

Online Appendix to:

How to get better survey data more efficiently

Mollie J. Cohen* and Zach Warner†

January 1, 2020

*Assistant Professor, University of Georgia. email: mj.cohen@uga.edu, web: <http://www.molliecohen.com>.

†Postdoctoral Research Fellow, Cardiff University. email: WarnerZ@cardiff.ac.uk, web: <http://www.zachwarner.net>.

Contents

A.1	Studies of quality control procedures	3
A.2	Designing an efficient quality control system	11
A.2.1	Case 1	11
A.2.2	Case 2	12
A.3	Quality control procedures	13
A.4	Models studied	29
A.5	Models with 30 versus 141 quality control procedures	31

A.1 Studies of quality control procedures

The following table describes the key contributions of a broad selection of papers that introduce quality control strategies. Since this literature is vast and spread across a number of disciplines (and languages), this Table is not exhaustive, and papers that describe or apply existing checks are omitted. Literature reviews are also excluded here, but see Bredl, Storfinger, and Menold (2011), Mneimneh et al. (2018), and Robbins (2018) for examples.

A check in the “test” column indicates that a study explicitly assesses a method’s effectiveness at detecting or deterring low quality interviews, rather than simply describing it (or using it to examine whether a dataset contains any fabricated or low-quality data). Many of these tests are informal, and only rarely are checks compared against one another for their efficiency in uncovering low-quality interviews.

Entries marked as *post hoc* were quality control procedures applied to existing surveys, e.g., when detecting anomalies in response distributions years after a survey is completed (as in Blasius 2018).

At the end of the entry, we provide the data source that allows us to examine each procedure or problem. For example, while the AmericasBarometer does not recontact participants, the auditor checks serve the same function: because auditors listen to subsets of interviews, they are able to verify that the respondent exists and participated in the survey, fulfilling the same purpose as a callback.

Table A1: Studies of quality control procedures

Study	Procedure(s) or problem(s) studied	Test
Bennett (1948)	Monetary incentives and feedback provided to enumerators in US market research surveys. Also used cards to verify interviews through recontacts. [Auditor checks]	
Bhuiyan and Lackie (2016)	GPS captures, audio recording, and interviewer retraining used to improve data quality in real time in three villages in Bangladesh census. [Automated GPS captures and auditor checks]	
Biemer and Stokes (1989)	Callbacks used to verify interviews based on enumerator characteristics such as experience and urban locations, based on the 1982 US census. [Auditor checks, cluster scripts, and enumerator IDs]	
Blasius (2018)	Statistical checks of response frequencies to detect straightlining, duplication, and abnormal response patterns, in PISA 2012 and PIAAC surveys (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Blasius and Thiessen (2012)	Statistical checks of response frequencies to detect straightlining, duplication, and abnormal response patterns in multiple surveys (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Blasius and Thiessen (2013)	Statistical checks of response frequencies to detect straightlining, duplication, and abnormal response patterns in the German General Social Survey 2008 (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	

Table A1 (continued): Studies of quality control procedures

Study	Description	Tests?
Blasius and Thiessen (2015)	Statistical checks of response frequencies to detect straightlining, duplication, and abnormal response patterns in the PISA 2009 (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Blasius and Thiessen (2018)	Statistical checks of response frequencies to detect straightlining, duplication, and abnormal response patterns in the European Social Survey 2010 (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Bredl, Winker, and Kötschau (2008)	Cluster analysis of non-response ratios, extreme responses, “other” responses, and Benford’s Law, using data from an unnamed non-OECD country (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	✓
Bushery et al. (1999)	Interview timers to focus re-interviews in the NHIS. [Scripts for interview timing and enumerator participation rates]	
Case (1971)	Recontacts to verify interviews took place in US market surveys. [Auditor checks]	
Cho, Eltinge, and Swanson (2003)	Benford’s Law to detect curbstoning in the Consumer Expenditure Survey (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Crespi (1945)	Recontacts to verify interviews in US public opinion surveys (suggested, not implemented, <i>post hoc</i>). [Auditor checks]	
De Haas and Winker (2014)	Cluster analysis of response frequencies to detect straightlining, duplication, and abnormal response patterns in simulated data. [Scripts including Percentmatch and completion percent]	✓

Table A1 (continued): Studies of quality control procedures

Study	Description	Tests?
De Haas and Winker (2016)	Cluster analysis of response frequencies to detect straightlining, duplication, and abnormal response patterns in simulated data. [Scripts including Percentmatch and completion percent]	✓
Evans (1961)	Callbacks used to verify interviews in a small US survey. [Auditor checks]	
Finn and Ranchhod (2017)	Benford's Law, GPS captures, statistical patterns in response distributions, and enumerators' interview success rates, all examined on enumerators known to have cheated in a South African survey (<i>post hoc</i>). [GPS captures and scripts for enumerator participation rates, Percentmatch, completion percent, and others]	
Gomila et al. (2017)	Audio recording to detect fabrication in a survey corruption reporting among southern Nigerians. [Auditor checks]	✓
Hicks et al. (2010)	Audio recording to detect fabrication in a 2007 US health survey. [Auditor checks]	
Hood and Bushery (1997)	Metadata such as interview length and interviews per day by enumerator, as well as geographic characteristics, to focus re-interview efforts (in real time) in the US census. [Scripts for enumerator participation rates, completion percent, and cluster characteristics]	
Hülser (2013)	Statistical checks of response frequencies for duplication and abnormal response patterns, as well as intra-interview consistency, in German market research. [Scripts for completion percent and Percentmatch, and flags for intra-interview consistency]	

Table A1 (continued): Studies of quality control procedures

Study	Description	Tests?
Judge and Schechter (2009)	Benford's Law to detect fabrication in national surveys which are widely used in Economics, including of Mexico, Paraguay, and Peru (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Kemper and Menold (2014)	Statistical checks of response frequencies for abnormal response patterns in a simulated sample (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Krejsa, Davis, and Hill (1999)	Metadata such as interview length and interviews per day by enumerator, to focus re-interviews in the US census (<i>post hoc</i>). [Scripts including enumerator participation rates and interview timings, as well as auditor checks]	✓
Kuriakose and Robbins (2016)	Duplicates and near-duplicates via Percentmatch in a number of large surveys and simulated data (<i>post hoc</i>). [Scripts for Percentmatch.]	
Landrock (2017)	Statistical checks of abnormal response patterns in simulated data (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Menold and Opper (2013)	Statistical checks of abnormal response patterns in simulated data (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Menold et al. (2013)	Cluster analysis of response frequencies to detect straightlining, duplication, and abnormal response patterns in simulated data. [Scripts including Percentmatch, completion percent, and cluster characteristics]	

Table A1 (continued): Studies of quality control procedures

Study	Description	Tests?
Menold and Kemper (2014)	Statistical checks of abnormal response patterns in simulated data (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	✓
Mitchell, Fahrney, and Strobl (2009)	Audio recording, callbacks, and third-party observation to verify interviews in the SCFL. [Auditor checks]	
Murphy et al. (2004)	Statistical checks of abnormal response patterns, item-nonresponse, and response timings in the US NSDUH survey. [Scripts including Percentmatch, completion percent, and interview timing]	
Philipson and Malani (1999)	Monetary incentives to reduce erroneous entry among a sample of US physicians. [Auditor checks]	
Porras and English (2004)	Benford's Law and statistical checks of abnormal response patterns in a large-scale US health study (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Schäfer et al. (2005)	Benford's Law and statistical checks of abnormal response patterns among the German SOEP (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Schräpler and Wagner (2005)	Recontacts, intra-interview consistency, and patterns in enumerator characteristics to detect wholesale interview fabrication in the German SOEP (<i>post hoc</i>). [Auditor checks, flags for intra-interview consistency, and enumerator IDs]	
Schreiner, Pennie, and Newbrough (1988)	Recontacts to detect interview fabrication in the US census. [Auditor checks]	

Table A1 (continued): Studies of quality control procedures

Study	Description	Tests?
Simmons et al. (2016)	Duplicates and near-duplicates via Percentmatch in a number of surveys, including the 2012 ANES, and simulated data (<i>post hoc</i>). [Scripts for Percentmatch]	
Slomczynski, Powalko, and Krauze (2017)	Duplication checks in 1,721 national surveys across 142 countries (<i>post hoc</i>). [Scripts for Percentmatch]	
Stokes and Jones (1989)	Contextual data and a stratified sample of re-interviews to detect wholesale fabrication in the US census (<i>post hoc</i>). [Scripts for cluster characteristics and auditor checks]	
Storfinger and Winker (2011)	Cluster analysis of of response frequencies to detect straightlining, duplication, and abnormal response patterns in an unnamed non-OECD country. [Scripts including Percentmatch, completion percent, and cluster characteristics]	✓
Storfinger and Winker (2013)	Cluster analysis of of response frequencies to detect straightlining, duplication, and abnormal response patterns in simulated data. [Scripts including Percentmatch, completion percent, and cluster characteristics]	✓
Swanson, Cho, and Eltinge (2003)	Benford's Law to detect fabrication in US Consumer Expenditure Survey data (<i>post hoc</i>). [Scripts including Percentmatch and completion percent]	
Turner et al. (2002)	High completion rates, callbacks, and suspicious response patterns to identify fabrication in a Baltimore healthcare survey (<i>post hoc</i>). [Auditor checks and scripts including Percentmatch and completion percent]	

Table A1 (continued): Studies of quality control procedures

Study	Description	Tests?
Vanden Eng et al. (2007)	GPS data captured in real time, from health surveys in Togo and Niger. [GPS captures and auditor checks]	
Waller (2013)	Interviews with experienced surveyors to detect strategies used to falsify interviews, including use of stereotypes to complete surveys, in Jamaica (<i>post hoc</i>). [Auditor checks and scripts for Percentmatch and enumerator participation rates]	

A.2 Designing an efficient quality control system

As discussed in the main text, we do not indicate which quality control procedures are most cost-effective, since costs vary so widely by context and by survey research organization. The “Efficient-Quality-Control-Worksheet” spreadsheet provides a worksheet for researchers to enter their own expected costs and design a quality control workflow accordingly. This section provides two hypothetical examples for how to do so. In each case, we only use five procedures for the sake of parsimony.

A.2.1 Case 1

Suppose a graduate student is designing a survey in Ghana for her dissertation fieldwork. She is an experienced programmer but has not worked on a survey before, and so sets herself a budget of \$1,000 for quality control. Due to linguistic barriers, she expects to have to hire locals to audit interviews, but will be able to code R scripts and real-time flags herself. She also already has a secure data storage server to hold interviews, related attachments, and metadata, and so does not need to factor these into her costs. She therefore uses the following formula to compute her expected costs for each procedure:

$$\text{Expected cost (A)} = (\text{Auditor hourly wage}) \times (\text{Expected hours auditing}), \text{ and} \quad (\text{A1})$$

$$\text{Expected cost (S, F)} = (\text{Her hourly wage}) \times (\text{Expected hours coding}), \quad (\text{A2})$$

where “A” indicates auditor checks, “S” R scripts, and “F” real-time flags that she will need to manually implement in the survey code.

After discussing with potential auditors and considering the complexity of coding these scripts, she identifies her expected costs for each of the quality control procedures. She follows the instructions in the worksheet, giving the result in Figure A1.

Although computing the completion percentage of each interview is the most informative procedure, she knows it will take nearly twice as long to code as a simple interview duration timer. As a result, her expected cost for these procedures are

	A	B	C	D	E
1	quality_control_procedure	marginal_benefit	marginal_cost	mb/mc	total_expenditure
2	netduration_script	49.78210274	60	0.829701712	60
3	completion_pc_script	61.02432027	100	0.610243203	160
4	consent_notread_auditor	48.1621383	300	0.160540461	460
5	cluster_too_big_script	56.8455304	500	0.113691106	960
6	question_oneskip_auditor	45.95325181	1000	0.045953252	1960

Figure A1: Cost-efficiency calculations, Case 1.

\$100 and \$60, respectively, making the duration script a more cost-effective tool to implement. Similarly, she expects that checking for interviewers skipping questions will take over three times as long as having auditors check for enumerators not reading the consent form. Thus, it is far more cost-effective for her to check participants' consent than enumerators skipping questions. Finally, since her budget is just \$1,000, she can only implement the first four procedures, and must forego checking for skipped questions.

A.2.2 Case 2

Suppose now a professor is designing a small survey to be implemented in Mexico. She and her two research assistants all speak Spanish fluently, and they have previously worked together on similar surveys. However, neither she nor her assistants have experience programming, and so she would have to hire a contractor to implement real-time flags and scripts. Like the student in Case 1, she does not need to purchase any data storage or transport devices. Her budget is \$4,000.

She uses equations similar to Equations A1-A2 to estimate costs, but inserts her contractor's hourly wage for her own in coding scripts and flags. After estimating these costs and following the instructions in the worksheet, she arrives at the results in Figure A2.

Because she and her assistants are experienced auditors—and because there are not too many interviews to audit—she expects that auditor checks will be cheap for her to implement. Checking for consent being read and enumerators' skipping or interpreting questions turn out to be very cost-effective quality control tools for her. On the other hand, her contractor quoted her \$2,000 to implement Percentmatch, so although it is more informative than checking for auditors interpreting questions, it is a less efficient use of her money. Nevertheless, her budget allows for all of these checks to be implemented and leaves room for an additional \$1,450 in spending. Given the results presented in the main text, she can expect that these additional checks will yield substantial improvements in separating high- and low-quality interviews, and so we recommend she invest in more quality control.

	A	B	C	D	E	
1	quality_control_procedure	marginal_benefit	marginal_cost	mb/mc	total_expenditure	
2	consent_notread_auditor	48.1621383		50	0.963242766	50
3	question_oneskip_auditor	45.95325181		100	0.459532518	150
4	question_oneinterpreted_auditor	24.79841777		100	0.247984178	250
5	netduration_script	49.78210274		300	0.165940342	550
6	top_pc_match_script	31.39836126		2000	0.015699181	2550

Figure A2: Cost-efficiency calculations, Case 2.

A.3 Quality control procedures

The following list gives each quality control procedure name, description, and coding rules.

Auditor problems, missingness:

1. `no_respondent_auditor`: describes whether an auditor noticed that there was no respondent (from silent audio capture). Coded as 1 if no respondent heard and 0 otherwise.
2. `no_gps_auditor`: describes whether an auditor noticed that there was no GPS data available for the interview (from silent GPS capture). Coded as 1 if no GPS data available and 0 otherwise.
3. `no_camera_auditor`: describes whether an auditor noticed that there was no image captures uploaded with an interview (from various image captures). Coded as 1 if no images available and 0 otherwise.

Auditor problems, consent form:

4. `consent_wrongtime_auditor`: describes whether an auditor noticed the enumerator mis-stating the expected duration of the survey when asking the respondent for informed consent (from silent audio capture). Coded as 1 if the stated time is incorrect and 0 otherwise.
5. `consent_notgiven_auditor`: describes whether an enumerator began the survey despite the respondent not giving informed consent (from silent audio capture). Coded as 1 if informed consent was not given and 0 otherwise.
6. `consent_notread_auditor`: describes whether an enumerator began the survey without reading the consent form (from silent audio capture). Coded as 1 if the consent form was not read and 0 otherwise.
7. `consent_partread_auditor`: describes whether an enumerator began the survey after only partially reading the consent form (from silent audio capture). Coded as 1 if the consent form was partially read and 0 otherwise.
8. `consent_misread_auditor`: describes whether an auditor misread the consent form (from silent audio capture). Coded as 1 if the consent form was misread and 0 otherwise.

Auditor problems, improper people involved:

9. friend_interviewed_auditor: describes whether an auditor noticed that the respondent appeared to know the enumerator personally (from silent audio capture). Coded as 1 if the enumerator knew the respondent and 0 otherwise.
10. wrong_picture_auditor: describes whether an auditor noticed that the front-facing image uploaded with an interview captured a face that was not the enumerator's (from image captures). Coded as 1 if the face photographed was not that of the enumerator and 0 otherwise.
11. wrong_voice_auditor: describes whether an auditor noticed that the voice conducting the interview was not that of the enumerator (from silent audio capture). Coded as 1 if the enumerator's voice was incorrect and 0 otherwise.
12. self_interview_auditor: describes whether an auditor noticed that the enumerator interviewed him- or herself instead of a respondent (from silent audio capture). Coded as 1 if the enumerator interviewed him- or herself and 0 otherwise.
13. interviewer_interview_auditor: describes whether an auditor noticed that an enumerator interviewed another enumerator instead of a respondent (from silent audio capture). Coded as 1 if the enumerator interviewed another enumerator and 0 otherwise.

Auditor problems, suspicious behavior:

14. interviewer_abandon_auditor: describes whether an auditor noticed that the enumerator abandoned the interview for any reason (from audio and image captures, as well as the interview log). Coded as 1 if the enumerator abandoned the interview and 0 otherwise.
15. interviewee_abandon_auditor: describes whether an auditor noticed that the respondent abandoned the interview for any reason (from audio and image captures, as well as the interview log). Coded as 1 if the interviewee abandoned the interview and 0 otherwise.
16. wrong_location_auditor: describes whether an auditor noticed that the interview took place in a proscribed location, such as a supermarket, gas station, or university (from audio, image, and GPS captures, as well as contextual clues). Coded as 1 if the interview was conducted in a proscribed location and 0 otherwise.

17. `location_moved_auditor`: describes whether an auditor noticed that the interview began and ended in different locations (from audio, image, and GPS captures as well as contextual clues). Coded as 1 if the starting and ending locations were not the same and 0 otherwise.
18. `attempts_exhausted_auditor`: describes whether the auditor noticed that the interview uploaded had exhausted all attempts to interview without successfully completing an interview (from the logs). Coded as 1 if all attempts were exhausted and no survey questions were answered and 0 otherwise.
19. `tooshort_toolong_auditor`: describes whether the auditor noticed (from the log) that the interview was completed too quickly, or took too long to complete, based on country-specific thresholds (but typically less than 25 minutes or more than 2 hours, respectively). Coded as 1 if the interview was either too short or too long and 0 otherwise.
20. `airplane_mode_auditor`: describes whether the auditor noticed that the CAPI device was switched to airplane mode to prevent a network connection (from the log). Coded as 1 if the device was in airplane mode and 0 otherwise.

Auditor problems, misreading:

21. `interviewer_gaveopinion_auditor`: describes whether an auditor noticed that the enumerator gave his or her opinion on a survey question or answer to the respondent (from a silent audio capture). Coded as 1 if the enumerator gave his or her opinion and 0 otherwise.
22. `question_oneinterpreted_auditor`: describes whether an auditor noticed that the enumerator interpreted one survey question for the respondent (from a silent audio capture). Coded as 1 if exactly one question was interpreted and 0 otherwise.
23. `question_twointerpreted_auditor`: describes whether an auditor noticed that the enumerator interpreted two survey questions for the respondent, from a silent audio capture (from a silent audio capture). Coded as 1 if exactly two questions were interpreted and 0 otherwise.
24. `question_manyinterpreted_auditor`: This variable describes whether an auditor noticed that the enumerator interpreted three or more survey questions for the respondent (from a silent audio capture). Coded as 1 if three or more questions were interpreted and 0 otherwise.

25. `question_oneskip_auditor`: describes whether an auditor noticed that the enumerator skipped a survey question (from a silent audio capture). Coded as 1 if exactly one question was skipped and 0 otherwise.
26. `question_twoskip_auditor`: describes whether an auditor noticed that the enumerator skipped two survey questions (from a silent audio capture). Coded as 1 if exactly two questions were skipped and 0 otherwise.
27. `question_manyskip_auditor`: describes whether an auditor noticed that the enumerator skipped three or more survey questions (from a silent audio capture). Coded as 1 if three or more questions were skipped and 0 otherwise.
28. `question_onemisread_auditor`: describes whether an auditor noticed that the enumerator misread a survey question (from a silent audio capture). Coded as 1 if exactly one question was misread and 0 otherwise.
29. `question_twomisread_auditor`: describes whether an auditor noticed that the enumerator misread two survey questions (from a silent audio capture). Coded as 1 if exactly two questions were misread and 0 otherwise.
30. `question_manymisread_auditor`: describes whether an auditor noticed that the enumerator misread three or more survey questions (from a silent audio capture). Coded as 1 if three or more questions were misread and 0 otherwise.
31. `question_onefast_auditor`: describes whether an auditor noticed that the enumerator read a survey question too fast (from a silent audio capture). Coded as 1 if exactly one question was read too fast and 0 otherwise.
32. `question_twofast_auditor`: describes whether an auditor noticed that the enumerator read two survey questions too fast (from a silent audio capture). Coded as 1 if exactly two questions were read too fast and 0 otherwise.
33. `question_manyfast_auditor`: describes whether an auditor noticed that the enumerator read three or more survey questions too fast (from a silent audio capture). Coded as 1 if three or more questions were read too fast and 0 otherwise.

Auditor problems, other:

34. `other_qac_abandoned_security_auditor`: describes whether the auditor abandoned the interview due to security concerns for him- or herself. Coded as 1 if the auditor abandoned the interview due to security concerns and 0 otherwise.

35. `other_qac_potential_fraud_auditor`: describes whether the auditor noticed anything in reviewing the interview that suggested potential enumerator fraud. Coded as 1 if the auditor suspected fraud and 0 otherwise.
36. `other_qac_potential_fraud_loc_auditor`: describes whether the auditor noticed anything in reviewing the interview that suggested potential enumerator fraud based on the location of the interview. Coded as 1 if the auditor suspected fraud due to location and 0 otherwise.
37. `other_qac_potential_fraud_quota_auditor`: describes whether the auditor observed anything in reviewing the interview that suggested potential enumerator fraud due to demographic quota issues. Coded as 1 if the auditor suspected fraud for quota reasons and 0 otherwise.
38. `other_qac_potential_fraud_many_ppl_auditor`: describes whether the auditor noticed anything in reviewing the interview that suggested potential enumerator fraud due to multiple apparent respondents in the same interview. Coded as 1 if the auditor suspected fraud for this reason and 0 otherwise.
39. `other_qac_potential_fraud_other_auditor`: describes whether the auditor noticed anything in reviewing the interview that suggested potential enumerator fraud for reasons not otherwise noted. Coded as 1 if the auditor suspected fraud and 0 otherwise.
40. `other_qac_third_party_influence_auditor`: describes whether the auditor noticed anything in reviewing the interview that suggested the respondent was unduly influenced by a third party present during the interview. Coded as 1 if the auditor suspected such influence and 0 otherwise.
41. `other_qac_incomplete_read_auditor`: describes whether the auditor noticed any place where the enumerator did not completely read an item or response (not noted elsewhere). Coded as 1 if the auditor noticed such incompletes and 0 otherwise.
42. `other_qac_duration_problem_auditor`: describes whether the auditor noticed any other problem with the duration of the interview (not noted elsewhere). Coded as 1 if the auditor noted such a problem and 0 otherwise.
43. `other_qac_read_wrong_inc_auditor`: describes whether the auditor noticed the enumerator reading wrong response options for a question (not noted elsewhere). Coded as 1 if the enumerator misread response options and 0 otherwise.

44. other_qac_missing_attachments_auditor: describes whether the auditor noticed any other uploads missing (not noted elsewhere). Coded as 1 if the attachments were missing and 0 otherwise.
45. other_qac_record_wrong_ans_auditor: describes whether the auditor mis-recorded an answer given by the respondent (from a silent audio capture). Coded as 1 if the auditor mis-recorded a response and 0 otherwise.
46. other_qac_technical_error_auditor: describes whether the auditor noticed a the enumerator make a technological error during the interview. Coded as 1 if the enumerator committed a technological error and 0 otherwise.
47. other_qac_no_consent_heard_auditor: describes whether the auditor could not hear or understand whether the respondent gave informed consent (from a silent audio capture) Coded as 1 if the consent could not be heard or understood and 0 otherwise.
48. other_qac_ambient_noise_loud_auditor: describes whether the auditor noticed loud ambient noises making it difficult to hear the voices of the enumerator and/or respondent (from a silent audio capture). Coded as 1 if ambient noise was excessive and 0 otherwise.
49. other_qac_outside_geo_area_auditor: describes whether the auditor noticed that the enumerator conducted the interview outside of the assigned geofence for that interview (from GPS captures). Coded as 1 if the interview was outside the geofence and 0 otherwise.
50. other_qac_tech_problems_auditor: describes whether the auditor noticed a technological problem during the interview. Coded as 1 if there was a technological problem and 0 otherwise.
51. other_qac_one_geocerca_auditor: describes whether the auditor noticed that the interview was conducted in the wrong geofence assigned to that interview. Coded as 1 if the interview was conducted in the wrong geofence and 0 otherwise.
52. other_qac_interpreter_used_auditor: describes whether the auditor noticed that the enumerator relied on an intermediary to translate communications to or from the respondent. Coded as 1 if the respondent used an interpreter and 0 otherwise.

53. other_qac_age_quota_unmet_auditor: describes whether the auditor noticed that the respondent did not meet the age quota. Coded as 1 if the respondent did not meet the age quota and 0 otherwise.
54. other_qac_wrong_gender_quota_auditor: describes whether the auditor noticed that the respondent did not meet the gender quota. Coded as 1 if the respondent did not meet the gender quota and 0 otherwise.
55. other_qac_cant_speak_lang_auditor: describes whether the auditor noticed that the respondent could not speak the language of enumeration. Coded as 1 if the respondent did not speak the language of the survey and 0 otherwise.
56. other_qac_resp_read_questionnaire_auditor: describes whether the auditor noticed that the respondent read the questionnaire directly. Coded as 1 if the respondent read the questionnaire and 0 otherwise.
57. other_qac_quota_problem_auditor: describes whether the auditor noticed a problem with the interview with respect to fulfilling demographic quotas. Coded as 1 if there was such a problem and 0 otherwise.
58. other_qac_comment_error_auditor: describes whether the auditor noticed a miscellaneous enumerator error (not otherwise noted) Coded as 1 if there was such an error and 0 otherwise.
59. other_qac_bad_etiquette_auditor: describes whether the auditor noticed the enumerator displaying bad etiquette, such as answering his or her phone. Coded as 1 if there the enumerator displayed bad etiquette and 0 otherwise.
60. other_qac_consent_error_auditor: describes whether the auditor noticed a miscellaneous problem with the consent form (not otherwise noted). Coded as 1 if there was such a consent error and 0 otherwise.
61. other_qac_consent_read_wrong_auditor: describes whether the auditor noticed the enumerator make a miscellaneous error reading the consent form (not otherwise noted). Coded as 1 if such an error was made and 0 otherwise.
62. other_qac_consent_request_id_auditor: describes whether the auditor noticed the enumerator request identifying information from the respondent when asking for informed consent. Coded as 1 if such a request was made and 0 otherwise.

63. other_qac_other_error_auditor: describes whether the auditor noticed any enumerator errors not otherwise noted. Coded as 1 if there was such an error and 0 otherwise.
64. qac_otherproblem_auditor: describes whether the auditor noticed any problems not otherwise noted. Coded as 1 if there was such a problem and 0 otherwise.

STG flags:

65. early_termination_flag: describes whether STG flagged the interview as being terminated before completion, as indicated by the enumerator pressing the “early termination” button. Coded as 1 if the interview was terminated early and 0 otherwise.
66. set_as_complete_flag: describes whether STG flagged the interview as being manually set to “complete,” as opposed to automatically set to complete after the interview concluded. Coded as 1 if the interview was manually set to complete and 0 otherwise.
67. gender_consistency_flag: describes whether STG flagged the interview for having gender set to different values by the enumerator at the beginning and end of the interview. Coded as 1 if the gender was inconsistent throughout the interview to complete and 0 otherwise.
68. stopped_continued_flag: describes whether STG flagged the interview as stopped and restarted. Coded as 1 if the interview stopped and restarted and 0 otherwise.
69. uploaded_by_another_flag: describes whether STG flagged the interview as being uploaded to the server by an enumerator other than the one who conducted the interview. Coded as 1 if the interview was conducted and uploaded by different enumerators and 0 otherwise.
70. no_gps_continue_flag: describes whether STG flagged the interview as having no GPS data but the enumerator manually choosing to continue anyway. Coded as 1 if the enumerator manually over-rode the “lack of GPS data” warning and continued with the interview, and 0 otherwise.
71. outside_geofence_continue_flag: describes whether STG flagged the interview as being outside the geofence assigned to that interview but the enumerator manually chose to continue anyway. Coded as 1 if the enumerator manually over-rode the “outside of geofence” warning and continued with the interview, and 0 otherwise.

72. `outside_geofence_cancel_flag`: describes whether STG flagged the interview as being outside the geofence assigned to that interview and the enumerator manually chose to cancel it. Coded as 1 if the enumerator manually canceled the interview for this reason, and 0 otherwise.
73. `silent_attachments_flag`: describes whether STG flagged any audio captures as being completely silent. Coded as 1 if the interview attachments include at least one silent audio capture and 0 otherwise.
74. `stop_without_save_flag`: describes whether STG flagged the interview as being stopped without saving. Coded as 1 if the interview stopped without saving and 0 otherwise.
75. `version_changed_flag`: describes whether STG flagged the survey version as changing during the interview. Coded as 1 if the survey version changed over the course of the interview and 0 otherwise.

R scripts, completion and Percentmatch:

76. `completion_pc_script`: describes the completion percentage for the interview, generated automatically via an R script. Coded as a numeric bounded between 0 and 1 and computed as the proportion of substantive questions (i.e., survey items) to which the respondent gave a valid answer.
77. `enumerator_comp_pc_script`: describes the mean completion percentage across all interviews conducted by the enumerator, generated automatically via an R script. Coded as a numeric bounded between 0 and 1 and computed as the mean proportion of substantive questions (i.e., survey items) to which the respondent gave a valid answer for all interviews conducted by that enumerator.
78. `enumerator_comp_pc_ur_gap_script`: describes the gap between the mean completion percentage (in absolute value) across all interviews conducted by the enumerator in urban versus rural primary sampling units, generated automatically via an R script. Coded as a numeric bounded between 0 and 1 and computed as the mean proportion of substantive questions (i.e., survey items) to which the respondent gave a valid answer for all interviews conducted by that enumerator.
79. `top_pc_match_script`: describes the highest Percentmatch value for the interview (i.e., the maximum overlap with any other interview), generated automatically via an R script. Coded as a numeric bounded between 0 and 1, computed as the

maximum proportion of substantive questions (i.e., survey items) to which the respondent's response is identical to those of another interview.

80. `pc_match_top_decile_script`: describes whether the interview's top Percentmatch value is in the top decile of all Percentmatch values in the data, generated automatically via an R script. Coded as 1 if the interview is in the top decile and 0 otherwise.
81. `pc_match_bot_decile_script`: describes whether the interview's top Percentmatch value is in the bottom decile of all Percentmatch values in the data, generated automatically via an R script. Coded as 1 if the interview is in the bottom decile and 0 otherwise.

R scripts, participation rates:

82. `enumerator_noh_pc_script`: describes the proportion of interview attempts made by the enumerator that are marked as "no one home," generated automatically via an R script. Coded as a numeric bounded between 0 and 1.
83. `enumerator_int_pc_script`: describes the proportion of interview attempts made by the enumerator that resulted in successful interviews, generated automatically via an R script. Coded as a numeric bounded between 0 and 1.
84. `enumerator_ref_pc_script`: describes the proportion of interview attempts made by the enumerator that are marked as "refusal," generated automatically via an R script. Coded as a numeric bounded between 0 and 1.
85. `enumerator_noh_pc_ur_gap_script`: describes the (absolute-valued) difference in proportions of interview attempts made by the enumerator that are marked as "no one home" between urban and rural primary sampling units, generated automatically via an R script. Coded as a numeric bounded between 0 and 1.
86. `enumerator_int_pc_ur_gap_script`: describes the (absolute-valued) difference in proportions of interview attempts made by the enumerator that resulted in successful interviews between urban and rural primary sampling units, generated automatically via an R script. Coded as a numeric bounded between 0 and 1.
87. `enumerator_ref_pc_ur_gap_script`: describes the (absolute-valued) difference in proportions of interview attempts made by the enumerator that are marked as "refusal" between urban and rural primary sampling units, generated automatically via an R script. Coded as a numeric bounded between 0 and 1.

R scripts, cluster sampling:

88. `cluster_too_big_script`: describes whether the sampling cluster contained more interviews than fieldwork protocols requires, generated automatically via an R script. Coded as 1 if 10 or more interviews were in the cluster and 0 otherwise.
89. `cluster_too_small_script`: describes whether the sampling cluster contained fewer interviews than fieldwork protocols requires, generated automatically via an R script. Coded as 1 if 1 or fewer interviews were in the cluster and 0 otherwise.
90. `no_cluster_geo_variation_script`: describes whether there was any variation in the GPS coordinates across all interviews in the sampling cluster, generated automatically via an R script. Coded as 1 if there was no geographic variation across interviews in the cluster and 0 otherwise.
91. `no_upm_geo_variation_script`: describes whether there was any variation in the GPS coordinates across all interviews in the primary sampling unit, generated automatically via an R script. Coded as 1 if there was no geographic variation across interviews in the PSU and 0 otherwise.
92. `no_upm_cluster_variation_script`: describes whether there was any variation in the unique cluster identification numbers across all interviews in the primary sampling unit, generated automatically via an R script. Coded as 1 if there was no cluster variation across interviews in the PSU and 0 otherwise.
93. `cluster_dispersion_script`: describes the compactness and separation of clusters within primary sampling units, generated automatically via an R script. Coded as numeric, bounded between -1 and 1 , computed using the global average silhouette within a primary sampling unit.
94. `cluster_disp_top_decile_script`: describes whether the interview's cluster dispersion (silhouette) value is in the top decile of all cluster dispersion values in the data, generated automatically via an R script. Coded as 1 if the interview is in the top decile and 0 otherwise.
95. `cluster_disp_bot_decile_script`: describes whether the interview's cluster dispersion (silhouette) value is in the bottom decile of all cluster dispersion values in the data, generated automatically via an R script. Coded as 1 if the interview is in the bottom decile and 0 otherwise.
96. `cluster_disp_other_script`: describes whether there were any other problems measuring cluster dispersion (silhouette) value, generated automatically via an

R script. Coded as 1 if the interview encountered any error computing cluster dispersion not otherwise noted, 0 otherwise.

R scripts, timing:

97. `duration_script`: describes the absolute duration of the interview, generated automatically via an R script. Coded as numeric, in seconds, strictly positive and integer-valued.
98. `netduration_script`: describes the duration of the interview, net of screening questions, generated automatically via an R script. Coded as numeric, in seconds, strictly positive and integer-valued.
99. `duration_diff_script`: describes the difference in duration and net duration of the interview, generated automatically via an R script. Coded as numeric, in seconds, integer-valued.
100. `short_ave_question_time_script`: describes whether the average time between question prompts was too short, generated automatically via an R script. Coded as 1 if the mean question time across the interview is less than 5 seconds and 0 otherwise.
101. `short_ave_attempt_time_script`: describes whether the average time between interview attempts was too short, generated automatically via an R script. Coded as 1 if the mean time between attempted interviews is less than 5 seconds and 0 otherwise.
102. `long_run_time_script`: describes whether the total runtime of the interview was too long, generated automatically via an R script. Coded as 1 if the total runtime for the interview exceeds three hours and 0 otherwise.
103. `big_time_jump_script`: describes whether there are any large time jumps between questions in the interview log, generated automatically via an R script. Coded as 1 if any time jumps between questions exceeded 10 minutes and 0 otherwise.
104. `time_goes_back_script`: describes whether there are any backward time jumps in the interview log, generated automatically via an R script. Coded as 1 if any backward time jumps occurred and 0 otherwise.
105. `time_out_of_bounds_script`: describes whether there are any timestamps in the interview log that are outside the dates over which fieldwork was conducted, generated automatically via an R script. Coded as 1 if any timestamps are out of fieldwork dates and 0 otherwise.

R scripts, network connectivity:

106. `lss_gps_disabled_script`: describes whether GPS location service was set as “disabled” by the enumerator, generated automatically via an R script. Coded as 1 if GPS location service was disabled for the interview and 0 otherwise.
107. `lss_net_disabled_script`: describes whether network location service was set as “disabled” by the enumerator, generated automatically via an R script. Coded as 1 if network location service was disabled for the interview and 0 otherwise.
108. `mobile_disabled_script`: describes whether mobile data was disabled by the enumerator, generated automatically via an R script. Coded as 1 if mobile data was off for the interview and 0 otherwise.
109. `use_gps_altered_script`: describes whether the “use GPS” setting was set as “off” by the enumerator, generated automatically via an R script. Coded as 1 if “use GPS” was “off” for the interview and 0 otherwise.
110. `real_gps_altered_script`: describes whether the “use real GPS only” setting was set as “on” by the enumerator, generated automatically via an R script. Coded as 1 if “use real GPS only” was “on” for the interview and 0 otherwise.
111. `no_lss_gps_captures_script`: describes whether GPS location services settings were captured during the interview, generated automatically via an R script. Coded as 1 if no GPS location service settings were captured and 0 otherwise.
112. `no_lss_net_captures_script`: describes whether network location services settings were captured during the interview, generated automatically via an R script. Coded as 1 if no network location service settings were captured and 0 otherwise.
113. `no_mobile_captures_script`: describes whether mobile data settings were captured during the interview, generated automatically via an R script. Coded as 1 if no mobile data settings were captured and 0 otherwise.
114. `no_use_gps_captures_script`: describes whether any “use GPS” settings were captured during the interview, generated automatically via an R script. Coded as 1 if no use GPS settings were captured and 0 otherwise.
115. `no_use_real_gps_captures_script`: describes whether any real GPS coordinates (as opposed to approximate coordinates triangulated by WiFi or mobile connections) were captured during the interview, generated automatically via an R script. Coded as 1 if no real GPS coordinates were captured and 0 otherwise.

116. `multiple_iss_gps_captures_script`: describes whether multiple GPS location services settings were logged during the interview, generated automatically via an R script. Coded as 1 if GPS location service settings were logged more than once and 0 otherwise.
117. `multiple_iss_net_captures_script`: describes whether multiple network location services settings were logged during the interview, generated automatically via an R script. Coded as 1 if network location service settings were logged more than once and 0 otherwise.
118. `multiple_mobile_captures_script`: describes whether multiple mobile network data settings were logged during the interview, generated automatically via an R script. Coded as 1 if mobile data settings were logged more than once and 0 otherwise.
119. `multiple_use_gps_captures_script`: describes whether multiple ‘use GPS’ settings were logged during the interview, generated automatically via an R script. Coded as 1 if ‘use GPS’ settings were logged more than once and 0 otherwise.
120. `multiple_use_real_gps_captures_script`: describes whether multiple ‘use real GPS only’ settings were logged during the interview, generated automatically via an R script. Coded as 1 if “real GPS” settings were logged more than once and 0 otherwise.

R scripts, quotas:

121. `age_script`: describes the age of the respondent, generated automatically via an R script. Coded as numeric, counted in years (integer-valued and strictly positive).
122. `age_quota_invalid_script`: describes whether the age given was invalid for the quota, generated automatically via an R script. Coded as 1 if the respondent falls into an invalid age quota category, and 0 otherwise.
123. `female_script`: describes the stated gender of the respondent, generated automatically via an R script. Coded as 1 if the respondent is female and 0 otherwise.
124. `age_quota_young_script`: describes the age category of the respondent, generated automatically via an R script. Coded as 1 if the respondent falls into the (country-specific) quota category for the youngest third of respondents and 0 otherwise.
125. `age_quota_middle_script`: describes the age category of the respondent, generated automatically via an R script. Coded as 1 if the respondent falls into the (country-specific) quota category for the middle third of respondents and 0 otherwise.

126. `age_quota_old_script`: describes the age category of the respondent, generated automatically via an R script. Coded as 1 if the respondent falls into the (country-specific) quota category for the oldest third of respondents and 0 otherwise.
127. `female_age_quota_young_script`: describes the age-gender category of the respondent, generated automatically via an R script. Coded as 1 if the respondent falls into the (country-specific) quota category for the youngest third of respondents and is female, and 0 otherwise.
128. `female_age_quota_middle_script`: describes the age-gender category of the respondent, generated automatically via an R script. Coded as 1 if the respondent falls into the (country-specific) quota category for the middle third of respondents and is female, and 0 otherwise.
129. `female_age_quota_old_script`: describes the age-gender category of the respondent, generated automatically via an R script. Coded as 1 if the respondent falls into the (country-specific) quota category for the oldest third of respondents and is female, and 0 otherwise.

R scripts, comments:

130. `enumerator_comment_script`: describes whether the enumerator conducting the interview made any comments in the interview log, generated automatically via an R script. Coded as 1 if the enumerator made any comment and 0 otherwise.
131. `qc_comment_script`: describes whether an enumerator's fieldwork supervisor made any comments in the interview log, generated automatically via an R script. Coded as 1 if the supervisor made any comment and 0 otherwise.
132. `reviewer_comment_script`: describes whether an auditor made any comments in the interview log, generated automatically via an R script. Coded as 1 if the auditor made any comment and 0 otherwise.
133. `any_comment_script`: describes whether any project staff made any comments in the interview log, generated automatically via an R script. Coded as 1 if any staff made any comment and 0 otherwise.

R scripts, location:

134. `no_gps_captures_script`: describes whether no GPS data were captured during the interview, generated automatically via an R script. Coded as 1 if no GPS data are available and 0 otherwise.

135. `little_gps_change_script`: describes if many attempted interviews were made with very little variation in GPS captures, generated automatically via an R script. Coded as 1 if the ratio of attempts to unique GPS locations captured during the interview is greater than five and 0 otherwise.
136. `attempts_no_gps_change_script`: describes if many consecutive attempted interviews were made without a change in location, generated automatically via an R script. Coded as 1 if seven or more consecutive attempted interviews were made at the same GPS coordinates and 0 otherwise.
137. `big_gps_jump_script`: describes if large jumps in GPS coordinates were captured between attempted interviews, generated automatically via an R script. Coded as 1 if any jumps of 10 kilometers or more are observed between attempts and 0 otherwise.

R scripts, miscellaneous:

138. `broken_photo_script`: describes whether the front-facing image capture of the enumerator was missing, broken, or unable to be processed for quality, generated automatically via an R script. Coded as 1 if the image was unreadable and 0 otherwise.
139. `bad_photo_script`: describes whether the front-facing image capture of the enumerator contained very little pixel variation, generated automatically via an R script. Coded as 1 if the variation in pixel color was less than 0.1 on a scale from 0 to 1 and 0 otherwise.
140. `device_battery_script`: describes the device battery percentage at the time the interview began, generated automatically via an R script. Coded as a numeric value bounded between 0 and 1.
141. `ur`: describes whether the interview was conducted in an urban or rural primary sampling unit. Coded as 1 if urban and 0 if rural.
142. `enumerator_id_script`: uniquely valid enumerator identification numbers, generated automatically via an R script. Coded as a factor.
143. `device_id_script`: uniquely valid CAPI device identification numbers, generated automatically via an R script. Coded as a factor.

A.4 Models studied

The following list gives the caret name and description for each model studied in our classification task. We chose these models for their diversity of underlying approach and computational stability.

1. avNNet: Model Averaged Neural Network
2. bagEarthGCV: Bagged MARS using gCV Pruning
3. bagFDAGCV: Bagged Flexible Discriminant Analysis using gCV Pruning
4. bayesglm: Bayesian Generalized Linear Model
5. C5.0: C5.0
6. C5.0Rules: Single C5.0 Ruleset
7. C5.0Tree: Single C5.0 Tree
8. earth: Multivariate Adaptive Regression Spline
9. fda: Flexible Discriminant Analysis
10. gamSpline: Generalized Additive Model using Splines
11. gcvEarth: Multivariate Adaptive Regression Splines
12. glm: Generalized Linear Model
13. glmnet: glmnet
14. hdda: High Dimensional Discriminant Analysis
15. hdrda: High-Dimensional Regularized Discriminant Analysis
16. multinom: Penalized Multinomial Regression
17. naive_bayes: Naive Bayes
18. nb: Naive Bayes
19. nnet: Neural Network
20. pam: Nearest Shrunken Centroids

21. parRF: Parallel Random Forest
22. pcaNNet: Neural Networks with Feature Extraction
23. pda: Penalized Discriminant Analysis
24. pda2: Penalized Discriminant Analysis
25. rbfDDA: Radial Basis Function Network
26. rf: Random Forest
27. rpart: CART
28. rpart1SE: CART
29. rpart2: CART
30. RRFglobal: Regularized Random Forest
31. sda: Shrinkage Discriminant Analysis
32. sdwd: Sparse Distance Weighted Discrimination
33. slda: Stabilized Linear Discriminant Analysis
34. stepQDA: Quadratic Discriminant Analysis with Stepwise Feature Selection
35. treebag: Bagged CART
36. xyf: Self-Organizing Maps

A.5 Models with 30 versus 141 quality control procedures

Table [A2](#) provides indicators of predictive performance for the ten best-performing models using just 30 procedures. Values in parentheses indicate the change from equivalent models using all 141 variables. Large differences in performance would indicate that scholars should invest in a large number of quality control procedures, while smaller differences would suggest that a limited set of tools is more or less sufficient to recover a high-quality sample. For all indicators except RMSE, positive values reflect better performance. Taken together, these results suggest that researchers using the 30 best-performing quality control procedures may expect a sample that is not significantly lower-quality than a sample generated using all 141 quality control procedures.

Table A2: Predictive performance for the ten best models, 30 most informative variables only

Model	AUC	RMSE	Precision	Recall	Specificity	NPV
rf	0.95 (-0.03)	0.05 (0.02)	0.56 (-0.23)	0.79 (0.01)	0.95 (-0.03)	0.98 (0.00)
parRF	0.95 (-0.02)	0.05 (0.02)	0.62 (-0.20)	0.76 (0.04)	0.96 (-0.03)	0.98 (0.00)
C5.0	0.94 (-0.03)	0.06 (0.03)	0.56 (-0.15)	0.81 (-0.04)	0.95 (-0.02)	0.98 (-0.01)
RRFglobal	0.94 (-0.03)	0.05 (0.01)	0.58 (-0.22)	0.74 (-0.08)	0.96 (-0.02)	0.98 (-0.01)
pcaNNet	0.93 (-0.03)	0.06 (0.02)	0.49 (-0.14)	0.80 (0.01)	0.94 (-0.03)	0.98 (0.00)
multinom	0.92 (-0.04)	0.08 (0.05)	0.40 (-0.30)	0.72 (-0.01)	0.92 (-0.06)	0.98 (0.00)
glmnet	0.92 (-0.04)	0.08 (0.05)	0.43 (-0.28)	0.72 (-0.03)	0.93 (-0.05)	0.98 (0.00)
avNNet	0.91 (-0.05)	0.09 (0.05)	0.44 (-0.23)	0.76 (-0.03)	0.93 (-0.04)	0.98 (0.00)
bayeslgm	0.91 (-0.04)	0.08 (0.04)	0.43 (-0.22)	0.73 (-0.05)	0.93 (-0.04)	0.98 (0.00)
pda	0.91 (-0.04)	0.06 (0.02)	0.46 (-0.23)	0.70 (-0.09)	0.94 (-0.03)	0.98 (0.00)

All metrics are constrained to $[0, 1]$, with higher values indicating better performance for all except RMSE. Values in parentheses reflect the change in each measure of model performance compared to an equivalent model which uses all 141 available checks.

References

- Bennett, Archibald S. 1948. "Toward a Solution of the 'Cheater Problem' among Part-Time Research Investigators." *Journal of Marketing* 12 (4): 470–474.
- Bhuiyan, Muhammad F., and Paula Lackie. 2016. "Mitigating Survey Fraud and Human Error: Lessons Learned from a Low Budget Village Census in Bangladesh." *IASSIST Quarterly* 40 (3): 20–26.
- Biemer, Paul P., and S. Lynne Stokes. 1989. "The Optimal Design of Quality Control Samples to Detect Interviewer Cheating." *Journal of Official Statistics* 5 (1): 23–39.
- Blasius, Jörg. 2018. "Fabrication of Interview Data." *Quality Assurance in Education* 26 (2): 213–226.
- Blasius, Jörg, and Victor Thiessen. 2012. *Assessing the Quality of Survey Data*. London: Sage.
- . 2013. "Detecting Poorly Conducted Interviews." In *Interviewers' Deviations in Surveys: Impact, Reasons, Detection and Prevention*, edited by Peter Winker, Natalja Menold, and Rolf Porst, 67–88. Bern: Peter Lang.
- . 2015. "Should we Trust Survey Data? Assessing Response Simplification and Data Fabrication." *Social Science Research* 52:479–493.
- . 2018. "Perceived Corruption, Trust, and Interviewer Behavior in 26 European Countries." *Sociological Methods & Research*.
- Bredl, Sebastian, Nina Storfinger, and Natalja Menold. 2011. A Literature Review of Methods to Detect Fabricated Survey Data. Discussion paper no. 56, Zentrum für internationale Entwicklungs- und Umweltforschung, ZEU, Giessen.
- Bredl, Sebastian, Peter Winker, and Kerstin Kötschau. 2008. A Statistical Approach to Detect Cheating Interviewers. Discussion paper no. 39, Justus-Liebig-Universität Gießen, Zentrum für internationale Entwicklungs- und Umweltforschung (ZEU), December.
- Bushery, John M., Jennifer W. Reichert, Keith A. Albright, and John C. Rossiter. 1999. "Using Date and Time Stamps to Detect Interviewer Falsification." *Proceedings of the American Statistical Association (Survey Research Methods Section)*: 316–320.
- Case, Peter B. 1971. "How to Catch Interviewer Errors." *Journal of Advertising Research* 11 (2): 39–43.

- Cho, Moon J., John L. Eltinge, and David Swanson. 2003. "Inferential Methods to Identify Possible Interviewer Fraud Using Leading Digit Preference Patterns and Design Effect Matrices." *Proceedings of the American Statistical Association (Survey Research Methods Section)*: 936–941.
- Crespi, Leo P. 1945. "The Cheater Problem in Polling." *Public Opinion Quarterly* 9 (4): 431–445.
- De Haas, Samuel, and Peter Winker. 2014. "Identification of Partial Falsifications in Survey Data." *Statistical Journal of the IAOS* 30 (3): 271–281.
- . 2016. "Detecting Fraudulent Interviewers by Improved Clustering Methods: The Case of Falsifications of Answers to Parts of a Questionnaire." *Journal of Official Statistics* 32 (3): 643–660.
- Evans, Franklin B. 1961. "On Interviewer Cheating." *Public Opinion Quarterly* 25 (1): 126–127.
- Finn, Arden, and Vimal Ranchhod. 2017. "Genuine Fakes: The Prevalence and Implications of Data Fabrication in a Large South African Survey." *World Bank Economic Review* 31 (1): 129–157.
- Gomila, Robin, Rebecca Littman, Graeme Blair, and Elizabeth Levy Paluck. 2017. "The Audio Check: A Method for Improving Data Quality and Detecting Data Fabrication." *Social Psychological and Personality Science* 8 (4): 424–433.
- Hicks, Wendy D., Brad Edwards, Karen Tourangeau, Brett McBride, Lauren D. Harris-Kojetin, and Abigail J. Moss. 2010. "Using CARI Tools to Understand Measurement Error." *Public Opinion Quarterly* 74 (5): 985–1003.
- Hood, Catherine C., and John M. Bushery. 1997. "Getting More Bang from the Reinterview Buck: Identifying 'at Risk' Interviewers." *Proceedings of the Survey Research Methods Section of the American Statistical Association*: 820–824.
- Hülser, Oliver. 2013. "Automatic Interview Control of Market Research Studies." In *Interviewers' Deviations in Surveys: Impact, Reasons, Detection and Prevention*, edited by Peter Winker, Natalja Menold, and Rolf Porst, 103–116. Bern: Peter Lang.
- Judge, George, and Laura Schechter. 2009. "Detecting Problems in Survey Data Using Benford's Law." *Journal of Human Resources* 44 (1): 1–24.

- Kemper, Christoph J., and Natalja Menold. 2014. "Nuisance or Remedy? The Utility of Stylistic Responding as an Indicator of Data Fabrication in Surveys." *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences* 10 (3): 92–99.
- Krejsa, Elizabeth A., Mary C. Davis, and Joan M. Hill. 1999. "Evaluation of the Quality Assurance Falsification Interview Used in the Census 2000 Dress Rehearsal." *Proceedings of the American Statistical Association (Survey Research Methods Section)*: 635–640.
- Kuriakose, Noble, and Michael Robbins. 2016. "Don't Get Duped: Fraud through Duplication in Public Opinion Surveys." *Statistical Journal of the IAOS* 32 (3): 283–291.
- Landrock, Uta. 2017. "Investigation Interviewer Falsifications: A Quasi-experimental Design." *Bulletin of Sociological Methodology* 136 (1): 5–20.
- Menold, Natalja, and Christoph J. Kemper. 2014. "How do Real and Falsified Data Differ? Psychology of Survey Response as a Source of Falsification Indicators in Face-to-Face Surveys." *International Journal of Public Opinion Research* 26 (1): 41–65.
- Menold, Natalja, and Marie Kristin Opper. 2013. "Interviewer Experience and 'Quality' of Falsified Data." In *Interviewers' Deviations in Surveys: Impact, Reasons, Detection and Prevention*, edited by Peter Winker, Natalja Menold, and Rolf Porst, 161–172. Bern: Peter Lang.
- Menold, Natalja, Peter Winker, Nina Storfinger, and Christoph J. Kemper. 2013. "A Method for Ex-Post Identification of Falsifications in Survey Data." In *Interviewers' Deviations in Surveys: Impact, Reasons, Detection and Prevention*, edited by Peter Winker, Natalja Menold, and Rolf Porst, 25–48. Bern: Peter Lang.
- Mitchell, Susan, Kristine Fahrney, and Matthew Strobl. 2009. Monitoring Field Interviewer and Respondent Interactions Using Computer-Assisted Recorded Interviewing: A Case Study. Paper presented at the annual conference of the American Association for Public Opinion Research (AAPOR).

- Mneimneh, Zeina, Lars Lyberg, Sharan Sharma, Mahesh Vyas, Dhananjay Bal Sathe, Frederic Malter, and Yasmin Altwajiri. 2018. "Case Studies on Monitoring Interviewer Behavior in International and Multinational Surveys." In *Advances in Comparative Survey Methods: Multicultural, Multinational and Multiregional Contexts (3MC)*, edited by Timothy P. Johnson, Beth-Ellen Pennell, Ineke A. L. Stoop, and Brita Dorer, 731–770. Hoboken, NJ: Wiley.
- Murphy, Joe, Rodney Baxter, Joe Eyerman, David Cunningham, and Joel Kennet. 2004. A System for Detecting Interviewer Falsification. Paper presented at the annual conference of the American Association for Public Opinion Research (AAPOR).
- Philipson, Tomas, and Anup Malani. 1999. "Measurement Errors: A Principal Investigator-Agent Approach." *Journal of Econometrics* 91 (2): 273–298.
- Porras, Javier, and Ned English. 2004. "Data-Driven Approaches to Identifying Interviewer Data Falsification: The Case of Health Surveys." *Proceedings of the American Statistical Association (Survey Research Methods Section)*: 4223–4228.
- Robbins, Michael. 2018. "New Frontiers in Detecting Data Fabrication." In *Advances in Comparative Survey Methods: Multicultural, Multinational and Multiregional Contexts (3MC)*, edited by Timothy P. Johnson, Beth-Ellen Pennell, Ineke A. L. Stoop, and Brita Dorer, 771–806. Hoboken, NJ: Wiley.
- Schäfer, Christin, Jörg-Peter Schräpler, Klaus-Robert Müller, and Gert G. Wagner. 2005. "Automatic Identification of Faked and Fraudulent Interviews in the German SOEP." *Schmollers Jahrbuch: Journal of Applied Social Science Studies/Zeitschrift für Wirtschafts- und Sozialwissenschaften* 125 (1): 183–193.
- Schräpler, Jörg-Peter, and Gert G. Wagner. 2005. "Characteristics and Impact of Faked Interviews in Surveys: An Analysis of Genuine Fakes in the Raw Data of SOEP." *Allgemeines Statistisches Archiv* 89 (1): 7–20.
- Schreiner, Irwin, Karen Pennie, and Jennifer Newbrough. 1988. "Interviewer Falsification in Census Bureau Surveys." *Proceedings of the American Statistical Association (Survey Research Methods Section)*: 491–496.
- Simmons, Katie, Andrew Mercer, Steve Schwarzer, and Courtney Kennedy. 2016. "Evaluating a New Proposal for Detecting Data Falsification in Surveys: The Underlying Causes of 'High Matches' Between Survey Respondents." *Statistical Journal of the IAOS* 32 (3): 327–338.

- Slomczynski, Kazimierz M., Przemek Powalko, and Tadeusz Krauze. 2017. "Non-unique Records in International Survey Projects: The Need for Extending Data Quality Control." *Survey Research Methods* 11 (1): 1–16.
- Stokes, Lynne, and Patty Jones. 1989. "Evaluation of the Interviewer Quality Control Procedure for the Post-Enumeration Survey." *Proceedings of the American Statistical Association (Survey Research Methods Section)*: 696–698.
- Storfinger, Nina, and Peter Winker. 2011. Robustness of Clustering Methods for Identification of Potential Falsifications in Survey Data. Discussion Papers 57, Justus Liebig University Giessen, Center for International Development and Environmental Research (ZEU).
- . 2013. "Assessing the Performance of Clustering Methods in Falsification Identification using Bootstrap." In *Interviewers' Deviations in Surveys: Impact, Reasons, Detection and Prevention*, edited by Peter Winker, Natalja Menold, and Rolf Porst, 49–66. Bern: Peter Lang.
- Swanson, David, Moon Jung Cho, and John Eltinge. 2003. "Detecting Possibly Fraudulent or Error-Prone Survey Data Using Benford's Law." *Proceedings of the American Statistical Association (Survey Research Methods Section)*: 4172–4177.
- Turner, C., J. Gribbe, A. Al-Tayyip, and J. Chromy. 2002. Falsification in Epidemiological Surveys: Detection and Remediation. Technical Papers on Health and Behavior Measurement No. 53. Washington, D.C.: Research Triangle Institute.
- Vanden Eng, Jodi L., Adam Wolkon, Anatoly S. Frolov, Dianne J. Terlouw, M. James Eliades, Kodjo Morgah, Vincent Takpa, et al. 2007. "Use of Handheld Computers with Global Positioning Systems for Probability Sampling and Data Entry in Household Surveys." *American Journal of Tropical Medicine and Hygiene* 77 (2): 393–399.
- Waller, LLOYD George. 2013. "Interviewing the Surveyors: Factors which Contribute to Questionnaire Falsification (Curbstoning) among Jamaican Field Surveyors." *International Journal of Social Research Methodology* 16 (2): 155–164.