Tutorial

Advances in Methods and Practices in Psychological Science
2019, Vol. 2(2) 188–196

DOI: 10.1177/2515245919838479
www.psychologicalscience.org/AMPPS

TaskMaster: A Tool for Determining When Subjects Are on Task

Stephanie Permut, Matthew Fisher, and Daniel M. Oppenheimer
Department of Social and Decision Sciences, Carnegie Mellon University

Abstract

In this Tutorial, we introduce a tool that allows researchers to track subjects' on- and off-task activity on Qualtrics' online survey platform. Our TaskMaster tool uses JavaScript to create several arrayed variables representing the frequency with which subjects enter and leave an active survey window and how long they remain within a given window. We provide code and instructions that will allow researchers to both implement the TaskMaster and analyze its output. We detail several potential applications, such as in studies of persistence and cheating, and studies that require sustained attention to experimental outcomes. The TaskMaster is designed to be accessible to researchers who are comfortable designing studies in Qualtrics, but who may have limited experience using programming languages such as JavaScript.

Keywords

attention, self-control

Received 9/4/18; Revision accepted 2/26/19

In this Tutorial, we introduce a novel tool, the TaskMaster, that allows experimenters to quantify research subjects' thoughtful engagement and honest participation. The TaskMaster offers an alternative to existing quality checks for crowdsourced data, such as screening measures (Hauser, Paolacci, & Chandler, 2018) and reverse-worded items that can be analyzed for response patterns indicative of inattention (Curran, 2016; Thomas & Clifford, 2017). The rise of crowdsourcing platforms to recruit online research subjects has produced continued debate regarding the merits of Web-based samples as compared with more traditional research populations, such as university students. Although Web-based samples are more demographically diverse, some researchers have raised concerns about the relative attentiveness, honesty, and experience of online subject pools (Chandler, Mueller, & Paolacci, 2014).

Indeed, data from nondiligent subjects are noisy and can drastically reduce a study's statistical power, as well as the interpretability of one's results. Methodological innovations such as the instructional manipulation check (Oppenheimer, Meyvis, & Davidenko, 2009) are intended to address one such source of inattention: a form of satisficing in which subjects do not carefully read the instructions that are provided. Several studies have found that workers recruited using Amazon Mechanical Turk (MTurk) pass attention-check questions at rates that are similar and sometimes superior to those of student samples (Paolacci, Chandler, & Ipeirotis, 2010).

This Tutorial is designed to help researchers address a related concern regarding online research populations: off-task behavior. MTurk workers report high rates of multitasking, ranging from unrelated Internet browsing to use of mobile phones while working on a task (Clifford & Jerit, 2014). Some workers will work on multiple tasks concurrently (Chandler et al., 2014). Research on multitasking shows that divided attention that induces central cognitive bottlenecks can decrease performance quality and increase error rates (Borst, Taatgen, & van Rijn, 2010). Moreover, off-task behavior

Corresponding Author:
Stephanie Permut, Department of Social and Decision Sciences, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213
E-mail: spermut@andrew.cmu.edu

may be a source of noise within a study if subjects vary in the amounts of time they are on and off task. Whereas off-task or inattentive subjects are easily flagged in a laboratory setting, identifying off-task behavior is considerably more challenging in online environments. Indeed, a reviewer cited by Hauser et al. (2018) pointed to multitasking as a particularly pressing concern, stating, "There is nothing that can be learned from research on MTurk except that multitasking produces dumb answers" (p. 2).

Perhaps of greater concern is the fact that off-task behavior can jeopardize the descriptive validity of survey items assessing subjects' knowledge (Luskin & Bullock, 2011; Peer, Brandimarte, Samat, & Acquisti, 2017) and may confer unfair advantages to subjects during competitive tasks. For example, subjects could use an online anagram solver to place in the top percentile of a scrambled-word task or could consult external resources to boost their score on a political-knowledge questionnaire (Jensen & Thomsen, 2014). Existing quality checks are poorly suited to detect such behaviors, as they are designed to measure inattentive responding and low MTurk approval ratings, but not out-of-task behavior specifically.

We present a novel method for tracking subjects' activity using Qualtrics online survey software. Qualtrics is a widely utilized tool in psychology and behavioral-science research, and it is known for its straightforward user interface and extensive customizability. It is particularly well suited for questionnaire studies, as scalar, open-ended, and multiple-choice items are easily added and edited using the platform's graphical interface. Researchers with experience programming in HTML, CSS, and JavaScript may implement study designs that are more elaborate than those possible under the graphical interface using Qualtrics' JavaScript Question API. We have developed a tool that allows researchers to track how long subjects spend inside and outside a task window, thereby providing an unobtrusive and easily implemented measure of on- and off-task behavior.

Although existing tools such as Qualtrics' page timer allow researchers to monitor the amount of time subjects spend on a given page, they typically measure only the interval between the loadings of two pages, rather than the amount of time spent on and off a page within this interval.

Using Qualtrics' built-in programming infrastructure, we created a JavaScript-based tool that measures the amount of time users spend on and off each page. This tool can be added to any Qualtrics survey—even one that does not otherwise utilize JavaScript in its design. The TaskMaster uses standard JavaScript onfocus( ) and onblur( ) events, which occur when a specified element (in this case, the survey window) enters and leaves focus. Our tool is built using infrastructure similar to that of a previously validated tool implemented by Diedenhofen and Musch (2017) and of de Leeuw's (2015) jsPsych browser-based experimental research platform. Diedenhofen and Musch used focus and blur events to capture on- and off-task browsing behavior and found that off-task browsing is responsive to incentives in a fact-based "knowledge test." They observed an increase in off-page navigation when test items were easily searchable and when subjects faced large performance incentives.

We consider the TaskMaster an extension of Diedenhofen and Musch's (2017) work that is adapted for general use within the existing Qualtrics interface. Loading of a page activates a timer that tracks the amount of time the survey window remains in focus. Time spent with the task window out of focus—that is, time spent in other tabs or applications—is measured and summed. The TaskMaster generates an index of how much time a subject spends on and off task for each page of the survey, as well as the number of times the subject leaves and returns to each page of the survey. Whereas Diedenhofen and Musch's PageFocus tool only saves and displays the interval between the most recent defocusing and refocusing events, the TaskMaster produces an array of all on- and off-task activity for the duration of the survey. The timing application is unnoticeable to subjects as they complete the survey. Notably, although the TaskMaster enables the researcher to collect information about time spent on and off task, it does not offer insight as to how and where off-task time is spent. In situations in which the specifics of off-task activity are considered relevant, researchers may include additional questions that are presented to subjects whose off-task time exceeds a designated cutoff.

Furthermore, the TaskMaster is able to detect off-task browsing only on the device where the survey is taken. It cannot identify subjects who, for instance, consult their mobile devices while completing a task on a computer. Still, to the extent that task switching is a concern to researchers and reviewers and poses a threat to experimental validity (see Diedenhofen & Musch, 2017, who used focus and blur events to detect off-task browsing during an incentivized knowledge task), the TaskMaster may help to address some of these concerns about experimental validity.

Implementation

This tool is implemented by first adding to one's survey the code available at https://github.com/steve-permut/TaskMaster (and in this article's appendices). Researchers using Qualtrics' survey builder should paste the

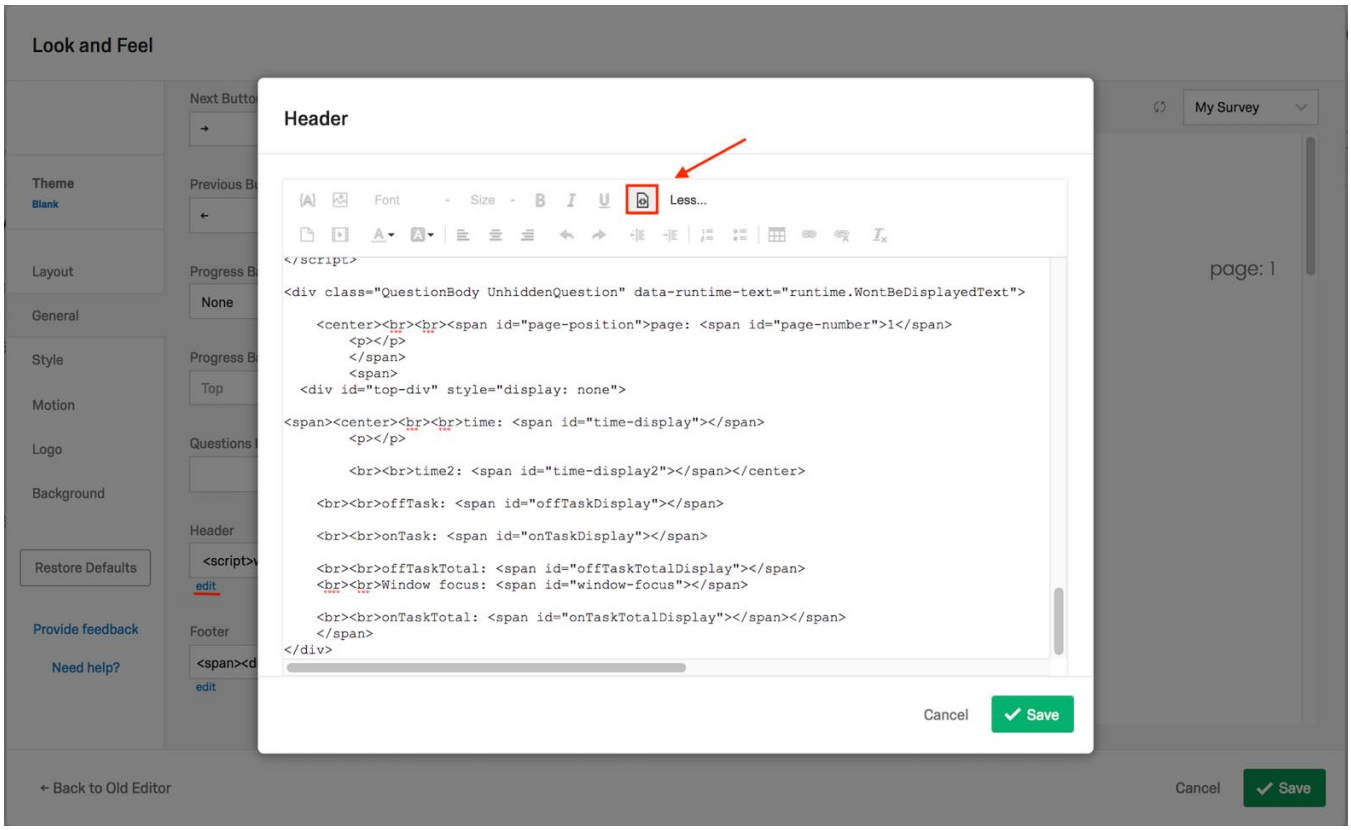

Fig. 1. Screenshot of Qualtrics' Look & Feel menu with the header source editor opened and the source code displayed. The TaskMaster code must be pasted in the header's source code, as the default rich-text editor does not save custom JavaScript. To paste in this source code, users should first access the header source editor by clicking on the "edit" link (highlighted by the red underline) under the box labeled "Header." Next, they should select the source icon (highlighted by the red box and arrow), and the source code will be displayed, as shown here. The TaskMaster's header script can then be pasted in the editor.

header code (https://github.com/steve-permut/TaskMaster/blob/master/HeaderCode.html or Appendix A) into the survey's header source html, which can be found in the General tab of the survey's Look & Feel menu (Fig. 1).

This code tracks time spent on and off each page, from the time the page has loaded to the time the user has moved on to the next page. When the survey data are downloaded from Qualtrics, time spent on and off task is represented as an array of positive (on-task) and negative (off-task) values, which can be summed to measure the total amount of time a subject spent on task or on a particular page. Moving to the next page of the survey automatically appends a "PAGEBREAK" string to the end of the array. The number of positive values and the number of negative values before each instance of PAGEBREAK represent the number of times the subject exited and reentered the work space.

The TaskMaster's footer code (https://github.com/steve-permut/TaskMaster/blob/master/FooterCode.html or Appendix B) allows researchers to track navigation data across an entire survey (rather than on a particular page). This code should be pasted in the source of the survey's footer html, which is also found in the General tab of the Look & Feel menu (see Fig. 1 for the equivalent field in the survey header).

The timing tool outputs five variables per monitored page:

- worktimeArray: an ordered array of the total time spent both on and off task
- onTask: an ordered array of the time spent on task; each item in the array represents a separate time the subjects' cursor entered the workspace and the amount of time the subject spent within the task window
- totalOnTask: the total amount of time on task (without subtracting time spent off task)
- offTask: an ordered array of the time spent off task; each item in the array represents a separate time the subjects' cursor left the work space and the amount of time the subject spent off task
- totalOffTask: the total amount of time spent off task

These variables are exported alongside other subject data in a Qualtrics-generated spreadsheet and must be

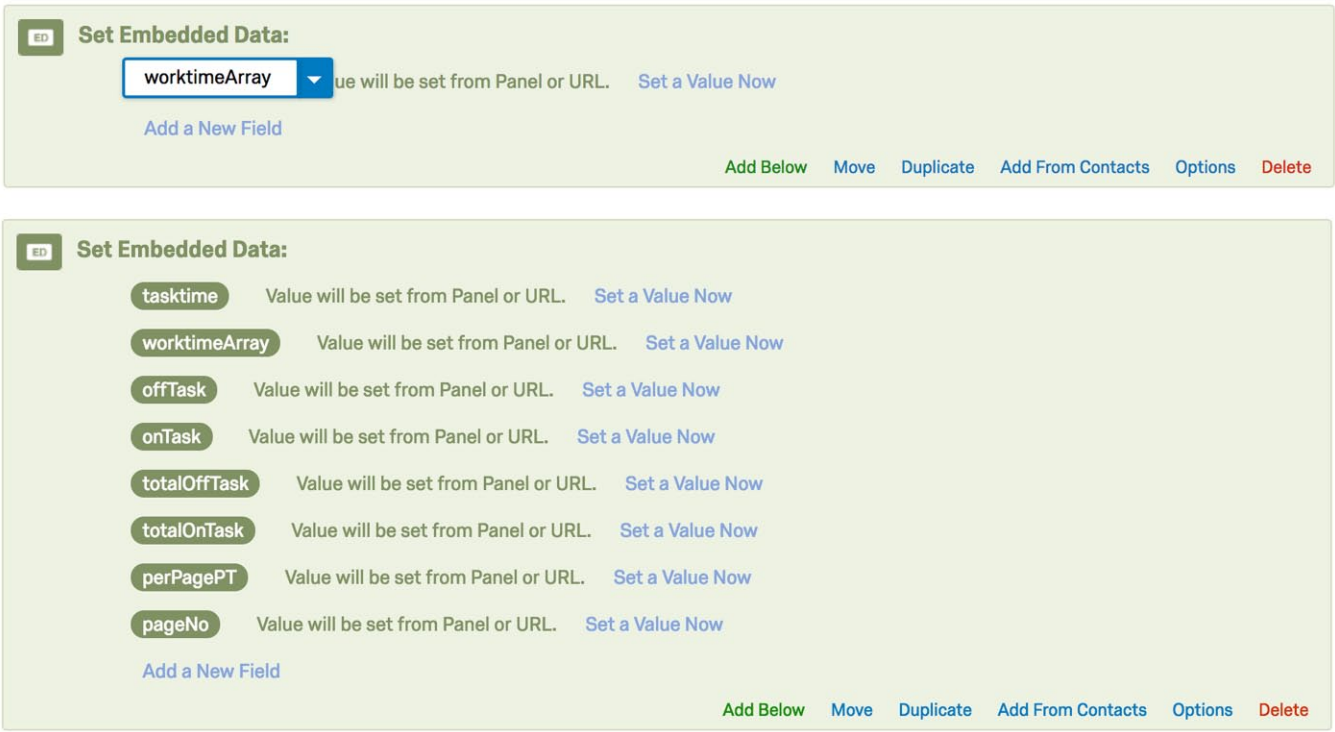

Fig. 2. Screenshots of Qualtrics' Survey Flow editor. The top image shows an embedded data field being set to the worktimeArray variable. The bottom image shows all of the embedded variables that must be input in order for the TaskMaster to run properly. Note that some of these fields are not discussed in the text because their output will not be of interest to the researcher.

manually added as empty embedded variables in Qualtrics' Survey Flow (see Fig. 2 for a list of all the embedded variables that must be entered). To add empty embedded variables, click on "+Add New Element" and then "Embedded Data." Next, click on "Create New Field or Choose from Dropdown" and input the first variable in the window under "Set Embedded Data" (see Fig. 2). Then click on "Add a New Field," below the window, and repeat the process for each of the remaining variables.

The variables must be labeled as just shown. For the script to run properly, study transitions (also accessed in a survey's Look & Feel section) must be disabled.

For ease of interpretation, we have developed an additional online tool (https://mfisher.shinyapps.io/expand_mouse_tracking_data) that researchers can use to upload their raw data from Qualtrics and have the timing data rendered in a more readable format. To use this Shiny app, begin in Qualtrics:

1. Click on "Data and Analysis"
2. Then click on "Export & Import"
3. Then click on "Export Data"
4. Select "as a CSV" and "using numeric values"

Next, go to the app (https://mfisher.shinyapps.io/expand_mouse_tracking_data):

5. Click on "Browse"
6. Upload the .csv file
7. When the file has been uploaded, click on "download"

This tool splits the single column of arrays outputted by Qualtrics into a set of per-page variables. Each row in the spreadsheet represents a single subject's on-task activity, and for each page of the survey, four columns are created (note that "N" in the variable labels is a placeholder for the specific page number; see Fig. 3 for an example of the app's output):

- Page_N: an ordered array of on- and off-task behavior at the page level. Negative numbers indicate the duration of intervals with the cursor outside the task window, and positive numbers indicate the duration of intervals with the cursor inside the task window.
- Page_N_ClickAways: the number of times the subject clicked away from the page. This variable corresponds to the count of negative values in the Page_N array.
- Page_N_TimeOffPage: the total amount of time spent off a given page (the absolute value of the sum of the negative values in the Page_N array).

	totalOnTask	Page_1	Page_1_ClickAways	Page_1_TimeOffPage	Page_1_TimeOnPage	Page_2	Page_2_Click	Page_2_Tim	Page_2_Tim
1	totalOnTask	perPagePT							
2	{"ImportId":'	{"ImportId":"perPagePT"}							
3	12	,-0.048,0.7	5	4.632	8.733	,-1.89,0.721	6	6.138	4.669
4	9	,-0.008,1.2	2	2.175	5.679	,-0.012,2.55	1	0.012	2.551
5	7	,-0.148,2.4	2	0.562	2.486	,-0.241,3.19	1	0.241	3.192

Fig. 3. Screenshot of the Shiny app output. Note that the columns are color coded here for ease of interpretation but are not color coded in the actual survey output. The white column to the left contains the totalOnTask variable, described earlier in the text.

- Page_N_TimeOnPage: the total amount of time spent on a given page (the sum of the positive values in the Page_N array).

To help researchers match the columns of app output with the correct pages of the survey, page numbers are displayed in the survey's preview mode. Display of page numbers is optional; they can be removed from the survey before launch.

Applications

We envision several uses for this tool. There are many cases in which researchers might be interested in subjects' on- and off-task behavior. The data provided by the TaskMaster could serve as a dependent measure or as an attention-check or quality-control mechanism.

The on-task and off-task time indices that are generated could be treated as supplementary measures of distraction that quantify interest in or attention to cognitively demanding tasks. If the data indicate that subjects reliably navigate away from a task during sections that require sustained attention, researchers could use this information to alter those sections in order to increase engagement. Diedenhofen and Musch (2017) were able to decrease off-task browsing through the inclusion of a pop-up, which was activated when the study window was not in focus.

Researchers might also treat off-task behavior as a stand-alone dependent measure—for example, to quantify the effectiveness of interventions designed to promote on-task behavior and cognitive control. Implementing our tool alongside a tedious exercise (e.g., a real-effort task) would allow researchers to establish baseline measures of behavior against which the effects of focused interventions might later be tested.

In addition, someone who is interested in studying task persistence without relying on self-report could do so unobtrusively using the time indices as dependent measures. The indices could also incentivize sustained on-task behavior if used to differentially warn or reward subjects who take many or few breaks over the course of a task.

We believe that the TaskMaster can help to shed light on an ongoing conversation within the behavioral sciences about ensuring the quality of online samples. It can be unclear whether low-quality open-ended responses are the work of automated bots or human subjects using virtual private servers (Dennis, Goodson, & Pearson, 2018). Human subjects may take generic text from online sources to use in place of self-generated responses. Because the TaskMaster monitors time spent on and off task, it could be used to diagnose the extent to which "suspicious" activity is human generated or fully automated.

Furthermore, researchers may use the TaskMaster to exclude subjects who appear to take excessive breaks over the course of a study. Although not every instance of off-task behavior should be considered excessive, it would be useful if experimenters could identify and flag cases in which more time has been spent off task than on.

A related use is to measure cheating behavior, as Diedenhofen and Musch (2017) did with their tool. Timing on- and off-task activity during a knowledge test can help to identify subjects who may have consulted outside resources to improve their scores. This application of the TaskMaster could improve the descriptive and construct validity of scales designed to measure prior knowledge or abilities.

Another, more direct cheating-detection application would be a "timesheet" study design wherein subjects self-report the amount of time spent on effortful tasks and receive a bonus payment based on the amount of time logged. Subjects' self-reported completion times could be compared against the totalOnTask variable, and large deviations would suggest deliberate misrepresentation.

Finally, in certain studies, researchers may want to validate that subjects have left or remained inside the survey window. For example, if subjects are instructed to read information from an external Web site, the TaskMaster can be used to verify if they have actually navigated away from the survey. Alternatively, some studies may require subjects to view stimuli for a certain period of time as part of an intervention or dependent measure (e.g., in a learning study, the amount of time subjects are exposed to materials is a key independent

variable). Although Qualtrics offers a handful of timing tools to prevent respondents from immediately advancing to the next page of the survey, there is no way to ensure that subjects remain on the intended page (as opposed to navigating to a different site).

Summary

We have described a simple tool that allows researchers to monitor time spent on and off task during Qualtrics surveys. The TaskMaster improves on existing task timers by generating an index of how much time subjects spend on and off each page of a survey. We have described several possible applications of this tool, including applications in studies of persistence and cheating, and studies in which sustained attention is relevant to experimental outcomes. By following the instructions provided here, researchers without programming experience can easily track subjects' activity in Qualtrics-based projects.

Appendix A: The TaskMaster's Header Code

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>
<script>
Qualtrics.SurveyEngine.addOnload(function()
{var jq = jQuery.noConflict();
  var pumpTimes = [jq('#timeArrayDIV')[0].innerHTML];
  var perPagePT = [jq('#perPagePTArrayDIV')[0].innerHTML];
  var duration = 0;
  var offTask = [jq('#offTaskDIV')[0].innerHTML];
  var onTask = [jq('#onTaskDIV')[0].innerHTML];
  var totalOnTaskPrev = [jq('#onTaskTotalDIV')[0].innerHTML];
  var totalOffTaskPrev = [jq('#offTaskTotalDIV')[0].innerHTML];
  var pageNo = parseFloat(jq('#page-no-DIV')[0].innerHTML);
  if (pageNo > 0) {
    jq('#page-number')[0].innerHTML = +pageNo + 1;
  }
    //creates a variable called timeOne containing date/time information on page load
    var timeOne = Date.now();
    jq(window).blur(function() {
      var window_focus = 0;
      jq('#window-focus')[0].innerHTML = window_focus;
      // creates a variable called timeTwo containing date/time information whenever
Ps leave workspace div
      var timeTwo = Date.now();
      // takes the interval between timeOne and timeTwo and converts to seconds (i.e.,
how long Ps spend in the task window). Steps below push this value to different arrays
      var duration = (timeTwo - timeOne) / 1000;
      // creates several arrays representing time spent on task and off task, time spent
on and off task on that page, and just time spent on task
      pumpTimes.push(duration);
      perPagePT.push(duration);
      onTask.push(duration);
      timeOne = Date.now();
        jq('#time-display2')[0].innerHTML = pumpTimes;
      var total = 0;
      for (var i = 0; i < pumpTimes.length; i++) {
        total += pumpTimes[i] << 0;
      }
      var totalOnTask = 0;
```

```
    for (var i = 0; i < pumpTimes.length; i++) {
      totalOnTask += onTask[i] << 0;
    }
    var OnTaskSum = (+totalOnTask + +totalOnTaskPrev)
    jq('#time-display')[0].innerHTML = total;
    jq('#onTaskDisplay')[0].innerHTML = onTask;
    jq('#onTaskTotalDisplay')[0].innerHTML = OnTaskSum;
  });
    jq(window).focus(function() {
      var window_focus = 1;
      jq('#window-focus')[0].innerHTML = window_focus;
    // performs equivalent series of steps, but measuring time spent off task instead
    var timeThree = Date.now();
    var duration = (0 - (timeThree - timeOne) / 1000);
    pumpTimes.push(duration);
    offTask.push(duration);
    perPagePT.push(duration);
    timeOne = Date.now();
    jq('#time-display2')[0].innerHTML = pumpTimes;
    var total = 0;
    for (var i = 0; i < pumpTimes.length; i++) {
      total += pumpTimes[i] << 0;
    }
    var totalOffTask = 0;
    for (var i = 0; i < pumpTimes.length; i++) {
      totalOffTask += offTask[i] << 0;
}
    var OffTaskSum = (+totalOffTask + +totalOffTaskPrev)
    jq('#offTaskDisplay')[0].innerHTML = offTask;
    jq('#offTaskTotalDisplay')[0].innerHTML = OffTaskSum;
   });
   Qualtrics.SurveyEngine.addOnPageSubmit(function() {
    var ValidationCheck = document.getElementsByClassName('ValidationError');
    var i;
    for (i=0; i < ValidationCheck.length; i++) {
       if (ValidationCheck[i].style.display == 'none') {
         var PassValidation = 1;
    } else {
      PassValidation = 0;
      break;
    };
  };
    if (PassValidation == 1) {
       pumpTimes = pumpTimes.toString();
       pumpTimes = pumpTimes.split(",@,")
       perPagePT.push(" PAGE BREAK ");
       perPagePT = perPagePT.toString();
       perPagePT = perPagePT.split(",@,")
       var total = 0;
       var total = parseFloat(jq('#time-display')[0].innerHTML);
       var totalOnTask = parseFloat(jq('#onTaskTotalDisplay')[0].innerHTML);
       var totalOffTask = parseFloat(jq('#offTaskTotalDisplay')[0].innerHTML);
       onTask = onTask.toString();
```

```
        onTask = onTask.split(",@,")
        offTask = offTask.toString();
        offTask = offTask.split(",@,")
        var pageNo = jq('#page-no-DIV')[0].innerHTML;
        pageNo++;
     Qualtrics.SurveyEngine.setEmbeddedData('worktimeArray', pumpTimes);
     Qualtrics.SurveyEngine.setEmbeddedData('perPagePT', perPagePT);

     Qualtrics.SurveyEngine.setEmbeddedData('tasktime', total);
     Qualtrics.SurveyEngine.setEmbeddedData('totalOnTask', totalOnTask);
     Qualtrics.SurveyEngine.setEmbeddedData('totalOffTask', totalOffTask);
     Qualtrics.SurveyEngine.setEmbeddedData('offTask', offTask);
     Qualtrics.SurveyEngine.setEmbeddedData('onTask', onTask);
     Qualtrics.SurveyEngine.setEmbeddedData('pageNo', pageNo);

} else {
  var pageNo = jq('#page-no-DIV')[0].innerHTML;
  pageNo --;
};
  });
    });
</script>

<div   class="QuestionBody   UnhiddenQuestion"   data-runtime-text="runtime.WontBeDis
playedText">
  <center><br><br><span id="page-position">page: <span id="page-number">1</span>
    <p></p>
    </span>
    <span>
<div id="top-div" style="display: none">
<span><center><br><br>time: <span id="time-display"></span>
    <p></p>
    <br><br>time2: <span id="time-display2"></span></center>
  <br><br>offTask: <span id="offTaskDisplay"></span>
  <br><br>onTask: <span id="onTaskDisplay"></span>
  <br><br>offTaskTotal: <span id="offTaskTotalDisplay"></span>
  <br><br>Window focus: <span id="window-focus"></span>
  <br><br>onTaskTotal: <span id="onTaskTotalDisplay"></span></span>
  </span>
</div>
```

Appendix B: The TaskMaster's Footer Code

```
<span><div><br></div>
<div id="bottom-div" style="display:none">
  Page:<div id="page-no-DIV"> ${e://Field/pageNo}</div>
<span><div><br></div>Time Array:<div id="timeArrayDIV"> ${e://Field/wtarray}</div>
<div><br></div>
On Task:<div id="onTaskDIV"> ${e://Field/onTask}</div><div><br></div>
Off Task <div id="offTaskDIV">${e://Field/offTask}</div><div><br></div>
Total On Task:<div id="onTaskTotalDIV"> ${e://Field/totalOnTask}</div><div><br></div>
Total Off Task:<div id="offTaskTotalDIV"> ${e://Field/totalOffTask}</div><div><br></div>
<div><br></div>Time Array:<div id="perPagePTArrayDIV"> ${e://Field/perPagePT}</div>
</span></div>
```

Action Editor

Mijke Rhemtulla served as action editor for this article.

Author Contributions

S. Permut came up with the idea for the TaskMaster and is responsible for the project's written code. M. Fisher wrote the accompanying app. S. Permut wrote the first draft of the manuscript; M. Fisher and D. M. Oppenheimer provided edits. All the authors approved the final submitted version of the manuscript. M. Fisher and D. M. Oppenheimer jointly generated the idea for a publicly available version of the TaskMaster.

Declaration of Conflicting Interests

The author(s) declared that there were no conflicts of interest with respect to the authorship or the publication of this article.

Open Practices

Open Data: not applicable
Open Materials: not applicable
Preregistration: not applicable

References

Borst, J. P., Taatgen, N. A., & van Rijn, H. (2010). The problem state: A cognitive bottleneck in multitasking. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *36*, 363–382.

Chandler, J., Mueller, P., & Paolacci, G. (2014). Nonnaïveté among Amazon Mechanical Turk workers: Consequences and solutions for behavioral researchers. *Behavior Research Methods*, *46*, 112–130.

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*, 120–131.

Curran, P. G. (2016). Methods for the detection of carelessly invalid responses in survey data. *Journal of Experimental Social Psychology*, *66*, 4–19.

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

Dennis, S. A., Goodson, B. M., & Pearson, C. (2018). MTurk workers' use of low-cost virtual private servers to circumvent screening methods: A research note. *SSRN*. doi:10.2139/ssrn.3233954

Diedenhofen, B., & Musch, J. (2017). PageFocus: Using paradata to detect and prevent cheating on online achievement tests. *Behavior Research Methods*, *49*, 1444–1459.

Hauser, D., Paolacci, G., & Chandler, J. J. (2018). Common concerns with MTurk as a participant pool: Evidence and solutions. *PsyArXiv*. Retrieved from https://psyarxiv.com/uq45c/

Jensen, C., & Thomsen, J. P. F. (2014). Self-reported cheating in Web surveys on political knowledge. *Quality & Quantity*, *48*, 3343–3354.

Luskin, R. C., & Bullock, J. G. (2011). "Don't know" means "don't know": DK responses and the public's level of political knowledge. *The Journal of Politics*, *73*, 547–557.

Oppenheimer, D. M., Meyvis, T., & Davidenko, N. (2009). Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology*, *45*, 867–872.

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

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

Thomas, K. A., & Clifford, S. (2017). Validity and Mechanical Turk: An assessment of exclusion methods and interactive experiments. *Computers in Human Behavior*, *77*, 184–197.