Berkeley Journal of Experimentation, Vol. 1, No. 1 (Dec. 2020)

Experiment of Effect of Date Pickers on Data Entry Speed

By Javed Roshan, Jorge Hernandez, Hao Wu, and Kevin Martin

Date pickers are a ubiquitous method of data entry on forms of all types. The goal of these entry methods is to speed data entry performance and reduce customer frustration. However, our review found no published experimental studies which evaluate their effectiveness. In this paper, we perform an experiment with 1543 participants from the amazon MTurk survey platform to examine the effect of using date pickers on entry time for dates within a month of the current date as well as dates with different months and years than the current date. We find that datepickers provide a significant decrease in entry time for both types of dates, but with a much larger effect on dates within a month of the current day. For detailed analysis and reproduction files see the research study github repository at: https://github.prod.oc.2u.com/kmartin/w241-final-project

KEYWORDS: User Experience, UX, Calendar, Date, Experiment

1. Introduction

Calendar date pickers are clickable interfaces that are used as an alternative date entry method on a computer. They are widely used on credit card applications, vacation rental forms, government services, trip planners, and other similar applications. Calendar date pickers have several advantages in terms of usability over manual date entry: They allow users to view day of the week for the given date being entered, they can convey different price information as is often the case on airline reservation websites, and date pickers can show availability as with websites that are used to book services. On the down-side, they can cause a host of accessibility issues for the visually impaired or those who need alternate data entry methods and a lack of uniformity in design can make them frustrating to use. Regardless, we were unable to find any published experiment verifying their effectiveness or for what cases they are more or less effective.

We decided to run a simple test of datepicker efficacy on the metric of time taken to enter the date. We ran an experiment using a custom built web page where users were randomly assigned to fill in dates with either a date picker or by “manually” typing in dates. Users were tasked with filling out a car rental form where they were required to fill in a variety of dates within the same month of the current date (**near dates**) and in other years than the current date (**far dates**). The time it took to enter each date was measured as the time that the given date field was in focus on the page. After, the times were compared to each other. The 2x2 multifactor setup allowed us to show that users are significantly faster filling out dates using a calendar date picker than by manually entering them. We also showed that the effect of the date picker is much more pronounced for near dates than for far dates.

2. Experimental Details

Brief overview

Participants for the experiment were recruited on Amazon MTurk. They were told that they were filling out a user experience survey of a car rental form. Users were directed to our custom web page which assigned them into either treatment or control. Treatment users filled out all dates on the page using a calendar datepicker. Control users filled out all dates on the page by “manually” typing them. All users had to fill out 4 dates in total. Two of the dates were a rental start and end date that users were instructed should be in the next month. The other 2 dates were a birthdate and a driver’s license date. These dates were provided via a photo of a hypothetical driver’s license. The entry time for each field was recorded and near times in treatment and control were compared separately from far times for treatment and control.

Our hypothesis going into the experiment:

For the near date questions, the response time should be in favor of the datepicker group (negative treatment effect, shorter response time for the treatment group), and for the far date questions, the response time will have compressed treatment effect in either direction (positive or negative treatment effect) relative to the near date questions.

2.1 Outcomes

Outcome Measurement

The measured outcomes were the entry times for the two “near dates” (the beginning and end of the hypothetical rental period) and for the two “far dates” (the birthday and driver’s license expiration). The entry times were measured as the total time that each given field had focus within the browser. This is the total time that the date picker widget was open for those in treatment or, for those in control, it was the total time that the cursor was active within a given date field. If a user left a field then reentered it (for example if they entered the wrong date for the driver’s license expiration date or the birthday and the form validator would not let them submit, they had to go back and correct the mistake), the time continued to accumulate for the given field from where they left off.

Outcome Transforms

The specific outcome measures used in our experiment are the natural log of the average entry times for the "far dates" and the natural log of the average entry time for the "near dates". We decided to transform our outcome variables based on the results of 2 pilot studies, of approximately 50 people each, that we ran in the weeks leading up to our experiment. We averaged the dates pairs in order to reduce variance in our outcome measures and to reduce our outcome to just a single measurement for each of our tests. We took the natural log of the outcomes in order to account for an observed right skew in the pilot study data. Additionally, transforming the outcome also helps to understand the results in simpler terms.

Exclusions

In order to preclude a single outlier time measurement from exerting too much influence on our average entry time for a given measurement, we did not include users who took more than 300 seconds (5 minutes) in any given date field in our data. The idea was that such users got distracted doing something else and were not actively participating in the survey. We determined that there was a 1/1000 chance of any particular observation exceeding this time threshold based on our analysis of the pilot study data and assuming that the log of the entry times followed a normal distribution. All of these transformations and exclusions were laid out in the pre-analysis report that we submitted prior to the start of the study.

2.2 Treatment and Control Condition Details

Treatment and control specifics

Participants in the treatment condition were required to fill in all dates on the form using a date picker. Treatment users were unable to manually type the date in using a keyboard. The specific implementation of the datepicker used was the "jQueryUI" with month and year dropdowns, due to it having a more straight-forward design in which the user is easily able to locate the month and year switching menu. Users in the control condition were required to fill in all dates on the form by typing in the dates. No date picker was provided to them.

Experimental Factors and Form Validation

Upon entry to the site, users were told that they were taking part in a user experience survey followed by questions about their experience at the bottom of the page. Participants were instructed to fill in 2 “near dates” within a month of the current date and were informed that these were the rental dates that they would have the car for. The “near dates" did not require specific dates meaning any date was acceptable, but they were suggested to match the format “mm/dd/yyyy” or the form would not submit. On the second half of the page, participants were asked to fill in the birthday and driver’s license expiration date for a hypothetical user. Both of these “far dates” were in a different month and year than the current date. They were required to match the date given on the hypothetical drivers license and use the format “mm/dd/yyyy” or else they would receive an error and the form would not submit until the mistakes were corrected.

2.3 Participant Recruitment and Generalizability

Participant Population

To determine the impact of date pickers on both near (within a month of the current date) and far (several years from current date) date entry times, we conducted an experiment using 1543 users of Amazon’s MTurk service. We hope that these results will generalize to the population of web users as a whole. However, it should be noted that according to a 2016 pew research study, MTurk users were largely US based, younger, and more educated than the population as a whole (Pew, 2016). We don't think that this should significantly affect generalizability, however it is possible that such a population would be more comfortable with the date picker widget than the general population. Additionally, our test data showed that our respondents only answered on desktop browsers whereas a lot of typical internet traffic takes place on mobile devices. These unique characteristics of the MTurk respondents don't invalidate the results, but they do suggest that more work will be required on future studies to generalize to a larger population.

Recruitment assignment to treatment, data collection

Participants in this study were recruited on the amazon MTurk platform. A request was put out which offered an average of $0.24 for completing an 8 minute survey. Participants were informed that they were completing a user experience survey for a car rental form. If they accepted the job, they were directed to a link to our custom built webpage where the experiment was conducted. The webpage was designed to mimic a simple car rental webpage (Figure 1). When the page was requested by a user, a simple javascript call randomly assigned users to control or treatment. The javascript was run on the user's local computer. Treatment and control assignment was given with a 50 percent probability for either condition. The form for the appropriate treatment condition was then loaded into the user's browser. As users completed the survey, a javascript program was running in the background tracking time spent on each field. The results of the survey were then collected by our web server when the participant clicked "submit".

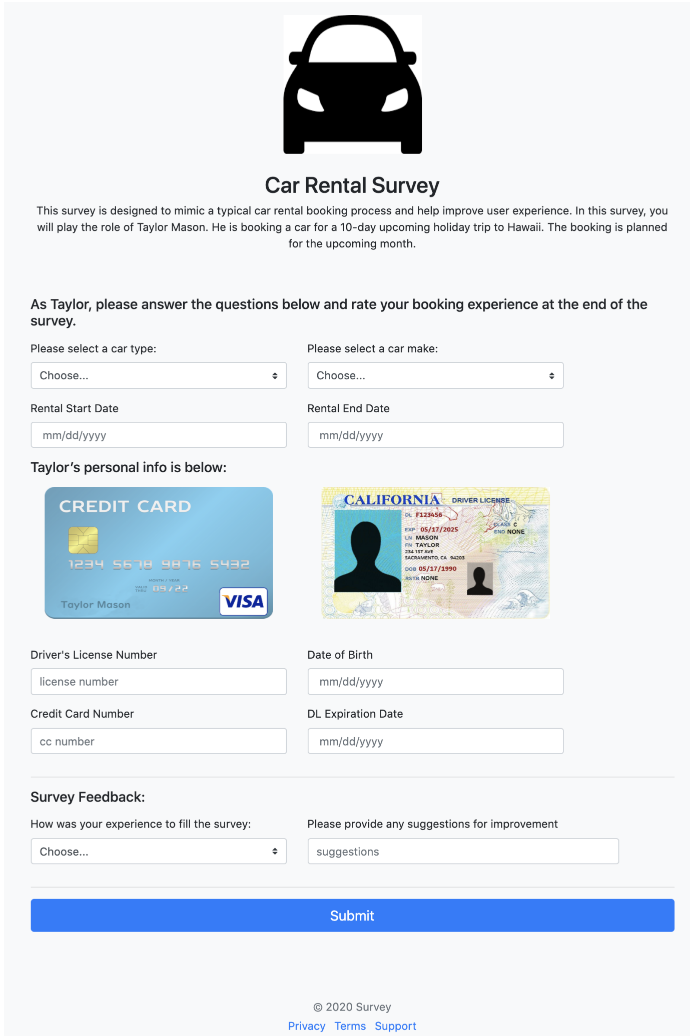


Figure 1 - Survey Page

Timing of study

The request for participation went out to Mturk on two separate batches. The first batch was submitted on November 15, 21:28 PST. All responses were received by November 16, 01:42 PST for a total of 250 responses. We used 50 of these as a pilot. We subsequently submitted a batch with a request for 1300 participants on November 19, 2020. However, after around 200 responses this batch was suspended by Amazon due to some system issue. We resubmitted this batch with a request for 1100 participants on November 22 at 22:00 PST. By November 24 at 08:07 PST, all requested respondents had participated in the survey. It should be noted that we thought it was important that participants be given the study near the middle of the month so that they could have all ten days of their hypothetical upcoming rental period in the same month as the current date. This would prevent them from having to scroll on the datepicker.

3. Analysis

Breakdown of observations and covariate balance checks

We requested 1543 people to fill out our survey on MTurk. Out of all those who claimed to have completed the survey, we actually got 1517 responses submitted to our web server. 2 of these were excluded from our results due to excessive response time, as laid out in our pre-analysis plan. Of the final 1515 responses included in our analysis, 821 were in control, 694 were in treatment. These responses are shown schematically in figure 2. Checking against a binomial probability distribution, the probability of a treatment imbalance this extreme given a p=0.5 assignment is less than 1/1000. This outcome is obviously a significant failure of covariate balance. The treatment control imbalance is fairly consistent across the hour of the day and browser type. Unfortunately, the pilot studies were not large enough to reveal this underlying issue. Potential causes and implications of this imbalance are discussed in the following paragraphs. In our opinion, the most likely cause of the imbalance is an error in javascript which caused the randomization to behave in an unexpected way.

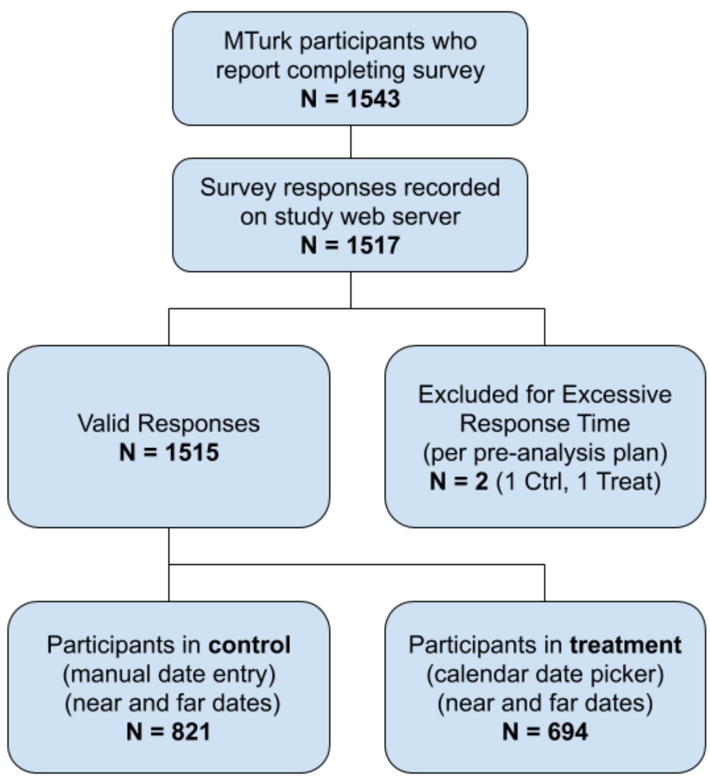


Figure 2 - Observation Flow Diagram (CONSORT)

3.1 Discussion of Covariate Imbalance

Speculation on Different Treatment, Control Assignment

Due to data collection design limitations, randomization happening on the user's local machine and having no data for a particular observation until a user submits the form, we can't be sure what caused the imbalance in treatment conditions. However, we can speculate on it. In our opinion, the cause of the imbalance was either some issue with the javascript randomization seed or could have been differential attrition. In our opinion, the javascript randomization error is more likely.

Differential Attrition

At first glance, differential attrition seems a likely cause of the treatment-control differential. Date pickers are anecdotally considered to be confusing. Additionally, they have well documented design flaws that makes them difficult to use for the visually impaired (Roselli, 2020). However, on closer inspection of the data, the differential attrition explanation looks less convincing. If differential attrition were the cause of the imbalance between treatment groups, then by definition, total attrition would have to be greater than the differential attrition. If we assume that the initial assignment was perfectly split 50-50, that would mean that a minimum of 127 people attrited from the treatment (datepicker) group. The total attrition number would increase if anyone had attrited from the control (manual entry) group. However, our estimate for total attrition is only 26 people; that's the difference between the number of people who reported completing the survey and the number of survey responses that we actually got back on the server.

There is of course a possibility that our estimate for total attrition is flawed and that participants would attrit without attempting to get paid via MTurk and reporting completion of work, but we find this unconvincing. We see no incentive for them to not attempt to receive payment for participation. Several participants who got paid and were apparently unable to complete the survey pasted error codes instead of the sessionid that they were supposed to provide to verify their participation. It is also very unlikely that participants received multiple versions of the page. The server assigned each user a unique “sessionid” cookie when they first logged into the site. Even if they were to refresh, they would be given the same version of the page (unless they cleared their cookies). Given that differential attrition seems unlikely, we move on to discuss the possibility of an error in the javascript random assignment.

Speculate on Possible Random Assignment Mechanism

While we don't know what the precise issue with the randomization might have been, we can speculate. Since the randomization function runs on the client side, there's just no way of telling what's going on. Certainly there have been reports in the past of a common browser extension disrupting the `Math` javascript package which we use for random assignment. The randomization function uses the system time as a seed, so maybe if multiple users have a system time that is absent or not accessible from the javascript interface that would potentially cause them to use the same default seed and come up with the same random number. Perhaps because the script needs to run for the page to load, there is some sort of browser optimization that just plugs in a certain value if communication is taking too long. Unfortunately, we can't know the precise mechanism without a lot more information on what the MTurk participants were doing.

Pessimism even if it isn't differential attrition

Even if we are right and the different assignment to treatment and control is not due to attrition, that still means that there is some kind of common browser plug-in or local compute condition that is interfering with the randomization process. This would mean that the random assignment is not **entirely** random. If that's the case, some subjects are being sorted into the control group (and out of the treatment group) because of some shared attribute that we were unable to identify. Users who share such attributes may be faster or slower at entering dates than the population in general. We have no way of knowing their outcome. Similar to how we noted that the MTurk population may not be a complete representative of internet traffic as a whole, this doesn't necessarily invalidate our results, but it does mean that we should take them with a grain of salt when applying them to new conditions.

3.2 Model Results Assuming No Differential Attrition

Headline Regression Numbers

In this section we proceed with the calculation of effect sizes as laid out in the pre-analysis plan assuming that the unbalanced assignment to treatment condition is pure random chance and it has no systematic effect on results. (In a later section, we examine the consequences to our results if the treatment control imbalance is due to differential attrition.) We see that the date picker results in a significantly reduced log date entry time for both the near dates and the far dates. Use of the date picker resulted in date entry times that were 64.4 percent lower for dates within a month of the current date and date entry times were 14.2 percent lower for dates in a different year than the current date. (See (Ford, 2018) for rules on interpreting logged outcome variables)

Interpretation and commentary on the numbers

The result generally fits with our intuition and predictions made prior to the experiment where we thought near dates would be faster with the date picker. We also thought that going to other years within the datepicker widget is enough trouble that it might or might not be significant. Indeed, there was a very large effect on the near dates and the effect on the far dates was much smaller in comparison. Luckily, thanks to the large number of observations in our sample, we were able to get quite significant results out of both date types in our experiment.

Table 1: Regression Results Assuming Treatment Imbalance is Due to Chance

Table 1: Nominal Regression Results

	Dependent variable:	
	ln_nea_avg	ln_far_avg
datetype	-1.034^{***}	-0.153^{***}
	(0.038)	(0.029)
Constant	9.822^{***}	9.944^{***}
	(0.493)	(0.257)
Browser Effects	Yes	Yes
Hour Effects	Yes	Yes
Adjusted P	4.38e-132	1.22e-07
Observations	1,515	1,515
R^2	0.380	0.090
Adjusted R^2	0.351	0.047
Residual Std. Error (df = 1446)	0.694	0.561
F Statistic (df = 68; 1446)	13.031^{***}	2.091^{***}

Note: *p<0.1; **p<0.05; ***p<0.01
ln-nea-avg: natrual log of average time to enter 'near' dates
ln-far-avg: natrual log of average time to enter 'far' dates
datetype: 1 is treatment (datepicker) and 0 is control (manual entry)
regression coefficeints shown above, standard errors in '()' below
'Adjusted P' is the Holm adjusted p value for 'datetype' coefficient

3.3 Other Interesting Trends in the Data

Introduction

In exploring the results of the analysis, we noted several interesting trends in addition to the specific treatment effect that we were interested in investigating which we had laid out in the pre-analysis plan. These other effects of the treatment point to interesting future studies that might be done on the subject of datepickers.

3.3.1 Total Time in Completing Form

Treatment Effect on Total Time to Complete Survey

We were able to calculate the total time taken to fill out the form and regress it on the treatment condition. Counterintuitively, those assigned to treatment took longer to fill out the form than those assigned to control. This is opposite of our findings for the date fields themselves. Specifically, the regression found that those assigned to treatment took 32.0 seconds longer (15.8 percent longer) to complete the survey than those in control. This finding doesn't fit well with our observations of the treatment outcomes since the time to complete those fields were quicker than the control.

Table 2: Total Time in Form Regression Results

	Dependent variable:	
	ttltime_s	log(ttltime_s)
datetype	31.950***	0.147***
	(6.075)	(0.026)
Constant	187.553***	5.110***
	(3.611)	(0.018)
Browser Effects	No	No
Hour Effects	No	No
Adjusted P	8.28e-08	2.44e-08
Observations	1,514	1,514
R^2	0.019	0.021
Adjusted R^2	0.018	0.020
Residual Std. Error (df = 1512)	115.658	0.505
F Statistic (df = 1; 1512)	28.677***	31.646***
Note:		*p<0.1; **p<0.05; ***p<0.01
		ttltime-s: time to complete the survey in seconds
		log(ttltime-s): natural log of time to complete the survey in seconds
		datetype: 1 is treatment (datepicker) and 0 is control (manual entry)
		regression coefficeints shown above, standard errors in '()' below
		'Adjusted P' is the Holm adjusted p value for 'datetype' coefficient

Connecting Unexpected Treatment Result to Randomization Error

The odd results of the regression analysis of treatment on total time to complete the survey may indicate a couple of different things. It might just be evidence that our random assignment has failed in some way and our treatment group has some unobserved difference that causes them to be slower in general at filling out web forms. We already know that our random assignment did not go as expected because our number in control and treatment were different. The increased total completion time for treatment seems to reinforce the idea that there was some “mechanical” issue with the randomization rather than a differential attrition. Our intuition is that those who are put off by calendar date pickers and attrit from the survey would be generally less technologically savvy and take longer to fill out a web form than those who felt comfortable enough to continue. Since we know that the treatment group took longer to fill out the non-datepicker parts of the survey, we feel reinforced in the belief that the treatment-control imbalance was more likely due to a compute environment issue. Furthermore, since we see that the people who ended up in the treatment are slower in general, we can feel confident of the robustness of the speed increases attributed to the datepicker.

Impact on Future Work

The implication that a datepicker might reduce time to enter dates in date fields specifically but lead to an increase in overall response time for a web form is intriguing and should be more thoroughly investigated in future research. We discussed a theory that the time discrepancy might reinforce our notion of a failed randomization that is connected to a certain user type or browser configuration, but it’s just a theory. In future work, better controls should be implemented at the various stages of the experiment to explore if this finding holds true in other situations. It might be that the datepicker is mentally exhausting to use or that switching context mentally from the datepicker back to manual entry has some sort of cost and the datepicker does in fact change how people respond to other fields. We won’t know for sure unless we run more studies.

3.3.2 Covariate Analysis

Covariates’ impact on treatment variable

Given the data we gathered through MTurk, we mainly explored two covariates, browser type and hour of the day. We suspected these two variables may have a significant impact on our treatment effect estimation. We explored the two covariates separately, using `vcovHC()` and `felm()` functions to account for heteroskedastic robust standard error and the statistical test “holm” to adjust for multiple comparison p-values. We also compared performance between the base model and the model with added covariates using a F-test.

For the Browser Type covariate:
In both the near date regression model and the far date regression model we did not see a significant impact on the date type coefficient when adding in the browser type covariates, but we did observe F-test results indicating the additional covariate was an improvement in model performance.

For Hour of the Day covariate:
In both the near date regression model and the far date regression model we did not see a significant impact on the date type coefficient when adding in the hour of the day covariates. We also did not observe indication of improvement with the additional covariates using a F-test.

Heterogeneous Treatment Effect Exploration

To identify heterogeneous treatment effects (HTE) for the two covariates we focused on we introduced interaction terms between the covariates and our treatment variable. After accounting for heteroskedastic robust standard error and holm multi-comparison p-value adjustments, we found that there is a marginally significant difference in treatment effect between opera browser responses and the base Chrome browser responses. Due to the adjusted p-value barely reaching the 5% significance level, we advise setting up a separate test to analyze the treatment effect if necessary. We did not find any evidence of HTE with regard to the hour of the day covariate.

The detailed regression tables are in the covariate analysis section and heterogeneous treatment exploration sections of the final-analysis.md file within the git repository linked in the abstract for reference.

3.3.3 Detailed Treatment Outcomes

Outcomes Measured and Potential Issues

In our final regression, we looked at an average measure of the effect of the datepicker on “near dates” and on “far dates”. This was helpful because it reduced the variance and got us down to one measure for each pair. It also reduced the number of tests being performed to help us with our family-wise error rate. However, the averaging also obscured some of the interesting finer details at the individual field level. Detailed histograms of participant response times can be seen in figures 3 and 4. We explore these detailed response times in this section.

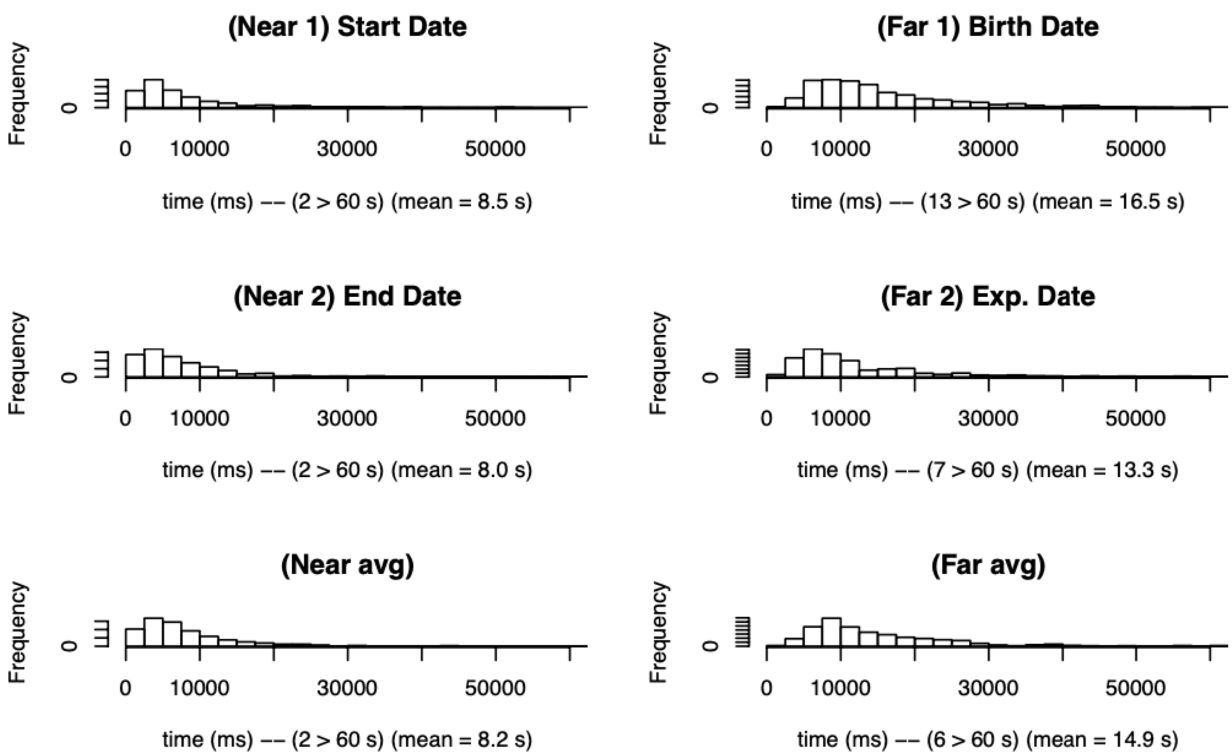


Figure 3 - Entry Times for Treatment (datepicker) Group

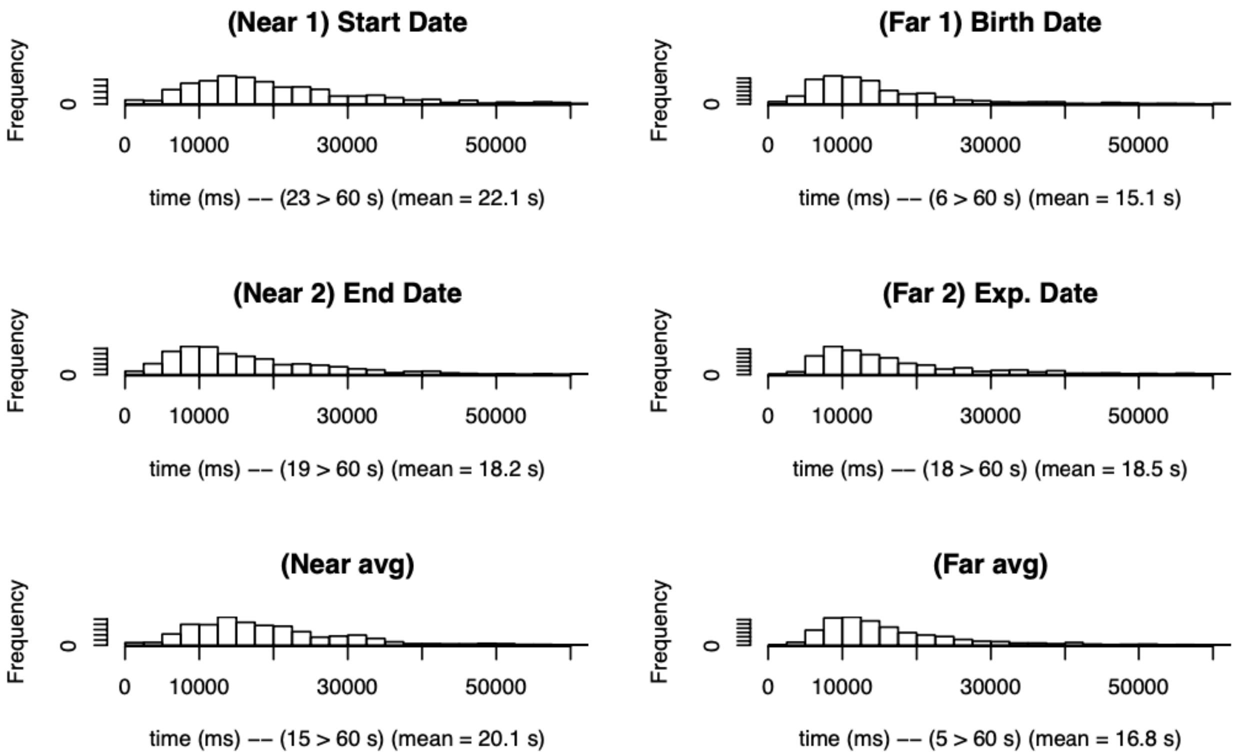


Figure 4 - Entry Times for Control (manual entry) Group

Learning Curve for the Datepicker

Looking at the detailed datepicker response times in figure 3, we see that for both the near-dates and the far-dates, the second field of the pair encountered was entered more quickly than the first field of the pair encountered. This suggests that there may be a learning curve associated with using the datepicker. The speed increase on the second field was particularly strong between the two far dates. The first date (birthday) had a mean time of 16.5 seconds. This individual date field actually took longer to enter than the manual entry did (15.1 seconds). The second date (expiration) had a mean time of 13.3 seconds, that's a 19.4% decrease in entry time from the birthday. The idea that there is a learning curve that is particularly strong for the far dates would make sense. To access different years the participants have to use a different part of the widget that may be non-intuitive. However, we also know that the birthdate was farther from the current date and might have required more scrolling and taken longer than the drivers license expiration date regardless of order. In future experiments it would be helpful to randomize the order of these types of fields to see if it is field type or order that is causing the differences. This could have implications for businesses that choose to implement a datepicker. If only one or two dates are required then the learning curve for users might reduce overall speed.

3.4 Extreme Value Bounds Analysis

Extreme Value Bounds Procedure

In this section we conduct an extreme value bounds analysis to examine our results looking at the possibility that differential attrition occurred. If we assume differential attrition due to treatment type then that changes our interpretation of events considerably. That might mean that people got frustrated with the datepicker and didn't complete the survey. A quick way of analyzing what might have happened if they had stayed instead of (hypothetically) attriting would be to assume that each of these observations took the maximum time to complete the survey that anyone in the treatment group did and see if the results are maintained. We assume that the entire 127 observation difference is due to attrition. We add that many observations to the treatment group each of which has the maximum average entry time that was seen in the set for the near dates among people in treatment and similarly the maximum for far dates.

Caveats for Extreme Value Bounds Analysis

This process outlined above is known as extreme value bounds analysis. It is generally considered a conservative way to estimate treatment effects in the face of differential attrition. It should be noted though that we can’t truly know the effects for those who attrited. It’s possible that everyone who attrited would have taken longer than the longest entry time in our set; in that case, our extreme value bounds estimates would be unconservative. Additionally it needs to be noted that this exercise isn’t able to attempt to correct for the possibility that unobserved qualities of the user or their compute environment are affecting our randomization process. In that case, we would just have to hope that those whose browsers or compute environments caused the error are not systematically different in their date entry speed behavior than those who didn’t suffer from the same randomization error.

Extreme Value Bounds Outcome and Interpretation.

We find that even with the fairly conservative assumptions of the extreme value bounds analyses, the treatment still shows a significant effect for near dates. The date entry time is 46.9% lower for those in treatment than those in control. Under this assumption, the far dates are no longer faster. The date entry time is 18.1% higher for those in treatment than those in control. The specific results of the regression are shown in table 2 below. This exercise shows the robustness of the near date result and gives us more confidence that the result is true regardless of the unforeseen circumstances of our imbalance treatment condition.

Table 3: Extreme Value Bounds Analysis Results

	Dependent variable:	
	ln_nea_avg	ln_far_avg
datetype	-0.634^{***}	0.166^{***}
	(0.045)	(0.038)
Constant	9.733^{***}	9.577^{***}
	(0.022)	(0.027)
Extreme Values	Yes	Yes
Browser Effects	No	No
Hour Effects	No	No
Adjusted P	8.79e-42	5.46e-06
Observations	1,642	1,642
R^2	0.106	0.012
Adjusted R^2	0.105	0.011
Residual Std. Error (df = 1640)	0.922	0.764
F Statistic (df = 1; 1640)	194.098^{***}	19.464^{***}

Note: *p<0.1; **p<0.05; ***p<0.01
ln-nea-avg: natrual log of average time to enter 'near' dates
ln-far-avg: natrual log of average time to enter 'far' dates
datetype: 1 is treatment (datepicker) and 0 is control (manual entry)
browser + hour covariates not used since extreme values are synthetic
regression coefficeints shown above, standard errors in '()' below
'Adjusted P' is the Holm adjusted p value for 'datetype' coefficient

4. Future Work

Overview of Potential Future Research Questions

Based on the results of this study, we learned a lot about how we would implement future studies on the question as well as additional questions that are interesting. Firstly, we would like to have a setup that allowed us to measure differential attrition if it were to happen. If datepickers cause people to attrite from the study, that would be a very significant finding. That would indicate that they lead to an actual drop off in usage for websites and would counterbalance the benefit to entry speed that we saw among those websites who chose to use it. Secondly, we would like to be able to test a third condition where users are given the option to manually enter a date or select one with a datepicker widget, a hybrid option. We anticipate that this would be the fastest option of the three and would potentially save off differential attrition if it is indeed happening. Thirdly, we would like to test if there is indeed a negative effect of datepickers on overall entry time (or if the result we saw in our exploratory analysis is non-representative). This would be a very interesting finding if it is true and identifying the mechanism that is triggering it could have wide-ranging implications for interface design. Lastly we would like to try to generalize these results. Right now we are dealing with the specific population of MTurk users, in the specific context of an auto rental form. “Near dates” were unconstrained by validation and “far dates” required specific days to be entered. We would like to see if the effect is significantly different if users are required to enter a specific “near date” or a less restricted “far date”. Additionally, we could come up with a variety of “far dates” and generate the image to randomly show a license id date within a certain range to make sure that there is not something special about the specific dates that we picked. Additionally, it might be helpful to randomize the order that fields appear on the form and see if a learning curve actually exists.

Experimental Infrastructure in Future Work

The number one lesson which we could apply to really any randomized study on the internet is to do more on the server side. That would allow us to measure differential attrition, and to ensure proper randomization. This would in turn give us more confidence in our study findings. In this experiment, we did not collect any information until the participant clicked submit. We were not able to collect information on anyone who started the survey but didn’t finish and which treatment group those folks were in. If we changed our web server design, we could have the user request the web page then assign them to treatment or control and give them a session id on the server side. We would be able to compare the session ids sent out to those that came back and identify how many participants attrited and what group they attrited from. Our current page implementation relies on javascript running on the participant’s local machine to assign them randomly to treatment and control. As we mentioned in an above section, we suspect that some common browser extension or configuration was causing the randomization algorithm for those in treatment to be different than those in control. The less we can have the critical functions of the treatment tied to the participants local computer environment the better.

With the variance we are seeing in our main experiment data and previous pilot tests, we may want to switch our current 2x2 factorial setup (near date/far date * date picker/ date typer) to a within subject design where we first create a baseline measurement and then assign the subject to treatment and control. In this way, we can control for individual fixed effects in our regression analysis for more precise estimation of treatment effect.

Dealing with MTurk in Future Experiments

In the future, we would be more suspicious of MTurk and more proactive in ensuring that we get quality responses. We would also build in some more exclusion restrictions into our pre-analysis plan. In multiple instances, we had 2 different users submit the same “sessionid”. The implication is that they were sharing this information among each other somehow. This eats away at the non interference assumption. In an ideal world, we would remove those observations from the data set, however, we did not in this case because we had trouble associating these MTurk IDs with our session ids. The survey participants appeared to be confused about what to enter. It would be better if we had just had participants input their mturk ids at the beginning and end of the survey into our page. Then we would have been able to keep better track of them. Lastly, we had a lot of users not follow the prompt with respect to the near dates. The prompt instructed them to pick 2 dates 10 days apart within the next month. They chose dates in the past or dates years apart. We would implement some sort of data validation for these instances. Similar to the validation that we used for birth dates and Expiration dates.

Generalizability of Interface

This study was limited in terms of generalizability due to the fact that participants only interacted with the page on desktop browsers. In future works, adding a variety of different interaction mediums such as mobile or tablet browsers can give a more generalized analysis of how datepickers affect users speed and overall experience. As of this study we know that datepickers are more favorable, in terms of speed, on the desktop browsers but we don't know how datepickers affect mobile users. If current trends continue, the way most people interact with the internet will be through mobile so understanding the effects of date pickers on this inference would be prime in understanding the most efficient way to have customers interact with one's website.

Other Generalizability Topics

Additionally on this topic of generalizability, our study one covered one example of utilizing a date picker, a rental form. We would like to expand the study to a variety of different examples to see if our outcome is consistent within these examples or if this outcome is only seen in rental type environments. Also the participants of the study can include more of a variety than those obtained through Mturk. As mentioned previously, Mturk may be a decent starting point but people on Mturk are more US-based, college educated and younger than the internet using public as a whole (Pew, 2016) so expanding the study to include a more generalized representation of the public will help give a more concrete outcome for the efficiency of a date picker.

5. Conclusion

Using a date picker for near dates (dates within a month of the current date) appears to be significantly faster than manual date entry. The naive regression (assuming that everything went well and our treatment-control imbalance is due to pure chance) suggests that date pickers lead to a 64% reduction in date entry time for near dates (within a month of the current date) and a 14 % reduction in date entry time for far dates (more than a year away from the current date). Even using an extreme value bounds analysis assuming a differential attrition, there was still a 47% reduction in date entry time for near dates. The conclusions are not as strong for far dates. The extreme value bounds analysis saw a 18% increase in date entry time for those dates. A refinement in experimental controls in future experiments of this type could reduce this uncertainty.

This experiment suffered from an unbalanced assignment between treatment and control in the results that we received back. Our current guess is that this is caused by some idiosyncrasy of a significant subset of the participants' computer environment or browser. However, we can't be confident that this is the case with just the data we've collected. It may be that there is a difference in attrition rates between treatment and control which would be an interesting outcome in itself. We propose tighter experimental controls and a different web-server setup to understand these potential effects in future research of this type.

References

When and why should date pickers be used? (2014, January 6). [Online Forum Post]. Stack Exchange.
https://ux.stackexchange.com/questions/49741/when-and-why-should-date-pickers-be-used?rq=1

Roselli, A. (2020, August 13). *Maybe You Don't Need a Date Picker*. AdrianRoselli.com.
https://adrianroselli.com/2019/07/maybe-you-dont-need-a-date-picker.html

Research in the Crowdsourcing Age, a Case Study. (2016, July 11). Pew Research Center: Internet, Science & Tech.
https://www.pewresearch.org/internet/2016/07/11/research-in-the-crowdsourcing-age-a-case-study/

JS Foundation - js.foundation. (n.d.). *Datepicker | jQuery UI*. Jqueryui.com.
https://jqueryui.com/datepicker/#dropdown-month-year

Arleo, A. (2015, March 16). *Math.random() javascript function *undefined* on Chrome* [Online forum post]. Stack Overflow.
https://stackoverflow.com/questions/29072744/math-random-javascript-function-undefined-on-chrome

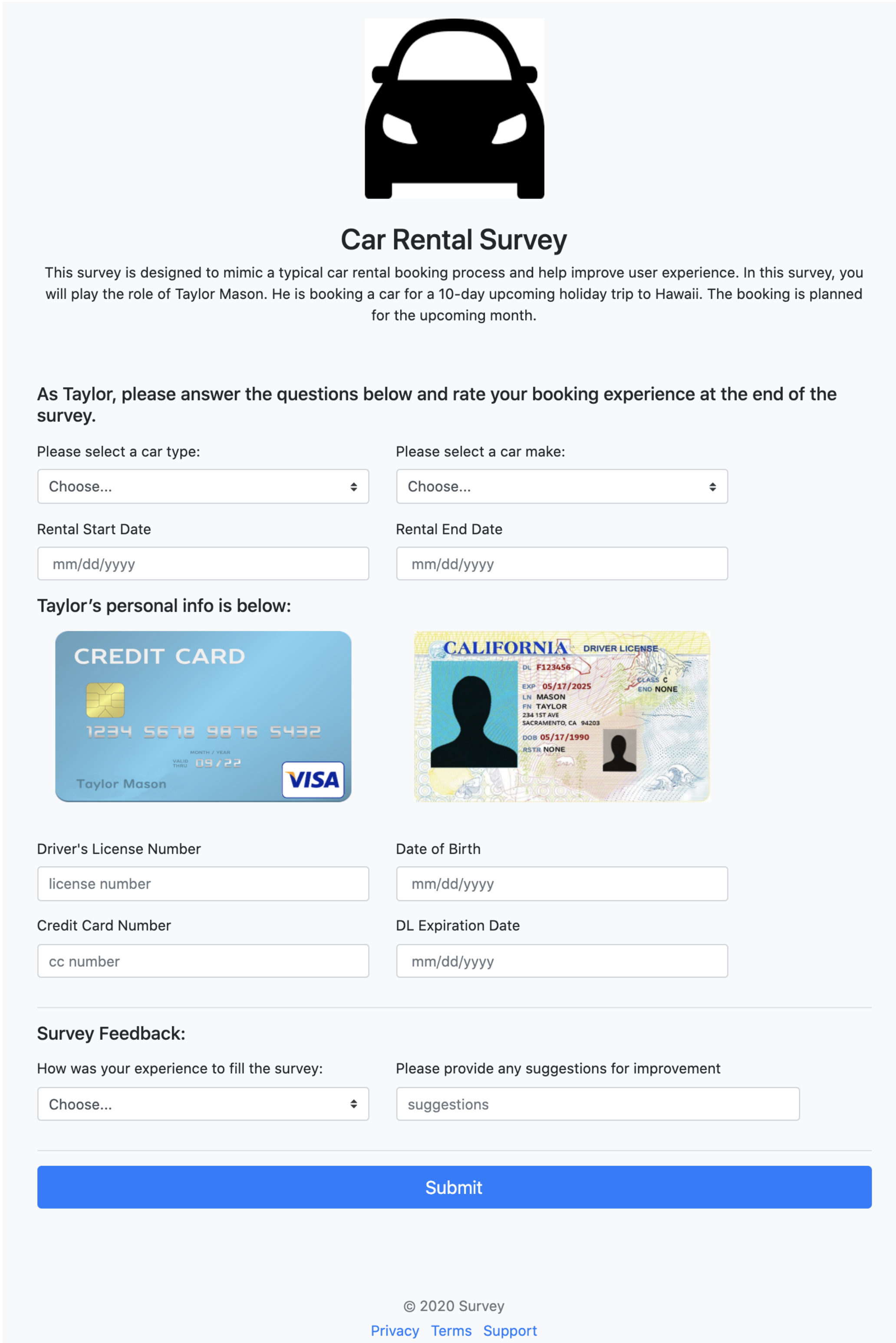
Mirko. (2018, May 27). *Javascript in HTML code: Random numbers are not evenly distributed* [Online Forum Post]. Stack Overflow.
https://stackoverflow.com/questions/50552855/javascript-in-html-code-random-numbers-are-not-evenly-distributed

Ford, C (2018, Aug 17). *Interpreting Log Transformations in a Linear Model*. University of Virginia Library.
https://data.library.virginia.edu/interpreting-log-transformations-in-a-linear-model/#rules_for_interpretation_3

Detailed Data, Analysis, and Reproduction Files

See the survey code repository on github:
https://github.prod.oc.2u.com/kmartin/w241-final-project

Appdx A. Survey Web Form



Car Rental Survey

This survey is designed to mimic a typical car rental booking process and help improve user experience. In this survey, you will play the role of Taylor Mason. He is booking a car for a 10-day upcoming holiday trip to Hawaii. The booking is planned for the upcoming month.

As Taylor, please answer the questions below and rate your booking experience at the end of the survey.

Please select a car type: Choose...

Please select a car make: Choose...

Rental Start Date: mm/dd/yyyy

Rental End Date: mm/dd/yyyy

Taylor's personal info is below:

Driver's License Number: license number

Date of Birth: mm/dd/yyyy

Credit Card Number: cc number

DL Expiration Date: mm/dd/yyyy

Survey Feedback:

How was your experience to fill the survey: Choose...

Please provide any suggestions for improvement: suggestions

Submit

© 2020 Survey

Privacy Terms Support

Figure A.1 Survey Layout

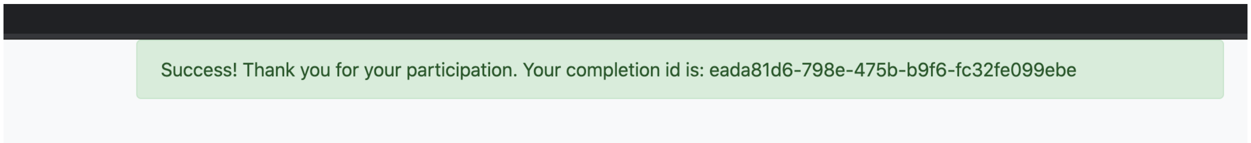


Figure A.2 Successful Submittal Page Response

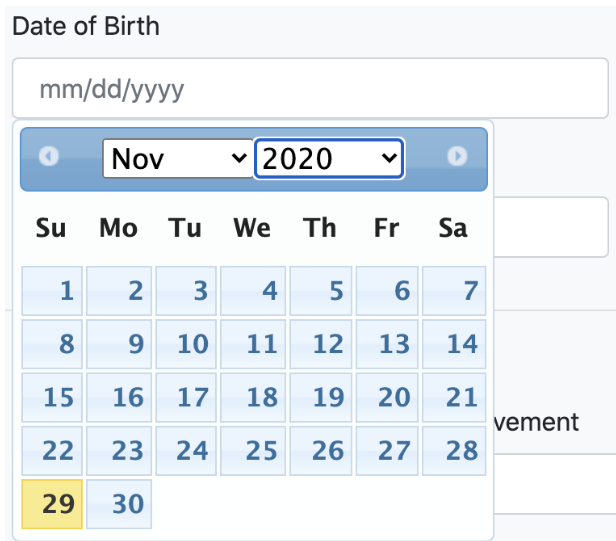


Figure A.3 The JQuery UI Datepicker Used in the Study

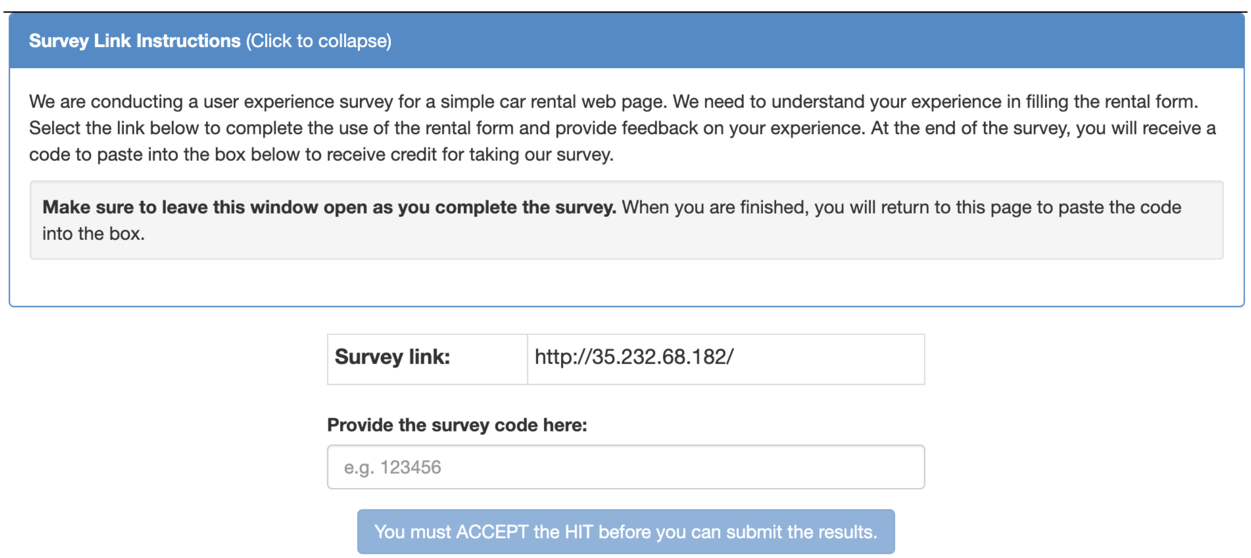


Figure A.4 Advertisement on MTurk

Data Field	Example Data	Description
browser	Chrome 80	Browser type of the user
sessionid	139bb617-0267-4a7a-aa5b-43da6924d55e	SessionId for each survey submission
startDT	2020-11-16T05:35:35.126Z	Survey start date/time
endDT	2020-11-16T05:39:34.686Z	Survey end date/time
datetype	1 **(Date Picker)**	1 = date picker; 0 = date typer
rentalstart_val	11/25/20	Rental start date value selected
rentalstart_et	4891 **(time in ms)**	Time taken to fill rental start date
rentalend_val	12/4/20	Rental end date value selected
rentalend_et	8412 **(time in ms)**	Time taken to fill rental end date
dob_val	5/17/90	Date of birth value selected
dob_et	15936 **(time in ms)**	Time taken to fill date of birth
cc_val	5/17/25	Driver's license expiry date selected
cc_et	44553 **(time in ms)**	Time taken to fill DL expiry date
model	2	Sedan(1), SUV(2), CUV(3), Minivan(4)
brand	4	Mercedes Benz(1), BMW(2), Audi(3), Toyota(4)
license	F123456	Taylor's License # F123456
card	1.23E+15	Credit Card Number: 1234 5678 9876 5432
rating	2	Best(1), Good(2), Average(3), Not that Bad(4), Bad(5)
comments		User comments
errors	-- Invalid License Expiry Date: 05/17/2020. Please fix it!	List of all errors

Figure A.5 Data Layout of the CSV file