Field Properties Values: A Tool to Identify and Adjust Missing Data from 'Relational' Extraction

Mohammad Mushtaq and April Beaulé, University of Michigan

1. Abstract

The Panel Study of Income Dynamics (PSID) has been using Blaise since 2003 and "ASCII-Relational" data export option to output data into SAS files. During the early stage of Blaise 5 development, the PSID applications development team has used reverse engineering to transform data from "wide" to "relational" format for "All Stars" pilot. Later, PSID staff worked closely with Blaise 5 development team to create tool to export Blaise 5 data into "relational" format, tested and provided feedback to Stats Netherlands during the development phase. The tool is now integrated with Blaise 5 Control Center and being used by the PSID in two surveys.

In a mixed-mode survey, it's important to identify the source and type of missing data values. The web instrument does not allow "Don't Know" and "Refused" whereas such values are allowed in CATI interviews. In web, the values could be missing due to "On-Route" (but not answered by the web respondent) vs. "Off-Route" (not presented to web respondent). Since the data from mixed-mode survey is delivered in a combined Blaise 5 data file, therefore, it's important to harmonize missing data values across both sources. This is done by using Field Properties Values file.

It was observed from pilot studies that Field Properties Values from "wide" are larger than that of "relational" extraction.

In this paper, we examine Field Properties Values from "wide" and "relational" Blaise 5 data extraction, identify missing observation and/or values from "relational" files. Develop methodology to complete the "relational" files where question was on-route but not answered by the respondent of web survey. This distinction is important for data processing team of the PSID and will be used to write explanation for missing values in the public release files. Also feedback to Statistics Netherlands about Blaise 5 "relational" export and possible improvement.

2. Introduction

The Panel Study of Income Dynamics (PSID) is a nationally representative longitudinal study of approximately 9,600 U.S. families. Since 2003, the PSID has used Blaise as its main software for its data collection. Due to the size and complexity of the instrument, PSID staff work with tables extracted in looped blocks or relational tables rather than a single flat file.

As the PSID moves from interviewer administered CATI data collection to self- administered web data collection, one of our significant challenges is to correctly document missing data values. PSID staff worked closely with the Statistics Netherlands team to create and refine the export option to generate relational tables. This extraction tool option is now integrated into the Blaise Control Center and is successfully being used in two PSID Blaise 5 surveys.

As a by-product of the reverse engineering method that used the fps file, we were able to identify and help debug one of our production instruments by observing the difference in size between the fps values from wide versus relational data extraction. This paper will discuss how the fps file was used to map with survey data from relational tables and how we were able to compare the fps file using both methods in order to correct the production instrument early in the production cycle.

3. Background

The PSID is a long running longitudinal study that began in 1968. From 1968-1992 the survey was collected using a paper and pencil instrument. Between 1992 and 1993, the survey transitioned to CAI and was originally programmed in SurveyCraft. The most recent major transition for the survey was when the survey was re-programmed in Blaise for the 2003 wave. For all versions of the survey from 1968-2019 the questionnaire has been interviewer administered. Once converted to CAI, the vast majority of the PSID interviews have been collected via decentralized CATI. Our next transition using Blaise 5 is in many ways, even more significant than the transition from paper to CAI in that our goal is to have a self-administered instrument. The move to a self-administered instrument requires not only significant changes to questionnaire wording and screen design but also to how we record missing data values.

In the PSID we collect information about all family members however, we collect this information from only one respondent. Our household roster has up to 24 loops. Although allowing for 24 individuals may appear excessive on the surface, due to the generational nature of the PSID, this number of roster rows is necessary. The PSID is a survey of related families, once a branch of the family breaks off and moves away, we begin interviewing independent families on their own survey line. However, the fluidity of families are complex and over time some generations of families may drift back together and share the same housing unit again. In these situations we continue to interview these distinct families as separate units; a concept we have termed "Two FU's in a HU" (Two Family Units in one Housing Unit). In order to capture situations where we have multiple families sharing the same Housing Unit, we have the interviewer list the other family members in related families and continuously remind the respondent throughout the interview to exclude these other family members from their responses to avoid double counting. In some waves of the PSID, we have two, and in rare cases three or more families sharing the same physical dwelling. Due to the nature of the PSID and the relatedness of our families and how they may be grouped, we require the 24 loops to list all of these individuals in the main roster.

The roster itself is then the anchor table that ties individuals to all other sections. We ask about jobs, marriages, children and so forth for each of our family members and each person may have several iterations of each. The design of the instrument requires a series of carefully crafted nested loops. This design feature of the PSID requires us to extract data in a relational format. In a wide extraction, we end up with numerous blank variable positions and an unwieldly amount of columns to handle. Understanding that our sample contains a fluctuating number of family members, and in turn a fluctuating number of their jobs, their children, and their marriages etc. requires that we process relational tables in these content domains.

In previous versions of Blaise 4 and earlier we extracted data in relational blocks using an extraction tool built by University of Michigan programmers based on Manipula. Block extraction has allowed us to handle the data editing, coding and release much more effectively and efficiently. For the PSID to continue its processing methods, relational extraction was not optional but rather, a necessity. Therefore, even though a relational extraction tool wasn't readily available at the early development stage of Blaise 5, the PSID programming staff was able to output data via the wide-extraction method and use information from that process to reverse engineer the wide table format into a series of smaller relational tables at the block level.

As an added complexity, in the web self-interviewing mode we also need to capture precisely the values of missing data. In the self-administered instrument, almost all fields are optional, meaning that they do not require a response from the respondent. Unlike CATI instruments where special answers like "Don't Know" or "Refused" can be invoked on almost any question, the web versions do not have special answers available to the respondent. In CATI interviewer mode each question on route requires a

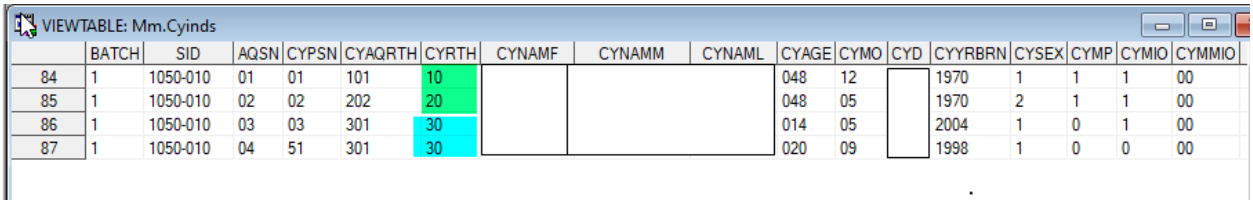
response including special answers as necessary. In contrast, the web self-administered mode allows an empty response on almost all screens and special answer options are not provided.

The prevailing reasoning given for the avoidance special answers for the respondent in a self-administration mode, is the belief that if they appear onscreen, the respondent may be more tempted to use them more readily in lieu of providing a valid response option. Given this significant difference between modes, the fps file becomes even more critical in determining the precise value of the missing information for each variable. For documentation purposes, we need to know if this question was on-route and simply not answered or whether it was never on-route. Knowing the difference between these two types of missing values is imperative for documentation and release.

4. Need for Relational Extraction for Coding/Processing and Release

The processing of the PSID takes many months using a team of eight experienced editors. Families are organized on the roster in order to prioritize the focal persons in the survey – they are the Reference Person and their Spouse/Partner. There is a set of rules for determining those two prominent people *(Figure A: CYRTH=10 and 20)* and many more questions are asked of these two individuals than OFUMs (Other Family Unit Members*) (Figure A: CYRTH = 30).* As editors comb through all the interviews in a related clan, they must determine the correct configuration of each family. During field work some individuals are listed more than once in different interviews and editors must decide where each person is located in each family. In order to maintain data integrity, a person may only appear once (in one single family) in any given wave. Due to the nature of this editing task, the PSID staff must manage many edits to the roster- moving people to different positions, adding or deleting people and so forth. In a long format, these types of transformations are straightforward and easy where rows are inserted, deleted or key values for individuals are updated as necessary. In a wide format this editing task would be extremely challenging and cumbersome as columns would need inserting, deletion and as individuals are shifted around positions, other individuals would need their information re-packed into existing columns.

Figure A: Example Roster PSID-Long Format


VIEWTABLE: Mm.Cyinds

	BATCH	SID	AQSN	CYPSN	CYAQRTH	CYRTH	CYNAMF	CYNAMM	CYNAML	CYAGE	CYMO	CYD	CYYRBRN	CYSEX	CYMP	CYMIO	CYMMIO
84	1	1050-010	01	01	101	10				048	12		1970	1	1	1	00
85	1	1050-010	02	02	202	20				048	05		1970	2	1	1	00
86	1	1050-010	03	03	301	30				014	05		2004	1	0	1	00
87	1	1050-010	04	51	301	30				020	09		1998	1	0	0	00

In order to easily manipulate individuals, their jobs, marriages, children and other domains, our editing system consists of a series of related tables with a set of primary keys. Our editors are able to easily insert, delete or reconfigure keys for tables organized in content areas.

5. The Filed Property Values (FPS) Files

The PSID Mixed Mode Pilot (MMP) data collection ended by the end of 2019, and we are using the data files from this survey to compare FPS files with survey data values.

The FPS file is cell level indicator whether a survey item (field question) is presented (on-route) to the respondents for the entire sample and questions in the survey. The file also includes remarks (F2 notes) data, therefore, it's important to determine whether the file is complete regardless of the method of data

extraction. To our surprise, the FPS file from "wide" extraction has 55,638 rows and the "relational" extraction has 50,993 rows. There were 179 "remarks" from "wide" and 174 from "relational" data extraction methods. Further checking of remarks data showed that one sampleid which was in "wide" file, it was missing from the "relational" file – something lost in transition. Since FPS file from "wide" data extraction has relatively more information on visited fields, therefore, the "wide" file was used to identify: a) any loss of data from "relational" data extraction, and b) to add or update data values of relational tables.

6. The Mapping of Survey Data and FPS Data Files

In order to compare FPS file with survey data, the first step is transform both datasets in way that they are linkable to each other and then compare "IsVisited" indictor with survey responses -- at cell level.

a) The Survey Data: From MMP survey, there are 139 tables and 6,279 variables from relational data extraction. Not all tables have data and keys variables are excluded from cell level transformation. So, there are 106 tables and 5,176 variables are used to create a "long" file. The cell level "long" data file has 1,187,455 rows where each data value is uniquely identified by set of four variables from the entire survey data collection. Below is the structure of "long" data file:

#	Variable	Type	Length	
1	PrimaryKey	Char	**7**	--+
2	Tablename	Char	**32**	| Key
3	TableInstance	Char	**75**	| Variables
4	Variablename	Char	**32**	--+
5	DataValue	Char	**40**	

During long transformation, all data values have been converted to character data type. Open text values were truncated to 40 characters. The value is used as flag to measure survey response for an item presented to the respondent.

b) The FPS Data: As explained before, we are going to use FPS from "wide" extraction in this linking. Below is the layout of raw FPS file:

#	Variable	Type	Length
1	PrimaryKey	Char	**7**
2	Path	Char	**100**
3	Property	Char	**10**
4	Value	Char	**200**

First, the data needs a lot of programming to create variable and instance level information from "path" column. The data in this field is delimited by a dot (".") and the last value is variable name. The FPS file is at .bmix level, therefore, multi mention (set of) variables need to be converted to match with survey data variables.

Second, create table instance (proxy for variable instance) from path column after removing the variable name from the text value. This is done by look at the "table instance" values of long file from Survey Data file created in step a) above.

Third, adjust array variables for an array range suffix and "set of" variables to match with variable name in the "long" data file. Then merge table and variable id from metadata created from .blax files and make

other adjustments manually as needed. The variable instance found in FPS file only may have missing values of key variables and missing values must be assigned valid values before merging FPS and survey data files. In the final merge, keep all rows from FPS file only, i.e., a left join on FPS file.

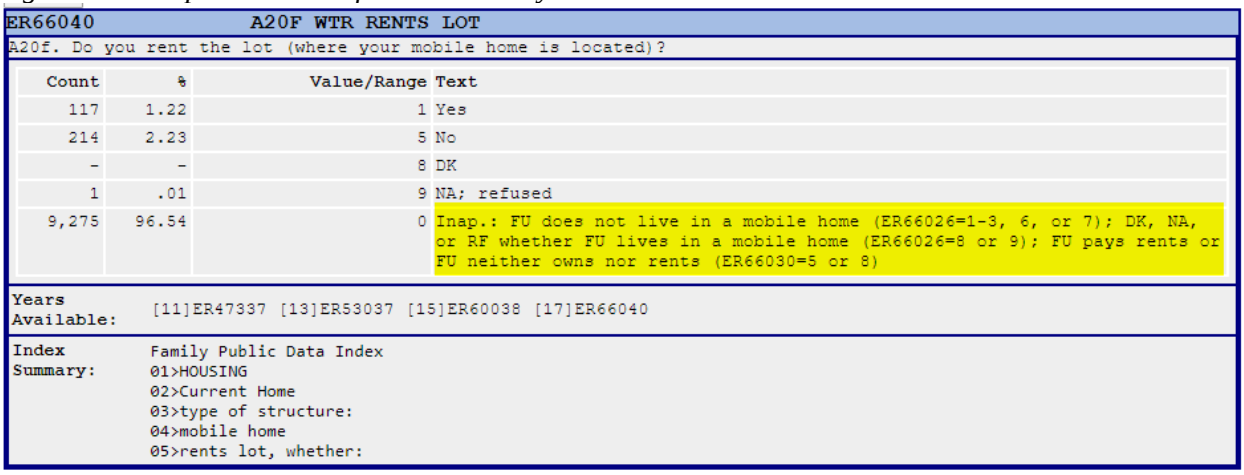
c) The Evidence: A total of 74,536 data out fields are flagged as visited by the FPS file, which is about 6.28% of all data cells in the survey data. Out of which: i) 55,458 have values in the data files, ii) 18,887 cells have indication of on-route from FPS file but no data in the data files, and iii) 191 new cells be added to the survey data files.

d) Alternative evidence: The FPS file is based on .bmix where "set of" variables have one row per sampleid-variable instance. A better approach is to rollup the "long" data file to match with variables from FPS file such that "set of" variables are counted once. The count of on-route/missing values is 4,481 compared to 18,887 reported in c-ii) above.

7. Use of FPS in missing data values: 'on-route' vs 'off-route'

As with all survey data, the ultimate goal of the study is clean, coherent data with fully documented variables. The PSID study has a long standing tradition of providing complete documentation and a description of the valid, missing and INAP (Inappropriate code) values. INAP values are assigned to items which were skipped or 'off route' for this particular person or family. The text description of the INAP values is the inverse of the universe *(Figure B).*

Figure B: Example Codebook for PSID Family Level Variable in CATI



ER66040 A20F WTR RENTS LOT

A20f. Do you rent the lot (where your mobile home is located)?

Count	%	Value/Range	Text
117	1.22	1	Yes
214	2.23	5	No
-	-	8	DK
1	.01	9	NA; refused
9,275	96.54	0	Inap.: FU does not live in a mobile home (ER66026=1-3, 6, or 7); DK, NA, or RF whether FU lives in a mobile home (ER66026=8 or 9); FU pays rents or FU neither owns nor rents (ER66030=5 or 8)

Years Available: [11]ER47337 [13]ER53037 [15]ER60038 [17]ER66040

Index Summary:
Family Public Data Index
01>HOUSING
02>Current Home
03>type of structure:
04>mobile home
05>rents lot, whether:

In CATI we capture the difference between missing (Don't Know or NA; Refused special answers) and INAP (system missing) because the variable is 'off-route'. Since all variables in the interviewer administered mode require a response, it is always clear which variables are on or off route. For self-administration mode where empty is allowed, this becomes increasingly difficult to determine the system missing values that are on-route or off route. In order to determine this critical difference, we turn to the fps file to help us make those assignments.

A three step approach is used to keep distinction between the data values which are on-route and missing vs. standard missing data.

1. In step one, data from Blaise are extracted with missing data codes (.D=Don't Know, .R=Refused, .A=Special Answer (997, 9997,…), .B=Special Answer (996, 9996, …), … more special answer codes. This step is labeled as "Blaise Data As Is".

2. In step two, another missing data code .V=Visited is used to indicate on-route and missing. If FPS to Survey Data mapping shows that a filed is visited but the data cell is empty then the cell will be updated to .V as special missing data code. If an entire is missing then a new row will be created with .V values appropriately. In this step all missing data values are represented by .D, R, .V, and all Special Answers (.A, .B, .C, .F, .G, .H, etc.).

3. In third and final step, the Don't Know and Refused missing data values are flipped to ISR standard DK/RF numeric values. The on-route missing (.V) are also converted to RF equivalent numeric values. The Special Answers missing data values are converted to their respective numeric equivalent. These files are saved in "Data Out" folder as final set of data files for further use of data with other data processing systems. The data frequencies are also calculated and uploaded to Oracle database for the use of in house applications.

With a three step approach, the process is able to keep backward linkage with Blaise data extracted in the first place (aka Blaise as is) and can used to write INAP data value explanation for the public release data file.

8. Conclusion and Summary

With Blaise 5 and mixed mode survey instrument, an identification on-route/missing data values is important for data processing and release of the PSID data. Regardless of the two approaches to identify on-route/missing data values in section 5 c) and d) above, the FPS file from "wide" and survey data "relational" extraction should be used to account for data leakage.

As described in section 5, FPS file from "relational" extraction has less data than the FPS file from "wide" extraction. The issue should be further investigated in collaboration with the PSID staff and the Blaise development team at the Statistics Netherlands.