Stark, Katharina; Zinn, Sabine

Working Paper

Using mathematical graphs for questionnaire testing in large-scale surveys

SOEPpapers on Multidisciplinary Panel Data Research, No. 1135

Provided in Cooperation with:
German Institute for Economic Research (DIW Berlin)

Suggested Citation: Stark, Katharina; Zinn, Sabine (2021) : Using mathematical graphs for questionnaire testing in large-scale surveys, SOEPpapers on Multidisciplinary Panel Data Research, No. 1135, Deutsches Institut für Wirtschaftsforschung (DIW), Berlin

This Version is available at:
https://hdl.handle.net/10419/234461

1135 2021

SOEP papers
on Multidisciplinary Panel Data Research

Using Mathematical Graphs for Questionnaire Testing in Large-Scale Surveys

Katharina Stark and Sabine Zinn

DIW SOEP

SOEPpapers on Multidisciplinary Panel Data Research at DIW Berlin

This series presents research findings based either directly on data from the German Socio-Economic Panel (SOEP) or using SOEP data as part of an internationally comparable data set (e.g. CNEF, ECHP, LIS, LWS, CHER/PACO). SOEP is a truly multidisciplinary household panel study covering a wide range of social and behavioral sciences: economics, sociology, psychology, survey methodology, econometrics and applied statistics, educational science, political science, public health, behavioral genetics, demography, geography, and sport science.

The decision to publish a submission in SOEPpapers is made by a board of editors chosen by the DIW Berlin to represent the wide range of disciplines covered by SOEP. There is no external referee process and papers are either accepted or rejected without revision. Papers appear in this series as works in progress and may also appear elsewhere. They often represent preliminary studies and are circulated to encourage discussion. Citation of such a paper should account for its provisional character. A revised version may be requested from the author directly.

Any opinions expressed in this series are those of the author(s) and not those of DIW Berlin. Research disseminated by DIW Berlin may include views on public policy issues, but the institute itself takes no institutional policy positions.

The SOEPpapers are available at **http://www.diw.de/soeppapers**

ISSN: 1864-6689 (online)

German Socio-Economic Panel (SOEP)
DIW Berlin
Mohrenstrasse 58
10117 Berlin, Germany

Contact: soeppapers@diw.de

Using Mathematical Graphs for Questionnaire Testing in Large-Scale Surveys

Katharina Stark
katharina.stark@lifbi.de, Leibniz Institute for Educational Trajectories, Germany

Sabine Zinn *(corresponding author)*
szinn@diw.de, German Institute for Economic Research & Humboldt University, Germany

Abstract

In this article, we present an automated test procedure for examining the filter structure and instructions implemented in electronic questionnaires, and for checking the fit of a questionnaire to the targeted sample. With our approach, we can represent and describe questionnaires using mathematical graphs and specify questionnaire properties in a formal and standardised way. It also allows us deriving mathematical graphs from empirical data. We can then compare the questionnaires (mathematical graphs) with the survey data in an automatable process. Our procedure also includes a test plan we developed for automatic testing. Our approach is complete, portable, and scalable: It is complete in that graphs are used to describe the questionnaires and questionnaire data. It is portable as a result of its generic structure, which is not limited to a specific questionnaire (type), and the provision of free, extensible open-source software. It is scalable through the use of a modular test structure and efficient, up-to-date graph representation and manipulation and display algorithms. We illustrate the functionality and broad applicability of our approach using a hypothetical example and two real examples from two large well-established survey studies (the German National Educational Panel Study, NEPS, and the German Socio-Economic Panel, SOEP).

Keywords: questionnaire testing, mathematical graphs, automated test procedure, large-scale surveys

1 INTRODUCTION

In this article, we introduce a flexible approach for electronic questionnaire testing that is easy to use, extensible, open-source, and that relies on state-of-the-art testing research. Our software is written in the statistical programming language R[1] and available at no charge.[2]

Many survey institutes still test their electronic questionnaires manually, despite the potential for comprehensive testing. For example, in the German Socio-Economic Panel (SOEP) and the German National Educational Panel Study (NEPS), it is common practice to test questionnaires using so-called mock interviews. In these interviews, testers take on the roles of interviewers and respondents and click their way through the questionnaires. Afterwards, the data output is checked for anomalies. Mock interviews are extremely helpful in pretesting, since they make it possible to identify issues such as mistakes in filter instructions and inadequate response options in an early stage of questionnaire development (Hansen and Couper 2004; Statistics Sweden 2004). However, comprehensive tests that take into account all possible routes through a questionnaire are difficult to carry out solely by means of manual testing, particularly with the long questionnaires used in large surveys such as SOEP and NEPS. Manual tests are simply too error-prone and time-consuming and require high personnel resources (Levinsohn and Rodriguez 2001). For comprehensive examination of the accuracy of questionnaires used in large-scale surveys automated tests are much more useful.

[1] https://cran.r-project.org/, R version 4.0.2
[2] See GitHub repository under https://github.com/KatharinaStark/MathematicalGraphsForQuestionnaireTesting.

1.1 Requirements for Automated Questionnaire Testing

For an automated testing approach to be applicable to large-scale survey studies, we argue that it must meet at least three criteria: it must be *complete*, *portable*, and *scalable*. *Complete* means that the testing approach must be able to check all paths through a questionnaire. *Portable* means that the approach and the corresponding software are not bound to a specific questionnaire structure or development system. *Scalable* means that the testing approach can handle long questionnaires with hundreds of items without aborting or requiring excessive amounts of resources. If a test approach does not fulfil all three of these minimum criteria, it remains a one-off solution for either a specific problem or a very specific test situation.

There is no lack of automated approaches to questionnaire testing. However, the existing approaches do not meet the three aforementioned criteria, and some approaches do not meet any of them. For example, RTI International developed a tool called RoboCAI, which creates test cases for questionnaire testing. The tool is directly tailored to the Blaise questionnaire development software[3] (Levinsohn and Rodriguez 2001). As it cannot be readily applied to questionnaires that were not produced with this software, it is not portable. Furthermore, as it does not test all the paths that respondents might take through a questionnaire (Levinsohn and Rodriguez 2001), it is not complete. Finally, it is not scalable, because the effort required to create test cases for questionnaires with 250 or more items increases disproportionately with the number of items (Rodriguez and Levinsohn 2003).

Bethlehem and Hundepool (2004) developed the Tool for the Documentation and Analysis of Electronic Questionnaires (TADEQ). It generates various statistics on the structure of a

[3] https://www.blaise.com/

particular questionnaire as a means of detecting possible errors. One attractive feature of TADEQ is that it uses a generic language (based on XML) to describe the structure and content of survey questionnaires. Hence, it is portable and not bound to a specific questionnaire development system like RoboCAI. Unfortunately, it is not possible to reconfigure the TADEQ software, e.g., to add further test functionalities, because its source code is not accessible. Since this prevents us from testing TADEQ—and since there is no information in the literature about the number of items it can handle—we cannot judge its scalability. Furthermore, TADEQ is not complete in the sense that it cannot identify erroneously added paths that occur due to an incorrectly programmed questionnaire. Therefore, neither RoboCAI nor TADEQ are promising tools for general and automated questionnaire testing in large-scale surveys.

Feeney and Feeney (2019) developed an intuitive questionnaire testing approach that is complete and portable. Concretely, they introduce a computer-readable representation of questionnaires that they call a progression table. Based on this representation, they develop an algorithm for testing the correct assignment of respondents to questionnaire strands and for testing the correctness of filter instructions. Their test approach is complete in the sense that it covers all possible paths through a questionnaire. The algorithm is implemented in R, and it is free and open-source. Thus, the approach is also portable. Unfortunately, the approach of Feeney and Feeney (2019) (despite having several very useful features) is not scalable, because it generates and uses a synthetic dataset that increases dramatically in size with the length of the questionnaire.

1.2 The Use of Mathematical Graphs

One way to overcome Feeney and Feeney's (2019) scalability problem is to extend their framework using mathematical graphs to describe survey questionnaires. Doing so means

defining questions as vertices and the path from one question to another as edges. The scalability of our test approach results, on the one hand, from (the large number of) scalable algorithms that exist for the description and manipulation of mathematical graphs (see, e.g., Zhang, Cui, and Zhu 2015; Kawai, Mukuta, and Harada 2019). On the other hand, it results from the use of an established test process (see Section 1.3).

The idea of using mathematical graphs for describing questionnaires is not new. Fagan and Greenberg (1988) used graph theory to identify whether respondents had failed to answer questions due to misleading or erroneously programmed filter instructions. Their approach is complete in the sense that it covers all questionnaire paths that can possibly be taken. As such, Fagan and Greenberg's (1988) approach is very useful for testing large-scale questionnaires. However, their approach lacks an automated link between questionnaire template and questionnaire graph and therefore requires manual translation of the questionnaire into the graph. This is cumbersome and error-prone, especially for large-scale surveys. Fagan and Greenberg (1988) give a summary of two computer programs written to implement their approach and also describe two illustrative applications. However, no source code is available, and therefore the implementation of the approach is not clear. Hence, the portability of the approach is, at a minimum, questionable.

Elliott (2012) uses mathematical graphs for questionnaire testing as well. He developed a documentation system in which questionnaires are defined by relational databases made up of two primary tables, one for the vertices of the questionnaire graph and one for its edges. These databases are used to derive graph properties, for instance, all paths that can possibly be traversed. Based on these properties, Elliott derives rules to test whether the questions are connected correctly, whether all filter instructions are implemented correctly, whether the order

of all components is correct, and whether all survey questions are reachable. Unfortunately, the programming of Elliott's documentation system tool is not open-source (but can be obtained from the author on request, see Elliott 2012, p. 18). Furthermore, it is unclear what questionnaire programming software the tool is tailored for. Testing is carried out in part by clicking through two questionnaire versions (a template and a programmed version) on a graphical user interface and comparing them visually. Such an approach is inefficient and prone to error. Finally, Elliott's implementation cannot handle long questionnaires due to scalability issues. Instead, he proposes using probability samples for all questionnaire components that are to be tested. As a result, Elliott's approach does not meet any of the minimum criteria formulated above (completeness, portability, scalability) for a testing approach that is suitable for large-scale surveys.

1.3 Embedding Questionnaire Testing in an Established Test System

Despite its shortcomings, Elliott's (2012) approach is innovative in one crucial respect: it is rooted in the theory of system testing used, for instance, in software testing. We know of no other approach to questionnaire testing that has made this connection, although it would seem obvious to do so: Electronic questionnaires implement sequences of instructions for conducting interviews that are carried out on a computer. Similarly, computer programs are collections of instructions that must be executed by a computer to perform certain tasks. Software testing is an established research field that has produced many sophisticated methods (see, for example, Ammann and Offutt 2016). The general V-Modell is a process model that has proven its usefulness in testing software (e.g., Mathur and Malik 2010). In our opinion, it is also very well suited for testing questionnaires. In the V-Modell, test and development are conducted in parallel throughout the different stages of product development. For questionnaire development, this means that the tests are carried out even on the smallest questionnaire units (the items). The

questionnaire is then tested successively across all of its levels of complexity. In the last step, the entire questionnaire is tested. Such a processing is possible because questionnaires can normally be broken down into items and question modules. At the final level of complexity, the questionnaire is then tested on the basis of the questionnaire modules (i.e., the question modules and their connection are examined here).[4] Thus, it is possible that testing runs simultaneously with questionnaire development. In this way, errors can be detected early in the development process. There are well-established and proven methods for testing the reliability and validity of the smallest questionnaire units, the items. However, to our knowledge, there are no efficient methods for testing questionnaire modules, i.e., for testing sets of questions within a specific topic area such as childcare or health behaviour. In this article, we present a test approach that closes this gap. By introducing a modular testing procedure, we address the problem of scalability that other questionnaire testing approaches have: We propose to test the questionnaire at a modular level and then bring the tested questionnaire modules back together.

Key to the approach is also a comprehensive test strategy (based on a test plan), which we implement together with the progression tables and the questionnaire graphs. We do not set any restrictions regarding the structure of the questionnaires. The only condition is that a questionnaire can be represented by a progression table and a mathematical graph. Thus, our approach meets all of the aforementioned minimum criteria for automated testing (*completeness*, *portability*, and *scalability*).

Our test approach is suitable for comparing empirical survey data with the questionnaire structure as well. The idea is to derive an empirical mathematical graph from the empirical

[4] This kind of testing is called an integration test (Ould and Unwin 1986, chap. 4.3.5).

survey data and to systematically compare it with the graph representing the questionnaire structure. In this way, we can investigate whether the questionnaire structure fits the target population under investigation.

The remainder of this article is structured as follows: First, in Section 2, we introduce the use of mathematical graphs to describe survey questionnaires and survey data. In Section 3, we detail the test plan including the test cases to be used for questionnaire testing. In Section 4, we apply our novel testing approach to two real questionnaire modules. We conclude our article in Section 5.

2 PROGRESSION TABLE AND MATHEMATICAL GRAPHS

To illustrate the different steps to test electronic questionnaires with our approach, we start with a simple questionnaire example, which can be seen as a module of a large-scale survey. This questionnaire is applied to a synthetic, self-generated sample.

2.1 Simple Example

Our example questionnaire contains ten questions about smoking behaviour. Figure 1 shows the questionnaire as a Nassi-Shneiderman diagram (Nassi and Shneiderman 1973; Shneiderman 2003). A detailed description of how to read this kind of diagram is given in the supplement S1. The sample in this example consists of N=200 respondents whose distribution of attributes is given in Table S2 in the supplement. To illustrate the capacity of our approach, we created two scenarios under which (1) the questionnaire is wrongly implemented and (2) the questionnaire does not match the sample because it contains redundant paths. Concretely, under Scenario 1, we assume that question 5 has erroneously been omitted (see Figure S3 in the supplement for the related Nassi-Shneiderman diagram).

<table>
<tr><td colspan="4">BEGIN</td></tr>
<tr><td colspan="4">Q1: Sex? (sex)
1: male; 2: female</td></tr>
<tr><td colspan="4">Q2: Highest educational achievement? (edu)
1: no school-leaving qualification; 2: school-leaving qualification from a special needs school;
3: Hauptschulabschluss[5]; 4: Mittlere Reife[6]; 5: Abitur[7]</td></tr>
<tr><td colspan="4">Q3: Subjective social class? (class)
1: lower class; 2: middle class; 3: upper class</td></tr>
<tr><td colspan="2">Q4: Current smoker? (smoker)
1: yes</td><td colspan="1">2: no</td><td>3: no answer</td></tr>
<tr><td colspan="2" rowspan="2">Q5: Smoking device? (smodev)
1: cigars; 2: cigarettes;
3: e-cigarettes/vaporizer</td><td colspan="2">Q8: Ever smoked? (smoever)</td><td rowspan="5"></td></tr>
<tr><td>1: yes</td><td>2: no</td></tr>
<tr><td colspan="2" rowspan="2">Q6: Smoking frequency? (smofreq)
1: irregular/less than once a week;
2: once/several times a week;
3: several times a day</td><td colspan="2" rowspan="2"></td></tr>
<tr><td colspan="2">Q7: Ever tried stop smoking? (smostop)
2: no | 1: yes</td></tr>
<tr><td></td><td colspan="2">Q9: Why tried to stop/stopped smoking? (smostopr)
1: health problems; 2: too expensive;
3: other reasons</td><td></td></tr>
<tr><td colspan="3">Q10: General state of health? (subhealth)
1: good; 2: poor</td><td></td></tr>
<tr><td colspan="4">END</td></tr>
</table>

Figure 1: Nassi-Shneiderman diagram of example questionnaire.
Authors' own representation.

Under Scenario 2, we assume that our sample contains only a few smokers (see Table S2 in the supplement for the related numbers). The questionnaire and sample without defects are called the baseline scenario. Using this example, we now illustrate how a questionnaire can be described in

[5] School-leaving qualification from the Hauptschule, which is a school for basic secondary education in Germany.
[6] School-leaving qualification from the Realschule, which is an intermediate secondary school in Germany.
[7] University entrance qualification in Germany.

a computer-readable format from which mathematical questionnaire graphs can then be derived in an automated way.

2.2 Progression Table

Table 1 depicts the progression table for the questionnaire in our example.

Table 1: Progression table of example questionnaire (baseline scenario).

ROW	FROM	FILTER	ANSWER OPTIONS	TO
1	BEGIN	ALL		sex
2	sex	ALL	1; 2	edu
3	edu	ALL	1; 2; 3; 4; 5	class
4	class	ALL	1; 2; 3	smoker
5	smoker	smoker = 1	1	smodev
6	smoker	smoker = 2	2	smoever
7	smoker	smoker = 3	3	END
8	smodev	ALL	1; 2; 3	smofreq
9	smofreq	ALL	1; 2; 3	smostop
10	smostop	smostop = 1	1	smostopr
11	smostop	smostop = 2	2	subhealth
12	smoever	smoever = 1	1	smostopr
13	smoever	smoever = 2	2	subhealth
14	smostopr	ALL	1; 2; 3	subhealth
15	subhealth	ALL	1; 2	END

Authors' own representation.

In its original form, a progression table has four columns: ROW, FROM, FILTER, and TO. The column ROW only enumerates the rows of the progression table and thus eases referencing. The FROM column gives the question just answered. The FILTER column displays the skip instructions that follow the FROM question. Only those respondents who have answered the FROM question and whose answers meet the criteria in the FILTER column will be directed next to the TO question. Each question in the questionnaire must appear at least once in the FROM or TO column. The skip instructions must always refer only to the FROM questions in the current or previous lines and must be mutually exclusive and exhaustive, so that each person can only ever receive one suitable follow-up question (see, e.g., lines 5-7 in Table 1). We have extended the original form of the progression table to include a further column, namely

ANSWER OPTIONS. This column contains all possible answer options of a FROM question.[8] This information helps later when testing whether all answer options are complete, have been used, or if some of them are even superfluous or missing. The rows of the table give the progression from one question to the next. They are sorted by the FROM questions. If two or more rows have the same FROM question (e.g., this occurs when, after filter questions, people are redirected to different TO questions) they are sorted by the TO questions (see, e.g., rows 5-7 in Table 1). The FROM and the TO columns are always sorted by the questionnaire order of the questions. Each progression table starts with a BEGIN in the FROM column and ends with an END in the TO column. By definition, the first line of the ANSWER OPTIONS column never contains a value (since there is nothing to answer at this point). The first and the last rows of the FILTER column also allow the use of progress tables for successive questionnaire modules: Here, criteria can be defined for the respondents who are to receive a module. The skip instruction ALL indicates that all respondents who answer the FROM question have to proceed to and answer the TO question. With the skip instruction *ALL* in our example (rows 1 and 15 in Table 1), we indicate that the questionnaire is for everyone (and a potential subsequent questionnaire module as well). Creating a progression table in the described form is the first step in our testing approach.[9]

[8] The answer options are seperated from each other with a semicolon and there must be no line breaks.
[9] For a questionnaire to be tested, a progression table has to be given in any format that can be imported to R, e.g., a simple text or Excel format. An example is given in the online supplement on GitHub, see footnote 2.

2.3 Questionnaire Graph

In a second step, a mathematical graph showing the structure of the questionnaire can be derived very easily with the help of the progression table that has been created.[10]

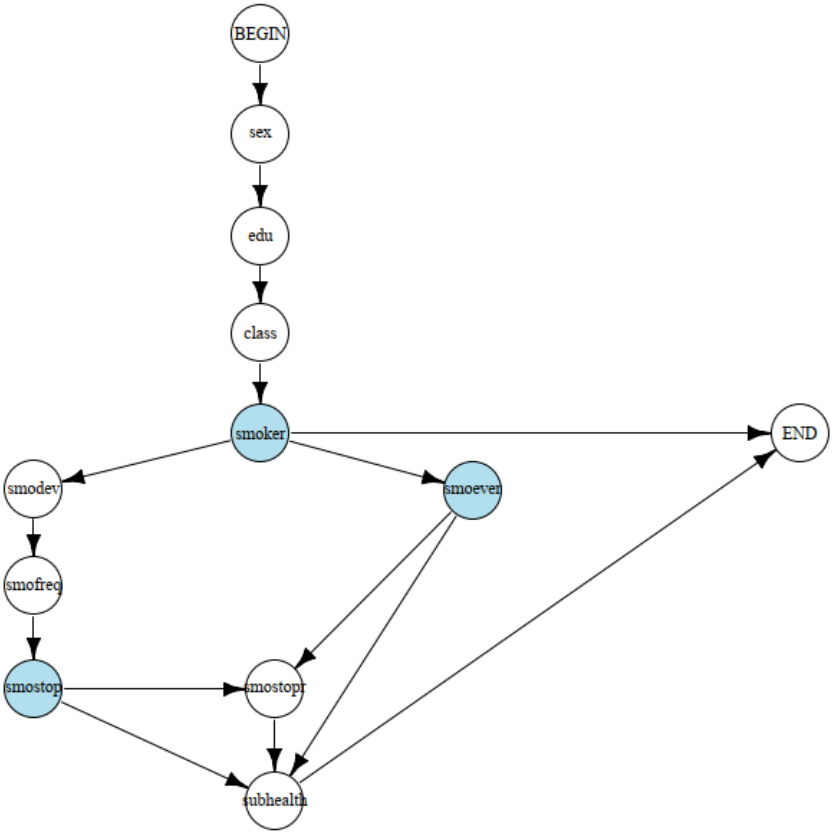

Figure 2: Questionnaire graph of questionnaire in the example (baseline scenario).
Authors' own representation.

Figure 2 shows the mathematical graph resulting from the progression table in the example. This graph consists of a finite set of vertices $V = \{v_1, v_2, \ldots, v_n\}$ representing the questions in the questionnaire and a finite set of edges $E = \{e_1, e_2, \ldots, e_p\}$ each representing a possible transition

[10] For this purpose, we suggest using the igraph package (Csárdi 2019) in R (R Core Team 2020). Concretely: Once we have imported the progression table in R as a data frame, we can use the *graph_from_data_frame* function to derive the mathematical graph.

from one question v_i to a subsequent question v_j. Each edge is defined by an ordered pair (v_i, v_j). Each row in the progression table corresponds to an edge in the graph. The edges are taken from the columns FROM and TO. We call this graph the questionnaire graph. The questionnaire graph G_{quest} in our simple example contains a total of 12 vertices, e.g., v_{BEGIN}, v_{sex}, etc., and 15 edges linking the vertices, e.g. (v_{sex}, v_{edu}). Variable names are vertex attributes; the filter instructions and all possible answer options are edge attributes (the latter is not shown in Figure 2 for reasons of clarity). Each graph representing a questionnaire has five properties (Elliott 2012): First, it contains a starting and an ending vertex, which are referred to here as BEGIN and END, respectively. Second, all vertices are connected except the BEGIN and END vertices. That is, for each vertex in the graph, there is at least one incoming edge coming from the BEGIN vertex and one outgoing edge leading to the END vertex. Third, all edges are directed, i.e., all edges lead only in one direction. Fourth, parallel edges are possible, leading from one vertex to the same subsequent vertex. They depict filter instructions of second order. Such filter instructions occur when the succession from one question to the next one does not only depend on the answer to the actual question but also on answers of preceding ones. (In our simple example, no parallel edges occur.) Fifth, vertices can appear more than once in a path, e.g., in the case of loops. (In our simple example, no loops exist.)

Once the questionnaire graph has been designed, two useful properties can be derived (which we will use later for testing, see Section 3): First, **all possible paths** through the (questionnaire) graph can be derived.[11] Here, a path is defined as 'a unique, ordered set of [vertices] which traverses an instrument from [BEGIN] to [END]' (Elliott 2012, p. 16). It thus indicates a possible

[11] E.g., with the function *all_simple_paths* from the package *igraph* in R. For further analyses, we transform parallel edges to one single edge and assign all related filter instructions and possible answer options as edge attributes to the new single edge. This step is necessary, because the *all_simple_paths* function ignores multiple and loop edges.

sequence of questions (and answers) that a respondent may go through in a questionnaire. Our simple example exhibits five possible paths (numbered with the path numbers 1 to 5, see Table 2). For example, a respondent starts at the BEGIN vertex, then answers the question about gender, her/his highest level of education and subjective class affiliation, and finally the question about her/his smoking status. A respondent who claims to be a non-smoker is then asked whether she/he has ever smoked. If the answer is no, she/he is asked to give her/his subjective health assessment and then exits the questionnaire (see line 4 in Table 2).

Second, the graph allows the derivation of (the number of) **all possible filter patterns** in a questionnaire. A filter pattern is defined as the combination of all paths that can be traversed in a graph. In our simple example, there exist 31 possible filter patterns. For example, the traversing of all five possible paths yields the filter pattern 1-2-3-4-5 (i.e., indicated by the path numbers). If we assume that only smokers are asked or that respondents make no statement about their smoking status, the filter pattern is 1-2-5 (since the paths 3 and 4 only refer to non-smokers and thus are not traversed by any respondent).

Table 2: All possible paths through the questionnaire (graph) in the example (baseline scenario).

Path Number	Path (indicated by the sequence of variable names of the questions passed)
1	BEGIN→sex→edu→class→smoker→smodev→smofreq→smostop→smostopr→subhealth→END
2	BEGIN→sex→edu→class→smoker→smodev→smofreq→smostop→subhealth→END
3	BEGIN→sex→edu→class→smoker→smoever→smostopr→subhealth→END
4	BEGIN→sex→edu→class→smoker→smoever→subhealth→END
5	BEGIN→sex→edu→class→smoker→END

Authors' own representation.

2.4 Empirical Graphs from Survey Data

Once data have been collected by means of a questionnaire (module), a second type of mathematical graph can be derived: graphs that show the sequence of questions and answers resulting from the actual survey of a given sample. We distinguish between graphs that are

derived from the data on a single respondent from graphs that are derived from an entire sample. We call the first type of graphs ***empirical individual graphs*** and the second type ***empirical sample graphs***. In their components, empirical graphs do not differ from questionnaire graphs: vertices mark questions, and edges mark paths from one question to another depending on the answer. However, the structure of the empirical graphs results directly from the survey data. There is an empirical individual graph for each respondent in the data set and an empirical sample graph that contains all vertices and edges that were traversed at least once by any respondent. We assign the filter instructions and the given answers as attributes to the edges of the empirical individual graphs. Respondent IDs, path numbers, and the actual filter pattern are stored in the empirical individual graphs as graph attributes. The summary of all paths and the filter pattern are stored as graph attributes in the empirical sample graph. From this graph, it is therefore possible to derive all the paths actually traversed by the respondents in the survey and thus the corresponding filter pattern of the entire sample. The procedure how we derive empirical graphs from survey data is described in detail in the supplement S4. Figure 3 shows three examples of empirical individual graphs for our simple example. It also depicts the corresponding sample graph. Under Scenario 1 "wrong questionnaire programming" in the empirical survey data, system-missing codes (i.e., NAs) appear erroneously under question 5 (i.e., smoking device, *smodev*). Thus, question 5 does not appear in the empirical sample graph.

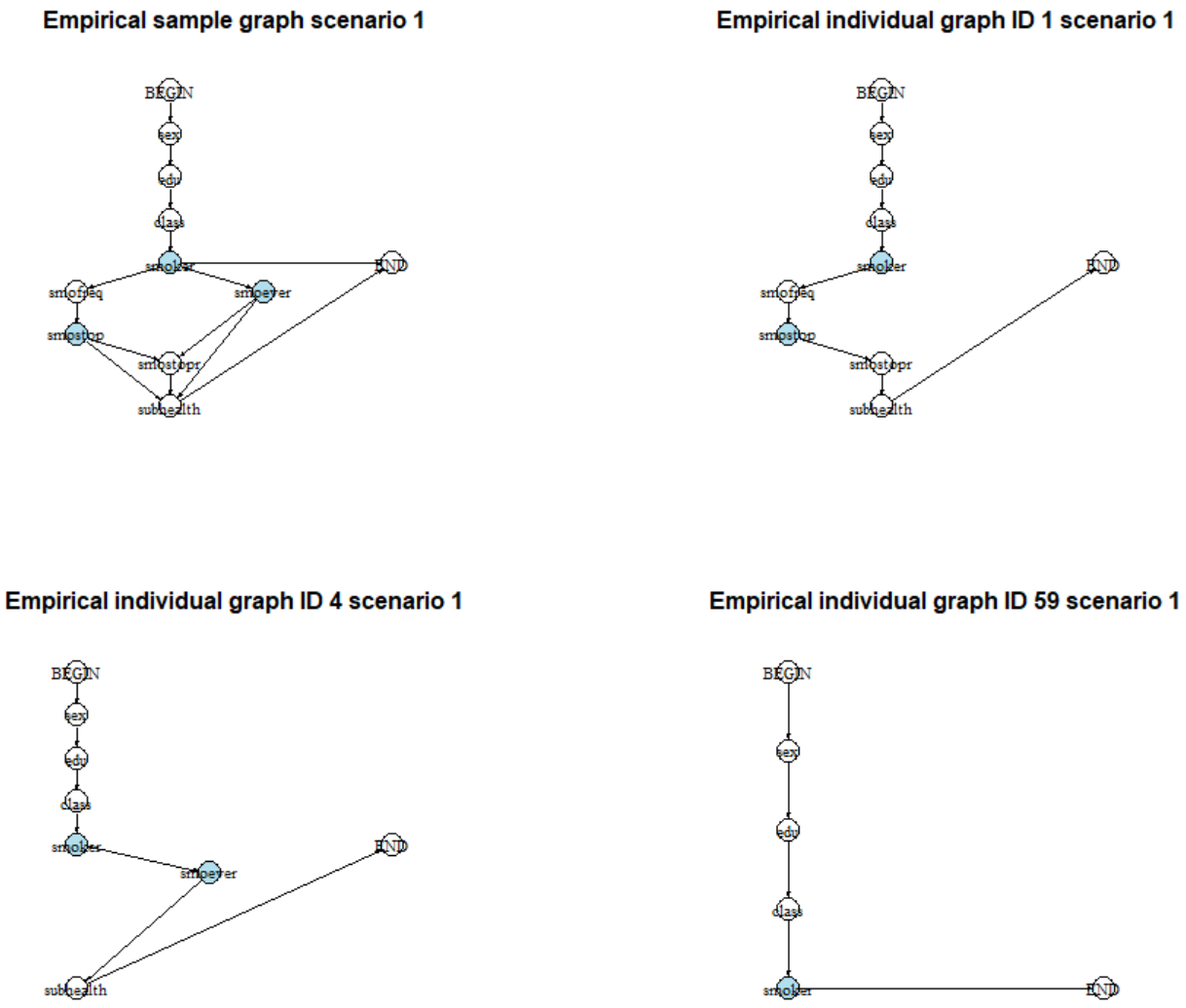

Figure 3: Empirical sample graph (top left) and three randomly chosen empirical individual graphs (top right and bottom line) for Scenario 1.
Authors' own representation.

Thus, the empirical sample graph for Scenario 1 already differs from the questionnaire graph (for the correctly implemented questionnaire) in terms of its structure. In contrast, the empirical sample graph for Scenario 2 "path redundancy" has the correct structure. Compared to the default setting, however, there are paths that are used very little.

3 QUESTIONNAIRE TESTING STRATEGY

We propose the use of a test plan to examine questionnaires in a structured, comprehensible, automatable, and repeatable way with regard to their programmed form and their fit to empirical data. A test plan specifies the process by which a product is to be tested, including partial test steps (*test cases*), preconditions for the partial test steps, measurements, and instruments used for testing, exemplifications, and implications of test results. The use of test plans is standard in quality planning (see, e.g., International Organization for Standardization 2015).

We have developed a test plan that makes use of the clear, formal structure of the questionnaire graph and the empirical graphs introduced in Section 2 (see Table 3). The first two columns of the test plan give the **Test Case ID** and **Test Case Specification**, which uniquely specify the exact issue under examination in the unique testing step. The third column, **Preconditions,** determines the conditions under which a certain test case applies. If the preconditions are not fulfilled, this test case is skipped in the testing process. The next column, **Test Data,** specifies first, the type of graph to be used in the test case, and second, the related graph or edge attributes. The column **Measure** indicates the conditions under which an anomaly in the questionnaire is present. If an anomaly is present, detailed instructions for further testing are given in the columns **Steps** and **Exemplifications**. The last column, **Implications**, shows the issue with the questionnaire if a detected anomaly proves to be a real problem.

Questionnaire testing follows this test plan, proceeding from one test case to the next. The test cases are processed regarding their assignment to the following graph characteristics: filter structure (FS), paths (P), vertices (V), and edges (E). As soon as a problem with the questionnaire or its fit to the data is detected, the *next steps* given in the column **Implications**

take effect. Our test plan does not claim to be complete, i.e., additional test cases can easily be integrated if they are necessary and appropriate to a questionnaire form or topic.

The first two test cases in our test plan concern the **filter structure**. Concretely, we test *impossible paths* (test case ID *FS1*) and *incorrect filter programming* (*FS2*). There are no preconditions for the associated test cases, since they are the first in a row. To test whether there are impossible paths (*FS1*), we use the questionnaire graph and the empirical sample graph as well as the graph attribute *filter pattern.* We check whether there are paths in the empirical graph that do not occur in the questionnaire graph (subsequently denoted "path(s) 0").[12] Impossible paths point to a mistake in the implementation of the questionnaire in the actual survey. If impossible paths are detected, the test procedure proceeds to the test cases *V1*, *V2*, *E1*, and *E2* (in that order) to determine why the problem occurred (e.g., due to missing or additional edges or vertices; see below). Otherwise, the testing procedure continues with the test case *incorrect filter programming* (*FS2*). This test case relates to the questionnaire programming: If a given answer is not included in the set of all possible answer options (of the questionnaire graph), a wrong skip instruction has been implemented. This leads to a correct follow-up question but guides the wrong sub-sample to this question. In this test case, the empirical individual graphs, the questionnaire graph, and the edge attributes *given answer* and *all possible answer options* are used: the empirical individual graphs and the questionnaire graph are compared for all *given*

[12] Due to the way filter patterns are defined, the computational effort required to derive all possible filter patterns grows exponentially with the number of filter questions. Thus, in order to make our approach applicable to complex questionnaires in a reasonable amount of time and computational effort, the procedure that we have implemented (in R) does not compare the empircal graph(s) and the questionnaire graph concerning filter patterns but concerning paths.

answers (attribute of empirical sample graph) and *all possible answer options* (questionnaire graph). If *all given answers* are not part of *all possible answer options*, there is an issue.[13]

The third test case examines the **paths** of the questionnaire graphs, or more precisely, whether there are *too-low path frequencies* (*P1*) indicated by less than 30 respondents traversing a path.[14] This test case presupposes that there are no impossible paths and no incorrect filter programming. If these conditions are not met, the test procedure ends in the case of incorrect filter programming and continues at *V1* in the case of impossible paths. To assess whether there is an issue with the path frequencies, the path distribution of the empirical sample graph is considered. If too-low path frequencies are detected, the questionnaire contains redundant paths that are likely to be hindering feasible statistical analyses of sub-samples. In this case, the test procedure ends and the questionnaire developer(s) should reconsider the questionnaire design.

The third set of test cases examines graph vertices and edges if impossible paths were detected in test case *FS1*. Specifically, we test at what point the problem of the impossible path(s) is triggered: Is it caused by the addition or omission of a vertex or edge in the empirical graph(s) compared to the questionnaire graph?

All four test cases *V1*, *V2*, *E1*, and *E2* use the empirical sample graph and the questionnaire graph. Test case *V1* examines whether there are *missing vertices*. For that purpose, it checks whether there are fewer vertices in the empirical sample graph than in the questionnaire graph by comparing the number of vertices in both graphs. Missing vertices in the empirical sample graph are caused by survey questions that are not answered by any respondent. Reasons are either

[13] In contrast to an issue detected by test case *FS1*, an issue detected by test case *FS2* does not mean that the structure of the empirical graphs is inconsistent with that of the questionnaire graph.
[14] This benchmark is a default value and can be changed in the algorithm/testing procedure at any time depending on the nature of the survey and the intended analysis, e.g., increased to 100 cases.

wrongly programmed skip instructions or that there was no one in the sample who fulfilled the entry conditions for the question. In the latter case, there is no issue. Test case *V2* examines the opposite case: whether there are *additional vertices*. To identify additional vertices, it checks whether there are more vertices in the empirical sample graph than in the questionnaire graph, again by comparing the number of vertices in both graphs. Additional vertices are caused by questions in the survey data that have no equivalent in the questionnaire (template). Testing of the test cases *E1, missing edges,* and *E2, additional edges,* is analogous to the testing of *V1* and *V2*. Missing edges point to skip instructions that were either not programmed or forgotten. Additional edges are an indicator of wrongly programmed skip instructions, leading the respondents to the wrong follow-up questions. Using the results of test cases *V1*, *V2*, *E1,* and *E2*, the problem of impossible paths (*FS1*) can be addressed and the questionnaire adapted accordingly. If at any time the testing procedure reveals problems with the questionnaire, i.e., with its design or programming or both, it must be revised in view of the errors and problems found. After revision, the test strategy should always be repeated in order to detect and, if necessary, correct any subsequent errors.

Table 3: Test plan.

Test Case ID	Test Case Specification	Preconditions	Test Data Graphs	Attributes	Measure	Steps	Exemplifications	Implications
Filter Structure (FS)								
FS1	Impossible paths	-	Question-naire graph and empirical sample graph	Graph attribute "Filter Pattern"	path(s) 0 ∈ Filter Pattern	1. Check whether the filter pattern of the empirical sample graph includes a "path(s) 0"	Path 0 subsumes impossible paths, i.e., paths in the empirical sample graph that do not occur in the questionnaire graph	*Questionnaire Programming:* There are deviations between the programming template and the actual programmed questionnaire *Next steps*: Continue with test case V1
FS2	Incorrect filter programming	-	Question-naire graph and empirical individual graphs	Edge attributes "All Possible Answer Options"/ "Given Answer"	$E_{empirical\ sample}$[Given Answer] ∉ $E_{questionnaire}$[All Possible Answer Options]	1. Check whether there are edges in the empirical individual graph connecting the correct vertices but its edge attributes "Given Answer" do not conform with the edge attributes "All possible Answer Options" in the questionnaire graph by comparing these two graphs 2. Identify these edges	-	*Questionnaire Programming:* Wrongly programmed skip instructions lead to the correct follow-up questions but guide the wrong sub-sample to these questions *Next steps*: Stop testing procedure & correct questionnaire programming
Paths (P)								
P1	Too-low path frequencies	No impossible paths and no incorrect filter pro-gramming	Empirical sample graph	Graph attribute "Path Distribu-tion"	Path(s) frequency < benchmark value	1. Check whether single path frequencies fall below a pre-specified benchmark value (default value is n=30)	Benchmark value depends on study population and intended analysis	*Questionnaire Design:* Potentially redundant paths and too-low case numbers hinder feasible statistical analyses of sub-populations *Next steps*: Stop testing procedure & rethink/adapt questionnaire design

Authors' own representation.

Table 3: Test plan (cont'd).

Test Case ID	Test Case Specification	Preconditions	Test Data Graphs	Test Data Attributes	Measure	Steps	Exemplifications	Implications
Vertices (V)								
V1	Missing vertices	Impossible paths	Empirical sample graph and question-naire graph	-	$\lvert V_{empirical\ sample}\rvert < \lvert V_{questionnaire}\rvert$	1. Check whether there are fewer vertices in the empirical sample graph than in the questionnaire graph by comparing the number of vertices of both graphs 2. Identify these vertices	-	*Questionnaire Programming:* Questions that are missing or could not be answered are due to either wrongly programmed skip instructions or inadequate sample *Next steps*: Stop testing procedure & correct questionnaire programming
V2	Additional vertices		Empirical sample graph and question-naire graph	-	$\lvert V_{empirical\ sample}\rvert > \lvert V_{questionnaire}\rvert$	1. Check whether there are more vertices in the empirical sample graph than in the questionnaire graph by comparing the number of vertices of both graphs 2. Identify these vertices	-	*Questionnaire Programming:* Questions should not be included in the questionnaire *Next steps*: Stop testing procedure & correct questionnaire programming
Edges (E)								
E1	Missing edges	Impossible paths	Empirical sample graph and question-naire graph	-	$\lvert E_{empirical\ sample}\rvert < \lvert E_{questionnaire}\rvert$	1. Check whether there are fewer edges in the empirical sample graph than in the questionnaire graph by comparing the number of edges of both graphs 2. Identify these edges	-	*Questionnaire Programming:* Skip instructions were not programmed or were forgotten *Next steps*: Stop testing procedure & correct questionnaire programming
E2	Additional edges		Empirical sample graph and question-naire graph	-	$\lvert E_{empirical\ sample}\rvert > \lvert E_{questionnaire}\rvert$	1. Check whether there are more edges in the empirical sample graph than in the questionnaire graph by comparing the number of edges of both graphs 2. Identify these edges	-	*Questionnaire Programming:* Wrongly programmed skip instructions lead to the wrong follow-up questions *Next steps*: Stop testing procedure & correct questionnaire programming

Authors' own representation.

To examine the applicability and suitability of our testing strategy for surveys, we have applied it to the two scenarios in our example (see Section 2, Scenario 1 "wrong questionnaire programming" and Scenario 2 "path redundancy"). Table 4 summarizes the results. The first column gives the test case ID and the test case specification. As additional information, the second line in the table shows the filter pattern and the corresponding path frequencies (i.e., how often each path was traversed by the respondents in the example). Tick marks indicate that the conditions in the Measure column (of Table 3) apply (i.e., the test procedure revealed issues in the questionnaire). In contrast, an X indicates that no issues have been detected. Dashes indicate that the corresponding test case was not executed because its preconditions were not met. The second-to-last line in Table 4 summarizes the results of the test run, and the last line indicates whether an issue was found at all.

Table 4: Identification of errors of Scenario 1 and Scenario 2.

Test Case ID: **Test Case Specification**	**Scenario 1**	**Scenario 2**
Filter pattern	0 – 3 – 4 – 5	1 – 2 – 3 – 4 – 5
Path frequency	84-33-50-33	11- 8-43-78-60
Filter structure (FS)		
FS1: Impossible paths	✓	✕
FS2: Incorrect filter programming	-	✕
Paths (P)		
P1: Too-low path frequencies	-	✓
Vertices (V)		
V1: Missing vertices	✓	-
V2: Additional vertices	✕	-
Edges (E)		
E1: Missing edges	(✓)	-
E2: Additional edges	(✓)	-
Implication	There is one missing question in the questionnaire, namely question 5 about the smoking device (*smodev*). All further deviations are due to this missing question.	Path 1 and 2 have been found to be redundant because there are too few smokers in the sample.
Issue detected	✓	✓

Authors' own representation.

Examining *FS1* for Scenario 1 reveals that the empirical sample graph comprises paths that do not occur in the questionnaire graph, i.e., the related filter pattern includes a "path(s) 0" value. In total, the erroneous ("impossible") paths are traversed by 84 of the N=200 respondents (in this example). The comparison of the correct filter pattern of the questionnaire graph (1-2-3-4-5) with the filter pattern in Scenario 1 (0-3-4-5) shows that "path(s) 0" replaces paths 1 and 2. The problem therefore lies in both of these paths. To examine which issue caused the erroneous path in Scenario 1, test case *V1* is executed. We find that in the questionnaire in Scenario 1, one vertex is missing. A closer look at the questionnaire graph shows that question 5 on the smoking device (*smodev*) is missing. Examining the test cases *V2*, *E1*, and *E2* reveals no further issues. Two missing edges (namely, *smoker* → *smodev* & *smodev* → *smofreq*) and one additional edge (*smoker* → *smofreq*) are due to the missing vertex "*smodev*".

After Scenario 1, we apply our test procedure to Scenario 2. The test cases *FS1* and *FS2* do not indicate any problems with the filter structure or the filter pattern. Examining *P1* reveals that the path frequencies of path 1 and path 2 fall below the default benchmark value of n=30, meaning that very few respondents traversed these paths. Both paths concern smokers. This means that the sample contains too few smokers and the design of the questionnaire is not adequate for the sample, or the sample is not adequate for the questionnaire.

Testing of the two scenarios shows that our test procedure works efficiently and effectively.

4 APPLICATIONS TO SURVEYS

The first empirical example investigates a (short) questionnaire module about private tutoring that was used in NEPS[15] Starting Cohort 2 "Kindergarten" (SC2) in wave 7 in 2016/17. With this module, parents of 10-11 year old children who received private tutoring in 2016/17 were asked about the subjects, the scope, and the contexts of the private tutoring, with a special focus on the school subject of German (see Aust, von der Burg, and Prussog-Wagner 2017). The questionnaire module contains 16 questions, including two filter questions. Table 5 shows the questions corresponding to the variable names. Figure S5 in the supplement shows its Nassi-Shneiderman diagram. Table 6 shows the progression table of the questionnaire module. From this progression table, a questionnaire graph is derived (see Figure 4, left hand side). It consists of 18 vertices and 19 edges. In total, there are three possible paths through the questionnaire graph, resulting in seven possible filter patterns, see Table 7.

[15] *This paper uses data from the National Educational Panel Study (NEPS): Starting Cohort Kindergarten, doi:10.5157/NEPS:SC2:7.0.0 . From 2008 to 2013, NEPS data were collected as part of the Framework Program for the Promotion of Empirical Educational Research funded by the German Federal Ministry of Education and Research (BMBF). As of 2014, NEPS is carried out by the Leibniz Institute for Educational Trajectories (LIfBi) at the University of Bamberg in cooperation with a nationwide network.*

Table 5: Assignment of the questions to the variable names in the scientific use files of NEPS (SC2) (LIfBi 2018).

Variable name	Question
p261100	Now I would like to move on to the topic of private tutoring. Does <target child's name> currently receive private tutoring?
p262101	And in what subjects is <target child's name> receiving private tutoring? – Mathematics
p262102	And in what subjects is <target child's name> receiving private tutoring? – German
p262103	And in what subjects is <target child's name> receiving private tutoring? – English
p262104	And in what subjects is <target child's name> receiving private tutoring? – French
p262105	And in what subjects is <target child's name> receiving private tutoring? – Latin
p262106	And in what subjects is <target child's name> receiving private tutoring? – Physics
p262107	And in what subjects is <target child's name> receiving private tutoring? – Chemistry
p262108	And in what subjects is <target child's name> receiving private tutoring? – Biology
p262109	And in what subjects is <target child's name> receiving private tutoring? – Other subject/subjects
pd0100n	What is the focus of your tutoring in German? – Spelling
pd0200n	What is the focus of your tutoring in German? – Reading and understanding texts
pd0300n	What is the focus of your tutoring in German? – Writing texts
pd0400n	What is the focus of your tutoring in German? – Speaking and oral comprehension
pd0500n	What is the focus of your tutoring in German? – Grammar
P261101	And how many hours in total per week does this private tutoring add up to in a normal school week?

Authors' own representation.

Table 6: Progression table for the NEPS questionnaire module "Private Tutoring".

ROW	FROM	FILTER	ANSWER OPTIONS	TO
1	BEGIN	ALL		p261100
2	p261100	p261100 = 1, -20	Yes; Child is receiving irregular private tutoring	p262101
3	p261100	p261100 = 2, -97, -98	No; Refused; Don't know	END
4	p262101	ALL	not specified; Specified; Refused; Don't know	p262102
5	p262102	ALL	not specified; Specified; Refused; Don't know	p262103
6	p262103	ALL	not specified; Specified; Refused; Don't know	p262104
7	p262104	ALL	not specified; Specified; Refused; Don't know	p262105
8	p262105	ALL	not specified; Specified; Refused; Don't know	p262106
9	p262106	ALL	not specified; Specified; Refused; Don't know	p262107
10	p262107	ALL	not specified; Specified; Refused; Don't know	p262108
11	p262108	ALL	not specified; Specified; Refused; Don't know	p262109
12	p262109	p262102 = 1	Specified	pd0100n
13	p262109	p262102 = 0, -97, -98	not specified; Refused; Don't know	p261101
14	pd0100n	ALL	not specified; Specified; Refused; Don't know; Don't want to talk about it	pd0200n
15	pd0200n	ALL	not specified; Specified; Refused; Don't know; Don't want to talk about it	pd0300n
16	pd0300n	ALL	not specified; Specified; Refused; Don't know; Don't want to talk about it	pd0400n
17	pd0400n	ALL	not specified; Specified; Refused; Don't know; Don't want to talk about it	pd0500n
18	pd0500n	ALL	not specified; Specified; Refused; Don't know; Don't want to talk about it	p261101
19	p261101	ALL	0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; 65; 66; 67; 68; 69; 70; 71; 72; 73; 74; 75; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 89; 90; 91; 92; 93; 94; 95; 96; 97; 98; 99; Refused; Don't know; Child is receiving irregular private tutoring	END

Authors' own representation.

Table 7: All possible paths through the questionnaire (graph) of the NEPS questionnaire module "Private Tutoring".

Path Number	Path (indicated by the sequence of variable names of the questions passed)
1	BEGIN→p261100→p262101→p262102→p262103→p262104→p262105→p262106→p262107→ p262108→p262109→pd0100n→pd0200n→pd0300n→pd0400n→pd0500n→p261101→END
2	BEGIN→p261100→p262101→p262102→p262103→p262104→p262105→p262106→p262107→ p262108→p262109→p261101→END
3	BEGIN→p261100→END

Authors' own representation.

In sum, N=379 parents answered the module (out of N=4,356 who were asked whether their child received private tutoring). From their answers, we constructed the empirical individual graphs and the empirical sample graph (see Figure 4, right hand side). The empirical sample graph consists of 14 vertices and 15 edges.

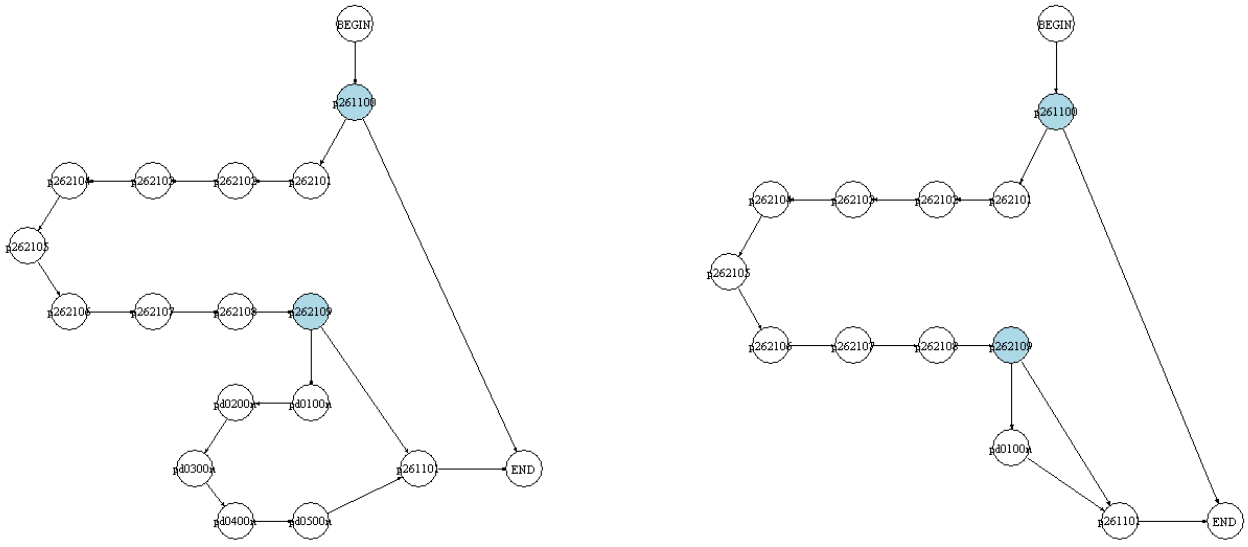

Figure 4: Questionnaire graph (left) and empirical sample graph (right) of the NEPS questionnaire module "Private Tutoring".
Authors' own representation.

Thereafter, we examined the module using the test strategy introduced in Section 3. Table 8 summarize the test results. *FS1* shows that the empricical sample graph contains impossible paths, i.e., the filter pattern includes “path(s) 0”. We find that in total, 215 of all respondents traversed the impossible path(s). Since path 1 is not part of the empirical filter pattern, the error lies in this path. Test case *V1* reveals that four vertices are missing in the empirical graph(s) as compared to the questionnaire graph (namely, pd0200n, pd0300n, pd 0400n and pd0500n). Further issues of missing and additional edges (indicated by *E1* and *E2*) are due to these missing vertices. (In Table 8, this dependency is marked by tick marks in paratheses). The missing vertices are items of an item battery that no respondent answered in the empirical data although the questions should have been asked according to the questionnaire template. Thus, our approach revealed differences between the questionnaire template and the actual empirical data collected.

The second empirical application uses the module "Jobs and Money" from the youth questionnaire of the SOEP. The testing procedure and its results are detailed in supplement S6. We found that the questionnaire does not match the sample.

Table 8: Identification of errors of the NEPS questionnaire module "Private Tutoring".

Test Case ID: **Test Case Specification**	**Outcome**
Filter pattern	0-2-3
Path frequency	215-164-3977
Filter structure (FS)	
FS1: Impossible paths	✓
FS2: Incorrect filter programming	-
Paths (P)	
P1: Too-low path frequencies	-
Vertices (V)	
V1: Missing vertices	✓
V2: Additional vertices	✗
Edges (E)	
E1: Missing edges	(✓)
E2: Additional edges	(✓)
Implication	There are four missing questions in the questionnaire, namely questions about the context of the school subject German. All further deviations are due to these missing questions.
Issue detected	✓

Authors' own representation.

5 Discussion and Conclusion

In this article, we present a procedure to automatically check questionnaire modules from (large-scale) surveys for programming errors in computer-based questionnaires and for the fit of the questionnaire to the sample under investigation. To this end, we use mathematical graphs and a test plan. The use of mathematical graphs in this context is not new, but the combination with a test plan is. The test plan enables us to conduct a structured examination of the functionalities that a questionnaire should fulfil. These functionalities are derived from the requirements of a questionnaire (e.g., its correct technical implementation). This means that before our test procedure can be applied, a list of requirements must be drawn up. This paper does not aim at checking the quality (e.g., the reliability or validity) of the questions contained in the questionnaire. Rather, our aim is to provide an efficient method of checking whether a

questionnaire that has been implemented is consistent with a questionnaire template (of whatever type). In addition, the use of mathematical graphs allows for the presentation of the entire structure of questionnaires and an easy-to-understand comparison with the final data. Our testing approach is complete, portable, and scalable. Completeness results from the use of mathematical graphs as a formal language for the holistic description of questionnaire structures. Portable means that our approach is not limited to a specific questionnaire or type of questionnaire, or to a specific software system. Again, this is made possible through the use of the very flexible formal language of mathematical graphs for describing questionnaires. Similarly, the implementation of our test procedure in the free software R means that our novel approach is available and accessible to anyone (link provided in the Introduction under footnote 2). Our approach is also scalable because it is modular. Large questionnaires can usually be broken down into smaller questionnaire modules. The functionality of these modules can then be tested with our approach in a first step. In a second step, the questionnaire modules are re-connected and a further test is carried out at an aggregated level. This kind of testing also makes our approach scalable. The use of mathematical graphs also makes our approach scalable: Even complex questionnaires and thus questionnaire structures can be examined efficiently and effectively using suitable (scalable) algorithms.

Using a hypothetical example and two real-world examples (using NEPS and SOEP questionnaires) we demonstrated the functionality and capacity of our test procedure. Our application of the procedure to real-world examples shows that even with established surveys like those in our examples, problems still occur with questionnaire implementation and fit to the sample, which probably could have been identified or corrected in advance. The problem with large studies such as NEPS and SOEP is the trade-off between a very broad and complex range

of questions on the one hand, and time and personnel constraints on the other, which make it difficult to fully test the range of questions and their programming (at least if this is done in large part manually). This is where our approach should be used more in the future.

A unique feature of our approach is that it extends the perspective of the Total Survey Error Framework (Groves et al. 2009) by introducing the broader perspective of a general test system. Our test system aims not only to minimize the TSE (by eliminating programming errors), but also to improve survey questionnaires in general (by comparing them with empirical survey data). Furthermore, the requirement of portability makes our testing approach generally usable and accessible. Thus, in addition to the TSE framework, our work is also oriented toward the fitness-of-use approach of Biemer and Lyberg (2003). The overall objective of any survey data provider is to generate high-quality survey data with high data analysis potential. Following the idea of the Total Quality Management (Lyberg 2012), this requires continuous review and adaptation of the data generation system. Both are essential for longitudinal large-scale studies such as SOEP and NEPS. Meeting the requirement of high-quality survey data with a high analysis potential across all survey waves is one of the main objectives of our testing approach.

Our work is not yet finished: So far, we have only applied our novel test procedure to small applications (or survey modules). It has not been put to use fully for questionnaire testing in large surveys—but this is planned for the SOEP. Of course, implementation will not be without challenges, and adjustments to our test approach will undoubtedly be necessary: The approach itself will be subjected to (customer) testing through this large-scale application. Nevertheless, we consider the application of the automated test procedure presented here to be necessary to prevent past mistakes being made in future surveys (e.g., the incorrectly programmed NEPS questionnaire module for private tutoring). Furthermore, the sometimes excessively long

questionnaires used in some surveys could be shortened, which would lead to a lower burden on respondents and (hopefully) to a higher willingness to participate at the unit and item level.

In addition, the number of test cases must still be increased in a sensible fashion. We are also constantly looking for further applications for our testing approach.

References

Ammann, P., and Offutt, J. (2016), *Introduction to Software Testing* (2nd ed.), Cambridge, New York, Melbourne.

Aust, F., von der Burg, J., and Prussog-Wagner, A. (2017), "Methodenbericht: NEPS-Startkohorte 2 (Elternbefragung) – Haupterhebung Frühjahr 2017 B120.," infas Institut für angewandte Sozialwissenschaft GmbH. Available at *https://www.neps-data.de/Portals/0/NEPS/Datenzentrum/Forschungsdaten/SC2/7-0-0/NEPS_FieldReport_SC2_W7_CATI.pdf*

Bethlehem, J., and Hundepool, A. (2004), "TADEQ: A Tool for the Documentation and Analysis of Electronic Questionnaires," *Journal of Official Statistics* [online], 20, 233–264. Available at *https://www.scb.se/contentassets/f6bcee6f397c4fd68db6452fc9643e68/tadeq-a-tool-for-the-documentation-and-analysis-of-electronic-questionnaires.pdf*

Biemer, P., and Lyberg, L. (2003), *Introduction to Survey Quality*, Hoboken, New Jersey, DOI: 10.1002/0471458740. Available at *http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10304975*

Elliott, S. (2012), "The Application of Graph Theory to the Development and Testing of Survey Instruments," *Survey Methodology* [online], 38, 11–21. Available at *https://www150.statcan.gc.ca/n1/pub/12-001-x/2012001/article/11681-eng.pdf*

Fagan, J., and Greenberg, B. V. (1988), "Using Graph Theory to Analyze Skip Patterns in Questionnaires," (Statistical Research Division Report Series). Washington D.C.: Bureau of the Census. Available at *https://www.census.gov/srd/papers/pdf/rr88-06.pdf*

Feeney, G., and Feeney, S. (2019), "On the Logical Structure of Census and Survey Questionnaires," *Genus* [online], 75, DOI: 10.1186/s41118-019-0065-y. Available at *https://link.springer.com/content/pdf/10.1186/s41118-019-0065-y.pdf*

Goebel, J., Grabka, M. M., Liebig, S., Kroh, M., Richter, D., Schröder, C., and Schupp, J. (2019), "The German Socio-Economic Panel (SOEP)," *Journal of Economics and Statistics* [online], 239, 345–360, DOI: 10.1515/jbnst-2018-0022. Available at *https://doi.org/10.1515/jbnst-2018-0022*

Groves, R. M., Fowler Jr., F. J., Couper, M. P., Lepkowski, J. M., Singer, E., and Tourangeau, R. (2009), *Survey Methodology* (2nd ed.), Hoboken, New Jersey. Available at *http://gbv.eblib.com/patron/FullRecord.aspx?p=819140*

Hansen, S. E., and Couper, M. P. (2004), "Usability Testing to Evaluate Computer-Assisted Instruments," In *Methods for Testing and Evaluating Survey Questionnaires*, eds. S. Presser, J. M. Rothgeb, M. P. Couper, J. T. Lessler, E. Martin, J. Martin and E. Singer, Hoboken, New Jersey: John Wiley & Sons, Inc, pp. 337–360, DOI: 10.1002/0471654728.ch17.

International Organization for Standardization (2015), "Quality Management Systems-Fundamentals and Vocabulary (ISO 9000: 2015)," ISO Copyright office. Available at *https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en*

Kawai, W., Mukuta, Y., and Harada, T. (2019), "Scalable Generative Models for Graphs with Graph Attention Mechanism,". Available at *http://arxiv.org/pdf/1906.01861v2*

Levinsohn, J. R., and Rodriguez, G. (2001), "Automated Testing of Blaise Questionnaires," In *Proceedings of the 7th International Blaise Users Conference*, pp. 1–12.

LIfBi (2018), "Questionnaires (SUF Versions): NEPS Starting Cohort 2 — Kindergarten: From Kindergarten to Elementary School. Wave 7 - 7.0.0," Research Data, Leibniz Institute for Educational Trajectories. Available at *https://www.neps-data.de/Portals/0/NEPS/Datenzentrum/Forschungsdaten/SC2/7-0-0/SC2_7-0-0_W7_en.pdf*

Lyberg, L. (2012), "Survey Quality," *Survey Methodology* [online], 38, 107–130. Available at *https://www150.statcan.gc.ca/n1/en/pub/12-001-x/2012002/article/11751-eng.pdf?st=7xXtj0UC*

Mathur, S., and Malik, S. (2010), "Advancements in the V-Model," *International Journal of Computer Applications* [online], 1, 29–34. Available at *https://www.ijcaonline.org/journal/number12/pxc387425.pdf*

Nassi, I., and Shneiderman, B. (1973), "Flowchart Techniques for Structured Programming," *ACM SIGPLAN Notices* [online], 8, 12–26, DOI: 10.1145/953349.953350. Available at *https://www.researchgate.net/publication/234805404_Flowchart_techniques_for_structured_programming*

Ould, M. A., and Unwin, C. (1986), *Testing in Software Development*, Cambridge.

R Core Team (2020). R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing, Vienna, Austria. Available at *https://www.R-project.org/*

Rodriguez, G., and Levinsohn, J. R. (2003), "A Random Walk Application for Blaise Instruments," In *Proceedings of the 8th International Blaise Users Conference* [online], pp. 175–181. Available at *http://www.blaiseusers.org/2003/papers/ibuc2003-proceedings.pdf*

Shneiderman, B. (2003), *A Short History of Structured Flowcharts (Nassi-Shneiderman Diagrams)*. Available at *https://www.cs.umd.edu/hcil/members/bshneiderman/nsd/*

Statistics Sweden (2004), "Design Your Questions Right: How to Develop, Test, Evaluate and Improve Questionnaires," Statistics Sweden. Available at *https://ec.europa.eu/eurostat/documents/7330775/7339614/DESIGN/7991980c-efc1-4b8e-87b8-b09aac449c19*

J. M. Rothgeb, M. P. Couper, J. T. Lessler, E. Martin, J. Martin and E. Singer, Hoboken, New Jersey: John Wiley & Sons, Inc, pp. 319–335, DOI: 10.1002/0471654728.ch16. Available at *https://onlinelibrary.wiley.com/doi/10.1002/0471654728.ch16*

Zhang, Z., Cui, P., and Zhu, W. (2015), "Deep Learning on Graphs: A Survey," *Journal of LATEX Class Files* [online], 14, 1–24. Available at *http://arxiv.org/pdf/1812.04202v3*

Supplementary material

Electronic source

Online link (to GitHub repository) to R programs implementing the testing procedure presented in this article:

https://github.com/KatharinaStark/MathematicalGraphsForQuestionnaireTesting

S1 Description of Nassi-Shneiderman diagram of illustrative, hypothetical example

The Nassi-Shneiderman diagram is a flowchart depicting all questions $Q_i, = 1, \ldots, 10$ (with variable names in parentheses) in sequence with their possible answers (the corresponding codes are given before the colons). The processing order of the questions is represented by blocks presented in vertical succession. Three of the ten questions are filter questions (Q4, Q7 and Q8, grey shaded). The related skip instructions follow from the order in the flowchart, with answer categories leading to specific questions (or not). Each respondent follows a path through the blocks of the diagram from top to bottom. In doing so, the respondent can only cross horizontal lines but not vertical lines. If a respondent runs into an empty block, she/he proceeds to the next non-empty block below. To keep the example simple, we excluded special data structures, e.g., loops.

The Nassi-Shneiderman diagram is clear and shows the structure of the questionnaire, but the diagram is not well suited as input for a test algorithm due to its graphical elements.

S2 Distribution of sample attributes in the hypothetical example

Table S2: Distribution of sample attributes in the example.

Variable	$N_{Baseline\ Scenario}$	$\%_{Baseline\ Scenario}$	$N_{Scenario\ 1}$	$\%_{Scenario\ 1}$	$N_{Scenario\ 2}$	$\%_{Scenario\ 2}$
V1: Sex						
1: male	95	47.5	95	47.5	99	49.5
2: female	105	52.5	105	52.5	101	50.5
V2: Highest educational achievement						
1: no school-leaving qualification	46	23.0	46	23.0	24	12.0
2: school-leaving certificate from a special needs school	32	16.0	32	16.0	36	18.0
3: Hauptschulabschluss[(i)]	35	17.5	35	17.5	50	25.0
4: Mittlere Reife[(ii)]	38	19.0	38	19.0	62	31.0
5: Abitur[(iii)]	49	24.5	49	24.5	28	14.0
V3: Subjective social class						
1: lower class	73	36.5	73	36.5	76	38.0
2: middle class	66	33.0	66	33.0	85	42.5
3: upper class	61	30.5	61	30.5	39	19.5
V4: Current smoker						
1: yes	84	42.0	84	42.0	19	9.5
2: no	83	41.5	83	41.5	121	60.5
3: no answer	33	16.5	33	16.5	60	30.0
V5: Smoking device						
1: cigarettes	41	20.5	0	0.00	11	5.50
2: cigars	22	11.0	0	0.00	2	1.00
3: e-cigarettes/vaporizer	21	10.5	0	0.00	6	3.00
missing	116	58.0	200	100.0	181	90.5
V6: Smoking frequency						
1: irregular/less than once a week	21	10.5	21	10.5	7	3.50
2: once/several times a week	47	23.5	47	23.5	11	5.50
3: several times a day	16	8.00	16	8.00	1	0.50
missing	116	58.0	116	58.0	181	90.5
V7: Ever tried stop smoking						
1: yes	46	23.0	46	23.0	11	5.50
2: no	38	19.0	38	19.0	8	4.00
missing	116	58.0	116	58.0	181	90.5
V8: Ever smoked						
1: yes	33	16.5	33	16.5	43	21.5
2: no	50	25.0	50	25.0	78	39.0
missing	117	58.5	117	58.5	79	39.5
V9: Why tried to stop/stopped smoking						
1: health problems	20	10.0	20	10.0	2	1.00
2: too expensive	59	29.5	59	29.5	52	26.0
3: other	0	0.00	0	0.00	0	0.00
missing	121	60.5	121	60.5	146	73.0
V10: General state of health						
1: good	66	33.0	66	33.0	51	25.5
2: poor	101	50.5	101	50.5	89	44.5
missing	33	16.5	33	16.5	60	30.0

Notes: (i) School-leaving qualification from the Hauptschule, which is a school for basic secondary education in Germany. (ii) School-leaving qualification from the Realschule, which is an intermediate secondary school in Germany. (iii) "Abitur" is entrance qualification for universities.

S3 Nassi-Shneiderman diagram for scenario 1 of hypothetical example

<table>
<tr><td colspan="4">BEGIN</td></tr>
<tr><td colspan="4">Q1: Sex? (sex)
1: male; 2: female</td></tr>
<tr><td colspan="4">Q2: Highest educational achievement? (edu)
1: no school-leaving qualification; 2: school-leaving qualification from a special needs school;
3: Hauptschulabschluss; 4: Mittlere Reife; 5: Abitur</td></tr>
<tr><td colspan="4">Q3: Subjective social class? (class)
1: lower class; 2: middle class; 3: upper class</td></tr>
<tr><td colspan="2">Q4: Current smoker? (smoker)
1: yes</td><td>2: no</td><td>3: no answer</td></tr>
<tr><td colspan="2" rowspan="3">Q6: Smoking frequency? (smofreq)
1: irregular/less than once a week;
2: once/several times a week;
3: several times a day</td><td>Q8: Ever smoked? (smoever)
1: yes</td><td rowspan="4">2: no</td><td rowspan="4"></td></tr>
<tr><td colspan="2">Q7: Ever tried stop smoking? (smostop)
2: no</td><td>1: yes</td></tr>
<tr><td></td><td colspan="2">Q9: Why tried to stop/stopped smoking? (smostopr)
1: health problems; 2: too expensive;
3: other reasons</td></tr>
<tr><td colspan="3">Q10: General state of health? (subhealth)
1: good; 2: poor</td></tr>
<tr><td colspan="4">END</td></tr>
</table>

Figure S3: Nassi-Shneiderman diagram of (erroneous) hypothetical example Scenario 1. *Authors' own representation.*

S4 derivation of the empirical graphs from survey data

We have developed a procedure that allows deriving empirical graphs from survey data. That procedure is implemented in R (the related source code is part of the R programs we provide in the GitHub repository to that article, see footnote 2 in the main text). Under the following three conditions, any kind of rectangular, cross-sectional dataset can automatically be converted into empirical graphs:

1. The dataset contains only variables and answer options that are also included in the progression table of the questionnaire. (This does not apply to identification variables, i.e., IDs.)
2. The order of the variables in the progression table and the dataset coincides.
3. All missing values that are missing due to the filters or due to the study design are marked as system missings, e.g., an NA in the programming language R. (We thus do not generate vertices and/or edges in the empirical graph for variables/values that are missing due to design or filtering.) To this end, response codes are replaced by the variable names; the key words BEGIN and END are added to the data frame; and in each line, NAs are replaced with the next non-NA value.

Figure S4 shows an extract of a dataset that has been transformed in this way and fits our simple example. (Identification variables are given at the beginning of each data line. They do not appear in the progression table or in the empirical graph. The IDs are only used to describe data lines (with potential defects).) In order to derive the empirical graphs from the described dataset, four steps are necessary. First, a progression table is created for each row in the dataset (related to a single respondent). From this table, the corresponding *empirical individual graphs* are derived (resulting in one graph for each respondent in the dataset; in R there are stored in a list of

length N). Second, we compress these empirical individual graphs into a single graph by combining all N individual graphs, and thus all vertex and edge attributes, into one single graph. We call this graph the *empirical sample graph*. This graph contains all vertices and edges that were traversed at least once by any respondent. Third, we derive all possible paths through the empirical sample graph and the corresponding filter pattern. (Note that all possible paths through the empirical sample graph correspond to all actual paths traversed by the respondents.) Fourth, we assign different attributes to the vertices and edges of the empirical sample graph and empirical individual graphs as well as to the graphs themselves. Specifically, in case of the empirical individual graphs, we assign the filter instructions and the given answers as attributes to the edges. Respondent IDs, path numbers, and the actual filter pattern are stored in the empirical individual graphs as graph attributes. In case of the empirical sample graph, we store variable distributions from the individual survey data as attributes in the vertices. The summary of all paths in all individual graphs and the filter pattern are stored as graph attributes in the empirical sample graph.

ID	BEGIN	sex	edu	class	smoker	smodev	smofreq	smostop	smoever	smostopr	subhealth	END
1	BEGIN	sex	edu	class	smoker	smoever	smoever	smoever	smoever	subhealth	subhealth	END
2	BEGIN	sex	edu	class	smoker	smodev	smofreq	smostop	subhealth	subhealth	subhealth	END
3	BEGIN	sex	edu	class	smoker	END	END	END	END	END	END	END
4	BEGIN	sex	edu	class	smoker	END	END	END	END	END	END	END
5	BEGIN	sex	edu	class	smoker	smoever	smoever	smoever	smoever	subhealth	subhealth	END
6	BEGIN	sex	edu	class	smoker	smoever	smoever	smoever	smoever	smostopr	subhealth	END
7	BEGIN	sex	edu	class	smoker	END	END	END	END	END	END	END
8	BEGIN	sex	edu	class	smoker	smodev	smofreq	smostop	smostopr	smostopr	subhealth	END
9	BEGIN	sex	edu	class	smoker	smoever	smoever	smoever	smoever	smostopr	subhealth	END

Figure S2: Data structure (mapping the dataset of the baseline scenario) to derive the empirical individual graphs and the sample graph.
Authors' own representation.

S5 Nassi-Shneiderman diagram of NEPS questionnaire module "Private Tutoring"

<table>
<tr><td colspan="3">BEGIN</td></tr>
<tr><td colspan="2">p261100
1: Yes; -20: Child is receiving irregular private tutoring</td><td>2: No; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262101
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td><td rowspan="12"></td></tr>
<tr><td colspan="2">p262102
1: Specified
0: Not specified; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262103
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262104
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262105
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262106
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262107
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262108
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td></tr>
<tr><td colspan="2">p262109
0: Not specified; 1: Specified; -97: Refused; -98: Don't know</td></tr>
<tr><td>pd0100n
0: Not specified; 1: Specified; -97: Refused; -98: Don't know; -20: none of it</td><td rowspan="5"></td></tr>
<tr><td>pd0200n
0: Not specified; 1: Specified; -97: Refused; -98: Don't know; -20: none of it</td></tr>
<tr><td>pd0300n
0: Not specified; 1: Specified; -97: Refused; -98: Don't know; -20: none of it</td></tr>
<tr><td>pd0400n
0: Not specified; 1: Specified; -97: Refused; -98: Don't know; -20: none of it</td></tr>
<tr><td>pd0500n
0: Not specified; 1: Specified; -97: Refused; -98: Don't know; -20: none of it</td></tr>
<tr><td colspan="2">p261101
Hours per week; -97: Refused; -98: Don't know ; -20: Child is receiving irregular private tutoring</td></tr>
<tr><td colspan="3">END</td></tr>
</table>

Figure S5: Nassi-Shneiderman diagram of the NEPS questionnaire module "Private Tutoring". *Authors' own representation.*

S6 Testing he Youth Questionnaire of the German Socio-Economic Panel Study (SOEP)

The second empirical application uses a questionnaire module and data from the SOEP.[16] The questionnaire module stems from the 2019 youth questionnaire, in which adolescents born in 2002 were surveyed in the CAPI survey mode about their biography, with questions covering their living conditions and friends, siblings, and parents. The questionnaire module under investigation here is "Jobs and Money". Adolescents were asked to answer whether they had worked at a job in the seven days before the interview. Figure S6 shows the corresponding Nassi-Shneiderman diagram with the variable names of the questions. Table S3 shows the questions corresponding to the variable names. In sum, the questionnaire module comprises 14 questions, including five filter questions.

In the SOEP, questionnaires are created on the basis of metadata. These metadata contain the variable names, types, response options, and filter instructions in tabular form.[17] Using the SOEP metadata makes it straightforward to create the progression table for the "Jobs and Money" module (see Table S4). However, it should be kept in mind that the metadata may also contain errors. That is, problems that arise during questionnaire testing may also indicate errors in the metadata. Consequently, the test procedure described here can also be used to check for errors in the SOEP metadata.

[16] The SOEP has been conducted annually since 1984 at the German Institute for Economic Research (DIW Berlin). It interviews individuals in private households about the household situation in general, see Goebel et al. (2019).
[17] These data were obtained from two csv tables that are not yet officially available because the 2019 data and related documentation had not yet been published at the time of writing. We are grateful to the SOEP Research Data Center, which provided us with these metadata and the corresponding empirical data for testing our procedure even before releasing the 2019 SOEP data (SOEP v36). So far, there is nothing comparable in the NEPS, where questionnaires are created on the basis of Word templates. As a result, progression tables have to be created manually which is no way efficient and is also prone to errors.

BEGIN		
j7tag 1: Yes; 2: No; -1: No answer / don't know		
jnerw10 1: Yes; 2: No; -1: No answer / don't know		
jalo 1: Yes; 2: No; -1: No answer / don't know		
jjob1 1: Yes; 2: No; -1: No answer / don't know		
jbrutt I earned gross: __ euros; -1: No answer / don't know		
jnett I earned net: __ euros; -1: No answer / don't know		
jjob2 1: As a part-time employee; 2: As a trainee or intern; 3: As a regular full-time employee; -1 No answer / don't know		
	jjob3 1: Yes; 2: No; -1: No answer / don't know	
jjob4 I was __ years old; -1: No answer / don't know		
jjob5 1: The work interested me; 2: Wanted to earn money; 3: Other reasons; -1: No answer / don't know		
jtg1 1: Yes; 2: No; -1: No answer / don't know		
jtg2_jtg3 **jtg2:** __ euros per week; -1: No answer / don't know **jtg3:** __ euros per month; -1: No answer / don't know		
jspar1 1: Yes, occasionally; 2: Yes, regularly; 3: No; -1: No answer / don't know		
jspar2_jspar3 **jtg2:** About __ euros per month; -1: No answer / don't know **jtg3:** Can't say, it's very irregular; -1: No answer / don't know		
END		

Figure S6: Nassi-Shneiderman diagram of the SOEP questionnaire module "Jobs and Money". *Authors' own representation.*

Table S3: Assignment of the questions to the variable names of the SOEP youth questionnaire module "Jobs and Money".

Variable name	Question
j7tag	Have you done paid work during the last 7 days, even if only for an hour or a few hours?
jnerw10	Have you actively looked for work within the last four weeks?
jalo	Are you officially registered unemployed at the Employment Office (“Arbeitsamt”)?
jjob1	Do you already have a job to earn own money?
jbrut	How much did you earn from your work last month? Please state both gross income, which means income before deduction of taxes and social security, and net income, which means income after deduction of taxes, social security, and unemployment and health insurance. - I earned: … euros (gross)
jnet	How much did you earn from your work last month? Please state both gross income, which means income before deduction of taxes and social security, and net income, which means income after deduction of taxes, social security, and unemployment and health insurance. - I earned: … euros (net)
jjob2	Do you earn the money …
jjob3	Have you ever done side jobs to earn money?
jjob4	How old were you when starting doing side jobs or earning money? - I was … years old
jjob5	Did you start those jobs because you were interested in the work, or to earn some money?
jtg1	What about now? Do you get an allowance or regular financial support from your parents or other relatives?
jtg2	How much do you get in allowance per week / per month? - … euros per week or
jtg3	How much do you get in allowance per week / per month? - … euros per month
jspar1	Are you able to save some money regularly (for vacations, larger purchases, etc.)
jspar2	How much do you save per month approximately? - About … euros per month
jspar3	How much do you save per month approximately? - Can't say, it's very irregular

Authors’ own representation.

Table S4: Progression table for the SOEP questionnaire module "Jobs and Money".

ROW	FROM	FILTER	ANSWER OPTIONS	TO
1	BEGIN	ALL		j7tag
2	j7tag	ALL	Yes; No; -1	jnerw10
3	jnerw10	ALL	Yes; No; -1	jalo
4	jalo	ALL	Yes; No; -1	jjob1
5	jjob1	jjob1 = 1, -1	Yes; No; -1	jbrut
6	jjob1	jjob1 = 2	Yes; No; -1	jjob3
7	jbrut	ALL	-1; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; …	jnett
8	jnett	ALL	-1; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; …	jjob2
9	jjob2	jjob2 = 1, 2, -1	As a part-time employee; As a trainee or intern; As a regular full-time employee; -1	jjob3
10	jjob2	jjob2 = 3	As a part-time employee; As a trainee or intern; As a regular full-time employee; -1	jjob4
11	jjob3	jjob3 = 1, -1	Yes; No; -1	jjob4
12	jjob3	jjob3 = 2	Yes; No; -1	jtg1
13	jjob4	ALL	-1; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; …	jjob5
14	jjob5	ALL	The work interested me; Wanted to earn money; Other reasons; -1	jtg1
15	jtg1	jtg1 = 1, -1	Yes; No; -1	jtg2_jtg3
16	jtg1	jtg1 = 2	Yes; No; -1	jspar1
17	jtg2_jtg3	ALL	-1; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; …	jspar1
18	jspar1	jspar1 = 1, 2, -1	Yes, occasionally; Yes, regularly; No; -1	jspar2_jspar3
19	jspar1	jspar1 = 3	Yes, occasionally; Yes, regularly; No; -1	END
20	jspar2_jspar3	ALL	-1; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; …	END

Authors' own representation.

Note: -1 represents the residual category "No answer / don't know"

Figure S7 shows the questionnaire graph derived from the progression table in Table S4. It contains 16 vertices and 20 edges. In total, there are 20 possible paths through the questionnaire graph (see Table S5) resulting in 1,048,575 possible filter patterns.

In sum, N=201 adolescents answered this questionnaire module. Using the data, we derive the empirical individual graphs and the empirical sample graph. The empirical sample graph (not shown) has the same structure as the questionnaire graph, i.e., it also has 16 edges and 20 vertices connected in the same way as the vertices and edges of the questionnaire graph. We applied our testing procedure to the graphs to see whether there are any issues with the SOEP "Jobs and Money" questionnaire module. Table S6 summarizes the results. Test case *FS1* reveals no impossible paths. There is also no incorrect filter programming (*FS2*). Because there are no impossible paths, the test cases *V1*, *V2*, *E1*, and *E2* are omitted (indicated by a dash). Test case *P1* (path redundancy) follows with a default value of n=20 for a feasible path frequency. We find that most of the paths through the questionnaire were taken by only a few respondents. For example, paths 11 and 13 were only taken by one respondent each and paths 1, 3 and 7 only by two respondents each.

Table S5: All possible paths through the questionnaire (graph) and the corresponding path frequencies of the SOEP questionnaire module "Jobs and Money".

Path Number	Path (indicated by the sequence of variable names of the questions passed)	Path Frequency
1	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jjob4→jjob5→jtg1→jtg2_jtg3→ jspar1→jspar2_jspar3→END	2
2	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jjob4→jjob5→jtg1→jtg2_jtg3→ jspar1→END	0
3	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jjob4→jjob5→jtg1→jspar1→ jspar2_jspar3→END	2
4	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jjob4→jjob5→jtg1→jspar1→END	21
5	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jtg1→jtg2_jtg3→jspar1→ jspar2_jspar3→END	0
6	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jtg1→jtg2_jtg3→jspar1→END	6
7	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jtg1→jspar1→jspar2_jspar3→ END	2
8	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob3→jtg1→jspar1→END	4
9	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob4→jjob5→jtg1→jtg2_jtg3→jspar1→ jspar2_jspar3→END	1
10	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob4→jjob5→jtg1→jtg2_jtg3→jspar1→ END	13
11	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob4→jjob5→jtg1→jspar1→ jspar2_jspar3→END	1
12	BEGIN→j7tag→jnerw10→jalo→jjob1→jbrut→jnett→jjob2→jjob4→jjob5→jtg1→jspar1→END	29
13	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jjob4→jjob5→jtg1→jtg2_jtg3→jspar1→jspar2_jspar3→END	1
14	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jjob4→jjob5→jtg1→jtg2_jtg3→jspar1→END	4
15	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jjob4→jjob5→jtg1→jspar1→jspar2_jspar3→END	3
16	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jjob4→jjob5→jtg1→jspar1→END	3
17	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jtg1→jtg2_jtg3→jspar1→jspar2_jspar3→END	75
18	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jtg1→jtg2_jtg3→jspar1→END	20
19	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jtg1→jspar1→jspar2_jspar3→END	7
20	BEGIN→j7tag→jnerw10→jalo→jjob1→jjob3→jtg1→jspar1→END	7

Authors' own representation.

This indicates that the questionnaire does not match the sample. One implication is that it is either shortened or applied to another sample. Since the SOEP is a panel study, where no major changes in the composition of adolescents are to be expected in the following wave, we recommend shortening the questionnaire.

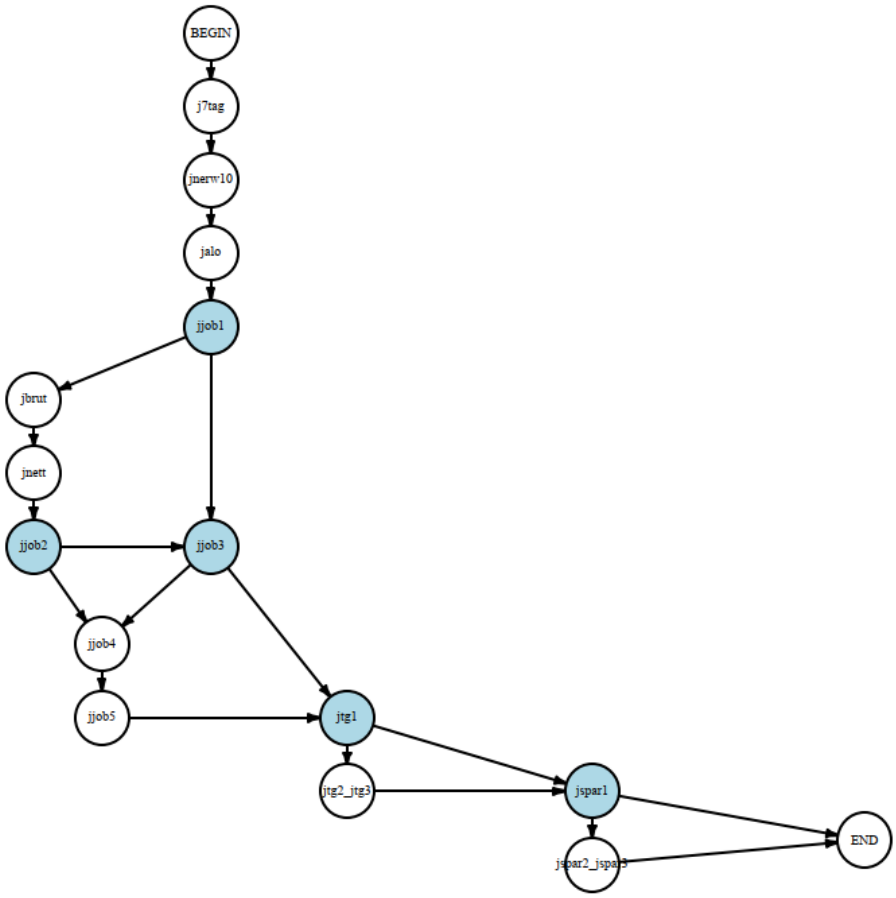

Figure S7: Questionnaire graph of SOEP questionnaire module "Jobs and Money".
Authors' own representation.

Table S6: Identification of errors of the SOEP youth questionnaire module "Jobs and Money".

Test Case ID: **Test Case Specification**	**Outcome**
Filter pattern	1-3-4-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
Path frequency	2-2-21-6-2-4-1-13-1-29-1-4-3-3-75-20-7-7
Filter structure (FS)	
FS1: Impossible paths	✕
FS2: Incorrect filter programming	✕
Paths (P)	
P1: Too-low path frequencies	✓
Vertices (V)	
V1: Missing vertices	-
V2: Additional vertices	-
Edges (E)	
E1: Missing edges	-
E2: Additional edges	-
Implication	There are many paths that have been found to be redundant.
Issue detected	✓

Authors' own representation.