Data Cleaning Primer with Worked Examples

Prepared for Blue Cross Blue Shield Minnesota
Center for Prevention
By

Annette Kavanaugh, M.S.
Research Analyst
Professional Data Analysts, Inc.

With contributions by

Sharrilyn Evered, PhD
Senior Research Consultant
Blue Cross Blue Shield Minnesota
Center for Prevention

Mike Krizan
Senior Research Analyst
Blue Cross Blue Shield Minnesota
Center for Prevention

Section 1.01 Overview

This document details a process for cleaning survey data when the data involve skip conditions. The process includes two steps: 1) verifying that the skip conditions were observed correctly, and 2) transforming system missing (blank) into numerical codes that are labeled to indicate the reason for the skip. Step 1 assures the analyst that the data are clean. Step 2 assures the *client* that the data are clean by accounting for each member of the respondent set and making clear which respondents are in the analysis (i.e., in the denominator). Step 1 should be used with any data set that contains skips. Step 2 is somewhat labor intensive and may be necessary only for surveys with complex skip patterns (i.e., dependent items are skipped based on the responses to two or more survey items).

Section 1.02 Genesis of this Document

This data cleaning primer was proposed by Sharrilyn Evered, Ph.D., to enhance the data cleaning and presentation process used by the Center for Prevention Measurement Team. Professional Data Analysts, Inc., was asked to prepare this document as a guide to a data labeling process developed and used by PDA over the past ten years,[1] referred to as the "Missing-Codes" approach. This labeling technique uses features specific to the SPSS statistical software (user-missing values and value labels) to replace "true missing" responses with "reason for missing" values. This allows analysts and report writers to see the full picture of the respondent set. It makes clear who *is* answering an item and who *is not* answering it and why. Analysts and report writers at PDA have found that labeling missing data is helpful in auditing data, in checking relationships between items, and in answering the number one question asked by report readers internal and external: "What is the base for this percentage?"

The data cleaning techniques currently in use by the Center for Prevention Measurement Team are comparable in rigor to those used in research settings such as the Census Bureau, the Centers for Disease Control and Prevention. The data are carefully reviewed for consistency, observance of skips, etc., and consistency errors, and corrections are captured in new "check variables." Original data are left intact, and data integrity is a prime concern. The data sets that reach the users have been vetted for cleanliness, and by design no trace of the cleaning process is left in the data. This approach is referred to as the "Check-Variables" approach.

This manual explains both approaches and indicates their advantages, disadvantages, and scope of applicability.

[1]Contributions to this method were made by the present author, Matthew Christenson and Chow-Hong Lin.

Survey Data Cleaning

Section 2.01 Step 1 of Data Cleaning

This most basic step of data cleaning involves examination of outliers, inconsistencies, and exaggerated patterns of response. At this stage, analysts assess whether or not to include given responses or patterns of response. One or more variables can be involved. For maximal purity, the analyst should do all consistency fixes on auxiliary variables, not on the original data.

Examples of problematic responses or patterns of response:

- Outlier / extreme response: A tobacco use survey asks "How many cigarettes per day did you smoke in the last 30 days?" An individual reports 300 cigarettes.
- Logically inconsistent response: Respondents report participating in an activity within the past 30 days, but not in the past 7 days.
- Suspect pattern of responses (suggest deceptive or careless responses): On a youth health survey, a handful of respondents endorse the maximum use category for several illegal drugs.

In many cases, Step 1 occurs at the beginning of analysis. In some cases, problems may appear when the analysis is already underway. This first step of data cleaning is not addressed in this manual.

Section 2.02 Step 2 of Data Cleaning

Here we make sure that skips are observed correctly. Usually, items that are skipped are those that are not applicable for certain respondents.

Example 1:

If a respondent indicates on item 6 that she doesn't own a car, she should definitely skip item 7, "What color is your car?" Incorrect observation of the skip could be of two kinds: (1) someone who *does* own a car doesn't get asked what color it is. In SPSS, this shows up as system-missing (the person didn't get asked the question, so there is no response at all), or (2) someone who *doesn't* own a car gets asked the color of the (nonexistent) car. In this case, there are data for this person in both variables, but the data are logically inconsistent.

If the survey were administered using a properly programmed CATI system, or on paper using well-trained interviewers, the data should be very clean at this level. Every question that should have been skipped would be missing, and all questions that should have been asked will contain valid responses (including "don't know" and "refused"). Programming error and/or interviewer error can produce violations of skip conditions. In either case, data must be reviewed for correct observation of skips.

The following sections provide an overview of the Check-Variables approach and instructions for verifying skip conditions (step 2 of data cleaning).

The Check-Variables Approach: Using Check Variables to Verify Skip Patterns

In this primer, we outline two complementary approaches to cleaning survey data that involve skips. We focus particular attention on the issue of verifying skip patterns. In discussing these approaches, the following definitions will be useful:

Switch item: Survey item whose values determine whether subsequent item(s) will be encountered by the respondent.

Dependent item: A survey item which may or may not be encountered, depending upon the value of a preceding switch item.

Skip condition: the value or combination of values of the switch item for which the dependent item(s) should be skipped.

Continuation condition: the value or combination of values of the switch item for which the dependent item(s) should be encountered by the respondent. The continuation condition is the negation of the skip condition. Most of the CATI surveys done by the Center for Prevention have been structured in terms of continuation conditions rather than skip conditions.

The table below shows the four possible interactions of a switch item with its dependent item(s). Error conditions are of two kinds: (1) the skip condition is satisfied but the dependent item is filled in, and (2) the continuation condition is satisfied but the dependent item is missing. Cells are numbered 1 through 4 for reference.

The Check-Variables approach detects both types of error conditions and valid responses. If a survey is administered using a CATI system, programming error can be detected by the presence of either of the error conditions.

Table 1. Relationship Between Skip Condition and Dependent Item

		Switch item: skip condition satisfied	
		Yes	No (continuation condition satisfied)
Dependent item: Filled in	Yes	**Cell 1: Error (invalid response)**	**Cell 2: OK (valid response)**
	No	**Cell 4: OK (valid nonresponse)**[2]	**Cell 3: Error (invalid nonresponse)**

The Check-Variables approach captures both error conditions (cells 1 and 3) and validates correctly non-missing values (cell 2). This provides an "automated" check for CATI programming problems. The first check variable (check variable 1) sorts all non-missing responses for the dependent item into the categories "valid" or "not valid," according to the indicated skip condition. The second check variable (check variable 2) provides a "corrected" version of the dependent item, which additionally detects invalid non-response. Frequencies on these two variables provide a view of the correctness of the CATI programming and the valid responses captured in the dependent item (check variable 2).

The Missing-Codes approach can be used to provide specific "reason for skip" codes for the cases included in cell 4. This approach is particularly helpful where a dependent item can be skipped based on skip conditions on multiple previous items, a pattern often encountered in surveys that follow a respondent through a process with multiple branch points.

[2] With the Check-Variables approach as outline here, we can consider cell 4 a valid nonresponse only in cases where a dependent item is subject to *only one* switch item. A dependent item may be subject to more than one switch items. Since the Check-Variables approach examines one switch item / dependent item pair at a time, we cannot make any global conclusion about validity of missing from a given switch item – dependent item pair.

Procedure for Creation of Check Variables

(current practice of Center for Prevention Measurement Team).

The Check-Variables technique discussed in this section allows the analyst to detect both error conditions in Table 1 and validate correctly non-missing values of the original dependent item. It does not document "correctly missing" values. PDA's Missing-Codes technique (described in later sections) addresses this issue.

With this technique, the analyst checks whether skip conditions have been followed correctly by creating "check" variables that take into account the information in both the switch and dependent items and return a value if the data are consistent with a correctly observed skip and another value if the data are not consistent with an appropriate skip. The following lists step-by-step instructions for this process.

Section 4.01 Procedure

Step 1. Produce frequencies on the switch item and the items to be skipped, in order to observe patterns of missing. At this point, the analyst is looking for possible inconsistencies between the count of cases where the skip condition is satisfied and the count of cases where dependent item(s) are missing. In Example 1, the analyst would look at how many respondents said they owned a car in relation to how many answered the question about the color of the car.

Step 2. For each switch-item/dependent-item pair, create Check Variable 1. Check Variable 1 deals only with cases that have data filled in for the dependent variable. Check Variable 1 distinguishes between cases that are "correctly filled in" (ok, cell 2) and cases that have data when they should not (error, cell 1). In other words, Check Variable 1 identifies any cases that should be set to missing for correct analysis.

a) Check Variable 1 is set to a "yes" value (typically 1) if the dependent item was answered and *should* have been answered given the value of the switch item(s). It is set to a "no" value (typically 2) if the dependent item was answered and should *not* have been answered. This captures "correctly permitted to answer the question" versus "incorrectly permitted to answer the question."
b) Check Variable 1 is empty (missing) for all cases where the dependent item is not answered.

NOTE: In creating Check Variable 1, the analyst should be careful account for all possible non-missing values for the skip item(s) so that no skip condition or continuation condition scenarios are omitted.

In Example 1, a person who said she owned a car and in the next item specified the color of the car would receive a value of 1 for Check Variable 1, which is equal to "yes" the skip was observed correctly. Check Variable 1 equal to 2 or "no" the skip was *not* observed correctly corresponds to respondents who do not own a car (or don't know or refused to say) who nonetheless answered the question about the color of the car.

Step 3. Crosstabulate Check Variable 1 by the switch variable (be sure to include missing) as a check on Check Variable 1's programming. In Example 1, Check Variable 1 would be "yes" for all who said yes, they owned a car; Check Variable 1 would be "no" for all who said they did not own a car, didn't know if they owned one, or refused to say but answered the dependent item anyway. Check variable 1 would be missing for all those who said they did not own a car (or dk or refused) and did not respond to the question about car color.

Step 4. For each switch-item/dependent-item pair, create Check Variable 2. Check Variable 2 is a cleaned version of the original dependent item. When it has been filled in as described below, Check Variable 2 is the variable that analysts would use for analysis and reporting (i.e., the working variable).

a) Check Variable 2 is initially a copy of the original dependent item. (Check Variable 2 = the Car Color variable without any recoding)

b) If Check Variable 1 is missing, assume that the dependent item is missing due to a correctly observed skip condition and assign Check Variable 2 the code -9 "Missing OK, skip observed." (This code can be replaced later with missing codes corresponding to the reason for missing using PDA's Missing-Codes approach). Note that this will never overwrite non-missing values of the dependent item, because Check Variable 1 will be non-missing and fail to trigger the overwrite if the dependent item is non-missing.

c) In cases where Check Variable 1 = "no" which indicates that a dependent item was incorrectly filled in (error, cell 1), Check Variable 2 is assigned an error code (-8 in current practice). This step flags and eliminates invalid answers. Presence of the value -8 may indicate CATI programming error.

d) If the Check Variable 2 still contains missing cases after steps a-c above have been completed, assign Check Variable 2 another error code to indicate unexplained missing (-7 in current practice by Center for Prevention Measurement Team). Presence of the code -7 may indicate CATI programming error (Cell 3 or an error in skips programming from earlier in the survey)

e) To summarize: Check Variable 2 is a "corrected" version of the dependent item, with codes assigned to account for missing and invalid data. Every case in Check Variable 2 will have a value; there should not be any system missing (i.e., blank or empty cases).

Step 5. Produce frequencies on the switch item, dependent item and the associated check variable 2; Crosstabulate Check Variable 2 by the switch variable.[3] Here, the analyst is verifying that Check Variable 2 is filled in with a valid answer only where the continuation condition is satisifed.

Step 6. If each Check Variable 2 appears consistent with its related switch item, set -8, -9 on each Check Variable 2 to missing. The data are now ready for analysis.

[3] If the skip condition is based on multiple (more than two) switch variables, the crosstabulation may not suffice as a check. In this case, sort the data in order by the switch variables and Check Variable 1, then list values.

Table 2. Values of Check Variable 1 and Check Variable 2 corresponding to each switch item / dependent item condition

		Switch item: skip condition satisfied	
		Yes	No
Dependent item: Filled in	Yes	**Cell 1: error** Check Variable 1 = no Check Variable 2 = -8 (error code for incorrectly filled in dependent item; 'overwrites' original value of dependent item)	**Cell 2: OK** Check Variable 1 = yes Check Variable 2 = original value of dependent item
	No	**Cell 4: No error detected** Check variable 1 is missing Check Variable 2 assigned code -9 (missing OK due to skip)	**Cell 3: error** Check Variable 1 missing Check Variable 2 = -9 (error code for incorrectly missing)

To summarize, for each skip, two check variables are created:

1. **Check variable 1 (binary skip check)** is a binary variable which checks whether the given dependent item is *correctly* non-missing, given that the switch item is non-missing. This variable identifies non-missing cases for the dependent item that should be excluded from analysis because the item was answered in error.

2. **Check variable 2 (reporting variable)** is assigned the value of the corresponding dependent item, but these values are overwritten by an error condition code (-8) if the item was *incorrectly* non-missing (i.e. the question was asked when it should have been skipped). If the dependent item was *incorrectly* missing, another error condition code (-7) is assigned. Correctly missing data remains system missing. Check Variable 2's may be used in analyses where it is desired to restrict attention to the cases in which skip conditions were correctly observed.

Original survey variables are never touched. Any resolution of inconsistent response patterns is done in auxiliary variables (e.g. Check Variable 2 or additional variables derived from it). This approach guarantees that original variables capture original data and that any additions, annotations, or corrections by the analyst are stored separately.

Advantages of Check-Variables Approach:

- In all versions of the data sets, the variables with the original raw data can always be compared against the analyst's work.
- The integrity of the original variables is guaranteed by the process itself. This can be crucial with certain types of data, e.g. billing records and surveys where the analyst's work and the original responses must be kept strictly distinguishable.

Disadvantages of Check-Variables Approach

- Proliferation of check variables, processing variables, and auxiliary variables. In general, each switch-item/dependent-item pair produces two check variables.
- In examining original or Check Variables, there is no direct documentation of the valid reason for missing. The base for analysis (the set of people who answered a question) has been checked for correctness, but there is no documentation of *why* those missing are missing. Definition of the base for analysis must be captured in documentation.

Section 4.02 Scope of the Check Variables method

This method was originally developed for use with CATI surveys with short sequences of skipped items. It may require revision for any of the following cases:

- Surveys administered on paper. Here, the interpretation of 'unexplained missing' data (cell 3 in Table 1) or 'incorrectly filled-in items' (cell 1) will be somewhat more difficult.
- Surveys with long or complex sequences of skipped items. Creation of Check Variable 1 for these items will be somewhat more complex, and creation of Check Variable 2 may need to be revised.

As presented, this method is by no means automatic and requires careful attention to programming of key variables, especially Check Variable 1. Needless to say, absence of error-condition codes does not guarantee error-free CATI programming, but the larger and more diverse the data set, the more likely that this is the case.

PDA's Missing-Codes Approach: Conveying the Valid Reason For Missing

In discussing this approach, the following definition will be useful:

Skips code or **missing code**: user-missing value assigned to a dependent item in place of the system-missing value to indicate *why* a response is missing.

Section 5.01 Scope of the Missing-Codes approach

The Missing-Codes approach provides full documentation for the cases in cell 4 of Table 1, where a dependent item is assumed (correctly) missing because one or more skip conditions have been observed. It also detects the error condition in cell 3 (incorrectly missing, skip condition not observed). It does not address the cases in cells 1 or 2. Therefore, this labeling technique when used in conjunction with the Check Variables approach fully accounts for all possible outcomes (shown in Table 1) involving skip conditions.

1) Focus is on missing data resulting from *correctly observed skips* (cases in cell 4 of table 1).
2) A 'reason for missing' code is assigned wherever CheckVariable 2 is assigned the value -9, where a and a skip condition has been properly observed.
3) This approach can be applied to Check Variable 2 or to a variable based upon it.

Section 5.02 Guidelines for Missing-Codes approach

1) In original practice at PDA, this approach is often applied to survey variables directly rather than to reporting variables. In this case, never assign a valid response (i.e. one from the survey itself) if the respondent did not endorse one, even if it can be logically inferred. If a valid response is entered in violation of a skip condition (i.e. if somebody got asked the question but shouldn't have) leave it alone and resolve the base issue using auxiliary variables (see item 3 below for detail). The Check-Variables approach (using the binary check variable 1) is useful in detecting such cases.
2) When implementing this Missing-Codes technique in SPSS, skip verification and missing-classification values are always negative integers. The entire range of values LO THRU -1 can then efficiently be set to user-missing. Information about the switch item and its value (i.e., the reason for the switch, such as an answer of "1" on Q10) can be embedded in the code by convention, though this is not always easy to do with precision. (See the Coding Convention section for details).
3) If applying the Missing-Codes approach directly to survey variables, consistency issues are ALWAYS resolved using auxiliary variables. For example, on a smoking cessation survey, we might ask respondents if they have had any tobacco past 7 days, and then ask if they've had any tobacco past 30 days. We generally use these questions to create auxiliary variables

'quit past 7 days' and 'quit past 30 days'. If 'quit past 30 days' is yes, then we force 'quit past 7 days' to be yes. This does not affect the original survey items. (This parallels the use of check variable 2 to eliminate responses filled in where the dependent item should have been skipped).

4) Skips codes are assigned in the order in which skip conditions are encountered in the survey. The effect of this is to "embed the flowchart" of the survey in the data. This is explained in more detail in part 6.

Advantages:

- If the Missing-Codes approach is applied to the original dependent items, there are no separate skip-checking variables; the number of variables on the set is kept under control while capturing skip-checking results.
- Only system-missing values are overwritten.
- The Missing-Codes approach can be applied on its own or to Check Variable 2 as an enhancement of the Check-Variables approach. The example in this manual takes the latter approach.

Disadvantages:

- It is vital to check carefully all cross-tabulations of switch item and skipped items.
- Because the cross-tabulations must be comprehensive, it is necessary to recode system missing with a non-missing value, conventionally a large negative number.
- Skips are checked in the order the switch items occur in the survey. The skips checking is not strictly independent under this method, and checking of intermediate cross-tabs can be confusing.
- All assignments of 'reason for missing' codes must be conditional on *the item being missing* AND *the skip condition being satisfied.*

Section 5.03 Relationship between the Check-Variables and Missing-Codes Approaches

When the Missing-Codes approach is used alone. The dependent item with missing codes added captures the same information as Check Variable 2, except it does not indicate or remove any responses that should have been missing.

Where Check Variable 1 indicates that a skip was not observed correctly, an enhanced reporting variable can be created from the dependent variable with missing codes by overwriting with an error code any responses that should have been missing due to skip. This accounts for all missing *and* retroactively enforces all skips.

Combining the two methods. First, apply the Check Variables approach. This handles cells 1, 2, and 3 in Table 3 below. Then create missing codes wherever Check Variable 2 is assumed correctly missing (cell 4). This provides full documentation in the last remaining case, cell 4 in Table 3, by giving all reasons that the dependent item is missing when the skip condition is observed.

Table 3. Relationship between skip condition and dependent item as captured by Check Variable 1 and Check Variable 2 (Check Variable 2 is filled in with missing codes)

		Switch item: skip condition satisfied	
		Yes	No
Dependent item: Filled in	Yes	**Cell 1: error** Check Variable 1 = no Check Variable 2 = -8 (error code for incorrectly filled in dependent item; 'overwrites' original value of dependent item)	**Cell 2: OK** Check Variable 1 = yes Check Variable 2 = original value of dependent item
	No	**Cell 4: OK** Check Variable 1 missing Check Variable 2 assigned additional missing codes to explain what skip condition was observed ("why missing")	**Cell 3: error** Check Variable 1 missing Check Variable 2 = -7 (error code for incorrectly missing in check values approach)

Section 5.04 Summary of hybrid Check-Variables / Missing-Codes approach (see Table 3):

1) Check Variable 1 classifies the non-missing cases for the dependent item as correctly filled in (cell 1 in Table 3) or incorrectly filled in (cell 2 in Table 3).

2) Check Variable 2 is a cleaned version of the original dependent item, with invalid responses overwritten and missing data identified as either correctly or unexplained missing.

3) The Missing-Codes approach can be used to fully document the cases in Cell 4 by assigning codes to Check Variable 2 that document why the original dependent item is not filled in.

Section 5.05 Data set version control: general principles

Original data sets should never be overwritten by subsequent versions. This rule should be followed without exception regardless of the data cleaning approach taken. For internal

purposes, well-written SPSS syntax may suffice for documentation. Such syntax would include, at a minimum:

- GET and SAVE statements for all files accessed and created by the syntax (to capture the 'inputs' and 'outputs' of data cleaning and modification syntax);
- Comments to indicate the purpose, authorship and version of the syntax.
- Names of additional supporting documentation (in syntax comments), as needed. For complex surveys, additional documentation may include a skips table and/or survey flowchart.

New versions of data may have modified names, or the same name and different locations. If *anything* is changed, save as a different set (with a different name and/or a different location). This ensures that all processing applied to the data can be both traced and (in case of error) reversed and corrected. In particular, *no data set is ever irrecoverably overwritten.*

- For example, Mike Krizan (Center for Prevention Measurement Team) reports that he keeps the original survey data in a folder titled "original data" and the working file (with the check variables added in) in a folder titled "working data." This approach works well where custody of the data remains fixed.
- Practice at Professional Data Analysts has been to require changes in file name for each version. Raw data from a survey vendor would retain the original name assigned by the vendor, with revised names assigned to each successive revision of the set. Data set names indicate content and revision number. For example: successive six-month follow-up survey data sets might be named "mo6-1.sav", "mo6-2.sav", etc. Explanatory suffixes can be added as necessary (e.g. "mo6-2auxAK", to indicate that it is version 2 with auxiliary variables added by analyst AK). This convention reflects historical practice at PDA, where data cleaning and/or analysis tasks may be split up between multiple individuals. However, proliferation of (typically large) files is a disadvantage to this approach.

Section 5.06 Data cleaning approaches and their impact on version control

The CheckVariables approach leaves no traces in the values of the original variables. If Check Variables are saved, this produces a new version of the data. However, it is a new version only in the sense that it includes additional variables. The content of the original survey variables is unchanged. For each dependent item, Check Variable 2 contains the cleaned data. Both the original dependent item and Check Variables 1 and 2 are retained, to enable checking of original data against check variables.

The Missing-Codes approach, as described above, *modifies* the original variables by replacing system-missing values with user-missing (with explanatory labels). Even if no additional variables are created, this is *de facto* a new version of the data and should be saved under a different name. Note that the variables can also be viewed as "new variables with the old names." The non-missing responses remain the same, and missing cases remain missing but new missing codes have been supplied to distinguish various reasons for missing.

NOTE: To avoid any modification of the data in the original variables, the Missing-Codes approach can be applied to Check Variable 2 rather than the original dependent item.

It cannot be overemphasized that correct use of the any data cleaning method approach is contingent upon good data set version control. In particular, the original data set should always remain intact.

Section 5.07 Missing-Codes Approach: Step-by-Step Instructions

NOTE: These instructions are written for the analyst who wishes to apply the Missing-Codes approach alone, independent of the Check-Variables approach. Tips for combining this approach with the Check-Variables approach are provided throughout.

Step 1. Recode system missing. A negative value is set aside as a temporary swap value for system-missing. This value is chosen so that it will precede (if large negative[4]) or follow (if small negative) any other missing value when frequencies are run on survey items, thus making it immediately clear where system-missing values have not yet been accounted for.

When applying the Missing-Codes approach to Check Variable 2, the value -9 stands in for system-missing, since it corresponds to values that are assumed to be missing because skip conditions are observed.

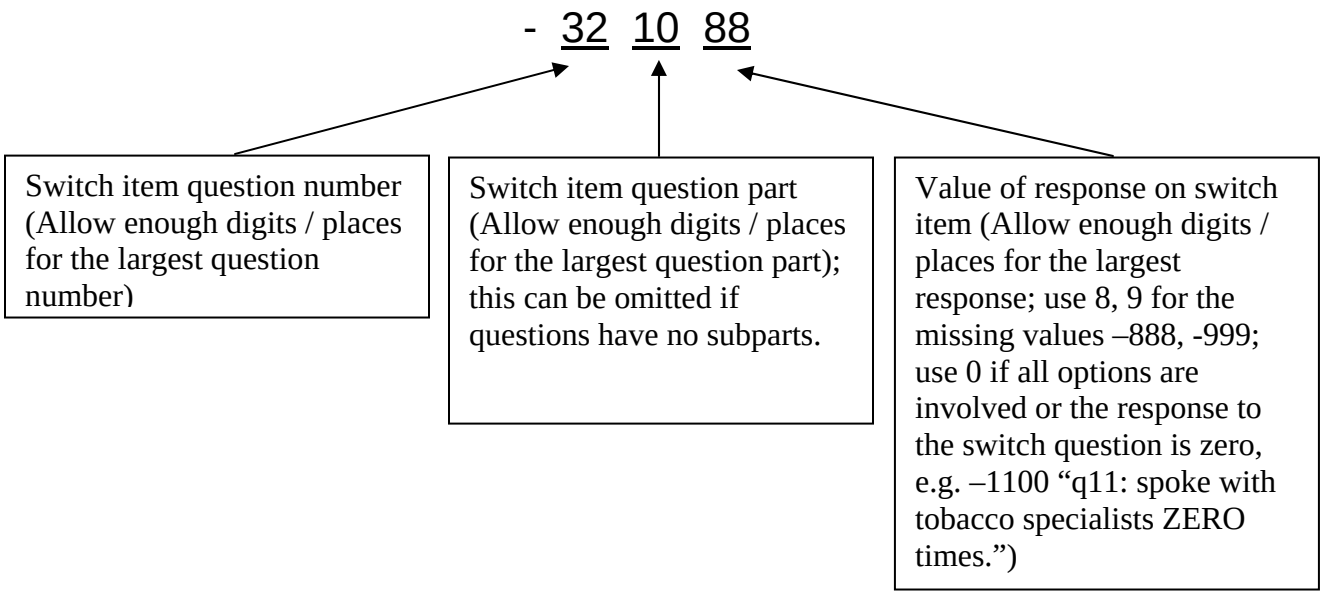
Step 2. Create skips codes. Skips codes are negative numbers. Ideally, they capture both the item number and the response code that explains why the dependent variable is not filled in. Negative numbers are preferred because, in general, nonnegative values (zero and positive numbers) are reserved for valid survey responses, making negative values easily distinguishable as a "reason for missing" code.[5] Also, using negative values for missing codes allows the analyst to be alert for "too small" results in means or medians for numerical (count) variables which will occur if the analyst fails to set the negative range to user-missing.

In the convention used at PDA, digits for missing codes are composed of three parts according to 1) the switch item question number (e.g., Q32), 2) the switch item question part, if applicable (e.g., part J), and 3) the value of the response (e.g., 88). These three values are combined in succession to form the missing code, so the analyst or client can infer the reason for the skip from the missing code itself (e.g., the missing code -321088 would indicate the response of 88 to part J of question 32 was the reason for the skip).

---

[4] For example, where skips codes do not exceed four digits, this value could be –88888 with label "Field not filled in"

[5] In an apparent exception, PDA conventionally codes "don't know" as -888 and "refused" as -999. Reasons are as follows: a) While these represent the respondent's actual answers they provide no information; b) in numerical (rather than categorical) items, e.g., age and other nonnegative counts, it is important that the codes for these "non-answers" be distinguished from any value in the range of possible answers.



Example A. Questions on the survey run from 1 up to 112; parts for a question can run A-Z (up to 26 parts), responses can have up to 2 places.

Set aside first/leftmost 3 digits for question number (1-112)
Set aside next 2 digits for question part (1-26)
Set aside last/rightmost two digits for response code

So if question 101, part B has response 77 "other" (and this is the value that triggers the skip), assign the missing code for this case as follows: -1010277 with a descriptive label, such as "Medications used: other."

Since the code captures the question number and response, the label does not need to include this information but can concentrate on describing the condition for the skip.

Example B. Questions on the survey run 1-45, parts for a question can run A-J (up to 10 parts), categorical responses are all one-digit.

So if question 6, part A has response 2 "no" and that triggers the skip, assign the missing code as follows: -06012

Example C (an unconditional skip). In some surveys, every respondent who is asked a particular item will then skip to some later part of the survey, regardless of the value of the response. It is most efficient in this case to assign a value of zero in the third part of the missing code.

For example, suppose that everyone who answers question 15 is skipped to question 17. Suppose that the survey is numbered 1-26, the questions have no subparts, and there are up to two digits for an answer code. If there is *any* answer to question 15, then missing values in

question 16 receives the code: -1500, with descriptive label “Unconditional skip: encountered Q15, skipped to Q17”.

Unconditional skips are often attached to questions that are to be answered by a selected subpopulation. For example, questions 15 and 17 may be answered by people who used medications for a given condition, with question 16 asking “Why didn’t you use medications?”. In this case, a better descriptive label would be -1500 “Unconditional skip: used medications”.

Example C (multi-select items). A code can be assigned to fill in a multi-select item where provision is made for the entire item being refused. This code (e.g., 9) should be assigned *following* skips code assignment for any skips that affect the multi-select item.

For example:
16. What kind of over-the-counter cold medications do you use?

__ refused entire question

A. pain relievers
B. antihistamines
C. decongestants
D. cough suppressants
E. combination medications
F. other

If an individual refuses to answer question 16 (the ‘refused’ option is chosen), then the following code is assigned to all the items q16A through q16E:

(Supposing two digits for question number, one digit for question part and one digit for response):
-1609 “Refused to specify kind of OTC cold meds”.

Another option would be to assign the “refused” code, -999, to each of the items in the multi-select question.

A similar approach can be applied for a global “don’t know” option.

To do full accounting for all missing, it may also be necessary to fill in a “not endorsed” option for multi-select items (e.g. where a respondent only endorsed choice #1 but did not endorse a response for choices #2 and higher). This should be done *following* the assignment of global “refused” or “don’t know” codes.

Some cautions.
1) Assigning negative values to dates is problematic in SPSS. In general, we do not assign skip codes to date variables.
2) This approach to missing codes assumes that all valid responses are coded as nonnegative numbers.

Detailed example of Check-Variables and Missing-Codes approaches

The combined Check Variables and Missing-Codes approach will be demonstrated using items 25-40 of the "***do*** Expansion" survey. The following steps are included:

1) Skips table: tabular flowchart listing the switch item, skip condition, items to be skipped if the skip condition is satisfied, and missing-codes to be assigned to the skipped items.
2) SPSS syntax for check-variables on the same range of "***do*** Expansion" survey items. (This is logically prior to the assignment of missing codes.)
3) SPSS syntax and output for the assignment and verification. Syntax is presented with annotation in comments or text. Output is presented in Appendix 1 (check variables) and Appendix 2 (missing codes).

Section 6.01 Detail of do *Expansion Survey items used in example*

"do Expansion" items used in the example

Q025 – ALL RESPONDENTS

Based on your understanding, what is the minimum number of days per week that an adult needs to do moderate physical activities, again, those activities that cause **some** increase in breathing or heart rate, in order to maintain good health?

_ ENTER DAYS PER WEEK (0-7)

77. DON'T KNOW / NOT SURE – **SKIP TO Q040**
99. REFUSED – **SKIP TO Q040**

Q030 – ONLY IF Q025 = 1-7

Finally, based on your understanding, on these days, what is the minimum number of minutes per day that an adult needs to do moderate physical activity in order to maintain good health?

1. One hour or more
2. 45 minutes
3. 30 minutes
4. 20 minutes
5. 15 minutes
6. OTHER TIME OVER 30 MINUTES **(SPECIFY)**
8. OTHER TIME UNDER 30 MINUTES **(SPECIFY)**

7. DON'T KNOW / NOT SURE
9. REFUSED

Q035 – ONLY IF Q030 = 1,2,3,6

Please tell me whether you think the following statement is true or false.

Being active for a total of <Q030> per day over several shorter periods, such as ten-minutes at a time, can be enough to maintain a person's health.

1. TRUE
2. FALSE

7. DON'T KNOW / NOT SURE
9. REFUSED

Q040 – ALL RESPONDENTS

The next questions ask you about the physical activities you do currently.

First, thinking about the vigorous physical activities you may do in a typical week outside of work, do you do vigorous activities for at least 10 minutes at a time, such as running, aerobics, heavy yard work, or anything else that causes large increases in breathing or heart rate?

1. YES
2. NO - **SKIP TO Q055**

7. DON'T KNOW / NOT SURE - **SKIP TO Q055**
8. NOT APPLICABLE **– SKIP TO Q055**
9. REFUSED - **SKIP TO Q055**

Section 6.02 Skips table

Skips coding syntax for a survey begins with the creation of a skips table. This is actually a flowchart in tabular form, with switch items (diamond boxes in flowchart) are listed in the order in which they are encountered in the survey. The skips table initially serves as specification for the syntax. In practice, it may contain errors and will have to be updated and corrected along with the syntax.

SWITCH ITEM	SKIP CONDITION	SKIP TO	ITEMS SKIPPED IF SKIP CONDITION IS SATISFIED	LABELS ON SKIPPED ITEMS
q025	q025 is 77 (DK) or 99 (refused)	Q040	q030 q035	-2577 "Min days per week: don't know" -2599 "Min days per week: refused"
q025	q025 outside the range (1 to 7) for days of the week	Q040	q030	-2500 "Min days per week: out of range [1,7]"
q030	Incorrect response: q030 eq 4,5,8 . Don't know: q030 eq 7 . Refused: q030 eq 9	Q040	q035	-3011 "Min exercise time per day for health: underestimated" -3007 "Min exercise time per day for health: don't know" -3009 "Min exercise time per day for health: refused"

Section 6.02 Syntax (Check-Variables)

This syntax is an excerpt from the data cleaning code for the "***do*** Expansion" survey written by Mike Krizan (Senior Research Analyst, Center for Prevention Measurement Team). Coding of error conditions was further refined by Sharrilyn Evered (Senior Research Consultant, Center for Prevention Measurement Team), to separate 'incorrectly filled in' from 'unexplained missing.' Explanatory text is added following each section.

Note: See Appendix 1 for output from this code.

Check Variable naming convention. Check Variables take their names from the dependent item. For example, for the Q025 (switch item) and Q030 (dependent item), check variable 1 would be called "q030_ck1" and check variable 2 would be called "q030_ck2."

Check Variable pairs

For each skip, two Check Variables are created:

1. **Check Variable 1 (binary skip check)** is a binary variable which checks whether the given dependent item is *correctly* non-missing, given that the switch item is non-missing.

2. **Check Variable 2 (reporting variable)** is assigned the value of the corresponding dependent item, but these values are overwritten by an error condition code (-8) if the item was *incorrectly* non-missing (i.e. the question was asked when it should have been skipped). If the dependent item is *incorrectly* missing, another error condition code (-7) is assigned. The code -9 is assigned if the original dependent item can be assumed missing because it was skipped. Check Variable 2 may be used in analyses where it is desired to restrict attention to the cases in which skip conditions were correctly observed.

In current practice at the Center for Prevention, these variables are captured in a separate "working" copy of the data. The number of Check Variables can become cumbersome if the survey contains a large number of dependent items. This can occur if there is either a large number of skips, or long sequences of dependent items associated with a given switch item.

The following pages contain actual syntax with annotations for coding using the Check-Variables approach for cleaning items Q025, Q030, and Q035.

```
** 1/30/2006 version (most recent).
** AK: add file get; SE: split out additional detail on missing codes.
GET
  FILE='K:\BCBS-MN\specialized training\merged_deID_PDA.sav'.


/*************************************/
/*         STEP 2                    */
/*      Code Qx_ck01                 */
/*   Create Skip Pattern Varis       */
/*************************************/

/************************************************/
/*  q030_ck1 -- KNOW MOD ACTIV        */
/*  Q030 – ONLY IF Q025 = 1-7         */

IF ( (MISSING (q030) NE 1)  & (q025 >= 1 & q025 <= 7) ) q030_ck1 = 1 .
EXECUTE .
IF ( (MISSING (q030) NE 1)  & (q025 < 1 | q025 > 7) ) q030_ck1 = 2 .
EXECUTE .

FORMATS
          q030_ck1 (F1.0).
VARIABLE LABELS
          q030_ck1  'KNOW MOD ACTIV -- q030<>Missing & (q025 >=1 & q025<=7)'.
VALUE LABELS q030_ck1
          1  'Yes-- Skip OK'
          2  'No- Not Valid Skip'.
EXECUTE .
```

Switch item: q025
Skip condition: doesn't know what moderate activity is (q025 is outside the range 1-7)
Dependent item: q030

Note that q030_ck1, the check variable 1 associated with the pair q025 and q030, is not defined where the dependent item q030 is missing. It only assesses whether q030 was *correctly filled in* (due to skip being correctly observed).

```
/************************************************/
/*  q035_ck1      KNOW T-F            */
/*  Q035 - ONLY IF Q030 = 1,2,3,6     */

IF ( (MISSING (q035) NE 1)  & (q030 = 1 | q030 = 2 | q030 = 3 | q030 = 6) ) q035_ck1 = 1 .
EXECUTE .
IF ( (MISSING (q035) NE 1)  & (q030 < 1 | q030 = 4 | q030 = 5 | q030 > 6) ) q035_ck1 = 2 .
EXECUTE .

FORMATS
          q035_ck1 (F1.0).
VARIABLE LABELS
          q035_ck1  'KNOW T-F -- q035_Chk1- q035<>Missing & (q030 =1,2,3, or 6)'.
```

```
VALUE LABELS q035_ck1
          1  'Yes-- Skip OK'
          2  'No- Not Valid Skip'.
EXECUTE .
```

Switch item: q030
Skip condition: q030 is NOT 1, 2, 3, or 6
Dependent item: q035

Note that q030_ck1, the check variable 1 associated with the pair q030 and q035, is not defined where the dependent item q035 is missing. It only assesses whether q035 was *correctly filled in* (due to skip being correctly observed).

```
/*************************************/
/*           STEP 3                  */
/*    Verify Skip Patterns           */
/*         FREQ & XTAB               */
/*************************************/

FREQUENCIES
VARIABLES=
          q030 q030_ck1
          q035 q035_ck1
/ORDER=  ANALYSIS .

/*****************************************/
/*   KNOW MIN              */

CROSSTABS
  /TABLES=
          q030  BY q030_ck1
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT
  /COUNT ROUND CELL .

/***********************************/
/*   KNOW T-F               */

CROSSTABS
  /TABLES=
          q035  BY q035_ck1
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT
  /COUNT ROUND CELL .

/*******************************/
```

Verify correct coding of check variable 1 (binary skip check) by means of frequencies and cross-tabulations.

```
/*************************************************************/
/*          STEP 4                                           */
/*       Code Qx_ck02                                        */
/*     Create Valid Skip Reporting Variable                  */
/*************************************************************/

/*********************************************************************/
/*  q030_ck2      KNOW MOD ACTIV --Reporting Vari      */

COMPUTE q030_ck2 = q030.
IF (MISSING(q030_ck1) = 1 ) q030_ck2 = -9.
IF (q030_ck1 = 2 ) q030_ck2 = -8.
IF (MISSING(q030_ck2) = 1) q030_ck2= -7.
EXECUTE.

FORMATS
        q030_ck2 (F2.0).
VARIABLE LABELS
        q030_ck2  'KNOW MOD ACTIV --q030_Chk2- q030_ReportingVariable'.
VALUE LABELS q030_ck2
        1       "One hour or more"
        2       "45 minutes"
        3       "30 minutes"
        4       "20 minutes"
        5       "15 minutes"
        6       "OTHER TIME OVER 30 MINUTES (SPECIFY)"
        7       "DON'T KNOW / NOT SURE"
        8       "OTHER TIME UNDER 30 MINUTES (SPECIFY)"
        9       "REFUSED"
        -7      "No data - please check"
        -8      "Invalid answer - data removed"
        -9      "Assume Valid Skip: Not Valid Knowledge of Moderate # of Days".
MISSING VALUES Q030_CK2 (lo thru -1).

EXECUTE .

/*********************************************************************/
/*  q035_ck2      KNOW T-F -- Reporting Vari                */

COMPUTE q035_ck2 = q035 .
IF (MISSING(q035_ck1) = 1 ) q035_ck2 = -9.
IF (q035_ck1 = 2 ) q030_ck2 = -8.
IF (MISSING(q035_ck2) = 1) q030_ck2= -7.
EXECUTE.

FORMATS
        q035_ck2 (F2.0).
VARIABLE LABELS
        q035_ck2  'KNOW T-F --  q035_Chk2- q035_ReportingVariable'.
VALUE LABELS q035_ck2
        1       "TRUE"
        2       "FALSE"
        7       "DON'T KNOW / NOT SURE"
        9       "REFUSED"
        -7      "No data - please check"
```

```
        -8      "Invalid answer - data removed"
        -9      "Assume Valid Skip: Not Valid Moderate Time Known".
MISSING VALUES Q035_CK2 (lo thru -1).
EXECUTE .

/*********************************************************/
```

Notes on the creation of Check Variable 2 (reporting variable):

- Check Variable 1 would be missing if the original dependent item were missing. In this case, Check Variable 2 is assigned the error code -9, "Not Valid Knowledge of Moderate # of Days". This refers to the skip condition based on the previous item, q030 (and implicitly assumes that this skip condition is the sole reason for the missing value). However, there could be additional reasons if skip conditions earlier in the survey cause the item to be skipped.
- Where the dependent item is *incorrectly* filled in, check variable 2 (reporting variable) has the error code -8 "Invalid answer - data removed" instead. Presence of this code indicates the need for further investigation (there may be a CATI error)[6]
- If Check Variable 2 is still missing after the above explanatory codes have been attached, the code -7 "No data - please check?" is assigned. Presence of this code indicates the need for further investigation (there may be a CATI error)
- All error codes are set to missing so that they do not show up in most SPSS statistical operations (cross-tabulations, frequencies, means etc.) unless specifically requested.

Check variable 2 coding therefore reflects what the dependent item would have looked like had the skip been correctly observed, *without* overwriting the data in the original. It can be used in place of the original dependent item for reporting purposes.

```
/*********************************************************************/
/*      STEP 5                                                       */
/*    DOUBLE CHECK                                                   */
/*    Verify Skip Patterns for REPORTING VARIABLE-                   */
/*********************************************************************/

FREQUENCIES
  VARIABLES=
        q030 q030_ck2
        q035 q035_ck2
/ORDER=  ANALYSIS .
EXECUTE.

/*******  XTABS  **********/

/*********************************/
/*  KNOW MOD TIME  */

CROSSTABS
  /TABLES=
```

[6] This assumes that Check Variable 1 has been programmed correctly. Be sure to confirm that it captures all cases corresponding to skip condition and continuation condition.

```
          q030  BY q030_ck2
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT
  /COUNT ROUND CELL .
EXECUTE.

/*********************************/
/*  KNOW T-F  */

CROSSTABS
  /TABLES=
          q035  BY q035_ck2
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT
  /COUNT ROUND CELL .
EXECUTE.

/*********************************/
```

Check variable 2 (the reporting variable) should contain the same information as the original dependent item, but substitute an error code where the item was filled in but should have been skipped. Verify against the dependent item by means of frequencies and cross-tabulations.

Section 6.02 Syntax (Missing-Codes)

This code snippet is based on the foregoing skips table. Comments are included for pedagogical purposes.

Note: See Appendix 2 for output from this code.

```
** filename "skip coding example v2.sps".
** author A Kavanaugh date 9/24/2006.

** Run Mike's cleaning first to produce the reporting variables.

title "Skips coding (PDA)".

** Sample skips, PDA-style.

** Restrict attention to the following items: q025 q030_ck2 q035_ck2 q040.
** We will be attaching skips codes to check variable 2 for each of q030 and q035, .
** rather than to the original survey data.

** We create codes to document all the ways in which a question is skipped.

** The "Field not filled in" value will be -9 (missing due to skip) for ease of handling.
** In survey sets embedded inside disposition data, this lets us see where a record belongs .
** ('true' sysmis indicates it's not in the survey set, negative/missing values .
```

```
** correspond to survey records) .

** for the purposes of this example, only the skipped items are recoded this way .


** make sure no missing values are set.
missing values q025 q030 q030_ck2 q035 q035_ck2 q040
 ().

freq q025 q030_ck2 q035_ck2 q040 .


** if q025 is 77 (DK) or 99 (ref) skip to q040.
** items skipped: q030_ck2 q035_ck2 .

crosstabs q030_ck2 q035_ck2 by q025
 /missing=include.

** Note that assignment of a skip code is done ONLY if the item was originally missing due to skip .
do repeat a= q030_ck2 q035_ck2.
if (q025 eq 77 and a eq -9) a=-2577.
if (q025 eq 99 and a eq -9) a=-2599.
end repeat.

** Labels can vary; usually we try to indicate content of the switch item.
** Code documents the item number and value filled in.
add value labels  q030_ck2 q035_ck2
 -2577 "Min days per week: don't know"
 -2599 "Min days per week: refused".

crosstabs q030_ck2 q035_ck2 by q025
 /missing=include.

** q030_ck2 asked only if q025=1-7 (o/w skip to q040, the next "all respondents" item).
** skip condition: (q025 lt 1 or (q025 gt 7 and q025 ne 77 and q025 ne 99)).
** skip to q040; items skipped: q030_ck2 .

crosstabs q030_ck2 q035_ck2 by q025
 /missing=include.

** Here we do "item 25, value 00" as a generalized "out of range" code.
do repeat a=q030_ck2 q035_ck2.
if ((q025 lt 1 or (q025 gt 7 and q025 ne 77 and q025 ne 99)) and a eq -9) a=-2500.
end repeat.

add value labels q030_ck2 q035_ck2
 -2500 "Min days per week: out of range [1,7]".

crosstabs q030_ck2 q035_ck2 by q025
 /missing=include.

** Note: we use the original value in the skip condition (rather than the check variable 2).
** This is because original survey value would have governed skip behavior.

** q035_ck2 asked only if q030=1,2,3,6 (o/w skip to q040, the next "all respondents" item).
** skip conditons fall into three different groups (we usually distinguish at least these) .
```

```
** Incorrect response: q030 eq 4,5,8 .
** Don't know: q030 eq 7 .
** Refused: q030 eq 9 .

crosstabs q035_ck2 by q030
 /missing=include.

** here we assign 11 to designate "incorrect responses" (it's not an actual survey value).
if ((q030 eq 4 or q030 eq 5 or q030 eq 8) and q035_ck2 eq -9) q035_ck2=-3011.

if (q030 eq 7 and q035_ck2 eq -9) q035_ck2=-3007.
if (q030 eq 9 and q035_ck2 eq -9) q035_ck2=-3009.
execute.
add value labels q035_ck2
 -3011 "Min exercise time per day for health: underestimated"
 -3007 "Min exercise time per day for health: don't know"
 -3009 "Min exercise time per day for health: refused".

crosstabs q035_ck2 by q030
 /missing=include.


** PDA-standard: missing codes are negative.
missing values q030_ck2 q035_ck2
 (lo thru -1).

** Do frequencies: we now see the effect of all skip conditions that skip OVER q030_ck2.
freq q025 q030_ck2 q035_ck2 q040  .
```