Survey Research Methods (2016)
Vol. 10, No. 1, pp. 49-64
doi:10.18148/srm/2016.v10i1.6273

ISSN 1864-3361
http://www.surveymethods.org

Helping Respondents Provide Good Answers in Web Surveys

Mick P. Couper
University of Michigan
Ann Arbor, USA

Chan Zhang
Fudan University
Shanghai, China

This paper reports on a series of experiments to explore ways to use the technology of Web surveys to help respondents provide well-formed answers to questions that may be difficult to answer. Specifically, we focus on the use of drop-down or select lists and JavaScript lookup tables as alternatives to open text fields for the collection of information on prescription drugs. The first two experiments were conducted among members of opt-in panels in the U.S. The third experiment was conducted in the 2013 Health and Retirement Study Internet Survey. Respondents in each of the studies were randomly assigned to one of three input methods: text field, drop box, or JavaScript lookup, and asked to provide the names of prescription drugs they were taking. We compare both the quality of answers obtained using the three methods, and the effort (time) taken to provide such answers. We examine differences in performance on the three input format types by key respondent demographics and Internet experience. We discuss some of the technical challenges of implementing complex question types and offer some recommendations for the use of such tools in Web surveys.

Keywords: Web survey; instrument design

1 Introduction

Since the advent of Web or Internet surveys almost 20 years ago, research has focused on exploiting the medium to enhance measurement relative to alternative modes of data collection (especially paper). A large literature has emerged on the optimal design of Web survey instruments for a variety of different question types (see, e.g., Callegaro, Lozar Manfred, and Vehohar, 2015; Couper, 2008; www.websm.org). In particular, research has focused on the interactive features of Web surveys to assist respondents in answering complex questions and to ensure that usable data are obtained (see, e.g., Tourangeau, Conrad, & Couper, 2013, chap. 6). One such line of research has focused on questions that would typically be asked as open-ended questions in interviewer-administered surveys and paper self-administered surveys, with coding done either on the spot by interviewers using some tool in computer-assisted interviewing or after the fact by trained coders. Examples of these types of questions include industry and occupation, product classification systems for pricing or establishment surveys, questions on field of study or educational qualifications, county of origin or residence, automobile make and model, and the like. There are a wide range of these open-ended question types (see, e.g., Couper, Kennedy, Conrad, & Tourangeau, 2011), and a variety of alternative tools for capturing this information in Web surveys. Our focus is on types of questions where there is a large number of response options, where an exhaustive (or near-complete) list of response options can be developed, and where the list can be organized in a meaningful way (e.g., alphabetically) to facilitate a search or look-up of the desired response. This is an area that has received very little research attention, yet is one where use of the appropriate tools may help both respondents and researchers.

Contact information: Mick P. Couper, P.O. Box 1248, Ann Arbor, MI 48109, U.S.A. (email: mcouper@umich.edu)

2 Background and Research Questions

In an early unpublished exploration of what they called dynamic forms, Funke and Reips (2007) tested the use of an autocomplete tool using JavaScript for collecting state of residence (Bundesländer) in Germany. They tested auto-complete and auto-suggest (like our lookup tool) versions programmed in JavaScript against a standard text box, a drop box, and a series of radio buttons. They lost 7.4% of the sample who did not have JavaScript enabled. They found significant differences in response times between conditions, with the drop down list and radio buttons taking the least time (there are only 16 German states). There were no significant time differences between the three text versions (text box, auto-complete and auto-suggest). There were no differences in item missing data or dropout by condition. In terms of providing exact (codable) answers, 83.7% of respondents provided exact answers in the auto-complete version, compared with 81.6% in the text box version and 80.6% in the auto-suggest version. They concluded that there was no benefit to the use of dynamic fields. One explanation may be the

shortness of the list, and another may be that respondents are familiar with providing this information.

More recently, Herzing (2015) experimentally tested three different versions of an educational attainment question in Germany. She compared 1) a list with 28 response categories, 2) a dynamic text field (lookup list) with 400 response categories, and 3) a search tree with 35 response categories. She found shorter median response times for the dynamic text field (32 secs) and search tree (35 secs) than the long list (43 secs). She found no significant differences in response quality (defined as consistency with responses in a prior wave of the panel) between versions.

Couper et al. (2011) compared drop boxes to one or three text fields for ascertaining data (month, day and year) of birth. They found that the drop box version had a significantly higher proportion of well-formed answers but also took significantly longer than the other versions. Several other studies have compared drop boxes to radio buttons for questions with relatively small numbers of responses. The findings are generally mixed. For example, Heerwegh and Loosveldt (2002) and Healey (2007) found longer completion times for drop boxes, while Couper, Tourangeau, Conrad, and Crawford (2004) did not find significant time differences. But Couper et al. (2004) and Healey (2007) both report differences in response order effects between the two formats. Healey attributed the finding to difficulty using a scroll wheel with the drop box (see also Gendall & Healey, 2008; Grondin & Sun, 2008). A review of these findings led Couper (2008, p. 66) to conclude that "drop boxes do have a use in Web surveys, but for specific types of questions." Recent research on drop boxes in mobile Web surveys has similarly produced mixed results, again for questions with limited response options (see, e.g., Couper, Antoun, & Mavletova, 2015; Peterson, LaFrance, Griffin, & Li, 2015).

Tijdens (2014, 2015) describes a three-step search tree to identify occupations from a database with more than 1,700 occupational titles (see www.wageindicator.org). She notes (Tijdens, 2014, p. 70) that "the closed response format is preferred over an OEQ [open-ended question] with office coding" in part because the OEQ "would have required a continuous and costly coding effort." However, this conclusion was not based on an experimental comparison to other approaches.

Despite the paucity of research, drop boxes and lookup lists are widely used in online surveys. Examples include state or country of residence or birth, automobile make and model, and so on. Other examples of complex hierarchies include the North American Product Classification System (NAPCS, see http://www.census.gov/eos/www/napcs/), educational qualifications in Europe (see Herzing & Schneider, 2014), medical diagnoses and medications (such as is captured in the National Ambulatory Medical Care Surveys; see http://www.cdc.gov/nchs/ahcd.htm), and the International Classification of Diseases (see http://www.who.int/classifications/icd/en/) which contains over 14,000 different diseases. As suggested by the above research, long and complex lists such as these preclude the possibility of presenting the full list using a series of radio buttons. Further, as the items on the list increase in length and complexity, the value of alternatives to text fields that require respondents to type the response may grow. With the rapid rise in respondents using mobile devices, and the different ways that mobile browsers handle drop boxes (see Buskirk, Michaud, & Saunders, 2014; Couper et al., 2015), the question of the best tool for this type of question is becoming more important.

As noted above, our research focuses on these types of complex coding tasks. In the more straightforward cases, respondents are likely to know how to find the relevant information on a list, and using an alphabetically-organized drop box or select list is likely to be more efficient (for both respondents and analysts) than typing the response in a text box. But lists can get much more complex than this, making more sophisticated tools necessary or desirable. Our specific focus is on prescription medications. There are over 6,000 medications and their generic equivalents on the market in the U.S. They often go by multiple names (e.g., pharmacological name, generic name, brand name). The names are difficult to spell and sometimes are made up of compound words (e.g., 8-Hour Acetaminophen E.R.). Further, the level of detail needed is sometimes unclear (e.g., there are 46 variations of Acetaminophen in the database we use).

Given the complexity of this task, we evaluate three different ways for respondents to provide information about prescription drugs:

1. A text box (TB) in which respondents type their responses for later lookup or coding.

2. A drop box (DB) or select list containing the full list of drug names sorted alphabetically. Respondents can either scroll through the list or type the first letter of a drug to jump to that part of the list.

3. A combo box[1] or lookup list implemented using JavaScript (JS). As the respondent types the name of the drug, the list of alternative narrows until the respondent can pick the relevant one or continue typing.

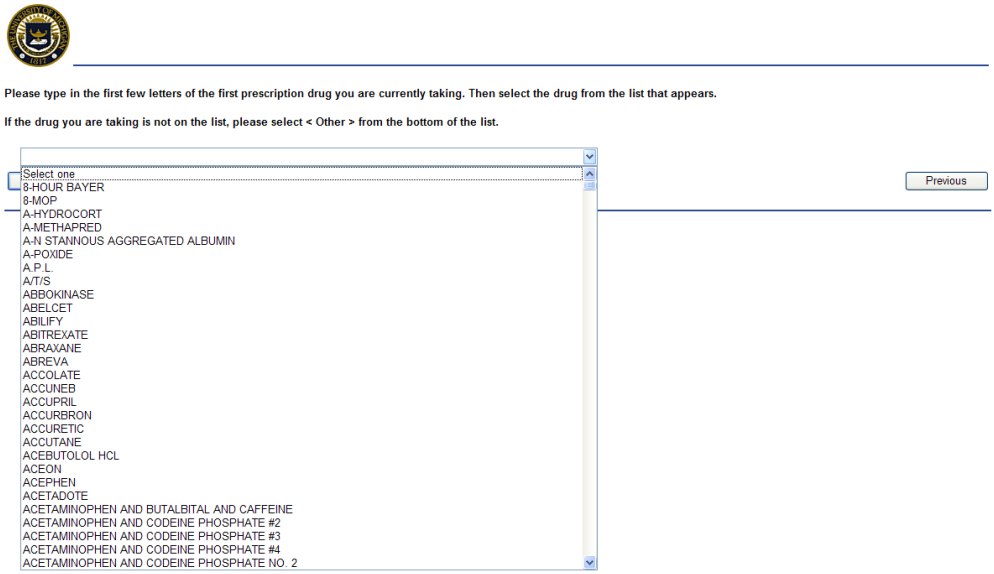
These three alternatives are illustrated in Figure 1. These can be implemented using standard HTML tools (text box or drop box) or using client-side scripting that is increasingly standard in modern browsers (e.g., JavaScript).

We expect that the text box option would be time-consuming and difficult for respondents as there is no guidance on what to enter and no feedback on the correctness of the response. This option also increases the likelihood of respondents mistyping or misspelling drug names, or providing

[1] So-called because it involves a combination of a text box and a drop box. Users can either type a response, or select from the list, or use a combination of approaches.

(a) Text box condition

(b) Drob box condition



(c) JavaScript lookup condition

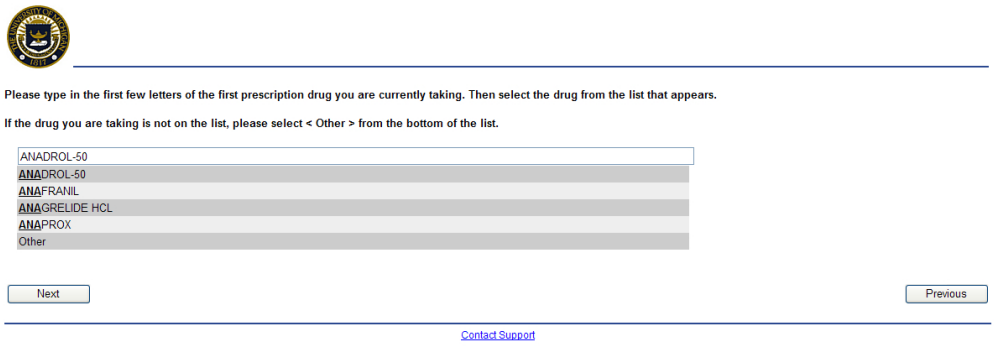


Figure 1. Experimental conditions in study 1

insufficient information. From a technical perspective, however, this alternative is the easiest to design, as it does not require JavaScript or loading the lengthy medication list into the Web page. The text box approach is also likely to require the most effort in terms of coding, and is more likely to yield uncodable responses given the lack of feedback to respondents. The drop box option provides the full list of medications, requiring the respondent to pick one. This obviates the need to type a response. This approach assumes that respondents are able to find a match in the list provided. However, given the length of the list, the Web page may take longer to download, and it may take respondents more time to find the desired response. This may increase frustration, leading to missing data and breakoffs. Finally, the JavaScript combo box potentially gives respondents several options for entering or selecting a response. By providing feedback (through

narrowing the list), this approach may make the task easier, and reduce time and effort in responding relative to the drop box, while still yielding higher levels of codability relative to the text box. We thus expect that the dynamic JavaScript approach may offer the best combination in terms of facilitating the respondent's answering task and providing codable data.

Below we describe the design of the three experiments in turn and describe the results. The first two experiments were conducted among members of opt-in or access panels. We conduct additional analyses on the third study, conducted in the 2013 Health and Retirement Study (HRS) Internet Survey. All three studies tested the same three approaches to eliciting this information, as described above. Variations in the implementation of the design are described in further detail below.

For each of the three studies, we are interested in several outcomes. First, we examine whether the experimental conditions resulted in different rates of dropout or breakoff on the affected items. Second, we examine response times as an indicator of the effort required to respond. There are often outliers in the time measures, some of which may reflect respondents taking time to look up the drug name (e.g., in the medicine cabinet or on a prescription), but others may simply reflect respondents taking a short break on that question. Given that we cannot distinguish reasons for such outliers, we uniformly trim the response times at 300 seconds (5 minutes) for all three studies. This is based on extensive analysis of outliers and examination of alternative cut points for trimming. We also examine logged response times and report on these where appropriate. Third, we are interested in item missing data rates across the treatment groups. Finally, we examine the codability of the responses provided.

We define codability as follows. In the text box condition, if an entry could be matched directly to an entry in the database, we considered this (machine) matchable. For the remaining cases, if a human coder was able to find a full or partial matching entry in the database, this was considered codable. Reasons for a non-matchable case being codable include misspellings, use of abbreviations, and partial information (e.g., the first word of a compound description, if it matches a unique entry in the database). Uncodable responses in the text box condition include explicit refusals or don't know responses (e.g., "Decline," "don't remember . . . for thyroid"), generic responses (e.g., "blood pressure," "cholusteral" [sic]) and other uncodable answers (e.g., "Chlorestyrene" "Alyssa" [Alustra?], "aldat"). Uncodable responses in the drop box and JavaScript conditions could occur when a respondent selected the "other drug not listed" option. Further, in the JavaScript (combo box) condition a respondent could type an answer that did not match to anything on the list. In the third experiment, an uncodable response could be produced following selection of the "other, specify" option or because of redirection to the text box condition (as described further below).

3 Study 1

3.1 Study 1 Design

Our first experiment was conducted as part of a series of methodological experiments administered to members of two different opt-in panels in the U.S. in September 2008 (see Tourangeau, Couper, Conrad, & Baker, 2008). The panels were Survey Sampling International's (SSI) Survey Spot Panel and Authentic Response's panel. Between the two panel sources, a total of 69,200 panelists we invited to the survey. A total of 3,213 panelists started the survey and 2,410 completed the survey. This represents a participation rate of 3.5%. Design of the survey instrument and data collection was managed by Market Strategies International (MSI), using SPSS mrInterview software (now owned by IBM).

Participants were randomized to one of three conditions described above (text box, drop box, JavaScript lookup). Respondents were first asked how many prescription drugs they were taking. A total of 2,964 respondents answered this question, of which 2,013 reported using one or more drugs. These 2,013 respondents were then asked for the names of up to three prescription drugs.

The database of drugs for the drop box and JavaScript versions contained a total of 4,768 prescription medications. These came from the Lexi-Data database (see http://www.lexi.com/). The Lexi-Data database provides drug information that includes drug names (brand name, generic name, and common abbreviations), therapeutic categories, and standard coding such including ICD-9.

Respondents assigned to the text box (TB) condition were instructed to "Please enter the name of the [first/second/third] prescription drug you are currently taking." Those in the drop box (DB) and JavaScript combo box (JS) conditions were instructed to "Please type in the first few letters of the [first/second/third] prescription drug you are currently taking. Then select the drug from the list that appears." They were further instructed that "If the drug you are taking is not on the list, please select < Other > from the bottom of the list." We did not ask a follow-up question to elicit the drug name from those choosing the "other" response.

At the time of the study, standard HTML drop boxes or select menus used single-letter lookup (see Couper, 2008, pp. 59–60). Typing a single letter would take the user to the first word starting with that letter in the list. Typing a second letter would take one to the first word starting with the second letter. The JS version was designed to use progressive or incremental lookup, in which typing several letters would narrow the search list. For example, in an HTML drop box of U.S. states, typing "M-I-C" would take one to Maine, then Idaho, then California. Typing "M-I-C" in the JavaScript version would first display all of the "M" states, then the "MI"

states, then Michigan (the only "MIC" state). Of course, the user could click on the menu and scroll to find the desired response (drop boxes are designed for scrolling rather than typing).

3.2 Study 1 Results

First, we examined breakoff rates on the drug questions across the three conditions. We found significant differences ($X^2 = 61.1$; $d.f. = 2$; $p < 0.001$) with 3.3% breaking off in the TB condition, 15.7% in the DB condition and 14.2% in the JS condition. We similarly found higher item missing data rates in the two alternative versions than in the text box version, as shown in Table 2. Across all three drugs, the missing data rate is about twice as high for the DB and JS conditions than for the TB condition.

Second, we see from Table 1 that it took significantly longer to provide a response in the DB and JS conditions than in the TB condition, among those who provided a response. For the first drug (for example), the DB condition took more than four times as long as the TB condition, and 1.5 times as long as the JS condition. While the time taken to answer the TB condition was relatively stable across the three drugs, a clear learning curve could be seen in the other two conditions. To check for selection effects in the time gradient, we restricted the data to those respondents who had answered all three drug items, and found a similar pattern in response times across the three drugs and experimental conditions. Despite the reductions in response times for the DB and JS conditions, the significant differences in response times between experimental conditions remain for the third drug. We fit a multilevel model using SAS PROC MIXED (not shown) to test the learning curve among those who answered all three drug questions, and find no significant change in response time in the text box condition across the three drugs, but significant negative slopes (showing reduced time across the three drugs) for the other two conditions, as seen in Table 1.

Given that there were fewer breakoffs and less missing data in the standard TB condition, and this condition took the least amount of time to answer, was there a penalty in terms of the codability of the responses? We see in Table 2 that the TB condition had the lowest machine-matchable rate, with only 63% of responses amenable to automated coding for the first drug. Further, we see that the matchable rate was significant higher for the DB than the JS condition for all three drugs. This means that more respondents in the JS condition selected "other" or typed in a response that did not match a drug in the database. If we include the codable cases in the comparison, we can see from Table 2 that 89.2% of responses to the first drug in the TB condition were matchable or codable, compared with 86.4% in the DB condition and 75.0% in the JS condition.

The results from the first experiment suggest that having respondents type in a response may require greater post-

Table 1
Response Times (in Seconds) by Drug and Experimental Condition, Study 1

	TB	DB	JS	F (d.f.(1)=2)
Drug 1 (1, 887)				
Mean	22.4[a]	84.7[b]	65.8[c]	196.1***
Std. Err.	1.1	2.8	2.6	
Median	14.0	61.0	41.0	
Drug 2 (1, 431)				
Mean	23.6[a]	71.9[b]	56.3[c]	92.5***
Std. Err.	1.6	3.3	2.7	
Median	13.0	44.0	37.0	
Drug 3 (1, 060)				
Mean	26.1[a]	69.0[b]	53.6[c]	55.3***
Std. Err.	1.9	3.8	3.1	
Median	15.0	41.0	37.0	

Note: Means with different superscripts are significantly different, $p < .01$, Bonferroni adjustment for multiple comparisons.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table 2
Missing Data Rates and Codability of Responses by Drug and Experimental Condition, Study 1

	TB	DB	JS	X^2	d.f.
Drug 1 ($n = 2,013$)					
Matchable (in %)	62.6	86.4	71.9		
Codable	26.6	0.0	3.1		
Uncodable	6.8	5.7	14.9	348***	4
Missing	4.0	8.0	10.1	18***	1
n	647	690	676		
Drug 2 ($n = 1,627$)					
Matchable (in %)	55.9	79.1	69.0		
Codable	25.2	0.0	3.1		
Uncodable	10.7	4.9	11.4	250***	4
Missing	8.2	16.0	16.5	19***	1
n	512	570	545		
Drug 3 ($n = 1,253$)					
Matchable (in %)	55.0	74.5	63.7		
Codable	25.6	0.0	3.6		
Uncodable	10.8	4.8	11.5	181***	4
Missing	8.5	20.6	21.2	30***	1
n	398	436	419		

The first significance test in the last column tests the distribution of non-missing codes across conditions. The second test contrasts whether item missing data rates are similar across conditions.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

survey effort (manual coding) but yields more usable (matchable or codable) responses and poses less burden on respondents (takes less time). Our attempt to provide alternatives to assist respondents in answering this set of questions appears to have failed.

Two possible explanations for the poor performance of the alternative approaches can be offered. First, the database of names was all in caps. Capitalization is known to make it harder to read (see Couper, 2008, pp. 154–155), which may explain the particularly poor time performance for the DB version. Second, the list of drugs contained a lot of detail – for example different dosages or dosage forms (caplets, liquid, tablet, etc.), or several variants of the same drug (see Figure 1b), Acetaminophen and Codeine Phosphate #2, #3, #3, No. 2, etc.). The length of the list may also slow down transmission of the information from the server to the client (browser) to populate the lists. Given these potential concerns, we ran a second experiment with some minor changes as described below.

4 Study 2

Study 2 is essentially a replication of Study 1, in a similar non-probability panel, but with improved designs of the alternative input forms.

4.1 Study 2 Design

The second experiment was also conducted using panels from SSI and Authentic Response. However, the SSI sample came from a river sampling approach, in which people receive a general invitation after which they are directed to specific surveys based on responses to profile questions. Thus, the number of invitations is unknown, but 1,655 SSI participants started the survey and 1,223 completed it. For Authentic Response, a total of 14,132 invitations generated 1,619 starts and 1,204 completes, for a participation rate of 8.5%. In total, 3,274 participants started the survey and 2,427 completed it. The survey was again programmed and deployed by MSI. Data collection took place in December 2011 (see Tourangeau, Couper, Conrad, & Baker, 2011).

As with Study 1, participants in the text box condition were instructed to: "Please enter the name of the first *prescription medication* you are currently taking." Those in the drop box condition were instructed as follows: "Please use the drop-down list below to select the [first/second/third] *prescription medication* you are currently taking." Those in the JavaScript condition were instructed: "Please type in the first few letters of the [first/second/third] *prescription medication* you are currently taking. Then select the drug from the list that appears." Both the DB and JS conditions included the following additional instruction: "If the drug you are taking is not on the list, please select '*Other drug not listed above*' from the bottom of the list" (see Figure 2).

A total of 2,458 respondents answered the initial question on how many prescription drugs they were taking, of which 1,429 reported using one or more prescription drugs. These 1,429 respondents were then asked the names of up to three prescription drugs, with randomization to one of the experimental conditions.

We used an updated list of 6,329 drugs and their generic equivalents, again obtained from Lexi-Comp. We merged these into a single list, and trimmed duplicates and excess detail, resulting in a list containing 5,007 unique drug names. We also converted all drug names to initial caps (as shown in Figure 2).

At the time of the second study, browsers were adapting the way they presented select lists (drop boxes). The then-extant version of Internet Explorer still used single-letter lookup, while Firefox and Chrome both permitted progressive lookup, depending on how fast one typed (any pause in typing would revert back to a single-letter lookup). We had no control over how each browser handled the standard HTML drop box element.

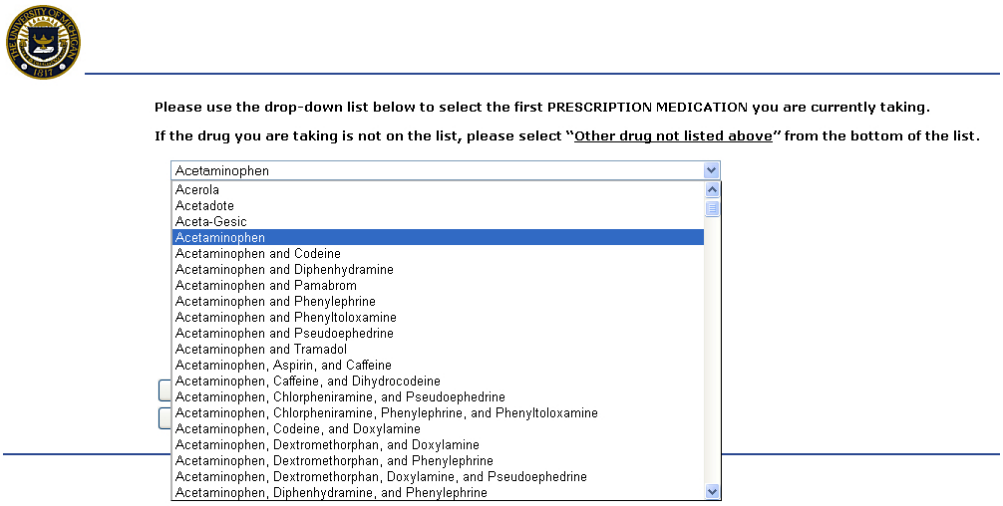
4.2 Study 2 Results

Breakoffs on the drug series were substantially lower (0.9% for TB, 1.1% for DB, and 0.9% for JS) than in Study 1, and did not differ significantly across conditions ($X^2 = 0.11, d.f. = 2,$ n.s.). We also see from Table 4 that item missing data rates were substantially lower than the first study, with none of the rates being significantly different across conditions (see second significance test in last column). Given that the sample was very similar to that used in the first study, we attribute this to the technical improvements described above. An alternative explanation is that increased user experience between 2008 and 2011 accounts for this improvement. We are unable to test this, but note that we have found no such improvements in respondent performance across a variety of other experiments in this time frame.

Table 3 shows the mean response times by condition for each of the three drug questions. First, we see that all three conditions take less time on average than in Study 1, again pointing to the benefits of the technical and design improvements. However, both the DB and JS conditions still took significantly longer to enter or select the first drug. For the second and third drugs, the JS condition no longer took significantly longer than the TB condition, but the DB condition still took significantly (and substantially) longer than the TB condition.

As in the first study, we see reductions in response time for the DB and JS conditions from the first to the second drug. Restricting the analysis to those who had provided an answer to all three drug items to check for selection effects, we see a similar pattern in response times. Multilevel models of response times show significant improvements across the

(a) Look up condition



(b) JavaScript condition

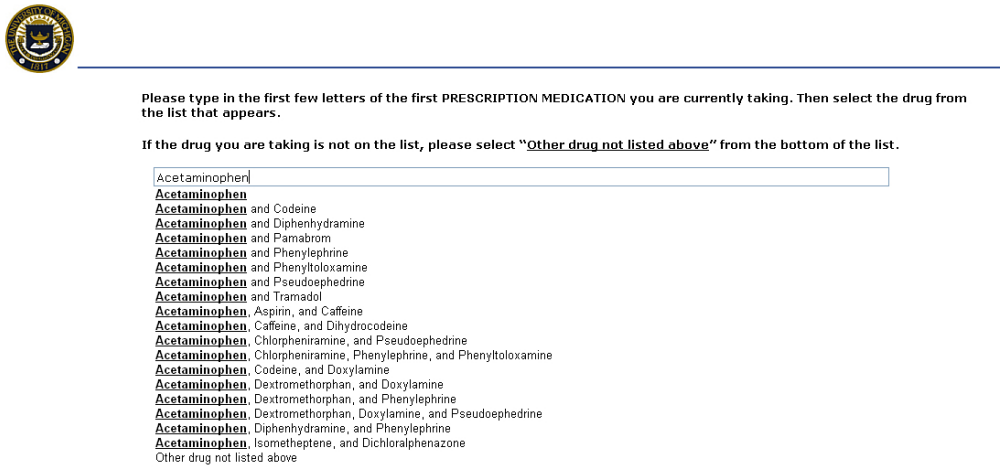


Figure 2. Experimental conditions in study 2

three drugs for all three experimental conditions.

Looking at the rate of codable and matchable responses in Table 4, we see that the DB and JS conditions have higher match rates than the TB condition for all three drugs reported, with the DB condition again providing the highest rate of machine-matchable responses, as expected. If we combine the matchable and codable categories, we can see from Table 4 that 81.9% of those in the TB condition provided a usable response to the first drug, compared with 90.7% for the DB condition and 77.4% for the JS condition.

As with Study 1, we found in Study 2 that the drop box condition yields more machine-matchable responses than the text box standard, but this comes at a cost of significantly longer completion times (by a factor of 3). We also see that the improved design resulted in lower missing data rates for both alternative conditions, compared to Study 1. However, the improved design of the Study 2 alternative input formats resulted in only marginal increases in match rates. Again, it appears that the two alternative approaches do not yield sufficient gains in quality to offset the longer response times. One possible reason for the high uncodable rates in the DB or JS conditions is that respondents were not given an opportunity to enter the name of a drug they could not find on the list – these "other, not specified" drugs were uncodable. We rectified this issue in the third study, described below.

5 Study 3

Study 3 tests the same three input formats in a probability sample of older Americans.

Table 3
Response Times (in Seconds) by Drug and Experimental Condition, Study 2

	TB	DB	JS	F (d.f.(1)=2)
Drug 1 ($n = 1,401$)				
Mean	18.9[a]	69.4[b]	32.9[c]	199.9***
Std. Err.	1.1	2.7	1.4	
Median	12.0	54.0	24.0	
Drug 2 ($n = 1,020$)				
Mean	15.4[a]	52.9[b]	21.3[c]	130.8***
Std. Err.	1.0	2.7	1.1	
Median	10.0	36.0	15.0	
Drug 3 ($n = 708$)				
Mean	16.6[a]	49.2[b]	23.0[c]	58.1***
Std. Err.	1.6	3.4	1.4	
Median	10.0	30.0	15.0	

Means with different superscripts are significantly different, $p < .01$, Bonferroni adjustment for multiple comparisons.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table 4
Missing Data Rates and Codability of Responses by Drug and Experimental Condition, Study 2

	TB	DB	JS	X^2	d.f.
Drug 1 ($n = 1,429$)					
Matchable (in %)	61.1	90.7	75.4		
Codable	20.8	0.0	2.0		
Uncodable	15.4	6.6	18.9	220***	4
Missing	2.7	2.7	3.7	1	2
n	486	484	459		
Drug 2 ($n = 1,040$)					
Matchable (in %)	54.4	84.9	72.5		
Codable	22.4	0.0	1.8		
Uncodable	18.2	11.3	20.4	163***	4
Missing	5.0	3.8	5.4	1	2
n	362	344	334		
Drug 3 ($n = 734$)					
Matchable (in %)	47.6	80.3	68.9		
Codable	28.8	0.0	3.7		
Uncodable	19.1	11.5	22.1	130***	4
Missing	6.5	8.2	5.3	2	2
n	246	244	244		

The first significance test in the last column tests the distribution of non-missing codes across conditions. The second test contrasts whether item missing data rates are similar across conditions.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

5.1 Study 3 Design

Our third experiment testing the same three input formats was conducted as part of the 2013 HRS Internet Survey (see HRS-IS, 2013). The sample differed from the first two studies in that participants are panelists in the Health and Retirement Study with Internet access. HRS is an ongoing panel study of American adults age 50 and older and their spouses. Core data collection by interviewer-administered survey is conducted every two years in the even-numbered years. Special studies (such as the Internet Survey) are conducted in the "off" years (odd-numbered years). In 2013 we embedded an experiment testing the same three input formats, but with a number of other modifications (for coded data from the experiment, see Couper, 2016).

First, in the first two studies we did not follow up on those who selected "other" to the DB or JS versions, so were unable to code whether we would have obtained codable data from these responses. For this study, we added an "other, specify" text box to capture these responses. Second, we increased the number of prescription drugs reported from 3 to 5, given the age of the population. Finally, we asked respondents if they were taking medications for ten common ailments (e.g., high blood pressure, high cholesterol, sleep problems) to see if respondents were giving plausible responses. The coding of the prescription drug responses to therapeutic classes to permit matching to common medical conditions is still ongoing, so we do not report on this here.

An updated database of prescription drugs was again obtained from Lexi-Comp. After merging the generic and brand name lists and cleaning duplicates, we ended up with a consolidated list of 11,354 unique medication names[2]. These were included in the DB and JS versions of the instrument, and used for matching and coding the text responses in the TB condition and "other, specify" responses in the other two conditions.

The 2013 HRS-IS instrument was developed and deployed by the Survey Research Center, using Datstat's Illume survey software. The sample was drawn from respondents who completed the 2012 core HRS interview and reported having Internet access. A total of 7,744 HRS panel members were invited by mail to complete the Internet survey, and offered a $25 incentive for doing so. Data collection took place from April to August, 2013. A total of 5,809 respondents completed all or part of the 2013 Internet Survey, for a simple response rate of 75%. Of these, 5,682 provided an answer to the question on the number of prescription drugs, and 4,632 (or 81.5% of those who answered this question) reported taking one or more drugs.

[2] The increase in medications over the first two studies is due in part to the inclusion of over-the-counter medications (as these can also be prescribed) and the inclusion of pediatric drugs.

5.2 Study 3 Results

Breakoffs were very low in the HRS-IS compared to the two opt-in samples. Overall, only 3.4% of those who started the survey did not complete it, and only 13 broke off during the experimental items (2 in TB, 9 in DB, and 2 in JS).

Next, we examined the device and browser used by respondents to start the survey. In terms of device, 4.4% used a tablet and 0.8% used a smartphone, while the balance used a PC (desktop or laptop). The most common browser used was Internet Explorer (MSIE), with 60.4% of respondents. However, about a third of these respondents (18.5% of all respondents) used MSIE version 8 or lower, while 19.4% used MSIE 9, and 22.5% used MSIE 10+. The remaining respondents used Chrome (13.7%), Firefox (13.3%), Safari (7.3%) or some other browser or a mobile browser (5.3%). One of the questions we explore is whether the different input methods worked equally well across all browsers.

About halfway through data collection, it was discovered that some respondents (especially those with older browsers) were having difficulty using the DB and JS versions. An "escape" question was added after the first drug entered, for those who reported taking two or more prescription drugs in these two conditions. Respondents were asked, "Did you have any technical difficulties in providing the name of the prescription drug?" If yes, they were asked to provide details on the problems and were then routed to the text box version for the remaining drugs. In investigating this issue, it was determined that the DB and JS conditions were not working reliably for those using older versions of MSIE (7.0 or earlier). A flag was subsequently incorporated in the instrument to route such cases to the TB version. These two modifications resulted in a total of 166 cases in the DB condition (57 for all 5 drugs and 109 for drugs 2-5) and 116 cases in the JS condition (55 for all 5 drugs and 61 for drugs 2-5) who were routed to the TB version. This represents 8.5% and 6.1% respectively of all cases in these conditions. For the rest of the analyses we include those respondents who were routed to the TB version with the originally-assigned experimental group (i.e., using an intent-to-treat analysis). Excluding them from the analysis does not change the main findings discussed below.

Item missing data rates are shown in the last row for each drug in Table 6. The differences in rates are significant ($p < 0.05$) across all drugs (see second significance test in last column). In this study, however, the TB and JS versions showed similar rates of missing data, while the DB version was higher across all drugs. We also see a steady increase in missing data across the number of drugs reported, suggesting a fatigue effect or the fact that those with more drugs to report may have more difficulty reporting.

The next question is whether response time differed between experimental conditions. As noted earlier, we truncated all item-level time measures at 300 seconds (about the 85th percentile). The resulting variable is still skewed, so we present in Table 5 both the means and medians of response times for these items among the non-missing responses. We also analyzed log-transformed times (not shown in Table 5). These lead to similar conclusions about differences between experimental conditions.

Table 5
Response Times (in Seconds) by Drug and Experimental Condition, Study 3

	TB	DB	JS	F (d.f.(1)=2)
Drug 1 ($n = 3,752$)				
Mean	47.4^{a}	121.3^{b}	70.7^{c}	479.3^{***}
Std. Err.	1.3	2.2	1.5	
Median	26.5	105.0	52.0	
Drug 2 ($n = 3,126$)				
Mean	32.7^{a}	90.3^{b}	45.9^{c}	401.7^{***}
Std. Err.	1.0	2.1	1.3	
Median	21.0	70.0	31.0	
Drug 3 ($n = 2,379$)				
Mean	32.1^{a}	82.5^{b}	44.6^{c}	239.5^{***}
Std. Err.	1.2	2.4	1.5	
Median	21.0	62.0	31.0	
Drug 4 ($n = 1,700$)				
Mean	31.0^{a}	78.6^{b}	38.5^{c}	194.2^{***}
Std. Err.	1.2	2.7	1.5	
Median	22.0	59.0	26.0	
Drug 5 ($n = 1,195$)				
Mean	31.2^{a}	83.6^{b}	42.8^{c}	145.4^{***}
Std. Err.	1.2	3.4	2.0	
Median	24.0	62.0	29.0	

Means with different superscripts are significantly different, $p < .01$, Bonferroni adjustment for multiple comparisons.
$^{*}\ p < 0.05$ $^{**}\ p < 0.01$ $^{***}\ p < 0.001$

Two observations can be made about the results in Table 5. First, the DB condition took consistently longer than the JS condition, which in turn was consistently slower than the TB condition, across all five drugs measured. Second, across all three input types, there appears to be evidence of a learning curve, with the second drug taking less time to complete than the first. However, this appears to level off after the second drug, which may suggest a floor effect.

We examined the effect of experimental treatment and drug number in a multilevel model (with drugs nested within respondents) using the SAS PROC MIXED procedure (see Appendix A). We found a significant main effect for experimental condition ($F = 609.65$; $d.f.(1) = 2$; $d.f.(2) = 4,456$; $p < 0.001$), a significant linear effect of drug number ($F = 937.71$; $d.f.(1) = 1$; $d.f.(2) = 9,453$; $p < 0.001$), and a significant interaction ($F = 68.39$; $d.f.(1) = 2$; $d.f.(2) = 9,453$;

$p < 0.001$). This supports the learning curve hypothesis, but suggests a steeper curve for the DB version than the other two versions. An alternative explanation for the time differences is that those who take more drugs are likely to be older and are disproportionately present in the later drug items. We return to this issue later.

Given that the two alternative input formats (DB and JS) had higher item missing data rates and took longer than the TB version, were there any advantages of these approaches in terms of the quality of the responses obtained? Here – as in the earlier studies – we define quality in terms of the codability of the response. To determine codability, we used the Lexi-Comp database as the source, using the same procedure to classify cases as matchable or codable as described earlier.

Among those in the DB condition, 51 respondents (3.3% of eligible respondents) selected the "other drug not listed above" option for the first drug. Of these, 21 (or 41%) provided a matchable response (i.e., an answer that appeared in the database), while a further 23 (56%) provided a codable response. In the JS condition, 146 respondents (9.6% of those eligible) selected this response for the first drug, of which 79 (or 54%) provided a matchable response and 51 (or 35%) a codable response. This points to the value of including an "other, specify" option for a complex task such as this. In subsequent analyses we keep these "other, specify" responses and those routed to the TB condition after reporting difficulty entering the first drug in their respective groups as experimentally assigned (i.e., using an intent-to-treat analysis).

The results are presented in Table 6. We include the missing data rates to show the overall loss of information (as discussed earlier). Significance tests are presented both for 1) differences in the proportion of eligible cases with missing data, and 2) differences in the codability of cases after excluding missing data.

The two alternative approaches (DB and JS) yielded higher rates of machine-matchable responses than the text box approach. This is not surprising, given that the information provided to the respondent increases the likelihood of an exact match (conditional on answering the question). What may be of more interest is the amount of usable data provided by respondents under each condition. To examine this, we combine the first two rows (matchable and codable) in Table 6, and contrast them with the last two rows (uncodable and missing). The resulting rates of usable responses are summarized in Table 7.

Across all five drugs, the JS version yielded significantly more usable responses than either the TB version or the DB version. However, these differences are relatively minor. Further, when testing these differences in a multilevel model (with drugs nested within respondents, using SAS PROC GLIMMIX), the effect of the experimental treatments was no longer statistically significant ($F = 1.87$; $d.f.(1) = 2$; $d.f.(2) = 10,564$; $p = .15$). Further, the slightly higher rate of usable responses for the JS version comes at a cost of somewhat longer completion times for respondents, but at lower coding costs for the survey organization.

Table 7
Usable Data Rates by Drug and Experimental Condition, Study 3

	TB (in %)	DB (in %)	JS (in %)	X^2 (d.f.=2)
Drug 1	87.8	88.0	90.5	7.1^{*}
Drug 2	86.2	85.2	88.9	8.5^{*}
Drug 3	84.2	80.0	88.1	24.2^{***}
Drug 4	83.7	76.4	86.5	27.1^{***}
Drug 5	83.6	75.2	83.5	15.5^{***}

$^{*}\ p < 0.05$ $^{**}\ p < 0.01$ $^{***}\ p < 0.001$

We noted earlier that answers can be selected in the DB version either by typing the first few letters, or by scrolling. To what extent do respondents attempt to type all or part of the response? Analysis of the client-side paradata for the first drug reveals that 92.9% of those in the DB condition did not type anything, i.e., they only used scrolling to select a response. This remains above 90% for the remaining drugs. Given that scrolling is the preferred action in the drop box (unlike the other two conditions where some amount of typing is required), does this lead to different distributiona of drugs selected? Specifically, do we see primacy effects – the selection of drugs earlier in the (alphabetized) order – in the drop box condition, to avoid lengthy scrolling? We examined this by converting the first letter of each drug to a numbered score (A or numeric=1, B=2, etc.). We find that the distribution of drugs reported in the drop box condition are significantly skewed toward the earlier end of the alphabet: the mean for the DB condition is 9.8, while that of the TB condition is 11.0 and the JS condition is 11.1 ($F = 50.26$; $d.f.(1) = 2$; $d.f.(2) = 13,047$; $p < 0.001$). More specifically, the effect is that of DB respondents selecting drugs starting with a number (e.g., 5 Benzagel) or the letter A at a much higher rate: 33.9% of DB respondents did so for the first drug, compared to 11.6% for the TB condition and 11.4% for the JS condition ($X^2 = 299.1$; $d.f. = 2$: $p < 0.001$).

This effect can be further seen in the distribution of major drug classes selected. Respondents in the DB condition reported taking antihypertensive or angiotensin drugs (for high blood pressure or HBP) at a lower rate (41.9%) than those in the TB (49.3%) or JS conditions (45.4%; $X^2 = 15.46$; $d.f. = 2$; $p < 0.001$). Similarly, DB respondents reported taking antilipemic drugs (for cholesterol) at a lower rate (34.9%) than those in the TB (45.6%) or JS conditions (37.4%; $X^2 = 36.4$; $d.f. = 2$; $p < 0.001$). We have some

Table 6
Missing Data Rates and Codability of Responses by Drug and Experimental Condition, Study 3

	TB	DB	JS	X^2	d.f.
Drug 1 ($n = 4,632$)					
Matchable (in %)	56.9	85.4	85.6		
Codable	30.8	2.7	4.9		
Uncodable	2.2	0.4	1.1	721^{***}	4
Missing	10.0	11.5	8.4	8^{*}	2
n	1,567	1,546	1,519		
Drug 2 ($n = 3,867$)					
Matchable (in %)	56.5	79.9	84.4		
Codable	29.6	5.2	4.5		
Uncodable	1.8	1.1	1.1	482^{***}	4
Missing	12.0	13.8	10.0	9^{**}	2
n	1,300	1,301	1,266		
Drug 3 ($n = 3,005$)					
Matchable (in %)	54.3	74.9	83.6		
Codable	29.9	5.1	4.4		
Uncodable	1.8	1.4	0.4	387^{***}	4
Missing	14.0	18.6	11.5	21^{***}	2
n	1,011	1,005	989		
Drug 4 ($n = 2,199$)					
Matchable (in %)	54.7	70.6	81.2		
Codable	29.0	5.9	5.2		
Uncodable	1.0	1.5	1.0	232^{***}	4
Missing	15.3	22.1	12.6	26^{***}	2
n	724	751	724		
Drug 5 ($n = 1,548$)					
Matchable (in %)	53.6	68.4	78.2		
Codable	30.1	6.9	5.4		
Uncodable	1.5	1.9	1.2	159^{***}	4
Missing	14.8	22.9	15.3	15^{***}	2
n	519	525	504		

The first significance test in the last column tests the distribution of non-missing codes across conditions. The second test contrasts whether item missing data rates are similar across conditions.
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

evidence that these lower rates may reflect lower-quality reporting. Respondents were asked elsewhere in the survey if they were taking drugs for a series of common medical conditions, including HBP and high cholesterol. The agreement between the self-report of taking HBP medication and identifying an antihypertensive or angiotensin drug is lower for the DB group ($\phi = 0.550$) than for the TB ($\phi = 0.668$) and JS ($\phi = 0.60$) groups. Similarly for antilipemic drugs, the agreement rate is lower for the DB group ($\phi = 0.692$) than for the TB ($\phi = 0.795$) and JS ($\phi = 0.753$) groups. However, no significant differences were found for other selected drugs classes (e.g., antidiabetic drugs, antianxiety drugs or antidepressants, antiarrhythmic drugs). Nonetheless, this provides partial evidence of lower data quality for the drop box condition.

5.3 Respondent differences in performance

Are there differences in performance across the three experimental conditions by respondent characteristics? To explore this question, we fit a series of multivariate models,

first examining the main effects of a set of covariates on performance on the task (missing data, response time, and matchable/codable responses). For those variables that have significant main effects, we then examine their interactions with the experimental conditions. We examine several sets of variables, including socio-demographic controls (age, gender, race/ethnicity, and education), technology-related variables (e.g., how much time the respondent spends online; the device, browser and operating system used to complete the survey), and health-related variables (e.g., number of prescription drugs taken, confidence filling out medical forms, whether the respondent needs help taking medications, respondent's rating of their memory and speed of thinking). The detailed models are not presented here, but selected multilevel models are presented in Appendix A.

We first examine covariates of item missing data. Of the set of variables mentioned above, race (1=African American, 0=other) was the only significant ($p < 0.05$) socio-demographic variable in the multilevel models, but did not interact significantly with experimental treatment. African Americans have higher odds of not providing a response to any drugs.

We next turn to socio-demographic covariates of response times and possible interactions with the experimental treatments. Here we find that race, education, age, amount of time spent on the Internet, confidence filling out medical forms, and the number of prescription drugs taken, are significant predictors of response time in the multilevel model for all drugs. Again, however, we find no interaction effects of these variables with the experimental treatment. The main effect of experimental treatment remains significant ($p < 0.001$) after controlling for the variables described above, with predicted mean response times of 51.5 seconds for the TB, 108.5 for the DB, and 67.3 for the JS condition. In general, African Americans take slightly longer than other races to answer these questions (80 seconds versus 72 seconds per drug). Education (4 categories) and confidence filling out medical forms are both negatively associated with response time (i.e., shorter times for better-educated respondents, and for those with greater confidence), while age (4 categories) and number of prescription drugs are positively associated (i.e., older persons and those who report taking more drugs take longer to respond). Since age and number of drugs have independent effects on response times and both have no interactions with the experimental treatments, this suggests that the reduced response times in later drugs is not due to changes in respondent demographics (i.e., older respondents reporting more drugs), but likely to reflect the learning effects among respondents in general.

Those who are more frequent Internet users take less time to answer these questions, suggesting that familiarity aids completion time. Response time is not significantly associated with the type of browser or device (PC versus tablet/smartphone) respondents used to complete the survey, but this may be due to the relatively small number who used mobile devices. We also remind the reader that some respondents with older browsers reported difficulties with the DB and JS versions, and were routed to the TB version. Finally, drug number remains significant in the multilevel models, indicating that response time decreases with each successive drug reported.

The final outcome we examine is whether a usable response was obtained or not. Here we find that race is again a significant predictor, with African Americans being less likely to provide a matchable or codable response. The number of drugs a respondent reported taking is also significant, even controlling for drug number in the models. This suggests that those taking a larger number of drugs may have a harder time reporting the prescription drugs they are taking. This may be associated with poorer health (although we control for health status in the models). Aside from drug number, none of the other predictors reaches statistical significance.

In summary, then, we find mixed results on the socio-demographic correlates of performance or data quality in answering these questions. Education and age affect response times, but not missing data rates or the codability of responses. Similarly, greater confidence in filling out medical forms and more frequent Internet use are associated with shorter response times across all three input formats, but not with missing data or codability. Further, we find no significant interactions with experimental condition, suggesting that the performance differences seen across the experimental conditions do not vary by socio-demographic characteristics, technology factors, or health status.

6 General Discussion and Conclusions

What have we learned across these three studies? First, the design of input tools makes a difference. The improvement in performance of the two alternative approaches (drop boxes and JavaScript lookup) from the first to the second study can be attributed to improvements in design, given that the samples are very similar[3]. The rate of usable data across all three drugs improved from 80.9% to 86.5% from Study 1 to Study 2 in the drop box condition, and from 72.1% to 75.2% in the JavaScript condition, while the average response time across all three drugs decreased from 76.8 seconds in Study 1 to 59.5 seconds in Study 2 for the drop box condition, and from 59.7 seconds to 26.9 seconds for the JavaScript condition. Further the break-off rates on these two versions dropped significantly, from Study 1 (15.7% for the drop box and 14.2% for the JavaScript condition) to Study 2 (1.1% for drop box and 0.9% for JavaScript). These results point to the value of careful design of complex tools such as these. In the third

[3] As noted earlier, an alternative explanation is that respondent familiarity with these tools has improved over time.

study, the JavaScript version took only slightly longer for respondents to complete, had lower rates of item missing data, and yielded comparable rates of codable responses compared to the standard text box version.

Second, respondents' performance using the alternative approaches appears to improve with practice, at least from the first to the second drug. While this suggests that respondents learn how to use these tools over the course of a survey, this does not help if the tools are only used rarely in surveys. Typing in a text box is an intuitive task that does not require much practice; however such an approach does not provide respondents with feedback on the appropriateness of the answers. The issue of learning time should be considered when using relatively unfamiliar tools. The fact that frequency of Internet use is negatively associated with response speed also suggests that those more familiar with and comfortable using the Internet may perform better with all input tools (that is, we find no interaction between Internet use and input type).

Third, the difference in performance between the drop box and JavaScript versions illustrates the trade-off of alternative design tools. The drop box (by definition) constrains the respondent to the set of items on the list. This results in longer response times (greater effort) to find the relevant item in a long list. It also results in more selections earlier in the list (primacy effects). The JavaScript combo box, on the other hand, may be more flexible in that it permits respondents to provide partial information (that does not yield an exact match) or type a response that does not match any item in the list. In both Study 1 and Study 2, the proportion of uncodable responses was higher for the JavaScript version than the drop box version. Giving the respondent the option of entering the response in a text box (by selecting "other, specify") reduces the rate of uncodable responses in the JavaScript version. This suggests that respondents may not know that they can type a response rather than select an item from the list in a combo box. Further, the improved design of the JavaScript version in Study 3 yielded comparable machine-matchable rates to the drop box version, with significantly lower average response times. This may suggest that 1) optimizing the design of the JavaScript lookup to facilitate respondents' behavior and 2) providing a text box alternative (or explaining how the text field works in a combo box) for those who were unsuccessful using the JavaScript lookup to find a response option should improve performance. The text box alternative is particularly helpful for the small number of users that do not have JavaScript enabled, and for those using older browsers or mobile devices with compatibility issues running JavaScript. However, we note that virtually all e-commerce and social media applications require some form of active scripting, so this is less of a concern for general Internet-user populations. We further note that JavaScript lookup is widely used in such applications.

Fourth, across all three studies, the vast majority of respondents provided usable (i.e., codable) responses to a fairly complex set of questions. For example, on average across all drugs, 64.3% of responses in the text box condition were codable in Study 1, 78.5% in Study 2, and 85.6% in Study 3. For the drop box condition, 80.9% of responses in Study 1, 86.5% in Study 2, and 82.7% in Study 3 were codable. Finally, for the JavaScript condition, the rate of usable (codable) responses ranged from 72.1% in Study 1 to 75.2% in Study 2 and 88.3% in Study 3. However, we have some evidence from Study 3 that the quality of the response in the DB condition may be lower than that for the other two conditions.

In summary, then, we see a clear trade-off in the use of these input tools in Web surveys. Offering an alternative design tool, with which respondents may be less familiar, may yield more automatically-codable responses (i.e., reducing coding effort), but comes at greater cost to respondents in terms of response time. The standard text field approach is easiest for respondents but may require more effort to code and match the responses to the database, and makes it harder to resolve ambiguous or insufficient responses.

Finding tools to optimize the completion of complex tasks in Web surveys, given that respondents may rarely encounter such tools or question types, remains a challenge. The choice of tool may depend on the objectives of the question being asked. If the question is designed to capture data that can be coded and analyzed later, then the standard text box seems sufficient. But if the question needs to yield an immediately-codable response (e.g., to direct follow-up questions or in e-commerce applications where a correct selection is required), then the drop box alternative may be needed; however, it takes longer and may yield lower-quality data. From a technical perspective, the text box is the easiest to implement, and works on all browsers and devices. For real-time complex coding tasks, we need to continue to refine the tools that optimally support respondents in providing the answers that we want.

7 Acknowledgements

Experiments 1 and 2 were supported by a grant from the National Institute for Child Health and Human Development (R01 HD041386-01A1, P.I.s Roger Tourangeau, Mick P. Couper, Frederick G. Conrad, and Reginald P. Baker). Experiment 3 was supported by a grant from the National Institute on Aging (R01 AG020638, P.I. Arie Kapteyn). We thank Karen Farris and Peter Batra of the University of Michigan's School of Nursing for invaluable assistance with the LexiComp database. We also thank the reviewers and Editor for their helpful comments on an earlier draft.

References

Buskirk, T., Michaud, J., & Saunders, T. (2014). *Swipe, snap & chat: mobile survey data collection using touch*

question types and mobile OS features. Paper presented at the Midwest Association for Public Opinion Research (MAPOR) conference, Chicago, November.

Callegaro, M., Lozar Manfred, K., & Vehohar, V. (2015). *Web survey methodology*. Los Angeles: Sage.

Couper, M. (2008). *Designing effective web surveys*. New York: Cambridge University Press.

Couper, M. (2016). Researcher contribution: prescription drug lookup experiment (PDLE). Retrieved from http://hrsonline.isr.umich.edu/index.php?p=shoavail&iyear=CG

Couper, M., Antoun, C., & Mavletova, A. (2015). *Mobile web surveys: a total survey error perspective.* Paper presented at the International Total Survey Error Conference, Baltimore, MD, September.

Couper, M., Kennedy, C., Conrad, F., & Tourangeau, R. (2011). Designing input fields for non-narrative open-ended responses in web surveys. *Journal of Official Statistics*, *27*(1), 1–22.

Couper, M., Tourangeau, R., Conrad, F., & Crawford, S. (2004). What they see is what we get: response options for web surveys. *Social Science Computer Review*, *22*(1), 111–127.

Funke, F. & Reips, U.-D. (2007). *Dynamic forms: Online Surveys 2.0.* Paper presented at the General Online Research Conference (GOR'07), Leipzig, March.

Gendall, P. & Healey, B. (2008). Asking the age question in mail and online surveys. *International Journal of Market Research*, *50*(3), 309–317.

Grondin, C. & Sun, L. (2008). *2006 Census Internet mode effect study.* Paper presented at the Joint Statistical Meetings, Denver, CO, August.

Healey, B. (2007). Drop downs and scroll mice: the effect of response option format and input mechanism employed on data quality in web surveys. *Social Science Computer Review*, *25*(1), 111–128.

Heerwegh, D. & Loosveldt, G. (2002). An evaluation of the effect of response formats on data quality in web surveys. *Social Science Computer Review*, *20*(4), 471–484.

Herzing, J. (2015). *Comparing interface designs of database lookups with traditional measurements.* Paper presented at the European Survey Research Association Conference, Reykjavik, Iceland, July.

Herzing, J. & Schneider, S. (2014). *Easy question, tricky answer: measurement quality of education questions.* Paper presented at the annual meeting of the American Association for Public Opinion Research, Anaheim, CA, May.

HRS-IS. (2013). Health and Retirement Study 2013 Internet survey data file, version 1.0, released July 2015. Retrieved from http://hrsonline.isr.umich.edu

Peterson, G., LaFrance, J., Griffin, J., & Li, J. (2015). *Smartphone participation in web surveys: choosing between the potential for coverage, nonresponse, and measurement error.* Paper presented at the International Total Survey Error Conference, Baltimore, MD, September.

Tijdens, K. (2014). Dropout rates and response times of an occupation search tree in a web survey. *Journal of Official Statistics*, *30*(1), 23–43.

Tijdens, K. (2015). Self-identification of occupation in web surveys: requirements for search trees and look-up tables. *Survey Methods: Insights from the Field*. doi:10.13094/SMIF-2015-00008

Tourangeau, R., Conrad, F., & Couper, M. (2013). *The science of web surveys*. New York: Oxford University Press.

Tourangeau, R., Couper, M., Conrad, F., & Baker, R. (2008). Web Design Experiment 7: 2008 (United States). Interuniversity Consortium for Political and Social Research, Ann Arbor, MI (distributor). doi:10.3886/E55245V1

Tourangeau, R., Couper, M., Conrad, F., & Baker, R. (2011). Web Design Experiment 9: 2011 (United States). Interuniversity Consortium for Political and Social Research, Ann Arbor, MI (distributor). doi:10.3886/E55420V1

Appendix
Multilevel Models for HRS-IS with Demographic Variables

Table A1
Response Time Model (DV=Response time in Seconds)

	Coef.	Std. Err.
Intercept	30.72***	3.45
Experimental treatment (Ref. cat.: *Text box*)		
Drop box	57.04***	1.38
JavaScript lookup	15.83***	1.35
Number of drugs taken	1.69***	0.36
Drug number	−9.71***	0.32
Gender (1=male)	−1.31	1.17
African American (1=yes)	8.34***	1.83
Hispanic (1=yes)	3.79	2.43
Education (Ref. cat.: *College graduate*)		
<High school	10.17**	3.54
High school or GED	5.90**	1.71
Some college	2.17	1.80
Age category (Ref. cat.: ≤ 59)		
60-69	5.45**	1.49
70-79	14.64***	1.59
80+	18.84***	2.20
Hours on Internet per week (Ref. cat.: *15+ hours*)		
<1 hour	7.93**	2.75
1-7 hours	2.96*	1.36
8-14 hours	2.65	1.55
Self-rated memory (Ref. cat.: *Fair/poor*)		
Excellent	0.14	3.40
Very good	−0.19	2.38
Good	−2.83	1.97
Self-rated speed of thinking (Ref. cat.: *Fair/poor*)		
Excellent	−5.30	3.29
Very good	−0.90	2.53
Good	0.96	2.15
Confid. filling out med. forms (Ref. cat.: *Extremely*)		
Quite	3.71**	1.34
Somewhat	6.90**	1.84
Little/not at all	4.80*	2.26
Need help taking meds (1=yes)	−1.52	2.27
Device type (Ref. cat.: *PC*)		
Tablet/smartphone	31.35	17.14
Browser (Ref. cat.: *Chrome*)		
Firefox	1.36	2.19
MSIE	2.31	1.72
Safari	6.10*	2.72
Other	−18.64	17.02

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table A2
Missing Data Model (DV 1=Missing, 0=Not Missing)

	Coef.	Std. Err.
Intercept	−10.66***	1.03
Experimental treatment (Ref. cat.: *Text box*)		
Drop box	0.52	0.33
JavaScript lookup	−0.13	0.36
Number of drugs taken	−0.54***	0.09
Drug number	0.79***	0.05
Gender (1=male)	0.24	0.29
African American (1=yes)	0.86*	0.39
Hispanic (1=yes)	0.14	0.57
Education (Ref. cat.: *College graduate*)		
<High school	0.48	0.82
High school or GED	0.29	0.45
Some college	0.19	0.47
Age category (Ref. cat.: ≤ 59)		
60-69	0.11	0.36
70-79	−0.004	0.40
80+	0.22	0.53
Hours on Internet per week (Ref. cat.: *15+ hours*)		
<1 hour	−0.16	0.58
1-7 hours	−0.31	0.62
8-14 hours	−0.44	0.61
Self-rated memory (Ref. cat.: *Fair/poor*)		
Excellent	0.82	0.79
Very good	0.13	0.59
Good	0.095	0.49
Self-rated speed of thinking (Ref. cat.: *Fair/poor*)		
Excellent	0.17	0.80
Very good	−0.044	0.63
Good	0.16	0.53
Confid. filling out med. forms (Ref. cat.: *Extremely*)		
Quite	0.20	0.35
Somewhat	0.53	0.44
Little/not at all	1.11*	0.48
Need help taking meds (1=yes)	0.20	0.52
Device type (Ref. cat.: *PC*)		
Tablet/smartphone	5.09	33.14
Browser (Ref. cat.: *Chrome*)		
Firefox	−0.24	0.55
MSIE	−0.16	0.41
Safari	0.047	0.65
Other	−4.54	33.13

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table A3
Usable Data Model (DV 1=Matchable/codable, 0=Not codable/missing)

	Coef.	Std. Err.
Intercept	9.55***	0.91
Experimental treatment (Ref. cat.: *Text box*)		
Drop box	−0.34	0.29
JavaScript lookup	0.25	0.31
Number of drugs taken	0.51***	0.083
Drug number	−0.72***	0.048
Gender (1=male) -0.22	0.26	
African American (1=yes) -0.87*	0.35	
Hispanic (1=yes) -0.14	0.51	
Education (Ref. cat.: *College graduate*)		
<High school	−0.57	0.73
High school or GED	−0.26	0.40
Some college	−0.19	0.42
Age category (Ref. cat.: ≤ 59)		
60-69	−0.080	0.32
70-79	0.10	0.35
80+	−0.042	0.48
Hours on Internet per week (Ref. cat.: *15+ hours*)		
<1 hour	0.29	0.52
1-7 hours	0.48	0.56
8-14 hours	0.61	0.55
Self-rated memory (Ref. cat.: *Fair/poor*)		
Excellent	−0.71	0.71
Very good	−0.097	0.53
Good -0.047	0.44	
Self-rated speed of thinking (Ref. cat.: *Fair/poor*)		
Excellent	−0.16	0.71
Very good	0.0095	0.56
Good	−0.12	0.47
Confid. filling out med. forms (Ref. cat.: *Extremely*)		
Quite	−0.18	0.30
Somewhat	−0.46	0.39
Little/not at all	−1.03*	0.44
Need help taking meds (1=yes) -0.26	0.47	
Device type (Ref. cat.: *PC*)		
Tablet/smartphone	−5.15	29.11
Browser (Ref. cat.: *Chrome*)		
Firefox	0.24	0.48
MSIE	0.22	0.37
Safari	0.17	0.59
Other	4.65	29.11

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$