Graduate Student Journal of Psychology
2022, Vol. 19
Copyright 2022 by the Department of Counseling and Clinical Psychology
Teachers College, Columbia University

Merging Multiple and/or Divergent Datasets using SPSS: A Method Review and Tutorial

Lillian M. Audette, Katherine A. Johnson, Marie S. Hammond, Jenna S. Lehmann, and Michael Oyeteju,
Department of Psychological Sciences & Counseling, Tennessee State University

For many reasons, researchers place their data in multiple electronic datasets and later find that they wish to combine those datasets for a particular analysis. This article reports on an analysis of the extant literature on combining multiple and/or divergent datasets and provides both a tutorial and related syntax that combines non-matching datasets (i.e., from different sources) in such a way that all data is contained in the final combined dataset with identification of data source. This syntax represents an improvement over the existing SPSS (v23-v27) combining dataset routines in terms of 1) expanding the types of datasets that can be combined, 2) completeness of the resulting dataset, and 3) facilitating management of multiple and/or divergent data sets. An additional benefit of using this process is the incorporation of a method to test the accuracy of the merged data and thus verify the data quality. The SPSS syntax along with instructions and examples are reviewed in the article.
Keywords: SPSS, syntax, datasets, merging, longitudinal data

Many researchers face problems merging multiple datasets with divergent or mismatched cases and/or variables to form a more complete dataset. This is particularly true when conducting research with humans in which data is obtained from multiple sources, such as self-report questionnaires, institutional databases, and/or collateral contacts. Challenges that occur in utilizing data from divergent sources include different structures, different formats, or that data is incomplete when compared to the original dataset.

The present article focuses on merging multiple datasets either generated longitudinally or containing different variables. Examples of the datasets to which we refer include pre-/post-testing for intervention research, longitudinal research utilizing two or more waves of data, or instances in which data from multiple sources related to the same case. These datasets are likely to be mismatched in that they may not contain the same variables (in the instance of adding cases) or may not include the same cases (in the instance of adding variables). Other relevant datasets might be considered "complex" in that the datasets contain different variables and have differences in cases. For example, it may be that graduate students working as a part of a larger research team are tasked with managing the data and will need to understand the process and steps to efficiently combine these datasets.

It should be noted that in order to align with current terminology, rather than describing these datasets as "complex," which has a specific meaning within the social sciences (Guha et al., 2009), we will use the term "divergent" to refer to datasets that do not match variable for variable, time frame by time frame, and/or case for case. Further, the term "merge" will be used in this article rather than "combining" in that combining implies a common data structure that facilitates bringing the datasets together, while the term "merging" typically relies on a "key" or "identifier" variable that provides the common link between the datasets. This is distinct from data fusion, which occurs when multiple data sources are integrated without full preservation of all data (Haghighat et al., 2016). The goal of the process to be described is that the datasets are merged in such a way that all data is retained from its original dataset in the final product.

When it comes to merging two or more datasets that contain mismatching and/or missing cases, several complications could occur – resulting in compromised analyses. For instance, mismatched data may indicate that data were added to the dataset incorrectly. Additionally, missing data can result in decreased analysis power or, depending on the analysis required, may prevent an analysis from being completed entirely. Accurate dataset merging is vital to prevent potential errors from being introduced, prevent data loss, and allow researchers to have more reliable and valid datasets with which to conduct analyses. Thus, the accurate merging of discrete datasets allows researchers to conduct analyses on a more accurate dataset, therefore producing results that have greater reliability, are more generalizable, and are more replicable. For the purposes of this article, the focus will stay on mismatched or divergent dataset merging using SPSS software.

Literature Review

To assess the need for an informative guide regarding how to merge mismatched datasets, the authors chose two separate approaches to conducting the lit

IBM v. Digital Doors,
IPR2023-00969, 7,339,987
DigitalDoors EX2032

erature searches. First, a literature search using the research databases EBSCOHost (which is an umbrella dataset encompassing approximately 70 individual publisher databases), Web of Science, Sage Premier, Science Direct, and Google Scholar was conducted. The Boolean search phrase utilized for this search was as follows: SPSS AND dataset AND (merg* OR combin* OR concatenat* OR mung* OR wrangl*). For databases that do not use the Boolean system exactly (e.g., Google Scholar), these search terms were added manually to the advanced search settings of each database. The terms mung and wrangle were incorporated once we discovered that these terms are used outside of psychology to describe the data-cleaning process, which includes dataset merging (Braun et al., 2018; Endel & Piringer, 2015; Rattenbury et al., 2017). The above search phrase and its individual search terms were used to identify articles containing these terms in their abstracts. The resulting list from EBSCOHost provided 27 peer-reviewed articles, while the Web of Science search produced 24. None of these results acknowledged the problem of merging mismatched or divergent datasets. A second literature search was conducted using Google and Google Scholar. Since Google Scholar uses a limited form of Boolean, the search phrase was changed to the following: SPSS AND dataset AND (merge OR merged OR merging OR combine OR combination OR combining OR combined OR mung OR munging OR munged OR concatenate OR concatenated OR concatenating OR wrangle OR wrangled OR wrangling). Other combinations of these search terms were also used (e.g., data wrangling, data munging, merging datasets). The results of this search found examples of both peer-reviewed articles and books that address dataset merging, but neither focused on mismatched dataset merging nor used SPSS for this purpose.

Previous research literature on merging datasets for analyses of psychological data primarily used the Statistical Analysis System (SAS) software (Foley, 1998; Foley, 2005) for this process, rather than SPSS. Parenthetically, research literature reviewed the steps to obtain a merged dataset with the specific variables and cases of interest. Again, however, no mention was made of merging mismatched or divergent datasets. Scholarly works and textbooks were reviewed to identify potentially helpful instruction in this area. Both works that did focus on the use of SPSS and works that did not specify which program they recommended using covered only basic dataset merging practices. These sources either did not provide information on the difficulties likely to be encountered in merging mismatched or divergent datasets (Stehlik-Barry & Babinex, 2017) or they only emphasized the importance of considering these challenges without providing guidance or details on its technicalities (Braun et al., 2018; Endel & Pringer, 2015; Rattenbury et al., 2017). Guides on merging datasets for non-psychological research purposes have also not focused on SPSS, but rather on software programs such as R and Python (Ojeda et al., 2014).

There does, however, appear to be plenty of gray literature (articles not formally published by commercial academic publishers) surrounding this topic in SPSS (Haddaway et al., 2015, p. 1). Examples of this gray literature include 1) websites that host questions, discussions, and video tutorials related to the common practices for merging datasets (Coyer, 2013; Truong, 2016), 2) academic websites with instructions (Coleman, n.d.; Glynn, 2002), 3) basic commands given by International Business Machine (IBM; IBM, n.d.); and 4) other SPSS tutorial websites (Spss-tutorials.com, n.d.). However, most of the gray literature provides limited basic commands such as MERGE DATASETS or COMPARE DATASETS, which do not adequately address specific divergent or mismatched dataset merging issues encountered by researchers.

Dataset Merging Methods Requirements

To address the above-mentioned issues, the extant literature previously identified was reviewed to surface existing guidelines and/or requirements. This type of information was found primarily in the gray literature (e.g., Coleman, n.d.; Coyer, 2013; DeCator, 2015; IBM, n.d.; Truong, n.d.). Six requirements of effective dataset merging methods to ensure the accuracy of data were identified from this literature. The first three requirements are basic expectations of any dataset merging method (Coleman, n.d.; Coyer, 2013). The next three requirements are more complex and are specific to merging datasets that contain overlapping variables as well as overlapping cases, and whose cases sometimes contain mismatched data (DeCator, 2015; IBM, n.d.; Truong, n.d.). These six requirements have implications for the structure and variables contained in the final created database, as will be discussed below.

IBM v. Digital Doors,
IPR2023-00969, 7,339,987
DigitalDoors EX2032

Requirement 1: Include All Cases
The merged database (finished product) should contain all the cases from Dataset 1 and all the cases from Dataset 2 (and any additional datasets). The case data from each respective dataset should be faithfully replicated in the merged database.

Requirement 2: Include All Variables
The merged database should hold all the variables present in all datasets, as well as all variables present in only one of the datasets. The width, number of decimal places, labels, value labels, missing specification, column width, alignment, measure specification, and role of each variable should be faithfully replicated in the merged database.

Requirement 3: Variable Settings Fidelity
The merging method integrates variables without changing the variable regardless of the type (numeric, string) and regardless of measure (scale, nominal, ordinal).

In addition to the three basic requirements listed above (Coleman, n.d.; Coyer, 2013), the merged database should provide researchers with three additional pieces of information. The following requirements are more complex, intending to assist researchers when the datasets to be merged are suspected of containing unique, overlapping, or mismatched cases, all of which should be included in the final database.

Requirement 4: Indication of Unique Cases by Dataset
A merged database should provide information as to when a case is unique to Dataset 1 or Dataset 2 (rather than being present in both original datasets), and which dataset it originates from. For example, a merged dataset should tell us if a case with ID 13 exists uniquely in Dataset 1 or Dataset 2.

Requirement 5: Indication of Overlapping Cases
A merged database should indicate when a case in Dataset 1 is also a case in Dataset 2. For example, a merged database should tell us if a case with ID 16 originated in Dataset 1 and whether it also exists in Dataset 2.

Requirement 6: Indication of Mismatched Data by Case
A merged database should tell us when a case in Dataset 1 and a case in Dataset 2 have a matching ID but contain mismatching data while retaining both instances of data in the merged database. Detecting mismatching data is a crucial requirement when merging datasets whose data, cases, or variables do not perfectly match using SPSS. For example, a merged database should tell us if a case with ID 16 in Dataset 1 has "Brief Cognitive Behavioral Therapy" entered for the "Treatment" variable, but a case with ID 16 in Dataset 2 has "Brief Object Relations Therapy" entered for the "Treatment" variable. In addition, both instances of data should be preserved within the merged database. For example, we should see both forms of cases with ID 16 in the merged database, one with "Brief Cognitive Behavioral Therapy" and one with "Brief Object Relations Therapy" entered for "Treatment." Thus, an indicator of match/mismatch and the retention of both instances of the data should appear in the merged database.

To clarify - several of the six requirements (discussed above) should be represented by one or more variables within the merged dataset. New variables within the dataset should be generated to represent at least one, if not more, of the six requirements. Each of the six requirements should be encoded into a variable within the dataset. The final merged dataset should include, in a specific variable, the dataset name from which each case was taken. This must be a unique variable, separate from the other variables. For example, for one of the authors' research projects, data was gathered from three different institutions. In this instance, in merging the data a new variable was created which identified the dataset/institution of origin. This variable, included in the merged dataset, provides information about the origin of the data and thus met this requirement. Thus, the dataset of origin variable should not rely on non-encoded methods (such as case origin based upon the cases' ordering - i.e., SPSS row number) within the dataset. If case origin is denoted by case order, and the final merged dataset is ever re-ordered, the information regarding case origin could be lost. To summarize, each of the six dataset merging requirements must be fulfilled by being encoded as separate variables within the dataset or in such a way that the information cannot be easily lost.

Conducting the Dataset Merge

Currently, SPSS does not include a built-in function that would perform a dataset merge that ensures all six requirements (above) are met. For example, the ADD FILES command does not identify cases that are unique to one of a researcher's original datasets,

nor is it able to identify duplicate cases that have mismatched data, thus, not meeting requirements 4 and 6. The MERGE DATASETS command does not meet requirements 4, 5, or 6. In order to meet requirement 1 using MERGE DATASETS, additional syntax is needed. Similarly, the MATCH FILES command does not meet requirements 1, 4, 5, or 6. Finally, the COMPARE DATASETS command does not meet requirements 1, 2, 3, or 4. In determining which syntax best meets all six requirements, a literature review was conducted which yielded five (5) known possible methods for combining datasets in SPSS. A total of 12 experiments were conducted using the five methods, with each method first being tested in its simplest form and then tested again in more complex forms. A summary of the experiments conducted, and their outcomes can be found in Table 1.

The final product (based on the outcome of experiment 12) was a set of SPSS syntax to successfully overcome the challenges associated with merging mismatched datasets using the least number of steps. It should be noted that the following steps were developed using SPSS v23 (IBM Corp., 2015) and tested using SPSS v25 and v27 (IBM Corp., 2017, 2020). It should be noted that conducting the merge using this syntax works across all versions including SPSS v28 (IBM Corp., 2021); however, changes made to v28 in the point-and-click options may create a mismatch between the instructions provided herein.

SPSS Syntax

The following instructions on how to use the syntax are applicable when researchers have two datasets that they wish to merge. It can also be used repeatedly if more than two datasets must be merged. Please refer to Appendix A for the syntax related to each step. It is also important to note that the figures provided throughout this section are simplified graphics of the analysis for demonstration only and were not conducted using the supplemental files. The supplemental files are provided for the purpose of practicing the procedure with a more realistic dataset.

Original Datasets

For variables shared by the two original datasets, the variable information should match perfectly (type, width, label, values, etc.). The following steps will erroneously identify matching variables as separate variables when they do not have identical characteristics (type, width, etc.). Before beginning the next steps, identify which dataset will be your "Dataset 1" and open both original datasets. See Figure 1 for an example of two original datasets to be merged. It is also necessary to clean the data from each of the datasets to be merged to the best of one's ability before beginning the dataset merging process.

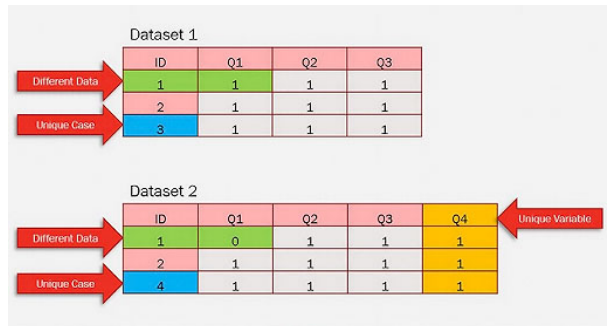

Figure 1: Simple Datasets Pre-Combination. This is a simple example of two separate, mismatched datasets.

Syntax

Step 1: Comparing Cases Between Datasets

The purpose of Step 1 is to create a new variable that encodes the number of unique cases present in Dataset 1, the number of cases present in both original datasets that are perfectly matched, and the number of cases present in both original datasets that contain mismatching data. The output will identify the location of the mismatched data in the dataset. Please note that the datasets will remain separate. Step 1 uses point-and-click to run the COMPARE DATASETS command (which can be found under the DATA menu). Before running the command, make the following point-and-click changes. In the "Compare" tab, put the ID variable in Case IDs, and put all "Matched Fields" into "Fields to Compare." In the "Attributes" tab, select "Do not compare the data dictionaries." In the "Output" tab select "flag mismatches in a new field" and name the new variable what you wish (we used "Mismatches"). Also, in the "Output" tab, unselect "Limit the case-by-case table." Click "paste" and run the resulting syntax on Dataset 1. See Figures 2A and 2B for example Datasets 1 and 2 after Step 1 as well as an example output after Step 1.

Dataset 1

ID	Q1	Q2	Q3	Mismatches
1	1	1	1	Mismatch
2	1	1	1	Match
3	1	1	1	Unmatched

Dataset 2

ID	Q1	Q2	Q3	Q4
1	0	1	1	1
2	1	1	1	1
4	1	1	1	1

Figure 2A: Simple Mismatched Datasets in SPSS after Step 1. This is an example of what the new variable created via Step 1 would look like once added.

Matched Summary

Results	Statistics	Datasets Active	Datasets Comparison
Cases	Count	3	3
Cases Compared	Count	2	2
	Percent	66.7%	66.7%
Cases Not Compared	Count	1	1
	Percent	33.3%	33.3%

Mismatched By Cases

Cases Compared	Count	2
Cases Containing Mismatches	Count	1
	Percent	50.0%

Mismatched By Variables

Variables	Mismatched Count	Mismatched Percent[a]
ID	0	0.0%
Q1	1	50.0%
Q2	0	0.0%
Q3	0	0.0%

a. Based on 2 cases compared

Case By Case Comparison

Case ID	Row				
ID	Active	Compare	Q1	Q2	Q3
1	1	1	(1) 1 (2) 0		

(1) is the Active Dataset and (2) is the Comparison Dataset

Figure 2B: Example Step 1 Output. This is an example of the output that would be generated after completing Step 1 which provides a summary of case comparisons between the two datasets.

Step 2: Merging Variables and Cases into One Dataset and Identifying Cases by Dataset

Step 2 merges all variables and cases into Dataset 1. From Dataset 1, select MERGE FILES then ADD CASES command. This is also found in the DATA menu. Before running the command, make the following point-and-click changes. Put all "Unpaired Variables" in "Variables in New Active Dataset." As needed, manually pair variables from the two original datasets. Select "Indicate case source as variable" and name the new variable that you wish to use (we used "SourceDataset2"). Click "paste" and run the resulting syntax on Dataset 1. This step will add all variables and all cases to Dataset 1. It will also identify which cases come from Dataset 1 or Dataset 2 in the newly created variable. See Figure 3 for the example Dataset 1 after Step 2. After completing this step, it may be helpful to label the values of this new variable, the syntax for which can be found in Appendix A. A value of 1 indicates that a case originates from Dataset2 while a value of 0 indicates that a case originates from Dataset 1.

Dataset 1

ID	Q1	Q2	Q3	Mismatches	Q4	SourceDataset2
1	1	1	1	Mismatch	.	from Dataset 1
2	1	1	1	Match	.	from Dataset 1
3	1	1	1	Unmatched	.	from Dataset 1
1	0	1	1	.	1	from Dataset 2
2	1	1	1	.	1	from Dataset 2
4	1	1	1	.	1	from Dataset 2

Figure 3: SPSS Data View After Step 2. This is an example of the variables added after step 2 which includes all variables existing in Dataset 2 that do not exist in Dataset 1 and a new variable that identifies the source of each case.

Step 3: Identifying Mismatches, Matches, and

Unique Cases Within Dataset 1

Step 3 duplicates the variable created in Step 1, which identifies mismatching, matching, and unique cases present in Dataset 1. The duplicated variable is the one that will be manipulated in Steps 4 and 5. Step 3 can be done with the syntax (provided in Appendix A) using your preferred variable names inserted (we used "CasesCompared"). The resulting duplicate variable only encodes information about matches, mismatches, and unique Dataset 1 cases for those cases from Dataset 1. The duplicated variable contains no information for cases added from Dataset 2, a problem that will be addressed in Step 4 and Step 5. See Figure 4 for the example Dataset 1 after Step 3. Once again, we suggest labeling the values of this new variable. A value of 1 in

dicates a mismatch between both datasets, 0 represents a match in both datasets, and -1 indicates that a case is unique to Dataset1. The syntax for creating these labels can be found in Appendix A.

Dataset 1

ID	Q1	Q2	Q3	Q4	Mismatches	SourceDataset2	CasesCompared
1	1	1	1	.	Mismatch	from Dataset 1	Mismatch between B...
2	1	1	1	.	Match	from Dataset 1	Match in Both Datasets
3	1	1	1	.	Unmatched	from Dataset 1	Unique to Dataset1
1	0	1	1	1	.	from Dataset 2	.
2	1	1	1	1	.	from Dataset 2	.
4	1	1	1	1	.	from Dataset 2	.

Figure 4: SPSS Data View After the Creation of a Variable that Identifies Mismatches, Matches, and Unique Cases Regarding Cases Originating From Dataset 1 (Step 3).

Step 4: Labeling Matches and Mismatches from Dataset 2

Step 4 uses the syntax we developed (utilizing the LAG function), and the variables created in Steps 2 and 3. Utilizing the LAG function and the "SourceDataset2" variable, Step 4 encodes for all cases within the newly created "CasesCompared" variable (from Step 3). Step 4 encodes information as to whether the case is a match between datasets, a mismatch between datasets, or unique to Dataset 1. This step is important because without it only cases from Dataset 2 would have information encoded in the "CasesCompared" variable. Again, the syntax is found in Appendix A. When running the syntax, be careful to change the variable names to your chosen variable names, and to run the sorting syntax first (we included examples to facilitate understanding). Your ID variable must be in numerical descending order (e.g. 1, 1, 2, 3, 4, 4), with cases from Dataset 1 listed or appearing before cases from Dataset 2. This syntax solves the problem from Step 3, however, the variable "CasesCompared" still does not encode when a case is unique to Dataset 2. See Figure 5 for the example Dataset 1 after Step 4.

Dataset 1

ID	Q1	Q2	Q3	Q4	Mismatches	SourceDataset2	CasesCompared
1	1	1	1	.	Mismatch	from Dataset 1	Mismatch between B...
1	0	1	1	1	.	from Dataset 2	Mismatch between B...
2	1	1	1	.	Match	from Dataset 1	Match in Both Datasets
2	1	1	1	1	.	from Dataset 2	Match in Both Datasets
3	1	1	1	.	Unmatched	from Dataset 1	Unique to Dataset1
4	1	1	1	1	.	from Dataset 2	.

Figure 5: SPSS Data View After Step 4 which Identifies Mismatches and Matches in Cases Originating from Dataset 2.

Step 5: Identifying Cases Unique to Dataset 2

Step 5 encodes in the "CasesCompared" variable when cases are unique to Dataset 2. The syntax can be found in Appendix A. Since the only cases with no data in "CasesCompared" are those which are unique to Dataset 2, this syntax identifies empty data in "CasesCompared" and encodes them as unique to Dataset 2. The syntax also includes a method of labeling this new value. See Figure 6 for the example Dataset 1 after Step 5.

Dataset 1

ID	Q1	Q2	Q3	Q4	Mismatches	SourceDataset2	CasesCompared
1	1	1	1	.	Mismatch	from Dataset 1	Mismatch between Both Datasets
1	0	1	1	1		from Dataset 2	Mismatch between Both Datasets
2	1	1	1	.	Match	from Dataset 1	Match in Both Datasets
2	1	1	1	1		from Dataset 2	Match in Both Datasets
3	1	1	1	.	Unmatched	from Dataset 1	Unique to Dataset1
4	1	1	1	1		from Dataset 2	Unique to Dataset2

Figure 6: SPSS Data View After Step 5 which Identifies Cases Unique to Dataset 2.

Discussion

There is significant literature that provides tutorials and instructions on the basic merging of files with identical variables and/or cases. Significant literature discussing these basic merging techniques can be found for programs such as R, SPSS, and STATA. What has not been discussed in the literature but may be of use to students and researchers using small- to medium-sized datasets, is a procedure that reliably merges datasets with missing or mismatched cases and/or variables within SPSS. While numerous software programs are used to conduct research, one of the more frequently used software programs to teach statistical analysis, particularly within the social sciences, is SPSS. Oftentimes students are introduced to data analysis using the drop-down menus within SPSS rather than syntax. This article has provided information on both the use of drop-down menus and syntax to conduct data merging. Depending on the individual's comfort level and the version of SPSS being used, either of these procedures may be more effective. In the case of an individual using version 27 or earlier, the drop-down menus provide point-and-click ease of conducting the analysis. While for those with greater comfort and/or facility using syntax, merging using the syntax is likely to be more comfortable regardless of the version used.

There are several advantages to using this procedure to merge divergent datasets. The first advantage is that it allows the researcher to store all data within a single dataset. This facilitates data analysis for most procedures. The second advantage is that by having all data in one dataset, analyses accounting for missing

data are more easily conducted from within one dataset rather than across multiple datasets. The third advantage is that this procedure allows one to conduct analyses that can only be performed within a single dataset. The fourth advantage is that embedded within the syntax is a data quality check which increases the likelihood that the merged data is accurate and represents the population of interest.

An additional consideration is related to the speed or efficiency of the procedure. It should be noted that the speed, efficiency, and amount of storage required to contain the dataset will be affected by the sizes of the datasets involved. Large datasets (such as those found in the National Center for Education Statistics, etc.), will require greater computing power and storage capacity than smaller datasets.

Conclusion

Merging datasets accurately is vital to ensuring that no data is altered or lost and that researchers can easily understand the sources of all their data. Presently, there is no consensus on best practices for merging SPSS datasets with overlapping and potentially mismatched cases. The present article introduces one method to handle such a situation. The recommended method results in a merged dataset that includes all cases and variables, in their original form, stemming from two or more original datasets. It creates an output file that identifies each instance of case mismatch and the location of each mismatch within the dataset. It also results in a single variable that encoded whether a case was 1) a perfect match between datasets 2) a mismatch between datasets with non-matching data 3) unique to one dataset and which original dataset contains that unique case. Alternative strategies are recommended when the datasets to be merged contain overlapping cases but no or few shared variables.

References

Braun, M. T., Kuljanin, G., & DeShon, R. P. (2018). Special considerations for the acquisition and wrangling of big data. *Organizational Research Methods, 21(3)*, 633-659. https://doi.org/10.1177/1094428117690235

Coleman, M. (n.d.). *Merging datasets in SPSS*. Retrieved December 12, 2017, from http://www.d.umn.edu/~mcoleman/tutorials/spss/merge.html

Coyer, L. (2013, April 20). *How can you merge two files in SPSS when the cases are not perfectly the same?* [Online forum post]. ResearchGate. Retrieved December 12, 2017, from https://www.researchgate.net/post/How_can_you_merge_two_files_in_SPSS_when_the_cases_are_not_perfectly_the_same.

DeCator, D. D. (2015, February 8). *SPSS Syntax Part 4: Double Entry Comparison.* Retrieved October 12, 2017, from https://www.ddecator.com/blog/2015/2/spss-syntax-part-4-double-entry-comparison

Endel, F., & Piringer, H. (2015). Data wrangling: Making data useful again. *IFAC-PapersOnLine, 48(1),* 111-112. https://doi.org/10.1016/j.ifacol.2015.05.197

Foley, M. J. (1998). Match-merging: 20 some traps and how to avoid them. In the P*roceedings of the 23rd Annual SAS Users Group International Conference* (pp. 277-286). Retrieved from https://support.sas.com/resources/papers/proceedings/proceedings/sugi23/Advtutor/P47.pdf

Foley, M. J. (2005). Merging vs. joining: Comparing the DATA step with SQL. In the *Proceedings of the 30th Annual SAS Users Group International Conference* (pp. 184–200). Retrieved from http://www.scsug.org/SCSUGProceedings/2005/Foley_Merging vs Joining - 184.pdf

Glynn, P. (2002). SPSS for Social Science Research - Using Syntax. Retrieved December 12, 2017 from http://staff.washington.edu/glynn/spssclas/

Guha, S., Kidwell, P., Hafen, R.P., & Cleveland W.S. (2009). Visualization databases for the analysis of large complex datasets. Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), USA, 193-200. Retrieved October 20, 2022 from https://proceedings.mlr.press/v5/guillory09a.html.

Haddaway, N. R., Collins, A. M., Coughlin, D., & Kirk, S. (2015). The role of Google Scholar in evidence reviews and its applicability to grey literature searching. *PLOS ONE, 10(9)*. https://doi.org/10.1371/journal.pone.0138237

Haghighat, M., Abdel-Mottaleb, M., & Alhalabi, W. (2016). Discriminant correlation analysis: Real-time feature level fusion for multimodal biometric recognition. *IEEE Transactions on*

IBM v. Digital Doors,
IPR2023-00969, 7,339,987
DigitalDoors EX2032

Information Forensics and Security, 11(9), 1984–1996. https://doi.org/10.1109/TIFS.2016.2569061

IBM Corp. (2015). IBM SPSS Statistics for Windows (Version 23.0) [Computer software]. IBM Corp.

IBM Corp. (2017). IBM SPSS Statistics for Windows (Version 25.0) [Computer software]. IBM Corp.

IBM Corp. (2020). IBM SPSS Statistics for Windows (Version 27.0) [Computer software]. IBM Corp.

IBM Corp. (2021). IBM SPSS Statistics for Windows (Version 28.0) [Computer software]. IBM Corp.

IBM Corp. (n.d.). Merging Data Files. Retrieved December 12, 2017 from https://www.ibm.com/support/knowledgecenter/en/SSLVMB_23.0.0/spss/base/idh_idd_add _gating.html

Ojeda, T., Dasgupta, A., Bengfort, B., & Murphy, S. P. (2014). *Practical data science cookbook.* Packt Publishing.

Rattenbury, T., Hellerstein, J. M., Heer, J., Kandel, S., & Carreras, C. (2017). *Principles of data wrangling: Practical techniques for data preparation.* O'Reilly Media.

Spss-tutorials.com. (n.d.). *Merging Data Files in SPSS.* Retrieved December 12, 2017 from https://www.spss-tutorials.com/merging-data-files/

Stehlik-Barry, K. & Babinec, A. J. (2017). *Data analytics with IBM SPSS statistics.* Packt Publishing.

Truong, D. [Dothang Truong]. (2016, July 21). *SPSS missing values.* [Video]. YouTube. https://www.youtube.com/watch?v=DNThjjqLz9Q

Table 1

Experiments Conducted to Determine Best Syntax for Merging Divergent Datasets

Requirement #	1	2	3	4	5	6	# of req. met	# of steps
Experiment 1 - Basic ADD FILES command, without additional syntax	X	X	X		X		4	1
Experiment 2 - ADD Files command + /BY, /MAP, /KEEP	X	X	X		X		4	2
Experiment 3 - MERGE DATASETS by point-and-click, variables first		X	X				2	1
Experiment 4 - MERGE DATASETS by point-and-click, cases first	X	X	X				3	1
Experiment 5 - MERGE DATASETS by point-and-click, then DO REPEAT, first attempt	X	X	X				3	3
Experiment 6 - MERGE DATASETS by point-and-click, then DO REPEAT, second attempt	X						1	3
Experiment 7 - MERGE DATASETS by point-and-click, then DO REPEAT, third attempt	X	X	X				3	3
Experiment 8 - Basic MATCH FILES command, without additional syntax		X	X				2	1
Experiment 9 - MATCH FILES command, with additional syntax		X	X				2	2
Experiment 10 - Basic COMPARE DATASETS command, without additional syntax					X	X	2	1
Experiment 11 - COMPARE DATASETS command, with additional syntax	X	X	X	X	X	X	6	6
Experiment 12 - COMPARE DATASETS command, with additional syntax, without identifying duplicate cases	X	X	X	X	X	X	6	5

Appendix A
Syntax for Steps One Through Five

Step 1:
```
***Step 1***
*compare datasets
DATASET ACTIVATE DataSet1.
SORT CASES BY ID.
COMPARE DATASETS
   /COMPDATASET = DataSet2
   /VARIABLES ALL
   /CASEID ID
   /SAVE FLAGMISMATCHES=YES VARNAME=Mis-
matches MATCHDATASET=NO
MISMATCHDATASET=NO
   /OUTPUT VARPROPERTIES=NONE CASE-
TABLE=YES TABLELIMIT=600.
```

Step 2:
```
***Step 2***
*merging all variables and cases into DataSet1
ADD FILES /FILE=*
   /FILE='DataSet2'
   /IN=SourceDataset2.
VARIABLE LABELS SourceDataset2
   'Case source is DataSet2'.
EXECUTE.

ADD Value Labels
SourceDataset2
1 'from Dataset2'
0 'from Dataset1'.
Execute.
```

Step 3:
```
***Step 3***
*creating new mismatches variable called CasesCompared
compute CasesCompared = Mismatches.
Execute.
ADD Value Labels
CasesCompared
1 'Mismatch between Both Datasets'
0 'Match in Both Datasets'
-1 'Unique to Dataset1'.
Execute.
```

Step 4:
```
***Step 4***
*duplicating CasesCompared encoded information for
cases from Dataset2
SORT CASES BY ID(A) SourceDataset2(A) Mismatches-
(A).
Execute.
IF ((ID = lag(ID)) AND (sysmis(CasesCompared)))
CasesCompared=lag(CasesCompared).
Execute.
```

Step 5:
```
***Step 5***
*encoding in CasesCompared those cases which are unique
to Dataset2
IF (sysmis(CasesCompared))
CasesCompared=-2.
Execute.
ADD Value Labels
CasesCompared
-2 'Unique to Dataset2'.
Execute.
```