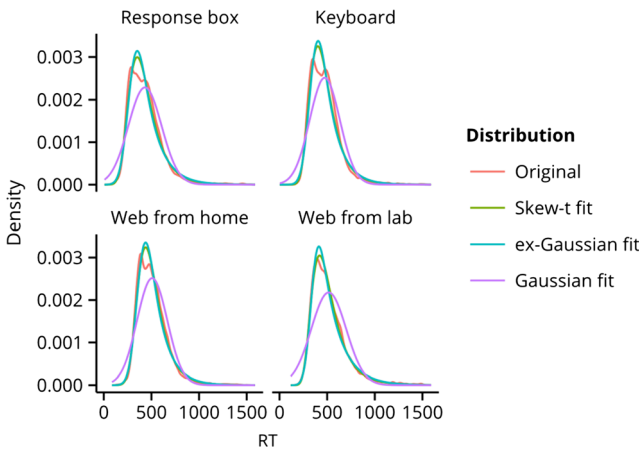

Fig.3 Densities of original data and fitted distributions by setting

Discussion

We used several approaches to look at the data, starting with repeated measures ANOVA and linear mixed-effects regression to analyze the means and standard deviations. We then used estimated parameters of the fitted distributions to look at the other possible effects of settings. The results demonstrate that RTs do depend on experimental setting: The location parameters of the fitted distributions (including means) were lower for the Cedrus response box than for a usual keyboard with PsychoPy, which in turn showed lower latencies than reactions collected online with JavaScript either within or outside the lab.

Other than the location parameters, τ_{EG} was higher for data collected with JavaScript in the lab than for data collected within the lab using PsychoPy and a keyboard. However, this

Table 3 Mean parameters of ex-Gaussian distributions (and their standard deviations)

Set Size	Response Box			Keyboard			Web From Lab			Web From Home		
	2	4	6	2	4	6	2	4	6	2	4	6
μ_{EG}	227	344	420	277	381	462	307	422	485	314	406	487
	(29)	(61)	(50)	(25)	(49)	(38)	(36)	(75)	(57)	(30)	(48)	(58)
σ_{EG}	31	36	42	25	30	48	27	34	42	29	31	46
	(14)	(16)	(10)	(11)	(12)	(13)	(10)	(18)	(13)	(10)	(19)	(16)
τ_{EG}	92	130	115	101	143	104	135	155	128	114	139	121
	(54)	(68)	(43)	(61)	(81)	(38)	(107)	(150)	(66)	(71)	(78)	(56)

effect turned out to be significant only for the largest set size. Data collected outside the lab with JavaScript did not differ in τ_{EG} or σ_{EG} from data collected within the lab. Accordingly, this effect could not be attributed to JavaScript.

Study 2

Some possible limitations of the present study were that we used preselected participants, and the variability of the computers they used at home may have been less than the variability that exists in real online samples. Thus, an additional online sample was recruited to assess the impact of hardware and software on the variability of RTs. These data cannot be directly compared to the data collected in the main part of the study, since the within-subjects design used in the main part created learning effects. However, the benefit of the online sample was a variety of computers that is difficult to produce in laboratory settings. Thus, we were able to evaluate the effects of hardware and software quantitatively rather than purely qualitatively.

Method

Participants A total of 284 participants were recruited through advertising in Russian online social networks (age: M = 25.11 years, SD = 13.37, Mdn = 24.00). They were not paid for participation. Six participants with accuracy below .90 were excluded, since the task provided was relatively easy and low accuracy indicated that these participants either did not understand the instructions or did not pay attention to the task.

Procedure The procedure was the same as in the "Web from home" setting. In addition, participants were asked to provide information about their CPU, GPU, amount of RAM, type of

Table 4 Comparison of distribution parameters using linear mixed-effects regression with sequential difference contrasts

Set Size	Effect	μ_{EG}			σ_{EG}			τ_{EG}		
		B	*SE*	*t*	*B*	*SE*	*t*	*B*	*SE*	*t*
2	A	281.4	5.0	56.3	28.2	1.7	16.7	110.7	11.2	9.9
	B–A	50.1	7.3	6.9	–5.2	3.1	–1.7	9.4	20.9	0.5
	C–B	30.1	7.3	4.1	2.1	3.1	0.7	34.2	20.9	1.6
	D–C	6.8	7.3	0.9	2.0	3.1	0.6	–21.0	20.9	–1.0
4	A	388.1	10.6	36.8	32.4	2.5	12.9	141.7	16.7	8.5
	B–A	37.3	13.0	2.9	–6.0	4.3	–1.4	12.9	24.0	0.5
	C–B	41.4	13.0	3.2	4.3	4.3	1.0	11.7	24.0	0.5
	D–C	–16.2	13.0	–1.3	–3.4	4.3	–0.8	–15.1	24.0	–0.6
6	A	463.5	10.2	45.4	44.6	2.2	20.8	116.9	9.5	12.4
	B–A	41.7	8.6	4.9	5.7	3.4	1.7	–11.6	10.8	–1.1
	C–B	23.8	8.6	2.8	–6.2	3.4	–1.8	24.7	10.8	2.3
	D–C	2.2	8.6	0.3	4.2	3.4	1.3	–7.7	10.8	–0.7

A, intercept, parameter value for "Response box"; B–A, difference between "Keyboard" and "Response box"; C–B, difference between "Web from lab" and "Keyboard"; D–C, difference between "Web from home" and "Web from lab." *B*, *SE*, and *t* show the regression coefficients, their standard errors, and Student's *t* criteria values, respectively

Table 5 Comparison of "Web from home" with all others settings

Set Size	Effect	μ_{EG}			σ_{EG}			τ_{EG}		
		B	*SE*	*t*	*B*	*SE*	*t*	*B*	*SE*	*t*
2	Response box	–87.06	7.32	–11.90	1.13	3.09	0.37	–22.68	20.93	–1.08
	Keyboard	–36.92	7.32	–5.05	–4.06	3.09	–1.31	–13.25	20.93	–0.63
	Web from lab	–6.80	7.32	–0.93	–1.99	3.09	–0.64	20.97	20.93	1.00
4	Response box	–62.43	13.03	–4.79	5.02	4.34	1.16	–9.53	24.00	–0.40
	Keyboard	–25.19	13.03	–1.93	–0.93	4.34	–0.21	3.39	24.00	0.14
	Web from lab	16.24	13.03	1.25	3.36	4.34	0.77	15.12	24.00	0.63
6	Response box	–67.69	8.57	–7.90	–3.67	3.37	–1.09	–5.33	10.77	–0.49
	Keyboard	–25.97	8.57	–3.03	1.99	3.37	0.59	–16.97	10.77	–1.58
	Web from lab	–2.16	8.57	–0.25	–4.21	3.37	–1.25	7.74	10.77	0.72

display (CRT, LCD, or other), and type of computer (laptop or desktop) before the task. The data on GPU and CPU were used to create a score indicating GPU and CPU performance based on the benchmarks provided by PassMark (www.cpubenchmark.net/cpu_list.php and www.videocardbenchmark.net/gpu_list.php). The information on the type of computer was used to resolve ambiguous cases. If several models fit the description provided by a participant, we used a median score. For example, if "AMD Athlon II X3" was entered as the CPU, the median of all AMD Athlon II X3 processor scores was used. Display refresh rate and resolution were determined automatically using JavaScript, and participants were asked to correct these data if necessary. In addition, with participants' agreement we used the user agent string reported by their browser to identify their browser and operating system.

Materials and apparatus Some of the participants did not report CPU ($N = 71$), GPU (or we were not able to obtain its score, $N = 117$), or RAM ($N = 47$). Chromium-based browsers were analyzed together, as well as different versions of Linux. Browsers other than Chromium-based ($N = 205$), Firefox ($N = 54$), and Safari ($N = 14$) were used by five participants and are not included in the software analyses. Participants mostly used Windows ($N = 230$), whereas Linux ($N = 18$) and Mac OS ($N = 30$) were less popular. Thus, Windows participants were further split by Windows version: Windows 7 ($N = 144$), Windows 8 or 8.1 ($N = 58$), or Windows XP/Vista ($N = 28$).

Results

Accuracy Answers with RTs above 10 s ($N = 29$) were discarded as outliers. We found a significant effect of set size on accuracy, $F(2, 554) = 44.51$, $p < .001$, $\eta^2_G = .065$. In contrast to Study 1, set size 4 and set size 6 resulted in the same accuracy, $t(277.0) = 0.56$, $p = .575$, whereas in set size 2 the accuracy was higher (Table 6).

Response times For all set sizes, a skew-*t* distribution (AIC = 516,515) provided a better fit than an ex-Gaussian (AIC = 520,641), which in turn fit better than a log-normal (AIC = 527,334) or a normal (AIC = 568,064) distribution. As in Study 1, we decided to use the ex-Gaussian distribution and obtained μ_{EG}, σ_{EG}, and τ_{EG} for each participant.

Because there were cases in which information on CPU, GPU, or RAM was missing, we first analyzed the effect of software (OS, browser) using the whole sample. We used robust linear regression provided by the robustbase package in R (Rousseeuw et al., 2015), controlling for age, gender, and display resolution as possible confounds. We did not include display type and refresh rate in the analyses because 257 out of 278 participants had LCD displays with a 60-Hz refresh rate.

The results indicated a significant effect of operating system, with Mac OS having a lower τ_{EG} than Windows 7, $B = –45.38$ (17.66), $t(258) = –2.57$, $p = .011$, and Windows XP/Vista having a higher μ_{EG}, $B = 29.03$ (13.34), $t(258) = 2.18$, $p = .030$, and σ_{EG}, $B = 16.17$ (6.92), $t(258) = 2.34$, $p = .020$. Browser effects were also present, with Firefox providing a lower μ_{EG}, $B = –18.43$ (8.84), $t(258) = –2.08$, $p = .038$, and σ_{EG}, $B = –11.56$ (5.24), $t(258) = –2.21$, $p = .028$, than Chromium-based browsers. In addition, we found a negative effect of age on μ_{EG}, $B = 2.88$ (0.86), $t(258) = 3.34$, $p < .001$.

We then used linear regression with hardware (CPU score, GPU score, RAM), controlling for the effects of OS, browser, and age. Gender and screen dimensions were not included, because the analyses above demonstrated that they did not affect RTs. Since there were cases in which information on CPU, GPU, or RAM was missing, and the three parameters

Table 6 Descriptive statistics for the online sample

	2	4	6
RT mean	501	713	697
RT *SD*	105	200	140
Share of errors	3.3	4.7	4.7

were correlated, we tested models that had only one of these three variables. CPU score had negative effects on μ_{EG}, B = –0.43 (0.19), $t(194)$ = –2.25, p = .026, and σ_{EG}, B = –0.26 (0.11), $t(194)$ = –2.46, p = .015 (Table 7). The amount of RAM and GPU score had no effect on distribution parameters. Importantly, the effect of Windows XP/Vista became nonsignificant when CPU scores were introduced, both for μ_{EG}, B = 22.19 (19.14), $t(194)$ = 1.16, p = .248, and σ_{EG}, B = 14.73 (9.08), $t(194)$ = 1.62, p = .107. As expected, a $t$test on CPU scores indicated that users of Windows XP/Vista had older CPUs, $t(36.0)$ = 9.61, $p < .001$. The Firefox browser effect also became nonsignificant, B = –15.93 (9.46), $t(194)$ = –1.68, p = .094. However, no difference in CPU scores between Firefox and the other browsers was found, $t(68.6)$ = –0.80, p = .426. Thus, whereas the effect of Windows XP/Vista was explained by a difference in CPUs, the lack of significance for browser effect on μ is probably explained simply by decreased sample size.

We then tested the effect of set size by repeating the analysis described above with the data fitted for each participant at each level of set size and with set size and its interactions with other predictors added to the regression equations. There were significant main effects of set size [on μ_{EG}, B = 108.21 (6.65), $t(592)$ = 16.27, $p < .001$, and σ_{EG}, B = 11.30 (2.11), $t(592)$ = 5.36, $p < .001$], but not of its interactions with other parameters.

Table 7 Regression models for ex-Gaussian distribution parameters in online sample

	μ	σ	τ
Constant	343.00***	77.00***	134.00**
	(25.00)	(15.00)	(47.00)
CPU Score	–0.43*	–0.26*	–0.04
	(0.19)	(0.11)	(0.34)
OS			
Linux/Windows 7	–10.00	–6.30	–16.00
	(11.00)	(6.10)	(27.00)
Mac OS/Windows 7	25.00	8.50	–57.00**
	(13.00)	(7.20)	(20.00)
Windows 8 or 8.1/Windows 7	15.00	1.40	14.00
	(11.00)	(5.50)	(17.00)
Windows XP or Vista/Windows 7	22.00	15.00	44.00
	(19.00)	(9.10)	(28.00)
Browser			
Firefox/Chrome	–16.00	–15.00**	–23.00
	(9.60)	(5.00)	(16.00)
Safari/Chrome	–23.00	–9.60	37.00
	(19.00)	(8.00)	(27.00)
Age	2.30*	0.04	3.80
	(0.98)	(0.56)	(2.10)

Windows 7 and Chrome were used as the reference levels, since they represented the largest group. * $p < .05$, ** $p < .01$, *** $p < .001$

Simulations

The results demonstrated that software and CPU score influence distribution parameters, whereas RAM and GPU score do not. What are the practical consequences of this influence? To answer that question, we used a modeling approach. Most of the commonly used statistical tests for RTs (e.g., ANOVA and $t$tests) depend on a Gaussian distribution. The Gaussian distribution parameters, μ_G and σ_G, are linked to the ex-Gaussian distribution parameters: $\mu_G = \mu_{EG} + \tau_{EG}$ and ${\sigma_G}^2 = {\sigma_{EG}}^2 + {\tau_{EG}}^2$. We used simulations based on the obtained results to find out how strongly the heterogeneity of CPUs, browsers, and operating systems would impact the μ_G and σ_G of individual participants and how this would in turn influence the sample SD and the statistical power of a $t$test for a preset difference in true means.

Statistical power measures the probability of a test to detect an effect of a given size—that is, $1-\beta$, where β is the probability of Type II error. This analysis should not be confused with estimating the real power of the study, since it does not account for individual differences and other factors not included in our regression models. Moreover, the statistical methods and experimental designs used may also influence statistical power. Rather, it provides an estimate of how big the drop in power is, due to the additional variance introduced by the factors that we analyzed.

Using the obtained regression models, we first estimated the parameters of an ex-Gaussian distribution for N subjects for each level of heterogeneity. Then those parameters were used to generate RTs for M trials for each subject, where N varied from 20 to 200 and M varied from 100 to 400. These data were used to estimate the power of a $t$test with α equal to .05 and a true difference in means equal to 10 ms (the details of our modeling approach are provided in the Appendix). This difference was chosen arbitrarily to represent a relatively small effect. For each parameter (CPU score, browser, and OS), we first describe its effect on statistical power in general and then describe a particularly negative case.

Figure 4 shows how the power of the $t$test changes as a function of the mean and SD of the CPU score distributions for different sample sizes and numbers of trials. It is clear that the heterogeneity of CPU scores has a detrimental effect on test power. Mean CPU score has more impact when the number of trials is relatively low—as one would expect, given its influence on σ_{EG}. With a sample size of 140 subjects or more, larger heterogeneity of CPU scores has only a small effect on power. For example, a sample of 140 subjects would have

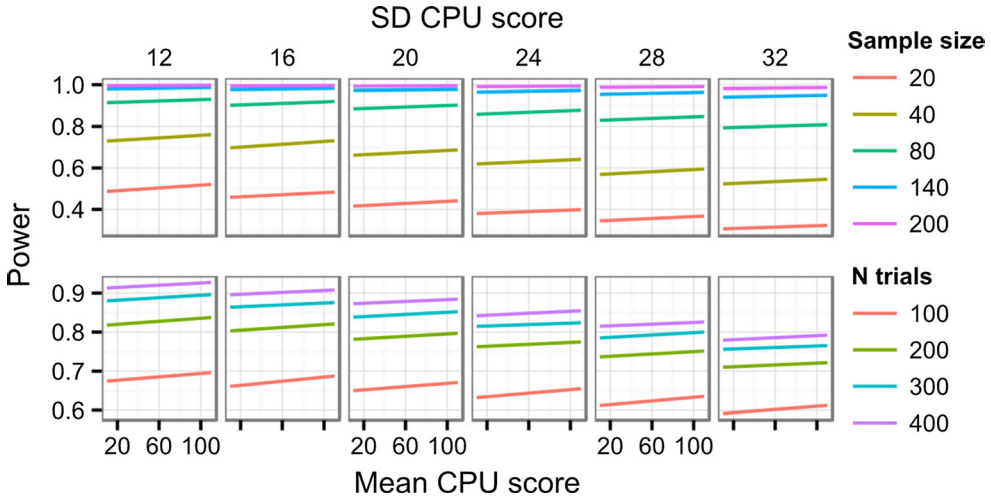

Fig. 4 Effects of CPU heterogeneity (*SD*, in columns) and average performance (means) on *t*test power to detect a 10-ms difference. The top row demonstrates the effect of sample size (*N* trials = 250) and the bottom row demonstrates the effect of trial numbers (sample size = 96)

only a 3 % decrease in *t*test power (from .98 to .95) with SD_{CPU} increasing from 12 to 32—that is, almost tripled. Of course, in a particularly negative scenario of a small sample ($N = 20$) and a low number of trials ($N = 100$), study power would be low (.28) even for the lowest SD_{CPU}, and would decrease even further (to .21) for the largest SD_{CPU}.

For browsers, the heterogeneity was modeled as the share of Chromium-based browsers in the sample. The situation was the most detrimental when the sample was split in half according to their choice of browsers (Fig. 5). In the worst-case scenario (20 subjects, 100 trials each, p(Chromium) = .5), power dropped to .19. However, increasing the number of subjects to 140 and the number of trials to 200 increased power to .91.

The most drastic decreases in test power were introduced by heterogeneity of operating systems, modeled as the share of OS X and Windows XP/Vista in the sample (Fig. 6). Even in the best-case scenario (200 subjects and 400 trials), power decreased to .5 when the share of OS X and Windows XP/Vista reached 100 % (i.e., 50 % each).

Discussion

The data from the online sample provide important insights into the influence of heterogeneity of hardware and software on the power of typical statistical tests. Amongst the hardware, only the diversity of CPUs (and not RAM or GPU) increased the variability of the obtained RTs. Specifically, the CPU score, representing its average performance, is inversely related to the mean and standard deviation of the Gaussian part of an ex-Gaussian distribution. Note that the CPU score represents not only the average performance, but also the probability that the CPU will be busy with some background tasks, since more powerful CPUs will spend less time on them. The effect of CPU score was relatively small, and it was remedied by increased sample sizes.

Both browser diversity and OS diversity have more pronounced negative effects. When the sample contains two major browsers (Firefox and Chromium-based) mixed in equal proportions, small samples suffer a large decrease in statistical power. However, increasing the number of trials and sample

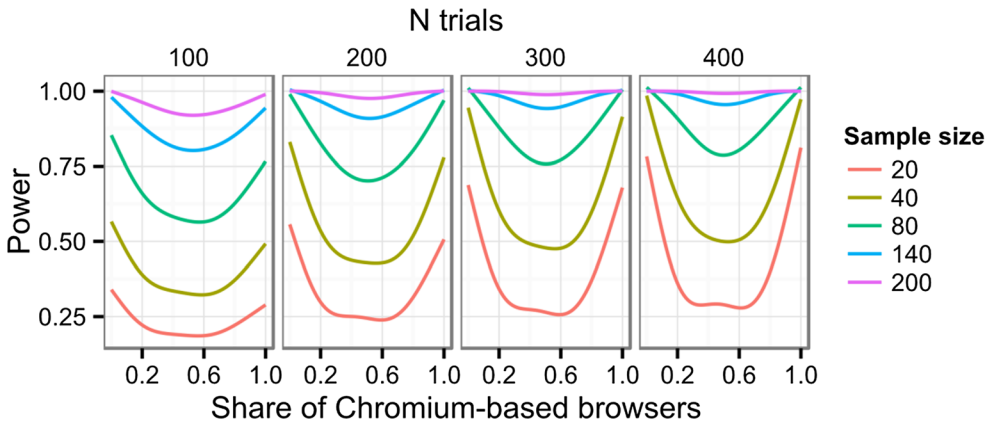

Fig. 5 Effects of browser heterogeneity on the power of *t*tests to detect a 10-ms difference. The remaining computers were assumed to use Firefox

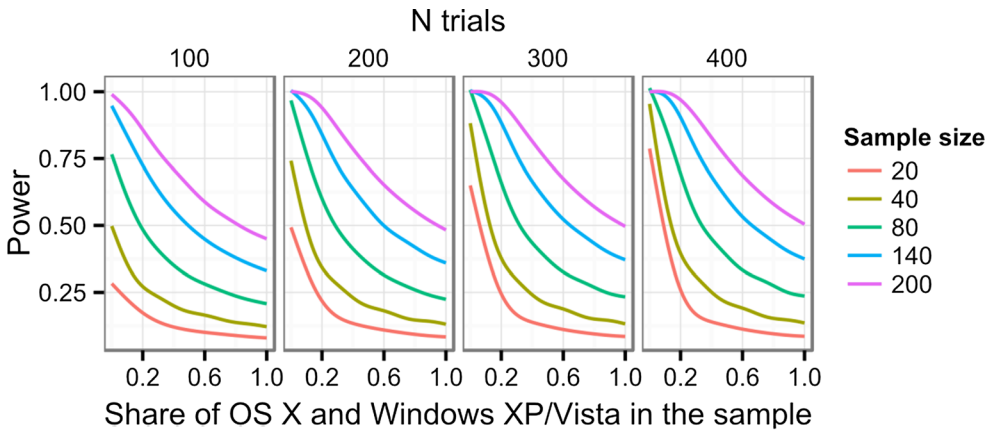

Fig. 6 Effects of operating system (OS) heterogeneity on the power of *t* tests to detect a 10-ms difference. OS X and Windows XP/Vista were supposed to have equal shares, and the remaining OSs were assumed to be Windows 7

size helps to avoid this. The influence of operating system diversity, on the other hand, is harder to compensate for. Even with large samples and large numbers of trials, it remains a significant threat to statistical power.

General discussion

Similar to previous studies aimed at comparing RTs in online and offline settings, we demonstrated that the experimental setting affects the location of the RT distribution. However, other distribution parameters that define its shape were not affected by the setting. To reiterate, within-lab data from a keyboard and PsychoPy or from a response box timer did not differ from data collected online outside the lab. Thus, our results concur with the results of Reimers and Stewart (2007) by showing that the location of the distribution depends on the experimental setting, but not its other parameters.

The first study also showed that task difficulty does not have an important role in the decision of whether or not to collect RT data through online experiments. Although set size did influence RTs, with longer RTs for larger set sizes, only in one case did we observe an interaction of set size with experimental settings. Specifically, the difference in the location of the distribution between a response box and a usual keyboard was smaller for larger set sizes. The response box seems to become less different from a usual keyboard when more than two keys need to be pressed. However, even with the simplest task, in which participants needed to discriminate between targets appearing on the left and targets appearing on the right, there were no effects of setting on the standard deviation (σ) of the Gaussian part or on the exponential part (τ) of an ex-Gaussian distribution. With increasing task difficulty, the influence of individual differences between observers will increase, and the impact of data collection method will further decrease, except for systematic errors of measurement. In other words, if we were not able to observe significant differences with one of the simplest tasks possible, it is unlikely that such differences would be found with more difficult tasks.

The results of this study are important, because researchers are usually interested not in the absolute values of RTs, which depend on the distribution location, but rather in the effects of some other variable—that is, the difference in RTs between experimental conditions. The probability of finding the latter depends on the magnitude of the effects in question and on the shape of the distribution. We found no differences between the three keyboard-based measurements, one using PsychoPy and two using JavaScript, in the parameters that defined the shapes of the RT distributions. It does not matter, then, according to the results of the first study, which method is used, as long as the magnitude of the effects does not depend on the experimental settings.

However, additional data collected from a larger online sample demonstrated that heterogeneity in CPUs, operating systems, and browsers may have a detrimental effect on the statistical power, due to differences in both the location and shape parameters of the RT distributions. These results raise concerns regarding the statistical power of online studies in the presence of high variability in software and hardware. Our simulations demonstrate that the negative impact of variability could be compensated for by larger sample sizes, except for the effect of high diversity of operating systems. In addition, the real sample did not show large diversity of operating systems. Remember that we simulated the effect of operating systems by increasing the shares of Windows XP/Vista and OS X in the sample. In our data, they provided about 10 % of the data each. Given that Windows XP and Windows Vista are outdated and no longer supported, we could expect that their use will continue to decrease. Modern versions of Windows do not differ from each other (though this can change in future). In any case, it is relatively easy to use the data provided by the browser to filter out users of specific operating systems

or to include OS in the analyses to account for its influence. The conclusions from the online sample are valid only to the extent that our sample represents variability comparable to that in future studies. For example, we did not include mobile devices, but they may become an essential part of online studies. Furthermore, it is hard to predict how new operating systems and browsers will differ from the existing ones.

The conclusions from Study 1 need to be considered with caution, as well. First, given that we wanted to perform four measurements on each participant in Study 1 on different days, our sample size was limited by practical considerations. This allowed us to use a Latin square design and also provided participants with enough training to reduce the between-subjects variability (as compared to the untrained participants from the online sample). It is possible that smaller differences in the distribution parameters would become significant with larger samples or even with more training. However, the practical consequences of such differences are questionable, given that larger samples are more easily available for online studies and that in many behavioral experiments in the laboratory 20 participants is still a usual sample size. Second, our testing environment in the laboratory was not an ideal reference environment. An ideal testing conditions would include a photosensitive element detecting the presentation of the stimuli and sending the signal directly into a response box, to minimize any delays and variation in timing. Nevertheless, our comparisons allowed us to analyze the effect of noise sources specific to online studies—namely, the variability of software and hardware in real-world online studies and the use of browsers and JavaScript for presentation of the stimuli and RT collection. For example, we did not account for the errors due to delays in the presentation of stimuli related to screen refresh rates in the "Web from lab" and "Web from home" conditions (unlike specialized software for stimulus presentation, JavaScript is unable to synchronize presentation with the refresh rate). Such errors may further decrease the accuracy of RTs, because they are measured starting from the moment when the program is commanded to present the stimuli, and the true RT begins only when the stimuli are actually presented. However, such errors are included in the overall errors in the "Web from home" and "Web from lab" conditions, and should make the difference between these and other conditions more pronounced. In addition, Garaizar and Vadillo (2014) tested PsychoPy's accuracy in the timing of stimulus presentation, but the accuracy of RT measurement was not assessed. However, we also used the Cedrus response box that had an internal timer independent of PsychoPy. Thus, the fact that we did not find a difference in measurement error between PsychoPy and the response box (in fact, RTs had lower variability with PsychoPy and the keyboard than with the response box) also validates the use of PsychoPy for further comparisons.

The results of the two studies may seem contradictory. The first study failed to find any impact of the condition except for the difference in the location parameters of the distributions. The second study demonstrated that variability plays an important role in determining the shapes of RT distributions in online studies. A straightforward conclusion from the first study would be that one can safely use the Web for experiments, whereas the second one tells a cautionary tale. We suggest that this contradiction is only apparent, and stems from the different questions asked by the two studies. The second study asked "How does variability of software and hardware in online studies influence the RT distributions?" Furthermore, our simulations concerned an idealized situation in which individual differences are absent. This is similar to studies that use a special device, such as the Black Box Toolkit or a simple photosensor connected to a key-pressing coil, to estimate the error in RT measurements. On the contrary, the first study asked "How much noise is added by online studies, relative to noise present in the usual conditions of offline study?" This question assumes that the usual offline environment is not ideal: Individual differences, operating systems and drivers, times of the day, tiredness and learning effects within the study, and many other factors add unwanted variability to the measurement. This study demonstrated that the effect of additional noise introduced in online studies is small, so that it cannot be detected in a simple location task with a relatively large number of trials. Thus, we suggest that the two studies complement each other. The first shows that RT data could be collected in online studies and that the shapes of RT distribution would not differ from the usual offline environment. The second study warns that one cannot simply assume that every online study will provide good data, and one needs to take into account the software used by participants.

In sum, online studies are an adequate source of RT data, comparable to a common keyboard-based setup in the laboratory. Higher sample sizes and ecological validity could even make such studies preferable. In some cases, though, high variability in software is hard to compensate for with a larger sample size or a higher number of trials. Thus, reporting the variability of software and, if possible, hardware should be a standard practice for future online studies.

Author note This study was supported in part by Saint-Petersburg State University (Research Grant Number 8.38.520.2013 to A.C.) and by the Russian Foundation for Basic Research (#15-06-07417, A.C.).

Appendix

We used the following procedure to model RTs as a function of software and hardware heterogeneity:

1. Using the obtained robust regression models, we estimated the predicted values of μ_{EG}, σ_{EG}, and τ_{EG} for N subjects, where N varied from 20 to 200 (samples of 20, 40, 80, 140, and 200 subjects were used), as a function of:

    a. CPU scores, with mean scores ranging from 100 to 11,000 in 100-point steps (our sample mean = 3,528) and *SD*s ranging from 1,000 to 3,000 in 200-point steps (our sample *SD* = 2,021);
    b. browser heterogeneity, operationalized as the share of Chromium-based browsers in the sample, from 0 to 1 in .05 steps [our sample: p(Chromium) = .75];
    c. OS heterogeneity, operationalized as the share of OS X and Windows XP/Vista in the sample, from 0 to 1 in .05 steps. OS X and Windows XP/Vista were assumed to have equal shares [our sample: p(OS X) = p(Windows XP/Vista) = .10]. Although the effect of Windows XP/Vista was not significant when CPU was included in the regression, we nevertheless decided to include it in the modeling. We reasoned that it might be useful to account for its effect when no data on CPU performance are available.
2. Predicted values were used to draw data from ex-Gaussian distributions for M trials for each subject, where M varied from 100 to 400 in 100-trial steps.
3. For each subject, a mean RT was calculated, and the *SD* of these means was used to calculate the power of a *t* test with sample size equal to N, α equal to .05, and a true difference in means equal to 10 ms.

Steps 2–3 were repeated 200 times for each level of CPU score, browser heterogeneity, and CPU heterogeneity. Note that heterogeneity of software and hardware can influence the accuracy of RT data in two ways. First, there could be some especially detrimental conditions. According to the obtained regression models, a low CPU score and Chromium-based browsers provided the highest σ_{EG}, and thus added the largest error to σ_G. Second, heterogeneity of software and hardware can be detrimental by itself, due to additional variance introduced by differences in μ_G between subjects, influenced by differences in μ_{EG} and τ_{EG}. With an increasing number of trials, the error of measuring μ_G due to σ_G (and hence σ_{EG}) will decrease, but not the error due to differences in μ_{EG} and τ_{EG}. The latter is accounted for only by increasing the sample size.

References

Baayen, R. H., Davidson, D. J., & Bates, D. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language, 59,* 390–412. doi:10.1016/j.jml.2007.12.005

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language, 68,* 255–278. doi:10.1016/j.jml.2012.11.001

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2013). *lme4: Linear mixed-effects models using Eigen and S4. R package version 1.0–4.* Retrieved from http://cran.r-project.org/package=lme4

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B, 57,* 289–300.

Cousineau, D. (2005). Confidence intervals in within-subject designs: A simpler solution to Loftus and Masson's method. *Tutorials in Quantitative Methods for Psychology, 1,* 42–45.

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:10.1371/journal.pone.0057410

Damian, M. F. (2010). Does variability in human performance outweigh imprecision in response devices such as computer keyboards? *Behavior Research Methods, 42,* 205–211. doi:10.3758/BRM.42.1.205

de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods, 47,* 1–12. doi:10.3758/s13428-014-0458-y

Garaizar, P., & Vadillo, M. A. (2014). Accuracy and precision of visual stimulus timing in psychopy: No timing errors in standard usage. *PLoS ONE, 9*(e112033), 1–8. doi:10.1371/journal.pone.0112033

Keller, F., Gunasekharan, S., Mayo, N., & Corley, M. (2009). Timing accuracy of Web experiments: A case study using the WebExp software package. *Behavior Research Methods, 41,* 1–12. doi:10.3758/BRM.41.1.12

Kristjánsson, Á., & Jóhannesson, Ó. I. (2014). How priming in visual search affects response time distributions: Analyses with ex-Gaussian fits. *Attention, Perception, & Psychophysics, 76,* 2199–2211. doi:10.3758/s13414-014-0735-y

McDonnell, J. V., Martin, J. B., Markant, D. B., Coenen, A., Rich, A. S., & Gureckis, T. M. (2014). *psiTurk*. New York, NY: New York University.

Morey, R. D. (2008). Confidence intervals from normalized data: A correction to Cousineau (2005). *Tutorials in Quantitative Methods for Psychology, 4,* 61–64.

Neath, I., Earle, A., Hallett, D., & Surprenant, A. M. (2011). Response time accuracy in Apple Macintosh computers. *Behavior Research Methods, 43,* 353–362. doi:10.3758/s13428-011-0069-9

Palmer, E. M., Horowitz, T. S., Torralba, A., & Wolfe, J. M. (2011). What are the shapes of response time distributions in visual search? *Journal of Experimental Psychology: Human Perception and Performance, 37,* 58–71. doi:10.1037/a0020747

Peirce, J. W. (2007). PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods, 162,* 8–13. doi:10.1016/j.jneumeth.2006.11.017

Plant, R. R., & Quinlan, P. T. (2013). Could millisecond timing errors in commonly used equipment be a cause of replication failure in some neuroscience studies? *Cognitive, Affective, & Behavioral Neuroscience, 13,* 598–614. doi:10.3758/s13415-013-0166-6

Reimers, S., & Maylor, E. A. (2005). Task switching across the life span: Effects of age on general and specific switch costs. *Developmental Psychology, 41,* 661–671. doi:10.1037/0012-1649.41.4.661

Reimers, S., & Stewart, N. (2007). Adobe Flash as a medium for online experimentation: A test of reaction time measurement capabilities. *Behavior Research Methods, 39,* 365–370. doi:10.3758/BF03193004

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:10.3758/s13428-014-0471-1

Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape (with discussion). *Journal of the Royal Statistical Society, Series C, 54,* 507–554. doi:10.1111/j.1467-9876.2005.00510.x

Rousseeuw, P., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., Verbeke, T., … Maechler, M. (2015). robustbase: Basic robust statistics [Software]. Retrieved from http://cran.*r*-project.org/web/packages/robustbase/index.html

Schubert, T. W., Murteira, C., Collins, E. C., & Lopes, D. (2013). ScriptingRT: A software library for collecting response latencies in online studies of cognition. *PLoS ONE, 8*, e67769. doi:10.1371/journal.pone.0067769

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