Survey Data Quality

Ioannis Andreadis[1], Andreas Andreadis[2],
[1]School of Political Sciences, Aristotle University of Thessaloniki, 54124, Greece
[2]School of Informatics, Aristotle University of Thessaloniki, 54124, Greece

Abstract
In this paper, we introduce the Survey Data Quality R Package, a library of functions that can be used for the assessment of the quality of survey data. Surveys offering rewards may attract careless respondents or even bots (automatic survey-takers) resulting in meaningless, careless, or fraudulent responses, (i.e. responses of lower quality) that we need to identify and probably remove in order to get a final cleaned dataset of high quality. Survey methodology scholars have used various methods to measure the attentiveness of the respondents and the quality of the collected data: item-nonresponse, mid-point responses in Likert-type scale items, straight-lining, the time spent on questionnaire items (speeding), etc. Using the aforementioned response quality indicators we can create an innovative multidimensional estimation of response quality for each completed questionnaire. Using this estimation, we can identify questionnaires that have been submitted by less attentive web survey respondents and we can decide to remove them or not depending on their quality score.

Key Words: data quality, careless responses, data cleaning, web surveys, speeding

1. Introduction

In this paper, we present a series of R functions that can be used to assess the quality of survey data. We always need to check the quality of survey data, because even the most motivated respondents may get tired or lose their interest and motivation and they may respond less carefully to the questions of a self-administered survey (i.e., with minimal cognitive effort). Especially surveys that offer rewards to participants may attract respondents who may not be interested in providing their best response. Instead, their motivation may be to complete the survey as quickly as possible (e.g., when they are after rewards/ incentives). In this case, we have to deal with careless respondents or even bots (automatic survey-takers) resulting in meaningless, careless, or fraudulent responses, (i.e. responses of lower quality) that we need to identify and probably remove in order to get a final cleaned dataset of high quality.

Survey methodology scholars have used many methods to measure response quality. Based on the theory of satisficing we use a series of indicators of lower quality responses to estimate the level of engagement of survey participants in answering the questionnaires: Item-nonresponse, Mid-point responses, Straight-lining and Speeding. Item nonresponse is a problem that in many cases has been related with the length of a web survey. Longer web questionnaires suffer from greater amounts of missing data on individual questions (Galesic, 2006; Galesic & Bosnjak, 2009; Peytchev & Tourangeau, 2005) Choosing a mid-point response in scales is also an indicator of low interest or low effort (Weems &

Onwuegbuzie, 2001). Respondents may choose mid-point responses when they do not process a question with the required effort. In addition, there is evidence that mid-point responses are similar to “No opinion” answers (Blasius & Thiessen, 2001). Non-differentiation in the answers to grid questions, the so-called straight-lining, is another indicator of satisficing behaviour and low response quality (Greszki et al., 2014; Schonlau & Toepoel, 2015) because it is assumed that respondents who straight-line do not pay the required attention to the questions.

From the data quality indicators used in this paper, the readers may be less familiar with the indicator that uses the minimum response time that respondents need to answer a survey question. In the relevant literature the interested reader may find various methods that use time as measure of survey data quality. For instance, some scholars use the time of the whole questionnaire instead of response times per question/page: The main problem in this case is that there are web survey respondents who temporarily stop answering the questionnaire (e.g., they may receive a telephone call, or they may interact with their social media accounts). As a result, a “normal” time for the whole questionnaire may be the sum of very short response times for many items plus one (or more) very long response time(s) due to break(s). Other scholars use percentiles of the time spent on each question or the whole questionnaire. These percentiles are arbitrary selected, and using the same percentile for all web surveys would have unpleasant consequences because the percentage of speeders depends on many factors (e.g., the age and education distribution of the sample, if incentives are offered or not, etc).

To calculate the minimum response time that respondents need to answer a survey question we use the following steps. First, we argue that before answering a survey question, a respondent needs to spend some: i) Time to Read and Comprehend the question and the available response options (TRC), and ii) Time to Select and Report an answer (TSR). Based on a method that was first developed more than ten years ago (Andreadis, 2012, 2014) for simple, single choice questions, each displayed on a separate page and has been recently expanded for matrix question (Andreadis, 2021), we can use the following formula to calculate the minimum response time that has to be spent on a question: MRT=MTSR*NS+MTRC=1.4*NS+NC/40, where NC are the number of characters in the question text and NS is number of sub-questions (in case of a matrix question).

2. Implementation of the data quality indicators

In this section we present the steps we have taken to implement the data quality indicators as function in the R package SurveyDataQuality. In addition to the data quality functions, the R package contains a data file with responses to the ISSP 2020 which we will be using for the example presented in this paper.

2.1 Item nonresponse

While implementing the item nonresponse quality indicator, we have to take into account the following issues:

- Some web survey packages offer the option to include a non-substantive response (such as “I do not know”) and set it as the default response option (i.e., it is recorded when the respondent has not selected any of the response options). In this case a “don’t know” is equivalent to item nonresponse. If “don’t know” has been used as the default response option, we might need to merge them with item nonresponse (e.g. transform “don’t know” to missing).

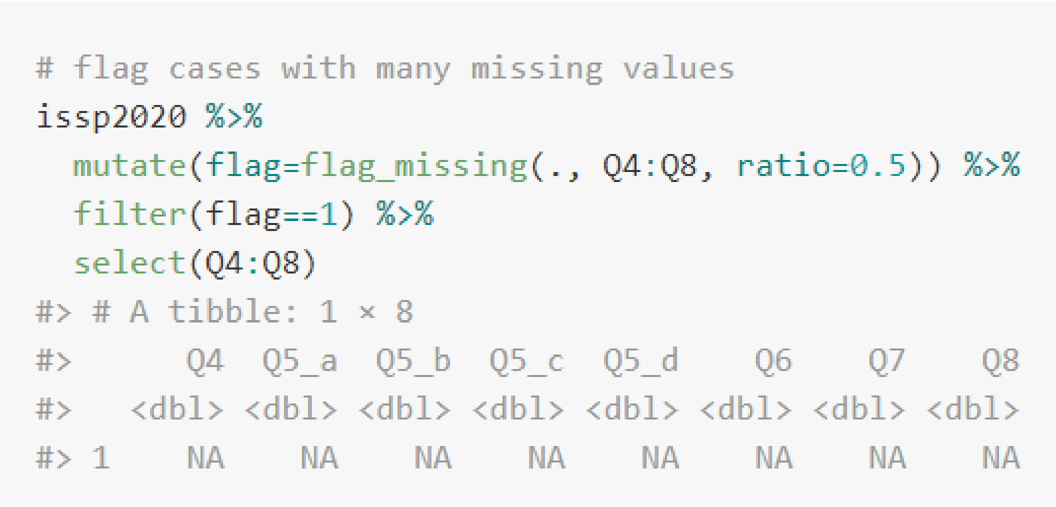
- When counting the number of missing values for each respondent, we should exclude items where a missing value would not indicate a low-quality response. Sometimes there is a good reason for a missing value e.g. conditional questions.
- For the item-nonresponse indicator, we need calculate the ratio of missing answers for each respondent and flag the case if this ratio is greater than (say) 0.33.

Our function flag_missing uses three arguments: data, vars and ratio (if not provided, its default value is 0.33) and it creates a binary vector.

- The argument data can be a tibble
- vars can be any expression that can be used with dplyr::select
- ratio can by any number between 0 and 1

Cases with a ratio of missing values greater than the provided (or the default) ratio, are flagged with ones, while cases with a smaller ratio of missing values have zeros

In Figure 1, we present how the function flag_missing can be used. We start with the ISSP 2020 data; we select a range of variables from Q4 to Q8; we look for missing values in these variables; if a respondent has not answered to more than half of these variables, the case is flagged. As we can see, this method has caught one respondent who has not answered any of these questions.



```
# flag cases with many missing values
issp2020 %>%
  mutate(flag=flag_missing(., Q4:Q8, ratio=0.5)) %>%
  filter(flag==1) %>%
  select(Q4:Q8)
#> # A tibble: 1 × 8
#>      Q4  Q5_a  Q5_b  Q5_c  Q5_d    Q6    Q7    Q8
#>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1    NA    NA    NA    NA    NA    NA    NA    NA
```

Figure 1: R code for flag_missing

2.2 Midpoint responses

While counting the number of midpoints for each respondent, we should use items where a midpoint would indeed indicate a low-quality response e.g. excessive use of the "Neither/nor" option.

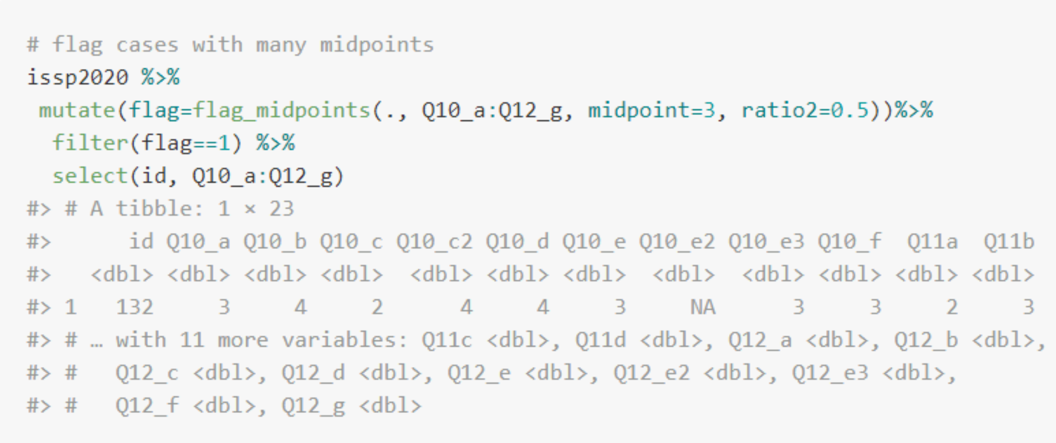
For the midpoint indicator we need to take the following steps:

- Select a list of survey items/variables
- Count how many times each respondent has selected the midpoint
- Calculate the ratio of midpoints for each respondent
- Flag if the ratio is greater than (say) 0.5.

Our function flag_ midpoint uses four arguments: data, vars, midpoint (if not provided, its default value is 3) and ratio (if not provided, its default value is 0.5)

- The argument data can be a tibble (or a compatible data structure)
- Vars can be any expression that can be used with dplyr::select
- Midpoint can be a number and ratio can by any number between 0 and 1

Cases with a ratio of midpoints greater than the provided (or the default) ratio, are flagged with ones, while cases with a smaller ratio of midpoints have zeros. In Figure 2, we show an example of how this function can be used.



```
# flag cases with many midpoints
issp2020 %>%
 mutate(flag=flag_midpoints(., Q10_a:Q12_g, midpoint=3, ratio2=0.5))%>%
  filter(flag==1) %>%
  select(id, Q10_a:Q12_g)
#> # A tibble: 1 × 23
#>      id Q10_a Q10_b Q10_c Q10_c2 Q10_d Q10_e Q10_e2 Q10_e3 Q10_f  Q11a  Q11b
#>   <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>
#> 1   132     3     4     2      4     4     3     NA      3     3     2     3
#> # … with 11 more variables: Q11c <dbl>, Q11d <dbl>, Q12_a <dbl>, Q12_b <dbl>,
#> #   Q12_c <dbl>, Q12_d <dbl>, Q12_e <dbl>, Q12_e2 <dbl>, Q12_e3 <dbl>,
#> #   Q12_f <dbl>, Q12_g <dbl>
```

Figure 2: R code for flag_midpoint

2.3 Straight-lining

To check straight-lining for each respondent, we need to choose items in grid question. We can be more confident about the low quality of the response if at least one of the items is not in the same direction with the other items of the grid.

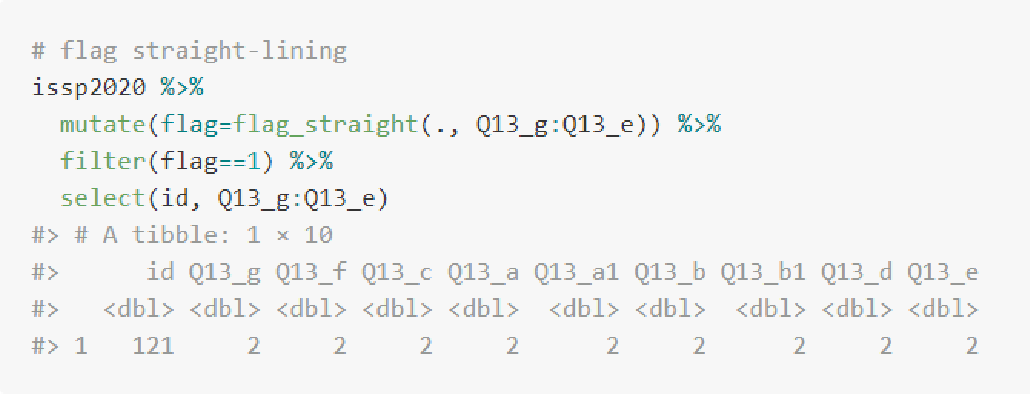
For each grid question we need to take the following steps:

- Count how many times each respondent has selected the same response
- If all responses are the same, we flag the corresponding case

Our function flag_ straight uses two arguments: data, and vars

- The argument data can be a tibble (or a compatible data structure)
- Vars can be any expression that can be used with dplyr::select and it should correspond to the items that have been displayed to respondents as a grid question

Straight-lining cases are flagged with ones, while the value of the rest of the cases is zero. In Figure 3 we show how this function has identified a respondent who has straight-line matrix question Q13 of in ISSP2020.

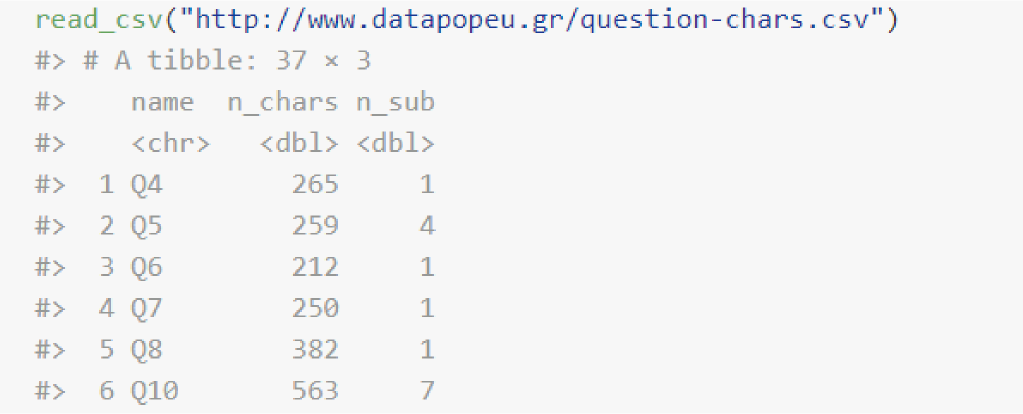


```
# flag straight-lining
issp2020 %>%
  mutate(flag=flag_straight(., Q13_g:Q13_e)) %>%
  filter(flag==1) %>%
  select(id, Q13_g:Q13_e)
#> # A tibble: 1 × 10
#>      id Q13_g Q13_f Q13_c Q13_a Q13_a1 Q13_b Q13_b1 Q13_d Q13_e
#>   <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl>
#> 1   121     2     2     2     2      2     2      2     2     2
```

Figure 3: R code for flag_ straight

2.4 Minimum Response Times

In order to apply this method, we first need: a) a web survey platform that enables us to capture the time spent on the questions and b) a table with the length of the question texts and the number of sub-questions, similar to the table presented in Figure 4.



```
read_csv("http://www.datapopeu.gr/question-chars.csv")
#> # A tibble: 37 × 3
#>    name  n_chars n_sub
#>    <chr>   <dbl> <dbl>
#>  1 Q4        265     1
#>  2 Q5        259     4
#>  3 Q6        212     1
#>  4 Q7        250     1
#>  5 Q8        382     1
#>  6 Q10       563     7
```

Figure 4: Length of the question texts and number of sub-questions

After preparing the table, we can calculate the thresholds for each question, according to the formula:

$$MRT=MTSR*NS+MTRC=1.4*NS+NC/40$$

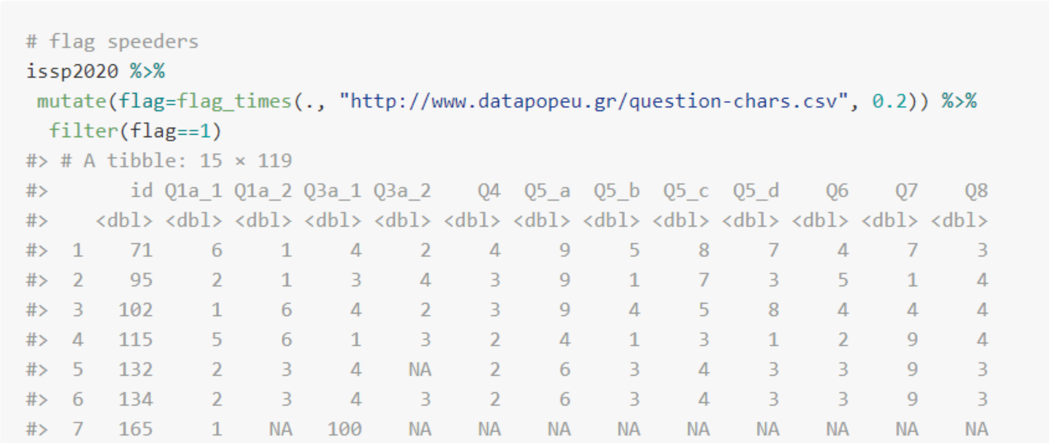
Then, we can bring in the data of the time spent on each question by each respondent and compare these times with the calculated thresholds. If a respondent has spent less time on a question that the minimum time required to understand and respond to the question, we can flag the corresponding case.

Our function flag_time uses three arguments: data, threshold_file and ratio (if not provided, its default value is 0.1) and it creates a binary vector.

- The argument data can be a tibble,
- threshold_file is the name of the file with variable names, number of characters and number of sub questions

- ratio can by any number between 0 and 1

Cases with a ratio of extremely fast responses greater than the provided (or the default) ratio, are flagged with ones, while cases with a smaller ratio have zeros. Figure 5 shows how this method has identified seven speeders

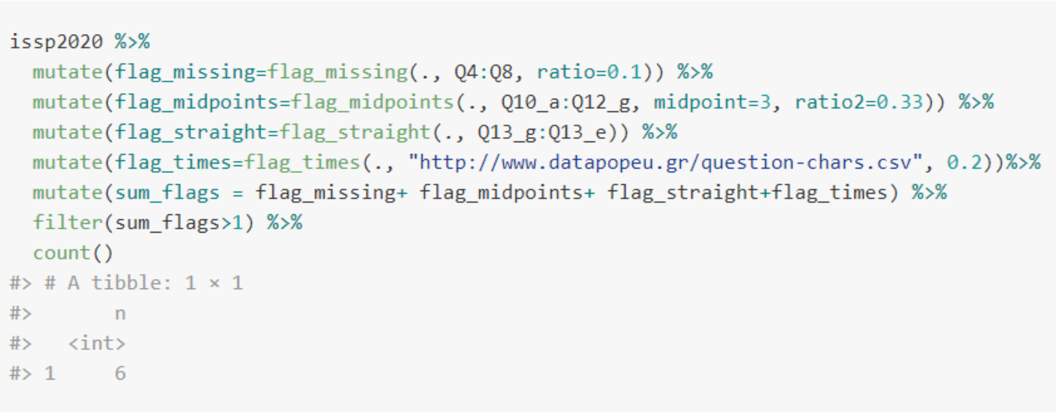


```
# flag speeders
issp2020 %>%
 mutate(flag=flag_times(., "http://www.datapopeu.gr/question-chars.csv", 0.2)) %>%
  filter(flag==1)
#> # A tibble: 15 × 119
#>        id Q1a_1 Q1a_2 Q3a_1 Q3a_2    Q4  Q5_a  Q5_b  Q5_c  Q5_d    Q6    Q7    Q8
#>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#>  1    71     6     1     4     2     4     9     5     8     7     4     7     3
#>  2    95     2     1     3     4     3     9     1     7     3     5     1     4
#>  3   102     1     6     4     2     3     9     4     5     8     4     4     4
#>  4   115     5     6     1     3     2     4     1     3     1     2     9     4
#>  5   132     2     3     4    NA     2     6     3     4     3     3     9     3
#>  6   134     2     3     4     3     2     6     3     4     3     3     9     3
#>  7   165     1    NA   100    NA    NA    NA    NA    NA    NA    NA    NA    NA
```

Figure 5: R code for flag_times

In order to show how the method of minimum response times works, we present the following example from the ISSP 2020 survey. One of the questions together with the answer options included in the questionnaire is the following: Which environmental problem, if any, do you think is the most important for Greece as a whole? [1] “Air pollution”, [2] “Chemicals and pesticides” , [3] “Water shortage”, [4] “Water pollution”, [5] “Nuclear waste”, [6] “Domestic waste disposal”, [7] “Climate change”, [8] “Genetically modified foods”, [9] “Using up our natural resources”, [10] “None of these”, [11] “Can’t choose”. One of the respondents has spent on this question 2.88 seconds only; this means that the respondent has answered the question before reading all the answer options included in it, and the flag_times function was flagged the corresponding case.

Now, we can combine all methods to reveal the respondents who have failed to pass at least two of the four quality checks we have presented. To do this we can combine all the methods, as we show in Figure 6.



```
issp2020 %>%
  mutate(flag_missing=flag_missing(., Q4:Q8, ratio=0.1)) %>%
  mutate(flag_midpoints=flag_midpoints(., Q10_a:Q12_g, midpoint=3, ratio2=0.33)) %>%
  mutate(flag_straight=flag_straight(., Q13_g:Q13_e)) %>%
  mutate(flag_times=flag_times(., "http://www.datapopeu.gr/question-chars.csv", 0.2))%>%
  mutate(sum_flags = flag_missing+ flag_midpoints+ flag_straight+flag_times) %>%
  filter(sum_flags>1) %>%
  count()
#> # A tibble: 1 × 1
#>       n
#>   <int>
#> 1     6
```

Figure 6: Combining all methods

3. Discussion

This paper provides a series of functions that can be used to flag responses of lower quality. We have used four different indicators. First, we use item-nonresponse (skipping) and we calculate the ratio of missing answers for each respondent. Then we use mid-point responses in Likert-type scale items: (e.g., "neither/nor") because respondents may choose mid-point responses when they do not process a question with the required effort and again we calculate the ratio of mid-point responses. We also use non-differentiation in grid questions (straight-lining). Finally, we use the minimum time needed to read and answer an attitudinal question given the length of the question text. We have included these functions in the R package "Survey Data Quality" that is being developed at: https://github.com/andreasa13/SurveyDataQuality and can be downloaded and used by everyone interested in using these functions to assess the quality of survey data.

Acknowledgements

The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 3572).

References

Andreadis, I. (2012). To clean or not to clean? Improving the quality of VAA data. *XXII World Congress of Political Science (IPSA)*. http://ikee.lib.auth.gr/record/132697/files/IPSA-2012.pdf

Andreadis, I. (2014). Data Quality and Data Cleaning. In D. Garzia & S. Marschall (Eds.), *Matching Voters with Parties and Candidates. Voting Advice Applications in Comparative Perspective* (pp. 79–91). ECPR Press. http://www.polres.gr/en/sites/default/files/VAA-Book-Ch6.pdf

Andreadis, I. (2021). Web Survey Response Times What to Do and What Not to Do. *Proceedings of the 2021 Joint Statistical Meetings*. 2021 Joint Statistical Meetings, Alexandria, VA.

Blasius, J., & Thiessen, V. (2001). Methodological Artifacts in Measures of Political Efficacy and Trust: A Multiple Correspondence Analysis. *Political Analysis*, *9*(01), 1–20. https://doi.org/10.1093/oxfordjournals.pan.a004862

Galesic, M. (2006). Dropouts on the Web: Effects of Interest and Burden Experienced During an Online Survey. *Journal of Official Statistics*, *22*(2), 313–328.

Galesic, M., & Bosnjak, M. (2009). Effects of Questionnaire Length on Participation and Indicators of Response Quality in a Web Survey. *Public Opinion Quarterly*, *73*(2), 349–360. https://doi.org/10.1093/poq/nfp031

Greszki, R., Meyer, M., & Schoen, H. (2014). The impact of speeding on data quality in nonprobability and freshly recruited probability-based online panels. In M. Callegaro, R. Baker, J. Bethlehem, A. S. Göritz, J. A. Krosnick, & P. J. Lavrakas (Eds.), *Online Panel Research: A Data Quality Perspective* (pp. 238–262). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118763520.ch11

Peytchev, A., & Tourangeau, R. (2005). Causes of Context Effects: How questionnaire layout induces measurement error. *Presented at the American Association for Public Opinion Research (AAPOR) 60th Annual Conference.*

Schonlau, M., & Toepoel, V. (2015). Straightlining in Web survey panels over time. *Survey Research Methods*, *9*(2). https://doi.org/10.18148/SRM/2015.V9I2.6128

Weems, G. H., & Onwuegbuzie, A. J. (2001). The Impact of Midpoint Responses and Reverse Coding on Survey Data. *Measurement and Evaluation in Counseling and Development*, *34*(3), 166–176. https://doi.org/10.1080/07481756.2002.12069033