PREPRINT
Not peer-reviewed

Online Timing Accuracy and Precision: A comparison of platforms, browsers, and participant's devices.

Alex Anwyl-Irvine[1,2], Edwin S. Dalmaijer[1], Nick Hodges[2], Jo K. Evershed[2]

Acknowledgements: AAI, NH, and JKE are employed by Cauldron, which operates the Gorilla experiment platform.

Abstract

Due to its increasing ease-of-use and ability to quickly collect large samples, online behavioral research is currently booming. With this increasing popularity, it is important that researchers are aware of who online participants are, and what devices and software they use to access experiments. While it is somewhat obvious that these factors can impact data quality, it remains unclear how big this problem is.

To understand how these characteristics impact experiment presentation and data quality, we performed a battery of automated tests on a number of representative setups. We investigated how different web-building platforms (Gorilla, jsPsych, Lab.js, and psychoJS/PsychoPy3), browsers (Chrome, Edge, Firefox, and Safari), and operating systems (mac OS and Windows 10) impact display time across 30 different frame durations for each software combination. In addition, we employed a robot actuator in representative setups to measure response recording across aforementioned platforms, and between different keyboard types (desktop and integrated laptop).

We then surveyed over 200 000 participants on their demographics, technology, and software to provide context to our findings. We found that modern web-platforms provide a reasonable accuracy and precision for display duration and manual response time, but also identify specific combinations that produce unexpected variance and delays. While no single platform stands out as the best in all features and conditions, our findings can help researchers make informed decisions about which experiment building platform is most appropriate in their situation, and what equipment their participants are likely to have.

Keywords: Online testing, Online studies, Stimulus presentation, Experiment builder, Psychophysics, Software, Big Data, Cyberpsych

[1] MRC Cognition & Brain Sciences Unit, University of Cambridge, Cambridge, UK

[2] Cauldron.sc: Cauldron Science, St Johns Innovation Centre, Cambridge, UK

Correspondence: Jo K. Evershed, Cauldron Science, St Johns Innovation Centre. jo.evershed@cauldron.sc

Introduction

Conducting behavioural research online has vastly increased in the last few years. For instance, the number of papers tracked by Web of Science with the keywords ‘MTurk’ or ‘Mechanical Turk’ (Amazon’s popular platform for accessing online participants or workers, available since 2005) was 642 in 2018, over a five-fold increase over five years from 121 publications in 2013 (Figure 1). While scientist do not exclusively use MTurk for psychological experiments (it is also used to gather training data for machine learning), it is indicative of a trend. For example, Bohannon (2016) reported that published MTurk studies in social science increased from 61 in 2011 to 1200 in 2015 – an almost 20-fold increase.

A unique problem with internet-based testing is its reliance on participants’ hardware and software. Researchers who are used to lab-based testing will be intimately familiar with their computer, stimulus software, and hardware, for response collection. At the very least, they can be sure that all participants are tested using the very same system. For online testing, the exact opposite is true: Participants use their own computer (desktop, laptop, tablet, or even phone), with their own operating system, and access experiments through a variety of web browsers.

In addition to participant degrees of freedom, researchers can choose between various options to generate experiments. These vary from programming libraries (e.g. jsPsych) to graphical experiment builders (e.g. Gorilla Experiment Builder), and come with their own idiosyncrasies with respect to timing, presentation of visual and auditory stimuli, and response collection.

This presents a potential problem for researchers: Are all of the unique combinations of hardware and software equal? Here, we first investigate what types of software potential participants use, and how common each option is. We then provide a thorough comparison of the timing precision and accuracy of the most popular platforms, operating systems, internet browsers, and common hardware. We specifically compare four frequently used platforms that facilitate internet-based behavioral research:

- Gorilla Experiment Builder (www.gorilla.sc)
- jsPsych (www.jspsych.org)
- Lab.js (lab.js.org)
- psychoJS (building in PsychoPy3, and hosting on www.pavlovia.org)

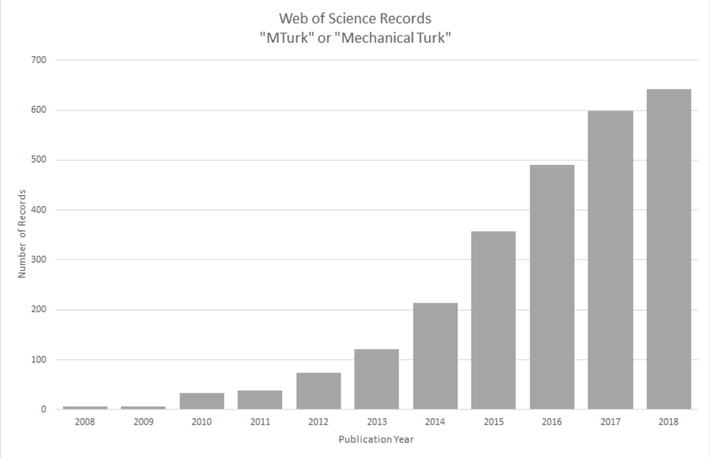


Figure 1. *Trends over time in papers mentioning mechanical turk, taken from Web of Science.*

A Brief History of Online Experiments

The almost exponential increase of papers citing MTurk is surprisingly recent. Despite the internet being available since the 1990s, and tools like MTurk existing since the mid-2000s, the adoption of online-research has started to grow at a more rapid rate in the last 5-10 years. There are, however, some early examples of online experimentation, for example: investigating spatial cognition (Givaty et al., 1998), visual motion extrapolation (Hecht et al., 1999), probability learning (Birnbaum & Wakcher, 2002), and establishment of labs dedicated to web experiments (Reips, 2001). In the late 1990s and early 2000s several guidance books and articles on the subject were published (Birnbaum, 2000; McGraw et al., 2000), with one 1995 review even coining the term ‘Cyberpsych’ to describe internet based psychological science (Kelley-Milburn & Milburn, 1995). Sadly, it appears that term did not catch on. Articles providing technical guidance published for running experiments, such as maintaining a web-server (Schmidt et al., 1997) and analysing server logs (Reips & Stieger, 2004) also emerged around this time. However, despite the availability of these tools and the promise of larger sample sizes, it took years to reach the current high levels of demand. There are several potential candidates for this apparent research adoption lag: the required level of technical ability, availability of personal devices, and concerns over data quality.

Building a research project online in the late 2000s required a much higher level of web-specific technical skills. Experimenters would have to know how to construct web pages and load resources (e.g. images and videos), capture and transmit participant data, configure and maintain a server to host the web pages and receive the participant data, and store the participant data in a database. Additionally, the capabilities of web applications at this time did not allow for much more than slow image and text presentation. Interactive animations and dynamic elements were inconsistent, and often slow to load for most users. There were survey tools available such as Qualtrics, Survey Monkey, and Lime Survey (Baker, 2013), but these really only permitted relatively simple experiments.

In the early 2010s, the situation began to change with better tools becoming available. In particular, the High Resolution Time API, which allowed for far betting timing accuracy than older methods such as setTimeout(), began appearing in browsers in 2013 (although it wasn’t supported in all major browsers until 2015 – www.caniuse.com/#feat=high-resolution-time). Running online research, allowing dynamic presentation of experimental trials and stimuli, and recording reaction times was possible through tools such as QRTEngine (Reaction Time engine for Qualtrics; Barnhoorn, Haasnoot, Bocanegra, & Steenbergen, 2015) and jsPsych (JavaScript Library for building and presenting experiments; de Leeuw, 2015), which appeared around 2013. As more tools and platforms have become available (for an overview, see Anwyl-Irvine, Massonnié, Flitton, Kirkham, & Evershed, 2019), the technical barrier to web-based research seems to have — at least partially — been alleviated, allowing more research to be conducted online.

The access individuals have to the internet via a personal or shared device has also increased over this time, and continues to increase relatively linearly. This is illustrated in Figure 2, using data provided by the United Nations International Telecoms Union. This pattern indicates a continuing increase in the potential reach of any web-based research to larger proportions of populations across the globe. This is particularly important considering a historical problem with under-powered research leading to unreliable results, where increased sample sizes provide one way to address this issue (Button et al., 2013).

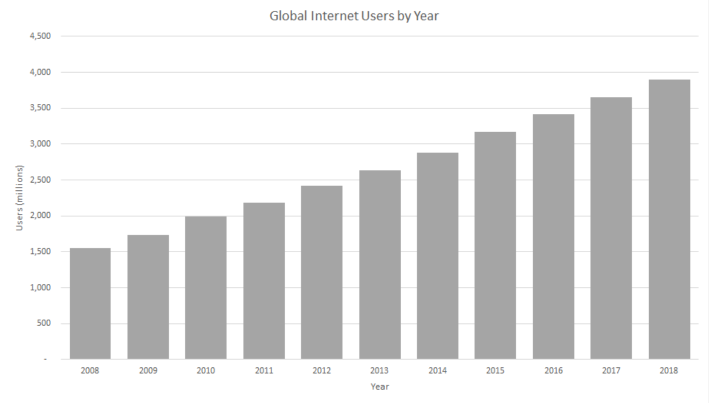


Figure 2. *Global internet users over time, data taken from the U.N International Telecoms Union (https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx)*

The Current State

Despite the potential availability of large samples online, there is a hesitancy to adopt certain types of tasks and experiments, particularly those that utilise short stimulus durations (e.g. visual masking experiments) or that need very accurate response time logging (such as an attentional flanker task). The relative noise from online studies can be characterized as coming from two independent sources:

1) Altered participant behaviour relative to a lab setting.
2) Software (OS, web browsers and platforms) and hardware (screens, computers, mobile devices).

The alteration of participant behaviour when taking part remotely is difficult to address systematically with software or hardware, and ultimately comes down to the design of the experiment, and utilisation of certain tools — such as webcam tracking. That being said, there *are* ways in which you can reduce this noise— a brief summary of how to improve the quality of data collected online is given by Rodd (2019), and is also discussed in Clifford & Jerit (2014). This paper, however, focuses on issues related to the second point: measurement error introduced by software and technology. This issue *can* be improved through the use of hardware and software, but also quantifying the introduced imprecisions would do much to help researchers utilise large samples easily in timing-sensitive experiments – which is the purpose of this paper.

There have been various claims made on the scientific record regarding the display and response timing ability of experimental setups using web browsers, for instance that timing can be good depending on device and setup (Pronk, Wiers, Molenkamp, & Murre, 2019), and that different techniques of rendering animations lead to reduced timing precision (Garaizar & Reips, 2019). Ultimately, though, the variance in timing reflects the number of different ways to create an online experiment, and the state of the software and hardware landscape at the time of assessment — all of these are changing at a fast rate. We previously undertook a discussion of the changing hardware and software ecosystem in Anwyl-Irvine et al. (2019). To address this variance, it is important to report any timing validation on a range of devices. To the authors knowledge, the largest number of devices tested with online software was undertaken by Reimers and Stewart (2015), where 19 Windows machine were assessed, and it is suggested that systems (OS and devices) contribute the greatest variability, with Windows XP displaying less variability than Windows 7. The justification for only testing Windows devices was that 85-90% of their participants used these. However, this has changed since 2015, see the demographics section of this paper for more details.

A vital issue with research into timing, is that it is tempting to interpret results from one (or a set of) studies, and extrapolate this to all ‘online research’. However, most online research is undertaken using different builders, hosting websites, and entire Software as a Service (SaaS) platforms — very little is made using written-from-scratch JavaScript. These different platforms and websites are separate software, each providing different animation, rendering, and response polling code. Just because good timing is possible using one particular JavaScript method in a specific scenario, does not mean that it will be great in all online studies. Hence, in this paper, we compare a variety of online study platforms.

A Realistic Approach to Chronometry

Researchers need to be furnished with the information they need to make sensible decisions about the limitations of browsers, devices, and operating systems. With this information, they can trade-off the size of their participant pool with the accuracy, and precision of the collected data. If we are to make any timing validation functionally informative to the users, we have to ensure that our methods are representative of the real-world setups that our participants will be using. Failure to do so could result in unexpected behaviour, even when running previously well-replicated experiments (Plant, 2016).

When researchers assess the accuracy of software in respect to timing, often the software and hardware setups are adjusted significantly in order to record optimum performance in the most ideal environment. These setups require the removal of keyboard keys and soldering on of wires (Reimers & Stewart, 2015), and include discrete graphics card (Garaizar, Vadillo, & López-de-Ipiña, 2014). This does not represent the average internet user's devices at all. For instance, in the first quarter of 2019 less than 30% of new PCs sold included discrete (i.e. non-integrated) graphics cards (Peddie, 2019), likely representing an even smaller number of online participants. Recently, Pronk et al. (2019), utilized a robotic actuator to press keyboard keys and touch-screens, a more representative assessment of RT recording. Testing on ideal-case setups, whilst vital for realising the frontier of what is possible with online software, is likely to poorly reflect the situation researchers face when collecting data online. Consequently, we have opted to take a more representative approach in our validation.

The first and second parts of this paper test the visual display and response logging performance of different software on different common browsers and devices, in order to give an indication of each setup's limits. The final part of this paper then provides an overview of the device demographics of online participants, with a snapshot sample over 200 000 of Gorilla participants taken in 2019. Pronk et al. (2019) use global web user data to select the browsers they use, but this may be different from the sub-population of those who engage in online research. Our approach is therefore well-suited to estimate the distribution and variability of devices and browsers within the online participant population.

For the testing sections, we selected a representational variety of devices. Windows and macOS operating systems cover the majority of the population for online testing (73% of our user sample). The devices we use are split between a Desktop PC with an external monitor, a Desktop Mac with an integrated monitor, a high-spec Windows ultra-book, and a lightweight Mac laptop. Further to this, the devices are assessed as they are with no steps taken to restrict the browsers or operating systems, increasing the likelihood they reflect users' actual setups. By taking this approach we properly acknowledge the variability present in the real-world. This can provide the online research community with information to make decisions about their online study given the sensitivity needs of their research.

In order to inform researchers about the likely accuracy of their set-ups, we have endeavoured to cover as many commonly used tools, operating systems, and devices as possible (given the number of trials needed for each test). We have assessed these using an external chronometry device that can independently capture the accuracy and precision of systems.

We also distinguish between the *average* accuracy of the timing of set-ups (e.g. on average, how close to the actual reaction time is a given set ups record), and the *variability* of this accuracy (i.e. will the reaction time error vary a lot within one experiment). Variability in presentation and reaction times increases the noise in the experiment. For example, a delayed – but consistent – reaction time record permits between trials & condition comparisons, whereas variability in this can potentially obscure small differences between conditions. These concepts are referred to respectively as *accuracy* and *precision.*

In all data reporting, we have intentionally avoided the use of inferential statistics, and chosen to show descriptive statistics, an approach previous studies have taken (Neath et al., 2011; Reimers & Stewart, 2015, 2016). We made this choice for two reasons. Firstly, the distributions of the data traces produced are highly irregular, deviations are either very small and frequent, or very large and infrequent – this makes formal comparison very difficult. Secondly, there is no ideal way to define a unit of observation. If we consider each sample within a condition, the large number of samples is likely to make any minor difference statistically significant, even if it is not practically meaningful. Alternatively, if we consider each device-browser-platform combination, comparisons would be severely under-powered. We thus report descriptive statistics, as well as the entire distribution of samples within each cell.

1. Visual Duration Accuracy

This experiment looks at how robust different web-based tools are when it comes to both response recording and display accuracy. We compare our platform, *Gorilla*, with three other web-based tools: *jsPsych*, *psychoJS/PsychoPy3* (produced from their builder and hosted on *Pavlovia.org*), and *Lab.js* (using their builder).

These implementations are tested in a variety of configurations, to represent some of the most common participant scenarios. Five browsers are used: *Chrome*, *Firefox*, *Edge*, *Safari*, *IE*, and two operating systems are used: *Windows 10* and *macOS Mojave*.

Methods

Visual Display Duration

The Visual Display Duration (VDD) experiment assessed the accuracy of the platform's visual display timing on the test rigs. A series of white squares were presented for a variable duration on a black background, with a 500ms/30 frame inter-stimulus interval. Stimuli were presented for a duration of 1-30 frames (1/60th a second to 1/2th a second) to create a profiling trace for each system. Each duration was repeated 150 times, for a total of 4500 presentations per hardware and software combination.

The duration of each white square was recorded using a photodiode/opto-detector connected to a Black Box Toolkit version 2 (BBTKv2) (Plant, 2014). This photo-diode was attached to the Center of each screen with an elastic strap, ensuring it was attached firmly and flatly to the screen. In line with the BBTKv2 user manual, an amplitude threshold was used that was relative to each screen. This was titrated beforehand with a continuously flashing square, and the highest threshold that permitted detection of the flashing white square was chosen.

Browsers

Browser versions were verified from the browsers themselves on each machine rather than via version tracking tools within testing platforms, as these were sometimes inaccurate, or used different versioning conventions (e.g. Edge 44 on Windows 10 Desktop was recorded as Edge 18.17763 by Gorilla — the first being version of the browser, and the second being the HTML engine version). The browser versions used were: Chrome 76 (Windows), Chrome 75 (macOS), Firefox 68 (Windows), Firefox 69 (macOS), Safari 12 (macOS) and Edge 44 (Windows).

At the time of testing psychoJS would not run on Edge on our set-ups, this compatibility has been fixed and we hope to include this data in a future version of this paper.

Devices

The two devices were:

1) Windows Desktop Running Windows 10 Pro, with an Intel Core i5-2500 3.3 GHz CPU, 8Gb of RAM, and an 60Hz ASUS VS247 23.6” Monitor with a 1920 x 1090 resolution.

2) 2017 Apple iMac with an Intel Core i5-7400 3.0 GHz CPU, a built in 21.5” monitor with a 4096 x 2304 resolution.

Platforms

All data were collected between June-September 2019. The Gorilla task was run on builds: 20190625, 20190730, and 20190828, the PsychoJS task was made with PsychoPy3 v3.1.5 and hosted on Pavlovia.org, The jsPsych task was made using v6.0.5. The Lab.js task was built using the GUI, and was made with version 19.1.0.

Data Processing

The metric of interest is the accuracy and precision of displaying the white square for the requested duration. This can be expressed as a delay score were the expected duration of the square and the actual recorded time from the photodiode are compared in milliseconds. Outliers (defined as more than 4 Standard Deviations from the mean) were removed, but their numbers are reported.

Visual Duration Delay

					Percentiles		
Platform	***Mean***	***Standard Deviation***	***Minimum***	***Maximum***	***25%***	***50%***	***75%***
Gorilla	13.44	15.41	-66.75	218.50	2.25	17.50	22.50
Lab.js	9.79	4.69	-0.89	16.49	6.17	6.90	15.70
PsychoJS	-6.24	12.99	-301.50	265.25	-13.00	-10.75	-1.00
jsPsych	26.02	17.40	-66.00	90.00	15.25	26.50	37.00

					Percentiles		
Browser	***Mean***	***Standard Deviation***	***Minimum***	***Maximum***	***25%***	***50%***	***75%***
Chrome	11.50	15.40	-301.50	265.25	0.50	7.22	22.75
Edge	18.72	17.88	-66.75	120.25	3.50	18.75	32.25
Firefox	22.58	19.80	-46.75	218.50	3.50	19.75	36.25
Safari	30.02	15.07	-27.75	86.50	17.50	29.25	42.50

					Percentiles		
Device	***Mean***	***Standard Deviation***	***Minimum***	***Maximum***	***25%***	***50%***	***75%***
Windows	12.43	17.11	-301.50	265.25	0.50	15.25	19.75
macOS	25.45	17.17	-38.25	218.50	14.25	23.50	34.50

Table 1. *Summary of Visual Duration Delay results. Visual Duration Delay is calculated as the difference in milliseconds between the requested duration of a white square and the duration that is recorded by a photodiode sensor. It is broken down by Platform, Browser and Device. All results are reported after outliers have been excluded*

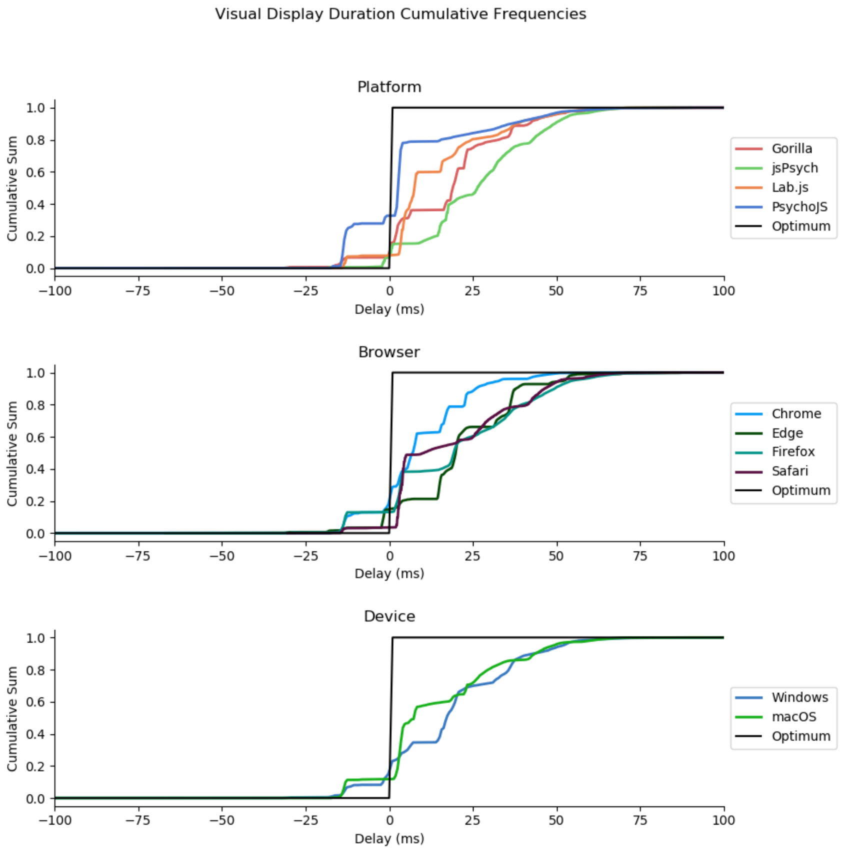


Figure 3. Cumulative Frequency plots for delays in visual display duration, separated by testing platform (top panel), browser (middle panel), and operating system (bottom panel).

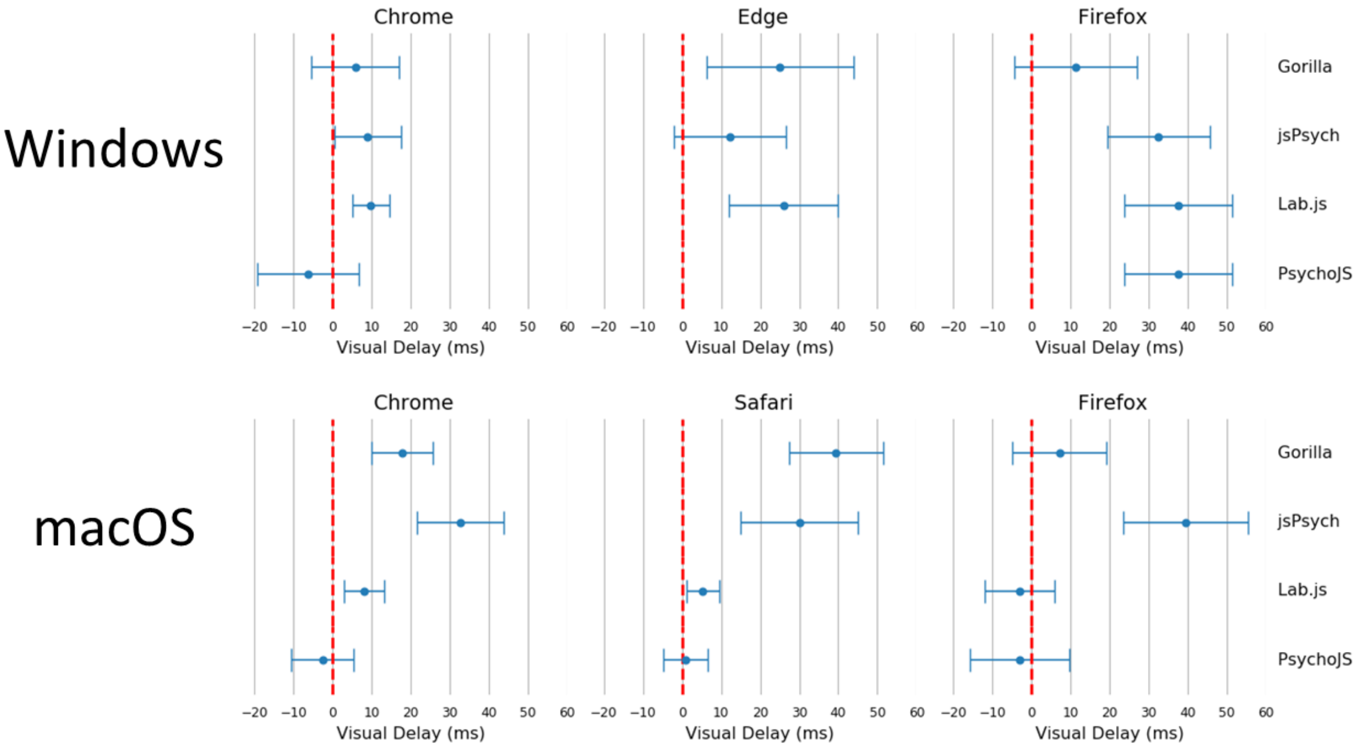


Figure 4. *Average visual delay across all frame lengths, broken down by browser, platform and operating system. Each point represents the average, with bars representing the standard error across all frames.*

Results

Summary statistics for this test are shown in **Table 1.** We have not converted all timings to frames, and have summarised the data in milliseconds for transparency – as the iMac screen appeared to not always stick to 60Hz. All platforms exhibited a positive delay (on average they overrepresented the duration of items), except for psychoJS, which both overestimated and underestimated. In terms of timing *Chrome* and *Windows* appear to show the smallest delay. In terms of variance, the smallest standard deviation was *Lab.js* which had a maximum delay of 16.49ms (one frame at 60Hz), an average of 9.8ms. The other platforms appear to exhibit almost equivalent delay. Browsers & platforms shown no superiority in terms of variance.

The most detailed overview of the results for VDD delay can be seen in **Figure 5**. The overall story is complex, with traces varying in shape, but some themes are apparent. In macOS, across different devices and platforms, jsPsych consistently showed a slight delay for requested durations between 3 and 20 frames. Firefox showed the largest amount of variance out of all the browsers, both between different frame lengths **(Figure** 5**)** and different platforms (**Figure 4**), leading to a more drawn out distribution in **Figure 6**. The best all-round browser was Chrome – it shows the least variance across devices and platforms – although it is more spread out between platforms on macOS (**Figure 4**).

The traces in **Figure 5** also tell us that delays persist in longer durations as well as shorter durations — in most platforms the error at one frame (16.66ms) was the same as the error at 30 frames (500ms). This is positive for users who wish to conduct research with different duration for different images — it means variability will be broadly equivalent between times. The exception to this is jsPsych, Firefox and Edge, which should probably be avoided in this scenario. Outliers are very rare, with 22 trials out of 103 500. They range from 95.75 to 265 ms. They are fairly equally distributed among some platforms (10 Gorilla, 9 PsychoJS, 4 lab.JS, 0 jsPsych), but it is difficult to draw inferences from so few instances. These are likely due to display or external chronometery anomalies – it is difficult to tell with such low rates of replication.

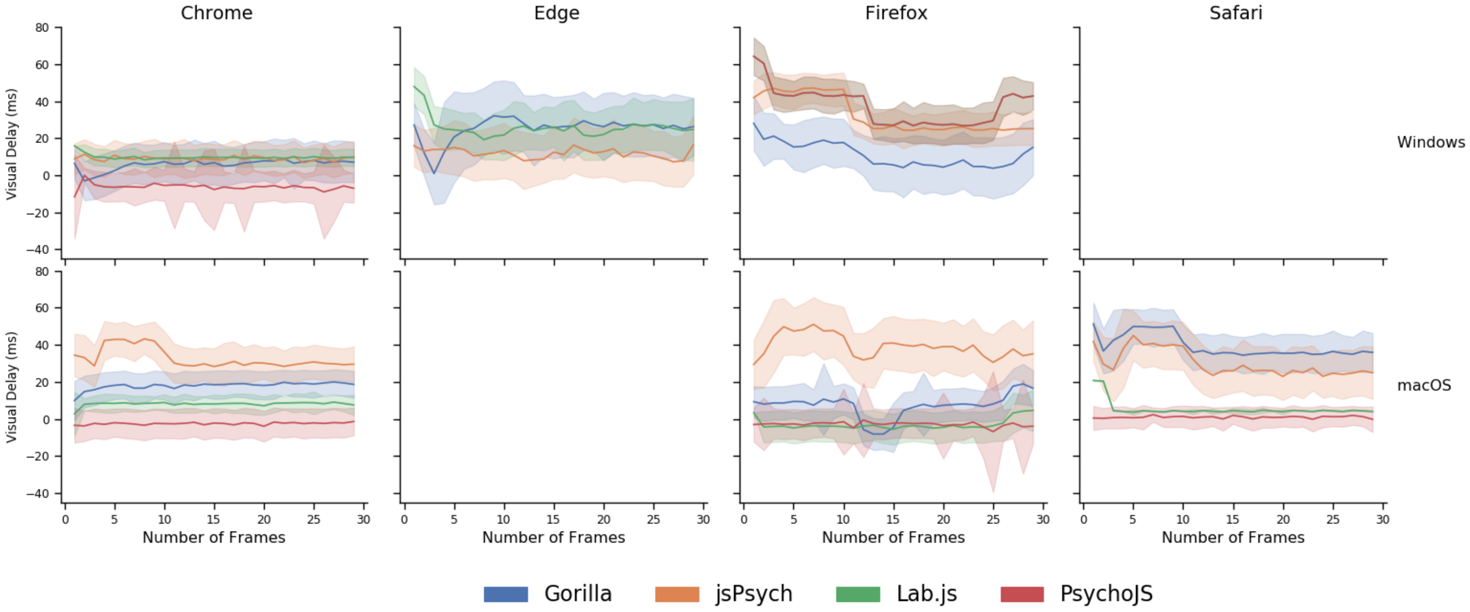


Figure 5. *Visual delay traces broken down by web browser, operating system and platform. Visual delay is the delta between requested and recording duration in milliseconds, this shown across 30 frames. The shaded errors represent standard error. Safari on Windows & Edge on macOS are not supported (so missing).*

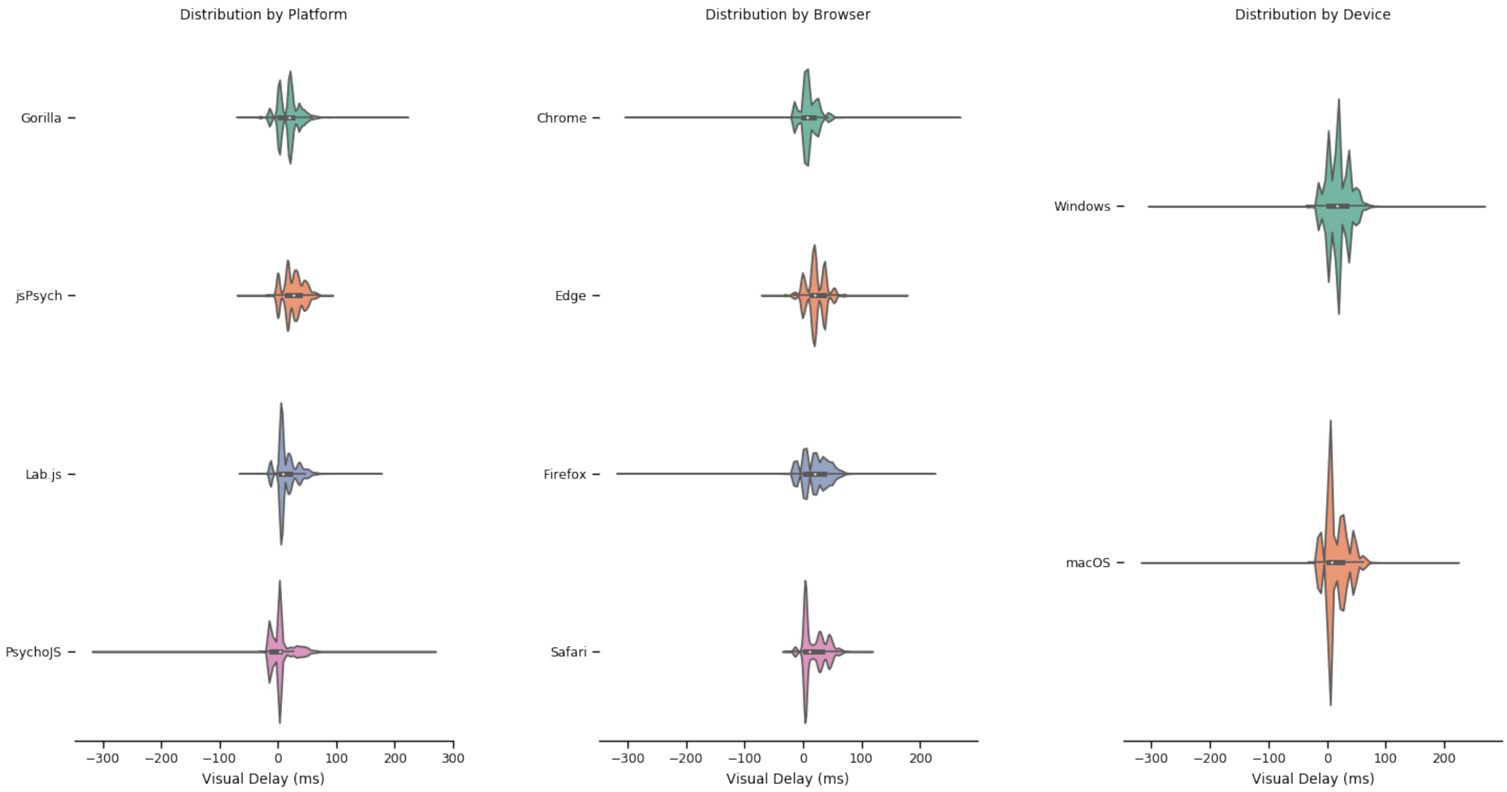


Figure 6. *Visual Delay Violin plots of data broken down by platform, browser and device. The shaded error represent the distribution density, the lines represent the span of times and the white dot represents the mean.*

2. Reaction Time Accuracy

This experiment assessed the accuracy of an entire system to record responses on a keyboard. The BBTK robotic actuator was programmed to press a space key in reaction to a white square at a pre-specified reaction time. This actuator uses a motor to propel a metal ‘finger’ with a foam tip onto the keyboard of the device. Once calibrated, it can deliver reaction times with sub-millisecond accuracy. We opted for using an actuator instead of deconstructing a keyboard to attach wires to the underlying board for two reasons: it enables us to easily test touch-screen devices in the future, and it more closely resembles what participants of online experiments do, without optimising for an unrealistic setup.

Methods

RT Task

As in the VDD experiment, an opto-detector was connected to each system on an elastic band, and connected to the BBTK. This detector acted as a trigger for the actuator to make a reponse, programmed with a fixed reaction time of either 100, 200, 300 or 500ms representing a reasonable range of fast responses by human participants. As in the VDD experiment, the opto-detector threshold was adjusted to suit each screen and setup. The actuator was calibrated before each testing run, using a TTL trigger button provided as part of the BBTK kit. 10 presses on this button give an initiation latency for the actuator, and this latency is accounted for when programming the key presses.

Some software tools force a full-screen by default on certain operating systems (e.g. Lab.js on Safari on macOS), which caused a white flash between setting the photodiode as a trigger and the experiment starting. This potential measurement problem was addressed by adding an extra single 10 ms press of actuator (not long enough to touch a key) before the main task, causing the initial flash to not impact the rest of the task. Very rarely (occurring only twice during all of the tests) the actuator would fail to be triggered by the opto-detector, or the keypress would not be registered. These trials were excluded from the analysis.

Browsers

As in the VDD test, we did not want to configure the browsers in any way beyond a standard user setup, so there was very minor variance in versions. The browser versions used were as follows. macOS Desktop: Chrome 76, Firefox 69, Safari 12, macOS Laptop: Chrome 75, Firefox 69, Safari 11. Windows Desktop: Chrome 76, Firefox 68, Edge 44, Windows Laptop: Chrome 75, Firefox 67, Edge 44.

At the time of testing psychoJS would not run on Edge on our set-ups, this compatibility has been fixed and we hope to include this data in a future version of this paper.

Devices

The two desktops were: 1) Windows Desktop Running Windows 10 Pro, with an Intel Core i5-2500 3.3 GHz CPU, 8Gb of RAM, and an 60Hz ASUS VS247 23.6” Monitor with a 1920 x 1090 resolution. 2) 2017 Apple iMac with an Intel Core i5-7400 3.0 GHz CPU, a built in 21.5” monitor with a 4096 x 2304 resolution.

Because laptops have different configurations of keyboards compared to desktops (i.e. they are connected internally rather than through USB), we employed two in this experiment. These were 1) Windows Surface Laptop 2, with an Intel core i7 CPU, 16Gb RAM, and an integrated touch-screen 13.5” 60Hz display with a 2256x1504 resolution. 2) Macbook Air early 2016, with an Intel Core m5 1.2Ghz CPU, 8Gb RAM, with a 12” 60Hz Retina display with a 2304 x 1440 resolution.

Platforms

The same versions of experiment software were used as in the VDD.

Data Processing

The delay scores were calculated as the difference between the known actuator reaction time and the recorded time on the software. No outliers (more than 4 Standard Deviations from the mean) were detected.

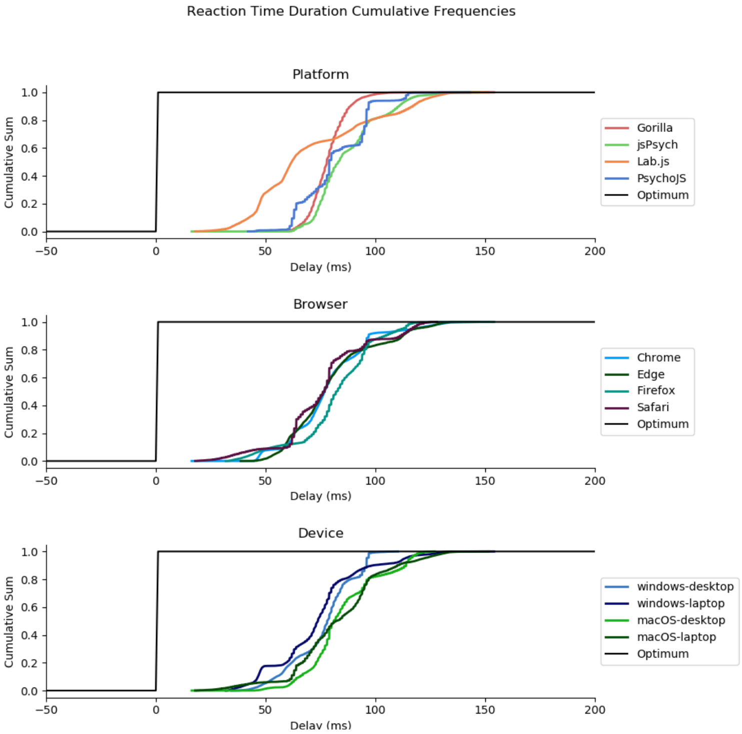
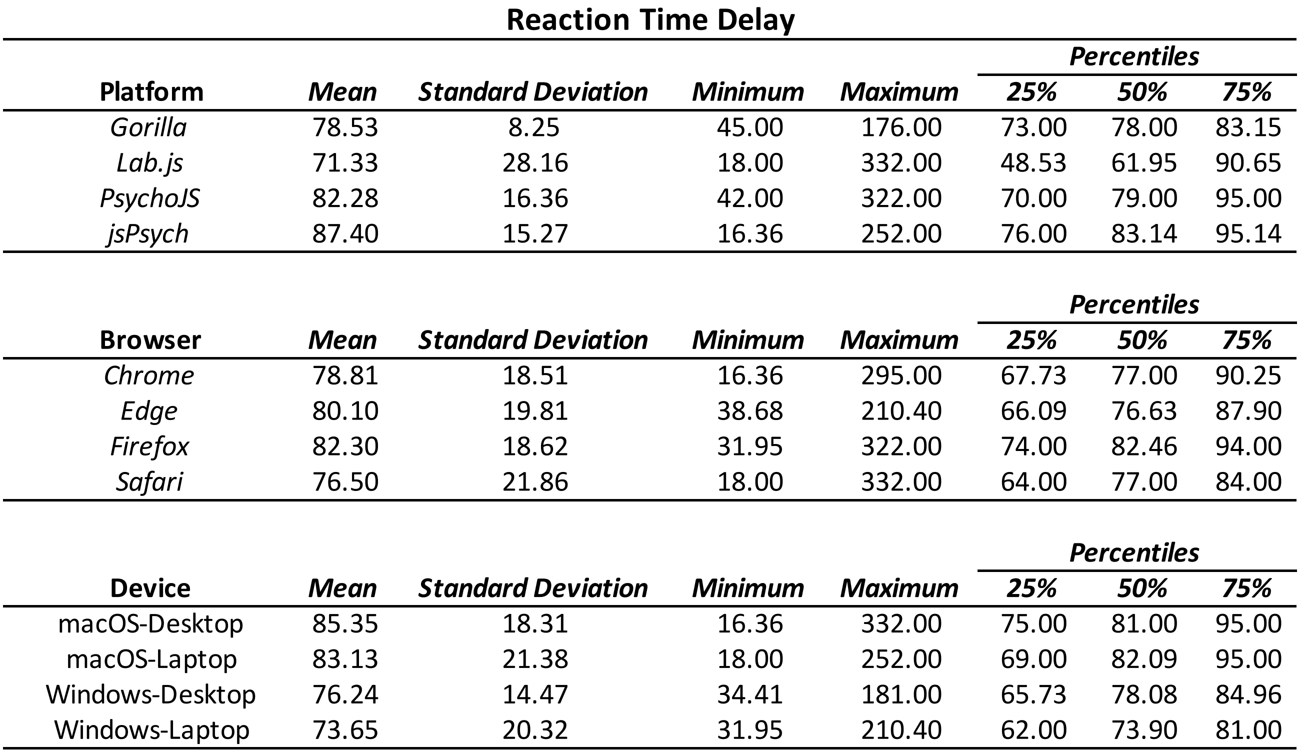


Figure 7. *Cumulative Frequency plots for delays in reaction time, separated by testing platform (top panel), browser (middle panel), and operating system (bottom panel).*



Reaction Time Delay

Platform	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Percentiles* *25%*	*50%*	*75%*
Gorilla	78.53	8.25	45.00	176.00	73.00	78.00	83.15
Lab.js	71.33	28.16	18.00	332.00	48.53	61.95	90.65
PsychoJS	82.28	16.36	42.00	322.00	70.00	79.00	95.00
jsPsych	87.40	15.27	16.36	252.00	76.00	83.14	95.14

Browser	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Percentiles* *25%*	*50%*	*75%*
Chrome	78.81	18.51	16.36	295.00	67.73	77.00	90.25
Edge	80.10	19.81	38.68	210.40	66.09	76.63	87.90
Firefox	82.30	18.62	31.95	322.00	74.00	82.46	94.00
Safari	76.50	21.86	18.00	332.00	64.00	77.00	84.00

Device	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Percentiles* *25%*	*50%*	*75%*
macOS-Desktop	85.35	18.31	16.36	332.00	75.00	81.00	95.00
macOS-Laptop	83.13	21.38	18.00	252.00	69.00	82.09	95.00
Windows-Desktop	76.24	14.47	34.41	181.00	65.73	78.08	84.96
Windows-Laptop	73.65	20.32	31.95	210.40	62.00	73.90	81.00

Table 2. *Summary of Reaction Time (RT) Delay results. RT Delay is calculated as the difference between known and recorded RT. It is broken down by Platform, Browser and Device. All results are reported after outliers have been excluded*

Results

Reaction Time delay (the difference between performed reaction time by the actuator and that recorded on the experiment platform), was first broken down by the requested reaction time (100, 200, 300 and 500 ms). This allows us to investigate if any particular duration led to more error in systems in general. This was not the case overall (**Figure 8**). There were also few differences between desktop and laptop computers, particularly on Windows. More importantly, experiment platforms did not all behave in similar ways.

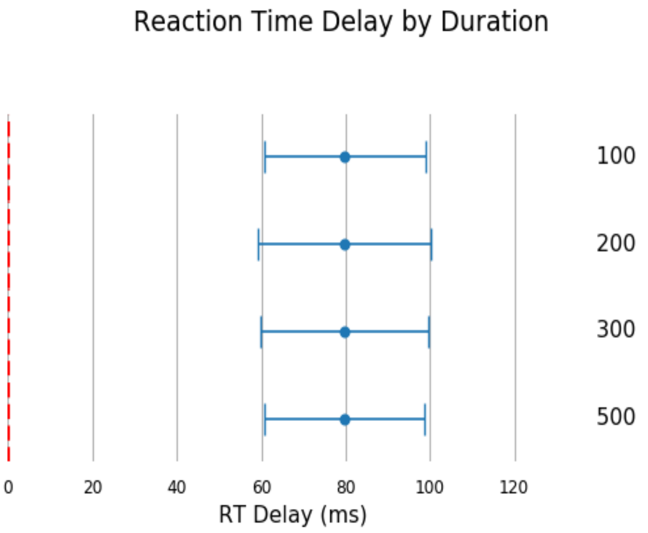


Figure 8. *Reaction time delay by requested duration. Points represent the mean, and error bars represent the standard deviation.*

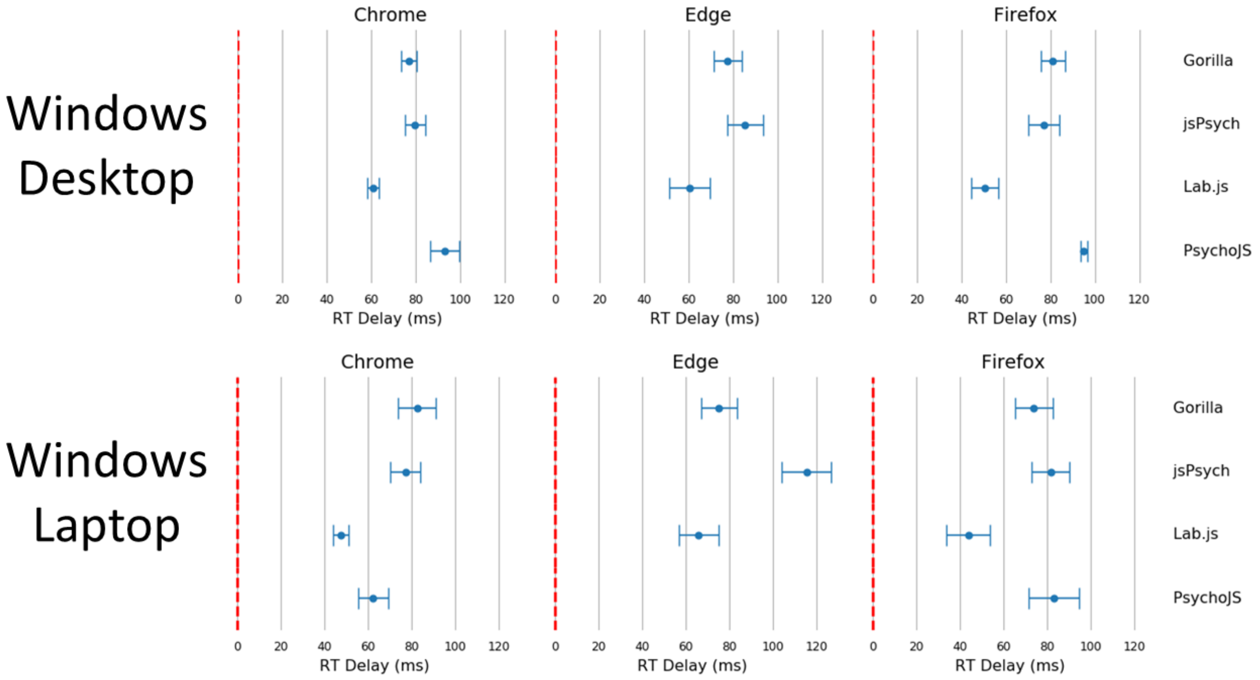


Figure 9. *Reaction time delay for Windows 10 devices broken down by browser, device and platform. Points represent the mean, and bars represent the standard deviation.*

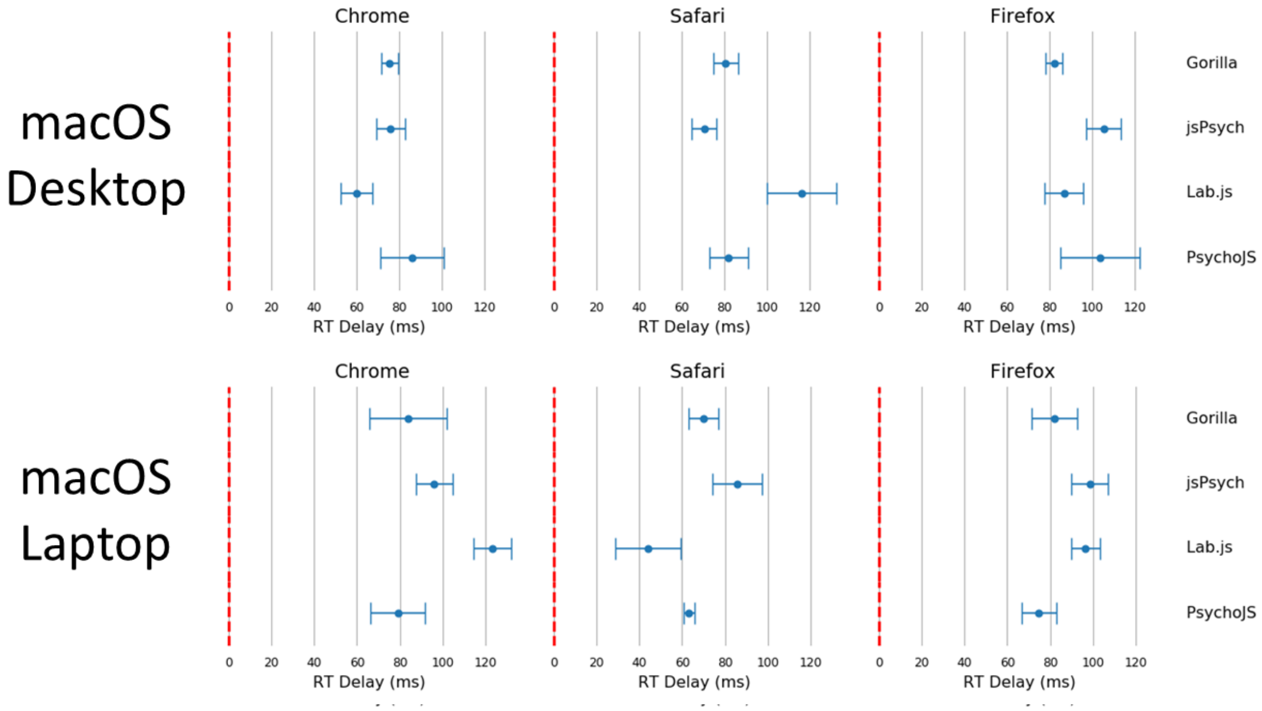


Figure 10. *Reaction time delay for macOS devices broken down by browser, device and platform. Points represent the mean, and bars represent the standard deviation.*

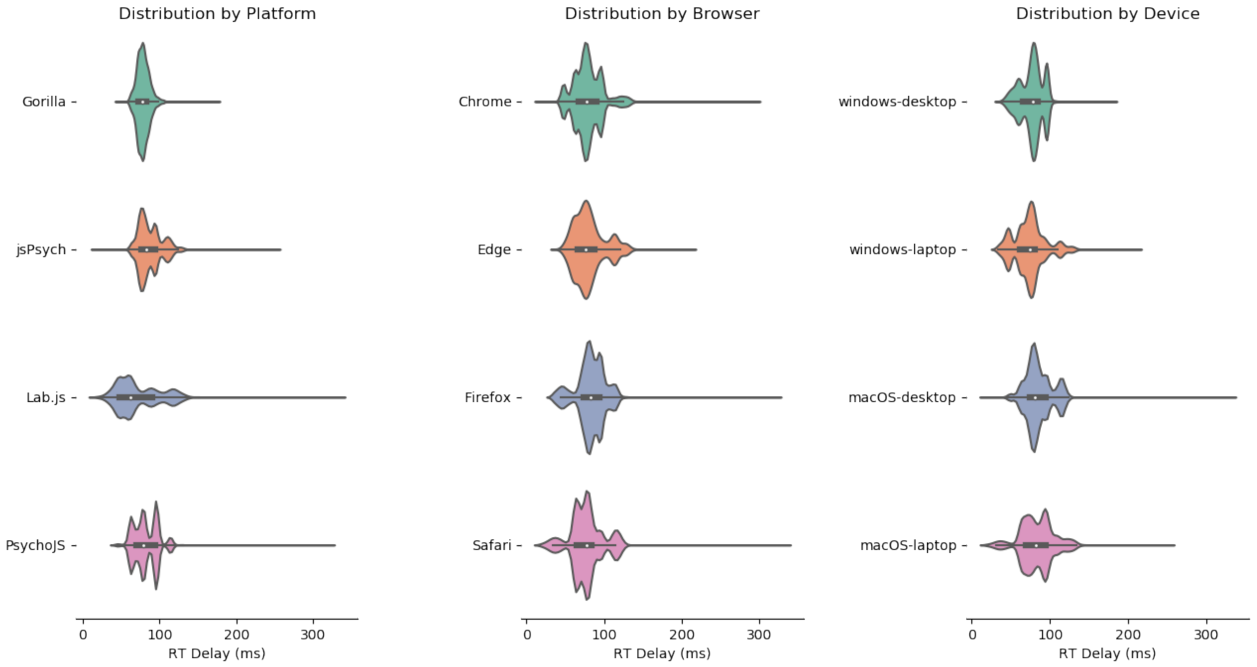


Figure 11. *Reaction time violin plots organized by platform, browser and device. Lines represent the maxima and minima, whereas the shaded error represents a distribution plot.*

Gorilla was relatively consistent overall, with around 80 ms of delay for all operating systems and device types. It also had good precision, with the lowest overall standard deviation out of all the platforms (8.25ms in **Table 2**). As discussed above, high precision in measuring reaction times permits a higher sensitivity for small differences between conditions. The platform also showed slightly higher standard deviations for laptops compared to desktop keyboards. However, this was in-line with the average results broken down by device type in **Table 2.**

jsPsych was consistent (around 70 ms) on desktop devices running Chrome or Safari, but less so for Firefox on macOS (desktop: around 110 ms, laptop: around 100 ms) and Windows (desktop and laptop: around 80 ms), and for Edge (desktop: around 85 ms, laptop: around 120 ms).

Lab.js showed a rather distributed pattern across all combinations of devices and browsers.
PsychoJS was relatively consistent (around 80 ms) on macOS, with the exceptions of Firefox on desktop (around 100 ms) and Safari on laptop (around 65 ms). It was also consistent on Windows desktop devices (around 95 ms) for Chrome and Firefox, but less so on the laptop (around 60 ms on Chrome, and 80 on Firefox). PsychoJS also shows clustering around 16ms increments, this likely due to RT logging within the animation loop at the time of testing. We understand in recent updates made in late 2019 have changed this.

3. Participant Survey

Operating Systems & Browsers

The first thing to consider is the types of devices users are utilising to access the internet. We found that 77% of these devices were desktop or laptop computers, whereas only 20% were mobile devices and just over 2% were tablets. A more detailed breakdown of the operating systems in use can be seen in **Figure 12**. The most common operating system was Windows,

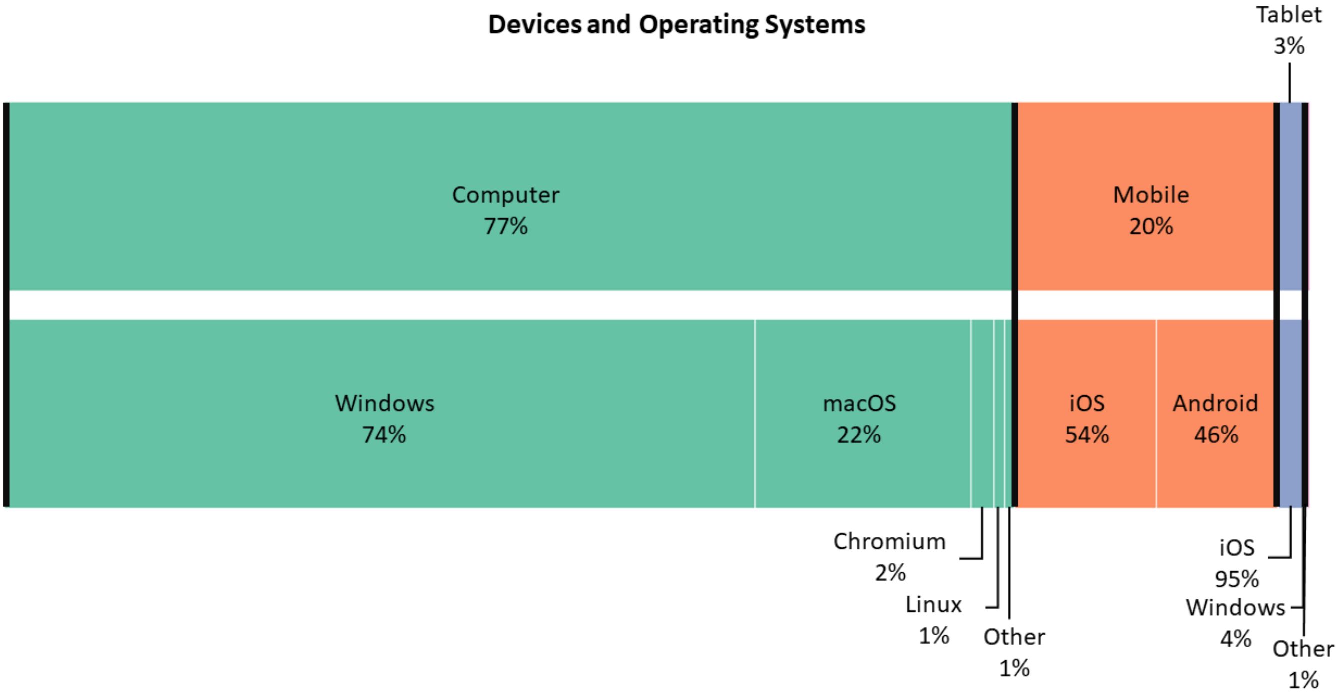


Figure 12. *Operating systems and devices, nested and stacked bar chart. Based on a sample of 202,600 participants. Percentages are rounded to the nearest integer.*

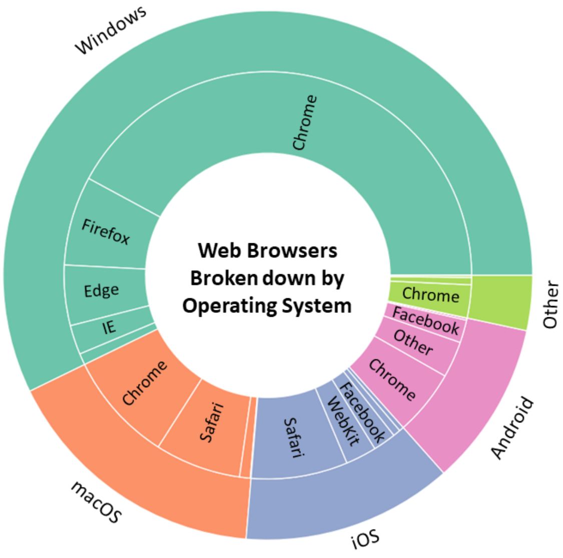


Figure 13. *Nested pie chart representing the breakdown of browsers within each operating system. For readability, wedges less than 3% are not labelled, but all are in the 'other' category.*

Operating System	*Chrome*	*Firefox*	*Safari*	*Edge*	*Internet Explorer*	*Facebook*	*Webkit*	*Other*
Windows	73.60%	12.40%	-	8.30%	4.10%	-	-	1.60%
macOS	53.10%	5.10%	41.40%	-	-	-	-	0.40%
iOS	-	-	61.20%	-	-	14.50%	19.60%	4.70%
Android	51.80%	1.70%	-	-	-	18.00%	-	28.50%
Other	77.80%	17.10%	-	0.40%	-	-	-	4.70%
Average	**59.00%**	**8.60%**	**14.50%**	**4.80%**	**2.30%**	**3.60%**	**2.40%**	**4.60%**

Table 3. *Browser percentage broken down by operating system. The average of each browser is taken from the total data, so is not an average of the operating systems – which have unequal number of users.*

followed by macOS. For mobile devices, users were roughly evenly split between iOS and Android, and the overwhelming majority of tablets were iPads running iOS.

Table 3 and **Figure 13** show the breakdown of participants' browsers by operating system. The most common browser was Chrome (59%), but this dominance varied depending on device (it was less popular on mobile operating systems). Overall, the average percentages for Chrome, Firefox, and Safari were in line with what we would expect from the global market share of 64.3%, 16.7%, and 4.5%, respectively ("Browser Market Share Worldwide," 2019). Where our sample differs, is in the use of the Facebook browser (3.6%), which is not listed in aforementioned market share statistics. It is likely to reflect researchers sharing studies in the mobile application Facebook Messenger, which opens links with it's built-in browser by default.

Screen Size & Window Size

The screen size of the devices, which limit the objective size of the item's presented on screen. Stimuli size, whilst less important for some phenomena, such as visual object priming (Biederman & Cooper, 1992) or perceptual learning (Furmanski & Engel, 2000), is important for others. In particular in some visual perceptual research – for example visual crowding (Tripathy & Cavanagh, 2002), where it can impact detectability of targets. We therefore looked at the variation and distribution of the participant's screen sizes.

It makes sense to analyze computers, mobiles and tablets separately – as experimenters interested in size

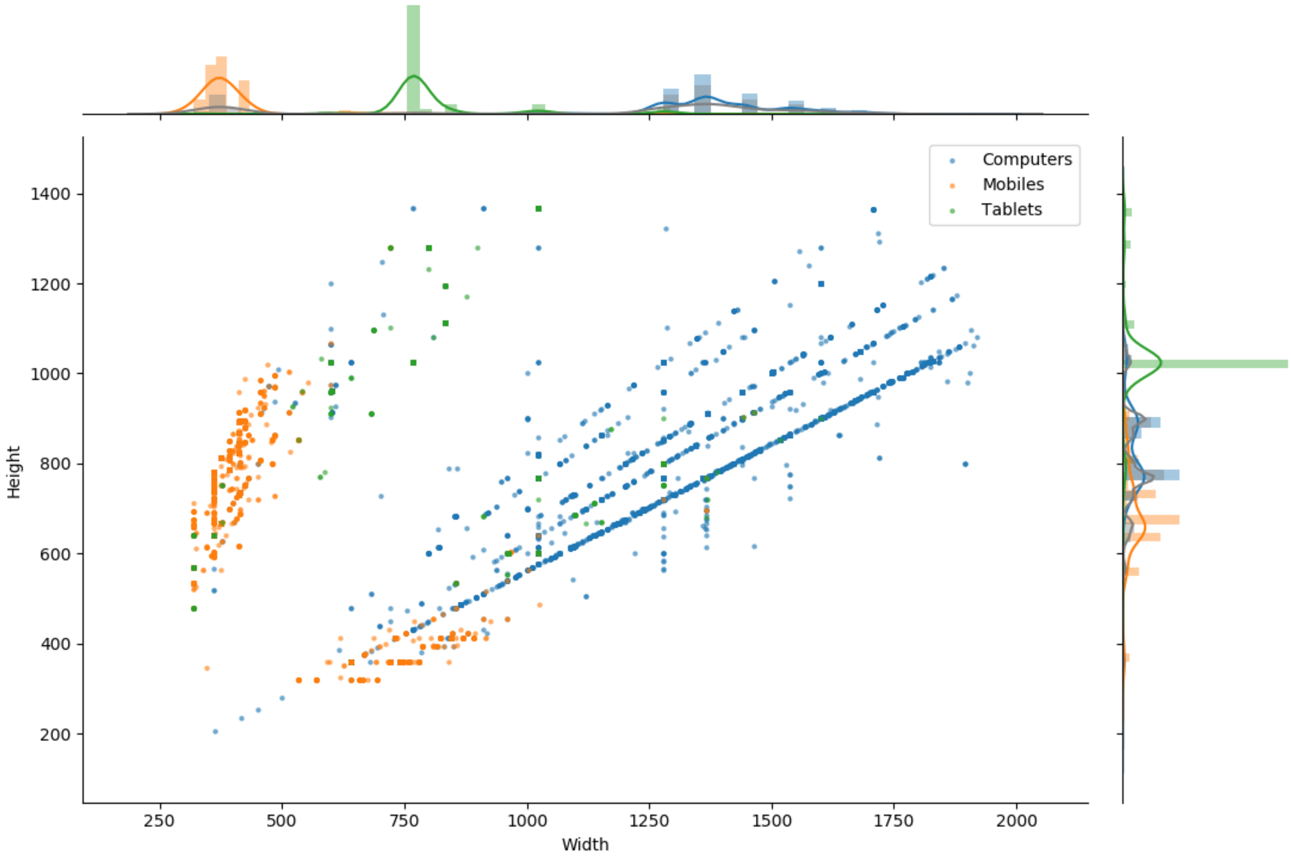


Figure 14. *Scatter graph of screen width and height, with histograms and kernel density estimation plots for each dimension. The diagonal lines represent the different aspect ratios.*

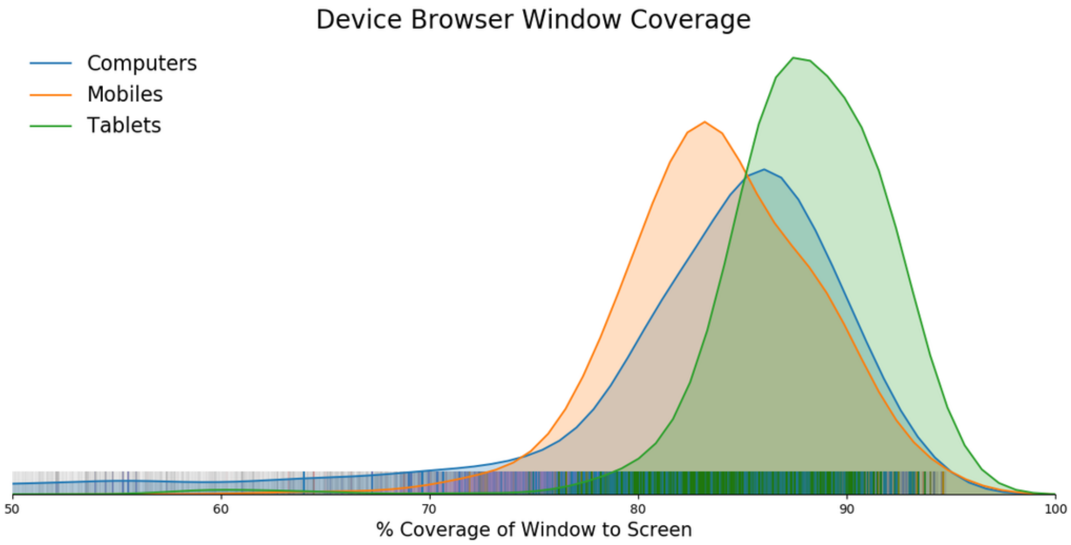


Figure 15. *Kernel Density Estimation of browser window coverage relative to screen size, with individual points as a carpet plot.*

are likely to restrict themselves to one of these categories. The two most common screen sizes for computers were 1366 x 768 pixels (23.2%) and 1920 x 1080 pixels (21.5%), for mobile devices these are 375 x 667 pixels (27.8%) and 360 x 640 pixels (18.5%) – both of in portrait mode, and finally tablets with 768 x 1024 (73.7%) – the logical resolution of all iPad minis and iPad airs.

Looking at the most frequent resolution combinations only tells part of the story, it becomes more interesting when we translate size into a continuous variable and look at the distribution of screen dimensions. This is illustrated in the scatter graph in **Figure 14**. The mean width of computer screens was 1387.6 pixels (SD = 161.9) and the height was 832.3 pixels (SD=99.5); mobile screens had a mean width of 393.2 pixels (SD = 92.4) and a height of 684 pixels (SD=109.5); tablets had a mean width of 811.7 pixels (SD=141.1) and the height was 1025 pixels (SD=128). The variance in tablets and mobiles is likely overestimated as the screen sizes are split between landscape and portrait users. This landscape/portrait split is illustrated in **Figure 14**, where tablets and mobile points appear to mirror each other in clusters.

Figure 14 also nicely shows the differing aspect ratios present in computers – with strong diagonal lines along those ratios (as the screens scale up with those rules). The most common aspect ratio was 16:9 / 1.77 to 1 – 41% of computers show this, and it scales up along the thickest blue line. There is also less clear aspect ratio lines for mobiles.
Screen size does not account for the entire presentation of stimuli on-screen. The browser window will always take up less than 100% unless the user is in full-screen mode. We quantified this in our sample by calculating the percentage coverage the browser window had on each screen. This can be seen illustrated in **Figure 15**. Computers have a longer tail of coverage, as users are able to scale the window with their mouse – something not as easy in tablets (highest coverage) and mobiles (slightly less).

Geography

We estimated geographical location from participants' timezone data. These were recorded in a standard format, and obtained using *moment.js*

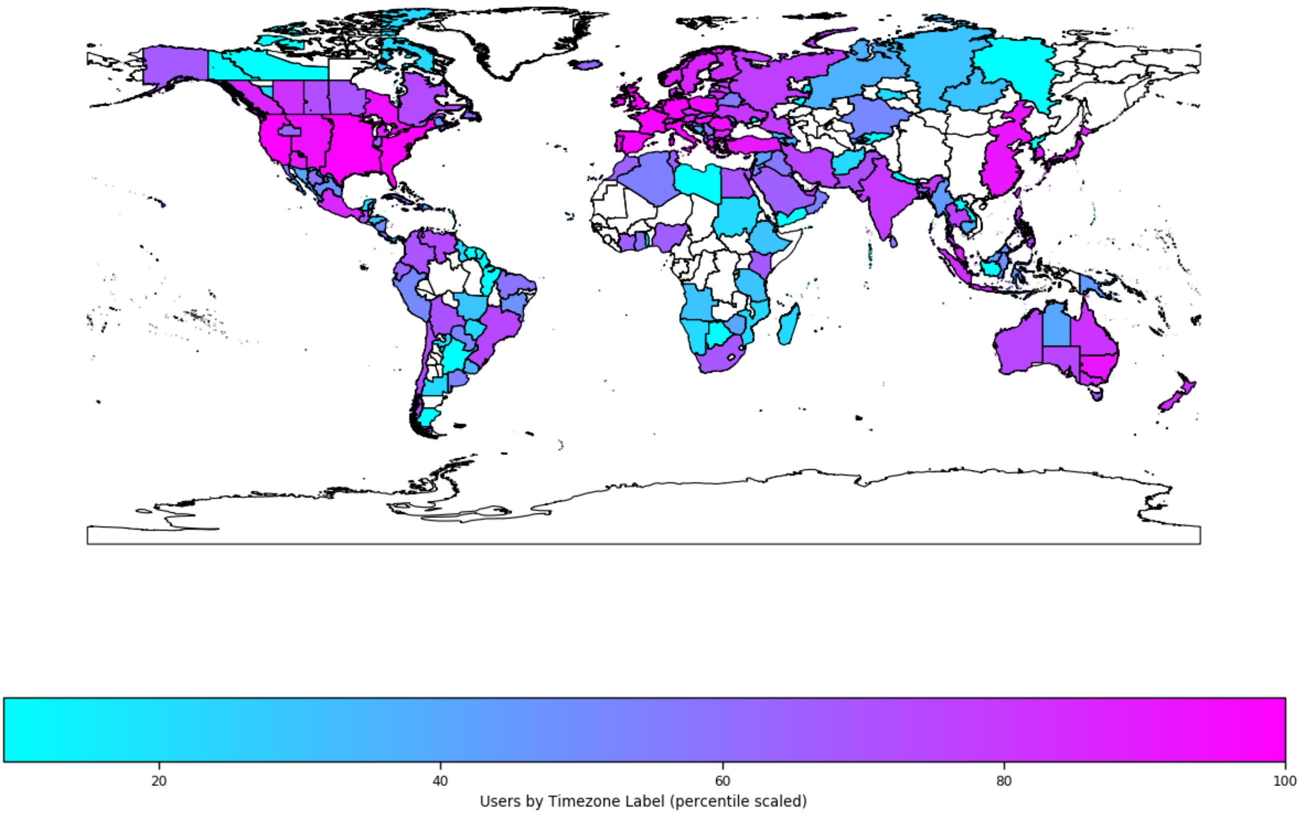


Figure 16. *Timezone of participants, scaled into percentiles for interpretability. Timezone areas are from the TZ Database (Lear & Eggert, 2012)*

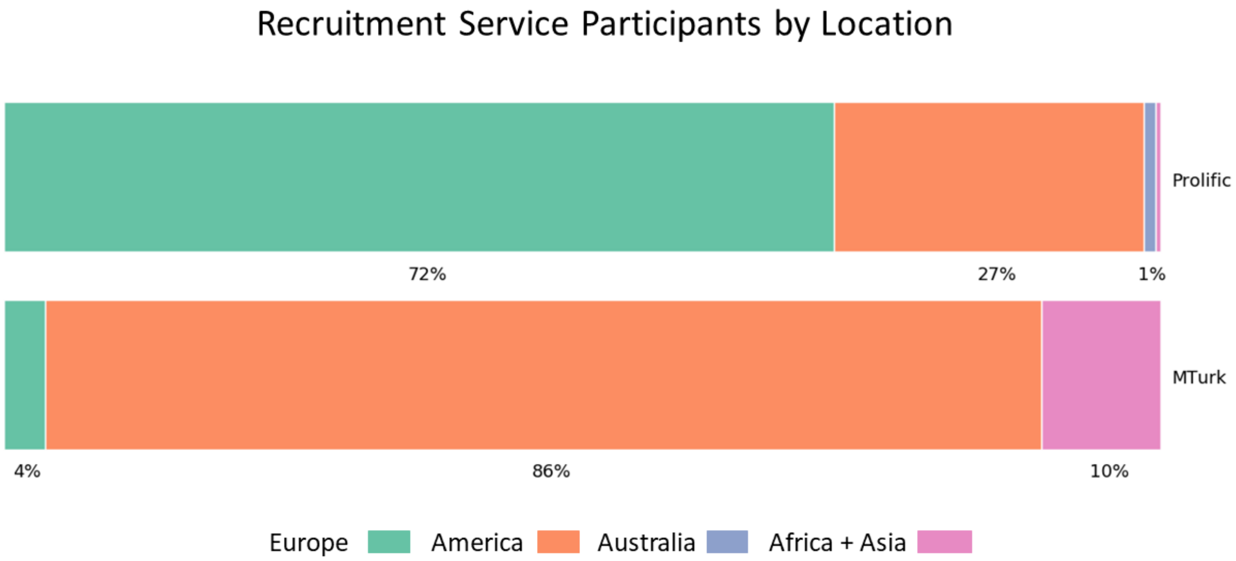


Figure 17. *Continent of participants from each recruitment platform. Africa and Asia are combined as they represent a relatively small number of participants.*

(https://momentjs.com/timezone/docs/). The labels produced refer to timezone localities, according to the TZ Database (Lear & Eggert, 2012). 70% (over 131 000) of the participants were based in Europe (mostly in UK: 53%, The Netherlands: 3%, Germany: 2%, and France: 1%), 23% (over 44 000) were based in the continent of America (mostly in TZ codes New_York: 10%, Chicago: 5%, and Los_Angeles: 3%). The distribution (**Figure 16**) is heavily biased towards westernised developed economies, which is not reflective of the broader internet-using population, the majority of which is based in Asia (57%) (International Telecommunication Union, 2019).

We were able to look at the geographical distribution of participants tested using different recruitment services, a breakdown between MTurk and Prolific is shown in **Figure 17**. Prolific (previously Prolific Academic) is an online study panel which specifically targets research participants rather than professional survey responders and human training for machine learning (Palan & Schitter, 2018) as such it's participant demographics in our sample (based on machine time-stamp) are more heavily skewed towards Europe and America – but with Europe being the dominant area. Whilst MTurk seems to show a large number of users from America, but also a relatively increased number from Africa and Asia. This difference could represent a difference in panel demographics, or it could reflect researchers criteria for recruitment within these websites.

Discussion

We undertook timing validation of presentation and response times on common browsers, platforms, and devices. Encouragingly all platforms are reasonably accurate and reliable for studies not needing <100ms Reaction Time accuracy or <2 frames presentation accuracy. However, we reveal complex patterns of variation between all set-up variables, and in general show that experiment platforms do not behave consistently between browsers and operating systems. We also conducted a survey of 200 000 online research participants, and found that some demographic factors do not overlap with the general online user population, and that choice of recruitment method impacts your population. The device, browser, and geographical distributions of online participants reported here could help researchers make sampling decisions.

We found that the choice of platform contributes greater variance than the device – contrary to earlier findings that systems introduced more variance than browsers (Reimers & Stewart, 2015). This is likely because browser technology has changed quickly in the last few years – as discussed in the introduction – and how platforms manage and render stimuli has also changed. Due to the huge number of trials we had to conduct, it was not feasible to undertake testing on more than the four devices assessed here, but it is perhaps worth replicating this analysis on a wider range of devices – such as touch-screen Android and iOS (despite these devices accounting for a smaller proportion of users in our sample). It is likely that the proportion of participants using these mobile devices will only increase. The relative small invariance of devices is good news for researchers, as the devices are often the variable that they are least able to control – this bodes well for the current state of online research.

In our findings, particularly noteworthy are the larger delays (compared to lab based software setups) in the recording of response times, which on average lag 80 ms, and extend to 100 ms on some setups. Researchers should keep this in mind when conducting reaction-time sensitive studies, by ensuring relative RT are used (Pronk et al., 2019). So, we recommend employing within-participant designs where possible to avoid having to make comparisons between participants with different devices, operating systems, and browsers.

The accuracy and precision differences between set-ups are relatively small, and for most researchers the guiding factor for platform selection should be individual preference and ease of use. For those interested in particularly niche and time sensitive tasks, platform selection strongly depends on the intended design and sample.

There is a clear – but small – difference in timing acuity between the presentation of stimuli and reaction times. Although, it is often the case that a task which demands an accurate and precise reaction time metric, also needs reasonable display metrics. For example, macOS devices appeared better for display accuracy than Windows devices but worse for reaction times (this can be seen in the device subplots of **Figure 3** and **Figure 7**) — so selecting this combination for achieving high presentation accuracy may unintentionally impact your reaction times.

To illustrate how these results may be used, it is helpful to provide a couple of vignettes. Consider a researcher, who does not want to limit their study to particular combinations of platforms and have good reaction accuracy, they could opt for using Gorilla, which performed with the highest precision across all setups for response time recording but was middle-of-field for visual accuracy and precision. In another example, a researcher could prioritise *only* response timing precision, and thus opt for using pyschoJS and only allowing participants who use Firefox on Windows desktop computers, which had the lowest standard deviation for response timing delay out of all combinations — but this combination performs relatively poorly in terms of average visual delay. A different researcher who needs to prioritise response time *accuracy* (i.e. low lag) could opt for using Lab.js, and only allowing participants who use Chrome, as this produced the lowest mean reaction time discrepancies.

In terms of informing researchers, this is the most comprehensive assessment of timing accuracy across different platforms, browsers and devices that has been carried out. For studies that require enhanced timing, we recommend using the plots in the results section to guide platform, browser and device choice. The interaction plots showing visual delay (**Figure 4**) and Reaction Time (**Figures 9** and **10**) are particularly informative for this purpose.

The best overall combination of browser and devsice to use – Chrome on Windows — was also the most common in our sample above (47.1%). This is good news – meaning a large proportion of your participants are likely to be using the most ideal equipment. Limiting participants' devices and browsers can be done programmatically in all tested platform, and via a graphical user interface in Gorilla (see https://gorilla.sc/support/reference/faq/requirements)

Conclusions

Whilst offering larger sample sizes, web experimentation introduces variation in participants' geographical locations and computer setups. We show that the accuracy and precision of display and response timing is not always consistent across different devices, operating systems, and experiment platforms; with no single platform standing out as the best. Our results also suggest that MTurk and Prolific participants are predominantly European and American, and that the best combination of browser and device (Chrome and Windows) is also the most common in use. Researchers who are keen to conduct online studies that include experiments for which timing is crucial would be wise to scrutinise the complex interactions between platforms, operating systems, and browsers; opt for within-participant designs, or potentially consider restricting participants' setup.

References

Anwyl-Irvine, A. L., Massonnié, J., Flitton, A., Kirkham, N., & Evershed, J. K. (2019). Gorilla in our midst: An online behavioral experiment builder. *Behavior Research Methods*. https://doi.org/10.3758/s13428-019-01237-x

Baker, J. D. (2013). Online Survey Software. *Online Instruments, Data Collection, and Electronic Measurements: Organizational Advancements*, 328–334. https://doi.org/10.4018/978-1-4666-2172-5.ch019

Barnhoorn, J. S., Haasnoot, E., Bocanegra, B. R., & Steenbergen, H. van. (2015). QRTEngine: An easy solution for running online reaction time experiments using Qualtrics. *Behavior Research Methods*, *47*(4), 918–929. https://doi.org/10.3758/s13428-014-0530-7

Biederman, I., & Cooper, E. (1992). Size Invariance in Visual Object Priming. *Journal of Experimental Psychology: Human Perception and Performance*, *18*(1), 121–133.

Birnbaum, M. H. (2000). *Psychological Experiments on the Internet*. Academic Press.

Birnbaum, M. H., & Wakcher, S. V. (2002). Web-based experiments controlled by JavaScript: An example from probability learning. *Behavior Research Methods, Instruments, & Computers*, *34*(2), 189–199. https://doi.org/10.3758/BF03195442

Bohannon, J. (2016). Mechanical Turk upends social sciences. *Science*, *352*(6291), 1263–1264. https://doi.org/10.1126/science.352.6291.1263

Button, K. S., Ioannidis, J. P. A., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S. J., & Munafò, M. R. (2013). Power failure: Why small sample size undermines the reliability of neuroscience. *Nature Reviews. Neuroscience*, *14*(5), 365–376. https://doi.org/10.1038/nrn3475

Clifford, S., & Jerit, J. (2014). Is There a Cost to Convenience? An Experimental Comparison of Data Quality in Laboratory and Online Studies. *Journal of Experimental Political Science*, *1*(2), 120–131. https://doi.org/10.1017/xps.2014.5

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

Furmanski, C. S., & Engel, S. A. (2000). Perceptual learning in object recognition: Object specificity and size invariance. *Vision Research*, *40*(5), 473–484. https://doi.org/10.1016/S0042-6989(99)00134-0

Garaizar, P., & Reips, U.-D. (2019). Best practices: Two Web-browser-based methods for stimulus presentation in behavioral experiments with high-resolution timing requirements. *Behavior Research Methods*, *51*(3), 1441–1453. https://doi.org/10.3758/s13428-018-1126-4

Garaizar, P., Vadillo, M. A., López-de-Ipiña, D., & Matute, H. (2014). Measuring Software Timing Errors in the Presentation of Visual Stimuli in Cognitive Neuroscience Experiments. *PLOS ONE*, *9*(1), e85108. https://doi.org/10.1371/journal.pone.0085108

Givaty, G., Veen, H. A. H. C. van, Christou, C., & Buelthoff, H. H. (1998). Tele-experiments—Experiments on spatial cognition using VRML-based multimedia. *Proceedings of the Annual Symposium on the Virtual Reality Modeling Language, VRML*, 101–105. https://pure.unic.ac.cy/en/publications/tele-experiments-experiments-on-spatial-cognition-using-vrml-base

Hecht, H., Oesker, M., Kaiser, A., Civelek, H., & Stecker, T. (1999). A perception experiment with time-critical graphics animation on the World-Wide Web. *Behavior Research Methods, Instruments, & Computers*, *31*(3), 439–445. https://doi.org/10.3758/BF03200724

Kelley-Milburn, D., & Milburn, M. A. (1995). Cyberpsych: Resources for Psychologists on the Internet. *Psychological Science*, *6*(4), 203–211. https://doi.org/10.1111/j.1467-9280.1995.tb00594.x

Lear, E., & Eggert, P. (2012). *Procedures for Maintaining the Time Zone Database*. Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc6557

McGraw, K. O., Tew, M. D., & Williams, J. E. (2000). The Integrity of Web-Delivered Experiments: Can You Trust the Data? *Psychological Science*, *11*(6), 502–506. https://doi.org/10.1111/1467-9280.00296

Neath, I., Earle, A., Hallett, D., & Surprenant, A. M. (2011). Response time accuracy in Apple Macintosh computers. *Behavior Research Methods*, *43*(2), 353. https://doi.org/10.3758/s13428-011-0069-9

Palan, S., & Schitter, C. (2018). Prolific.ac—A subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, *17*, 22–27. https://doi.org/10.1016/j.jbef.2017.12.004

Peddie, J. (2019). *Global GPU shipments mixed in Q1’19 reports Jon Peddie Research*. https://www.jonpeddie.com/press-releases/global-gpu-shipments-mixed-in-q119-reports-jon-peddie-research/

Plant, R. (2014). *Quick, quick, slow: Timing inaccuracy in computer-based studies means we may need to make use of external chronometry to guarantee our ability to replicate.* 44th Annual Meeting of the Society for Computers in Psychology (SCiP), Long Beach, California.

Plant, R. R. (2016). A reminder on millisecond timing accuracy and potential replication failure in computer-based psychology experiments: An open letter. *Behavior Research Methods*, *48*(1), 408–411. https://doi.org/10.3758/s13428-015-0577-0

Pronk, T., Wiers, R. W., Molenkamp, B., & Murre, J. (2019). Mental chronometry in the pocket? Timing accuracy of web applications on touchscreen and keyboard devices. *Behavior Research Methods*. https://doi.org/10.3758/s13428-019-01321-2

Reimers, S., & Stewart, N. (2015). Presentation and response timing accuracy in Adobe Flash and HTML5/JavaScript Web experiments. *Behavior Research Methods*, *47*(2), 309–327. https://doi.org/10.3758/s13428-014-0471-1

Reimers, S., & Stewart, N. (2016). Auditory presentation and synchronization in Adobe Flash and HTML5/JavaScript Web experiments. *Behavior Research Methods*, *48*(3), 897–908. https://doi.org/10.3758/s13428-016-0758-5

Reips, U.-D. (2001). The Web Experimental Psychology Lab: Five years of data collection on the Internet. *Behavior Research Methods, Instruments, & Computers*, *33*(2), 201–211. https://doi.org/10.3758/BF03195366

Reips, U.-D., & Stieger, S. (2004). Scientific LogAnalyzer: A Web-based tool for analyses of server log files in psychological research. *Behavior Research Methods, Instruments, & Computers*, *36*(2), 304–311. https://doi.org/10.3758/BF03195576

Rodd, J. (2019). How to Maintain Data Quality When You Can’t See Your Participants. *APS Observer*, *32*(3). https://www.psychologicalscience.org/observer/how-to-maintain-data-quality-when-you-cant-see-your-participants

Schmidt, W. C., Hoffman, R., & Macdonald, J. (1997). Operate your own World-Wide Web server. *Behavior Research Methods, Instruments, & Computers*, *29*(2), 189–193. https://doi.org/10.3758/BF03204809

Tripathy, S. P., & Cavanagh, P. (2002). The extent of crowding in peripheral vision does not scale with target size. *Vision Research*, *42*(20), 2357–2369. https://doi.org/10.1016/S0042-6989(02)00197-9