

# How can I compare two data sets in SPSS or check for consistency between data entered by two people?

Authored by  
**stats writer**

June 30, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I compare two data sets in SPSS or check for consistency between data entered by two people?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=161970>

The process of comparing two data sets in SPSS or checking for consistency between data entered by two people involves several steps. Firstly, both data sets must be imported into SPSS and organized into separate variables. Then, descriptive statistics such as mean, median, and standard deviation can be calculated for each variable in both data sets. This will provide an overall understanding of the data and identify any major discrepancies. Next, a visual comparison can be made by creating histograms or scatterplots to identify any patterns or outliers. Additionally, inferential statistics such as t-tests or ANOVA can be used to determine if there are significant differences between the two data sets. Finally, it is important to thoroughly check for data entry errors or inconsistencies by manually reviewing the data and cross-checking with the original source. This process will ensure the accuracy and consistency of the data, allowing for reliable analysis and conclusions to be drawn.

## **How can I compare two data sets in SPSS? or How do I check that the same data input by two people are consistently entered? | SPSS FAQ**

**There are times when you would like to compare two data sets to see if they are exactly the same. For example, if two people enter the same data (double data entry), you would want to know if any discrepancies exist between the two datasets (the rationale of double data entry), and if so, where those discrepancies are. We start by reading in the two datasets, one entered by person1 and the second by person2. The two data sets are identical, except**

that we created a missing value in the ninth row, second variable, in the first data set, and we changed the very last entry from 51 to 52 in the second data set.

After entering each data set, we need to sort the data set. In our example, we will sort the data set on all variables, starting with the first variable in the data set. We use the SPSS keyword `all` to do this.

We use this method because it is very general and will work in many situations.

(However, if you want to compare the files on only a few variables in the data set, you will need to list the variables in the same order in both sorts and on the `by` subcommand of the `update` command.) After sorting the data set, we save it. We do this for both data sets.

**data list list**

```
/id female race ses * schtype (A3) prog read write math
```

**science socst.**

**begin data.**

**147 1 1 3 pub 1 47 62 53 53 61**

**108 0 1 2 pub 2 34 33 41 36 36**

**18 0 3 2 pub 3 50 33 49 44 36**

**153 0 1 2 pub 3 39 31 40 39 51**

**50 0 2 2 pub 2 50 59 42 53 61**

**51 1 2 1 pub 2 42 36 42 31 39**

**102 0 1 1 pub 1 52 41 51 53 56**

**57 1 1 2 pub 1 71 65 72 66 56**

**160 . 1 2 pub 1 55 65 55 50 61**

**136 0 1 2 pub 1 65 59 70 63 51**

**end data.**

**sort cases by all.**

**save outfile "D:person1.sav".**

**data list list**

**/id female race ses \* schtype (A3) prog read write math**

**science socst.**

**begin data.**

**147 1 1 3 pub 1 47 62 53 53 61**

**108 0 1 2 pub 2 34 33 41 36 36**

**18 0 3 2 pub 3 50 33 49 44 36**

**153 0 1 2 pub 3 39 31 40 39 51**

```
50 0 2 2 pub 2 50 59 42 53 61
51 1 2 1 pub 2 42 36 42 31 39
102 0 1 1 pub 1 52 41 51 53 56
57 1 1 2 pub 1 71 65 72 66 56
160 1 1 2 pub 1 55 65 55 50 61
136 0 1 2 pub 1 65 59 70 63 52
end data.
sort cases by all.
save outfile "D:person2.sav".
```

Now we can use the update command to compare the two data files.

We need to use the SPSS keyword all on the by subcommand, because that is how we sorted the data sets. Also, we use the in subcommand to create a flag variable, which we called flag1, to indicate which rows match and which rows do not match. We use the label values command to add value labels to flag1, and finally we run a frequency on flag1.

As we can see, there are two mismatches.

**update file = "D:person1.sav"**

**/in = flag1**

**/file = "D:person2.sav"**

**/by all.**

**exe.**

**save outfile "D:combo.sav".**

**value labels flag1 0 'mismatch' 1 'match'.**

**freq var = flag1.**

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0 mismatch	2	16.7	16.7	16.7
	1 match	10	83.3	83.3	100.0
	Total	12	100.0	100.0	

**Finally, if we look at our new data set, combo, we see that we now have 12 rows of data instead of the original 10. A new row is added to the data set for each mismatched row, so that you can see where the mismatch is. If there are two mismatches in a row, the row is listed only once, so you will need**

to compare the values for each variable to find all of the mismatches.

**sch** **fla**

**id female race ses ype prog read write socst g1**

18.00 .00 3.00 2.00 pub 3.00 50.00 33.00 36.00 1  
50.00 .00 2.00 2.00 pub 2.00 50.00 59.00 61.00 1  
51.00 1.00 2.00 1.00 pub 2.00 42.00 36.00 39.00 1  
57.00 1.00 1.00 2.00 pub 1.00 71.00 65.00 56.00 1  
102.00 .00 1.00 1.00 pub 1.00 52.00 41.00 56.00 1  
108.00 .00 1.00 2.00 pub 2.00 34.00 33.00 36.00 1  
136.00 .00 1.00 2.00 pub 1.00 65.00 59.00 51.00 1  
136.00 .00 1.00 2.00 pub 1.00 65.00 59.00 52.00 0  
147.00 1.00 1.00 3.00 pub 1.00 47.00 62.00 61.00 1  
153.00 .00 1.00 2.00 pub 3.00 39.00 31.00 51.00 1  
160.00 . 1.00 2.00 pub 1.00 55.00 65.00 61.00 1  
160.00 1.00 1.00 2.00 pub 1.00 55.00 65.00 61.00 0

**Number of cases read: 12 Number of cases listed: 12**