

# How can I create a variable in SPSS that contains the number of reciprocal friends?

Authored by  
**stats writer**

June 30, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I create a variable in SPSS that contains the number of reciprocal friends?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162198>

In order to create a variable in SPSS that contains the number of reciprocal friends, you will need to follow a few simple steps. First, you will need to open your SPSS program and navigate to the "Variable View" tab. From there, you can create a new variable by clicking on the "Add Variable" button. Next, you will need to give the variable a name and specify its data type as "Numeric." Then, you can use the "Compute" function to calculate the number of reciprocal friends by using the formula:  $\text{reciprocal friends} = \text{total friends} / 2$ . Lastly, you can save the variable by clicking on the "Save" button and it will be added to your dataset. This variable will now contain the number of reciprocal friends for each individual in your data.

## How to create a variable that contains the number of reciprocal friends? | SPSS FAQ

When studying social networks, we might need to create a variable that contains the number of reciprocal friends for each person. We show a step by step example on this page using a wide data format. Click [here](#) to access the data. Here is how our data set structured:

**id friend1 friend2 friend3 friend4 friend5**

**44006 45611 55610 74158 55898 .**

**45611 44006 45623 45621 74158 55898**

**45621 71643 45611 . . .**

**45623 59642 71643 45611 73452 55610**

55610 45623 45611 44006 55898 71643  
55898 74158 55610 45621 . .  
59642 55898 71643 45621 45611 74158  
71643 55898 73452 59642 45611 44006  
73452 71643 45623 . . .  
74158 55898 45611 59642 45621 44006

Each person can nominate up to 5 friends. For example, focal person with id = 44006 has nominated 4 friends and we want to know how many of these 4 friends have also nominated 44006, that is the number of reciprocal friends initiated by 44006.

One way to accomplish this task is to first turn the data in long format so each focal person will have as many rows of data as the number of friends nominated. Then we merge back the original data matching focal id with friend.

At this point, we can simply identify by rows if the focal id and the friend is a reciprocal friend. Last, we aggregate the data back to

**get the total number of reciprocal friends.**

**Now let's go through the steps.**

**Step 1:**

**Turning the data to long format.**

**get**

**file='D:\workspsshttps://stats.idre.ucla.edu/wp-content/uploads/2016/02/friends\_wide.sav'.**

**list.**

**varstocases**

**/make friend from friend1 to friend5**

**/index = i.**

**\* list the first 15 observations to see the structure.**

**list /cases = from 1 to 15.**

**id i friend**

**44006 1 45611**

**44006 2 55610**

**44006 3 74158**

44006 4 55898  
45611 1 44006  
45611 2 45623  
45611 3 45621  
45611 4 74158  
45611 5 55898  
45621 1 71643  
45621 2 45611  
45623 1 59642  
45623 2 71643  
45623 3 45611  
45623 4 73452

Step 2.

Merging with the original data matching the variable friend in current data with the variable id in the original data. To this end, we need to rename variables and make sure that both data sets are sorted by id.

rename variables id = focal.

rename variables friend = id.

**sort cases by id(A).**  
**dataset name long.**

**get** **file**  
**= 'D:\workspsshttps://stats.idre.ucla.edu/wp-content/uploads/2016/02/friends\_wide.sav'.**

**sort cases by id(A).**  
**dataset name friend\_wide.**

**dataset activate long.**

**match files /file=\***  
**/table='friend\_wide'**  
**/by id.**  
**exe.**

**list /cases = from 1 to 15.**

**focal i id friend1 friend2 friend3 friend4 friend5**

**45611 1 44006 45611 55610 74158 55898 .**

**55610 3 44006 45611 55610 74158 55898 .**

**71643 5 44006 45611 55610 74158 55898 .**

**74158 5 44006 45611 55610 74158 55898 .**

**44006 1 45611 44006 45623 45621 74158 55898**

45621 2 45611 44006 45623 45621 74158 55898  
45623 3 45611 44006 45623 45621 74158 55898  
55610 2 45611 44006 45623 45621 74158 55898  
59642 4 45611 44006 45623 45621 74158 55898  
71643 4 45611 44006 45623 45621 74158 55898  
74158 2 45611 44006 45623 45621 74158 55898  
45611 3 45621 71643 45611 ...  
55898 3 45621 71643 45611 ...  
59642 3 45621 71643 45611 ...  
74158 4 45621 71643 45611 ...

What do we have here? Let's look at the first row. Focal person 45611 has nominated 44006 as a friend and 44006 has nominated 4 friends: 45611, 55610, 74158 and 55898. Since 45611 nominated 44006 and 44006 nominated 45611 in return, they form a reciprocal pair. So we can simply check by row if each pair of focal and id is a reciprocal friends by checking if the focal appears in the list of friends. This leads to our next step.

**Step 3.**

**Checking if focal and id are a pair of reciprocal friends.**

**To this end,**

**we use the do repeat to loop through the friend list.**

**compute rtie = 0.**

**exe.**

**do repeat f = friend1 to friend5.**

**if (focal = f) rtie = 1.**

**end repeat.**

**exe.**

**sort cases by focal(A).**

**list /cases = from 1 to 15.**

**focal i id friend1 friend2 friend3 friend4 friend5 rtie**

**44006 1 45611 44006 45623 45621 74158 55898 1.00**

**44006 2 55610 45623 45611 44006 55898 71643 1.00**

**44006 4 55898 74158 55610 45621 . . .00**

**44006 3 74158 55898 45611 59642 45621 44006 1.00**

**45611 1 44006 45611 55610 74158 55898 . 1.00**

**45611 3 45621 71643 45611 . . . 1.00**

**45611 2 45623 59642 71643 45611 73452 55610 1.00**

```

45611 5 55898 74158 55610 45621 . . .00
45611 4 74158 55898 45611 59642 45621 44006 1.00
45621 2 45611 44006 45623 45621 74158 55898 1.00
45621 1 71643 55898 73452 59642 45611 44006 .00
45623 3 45611 44006 45623 45621 74158 55898 1.00
45623 5 55610 45623 45611 44006 55898 71643 1.00
45623 1 59642 55898 71643 45621 45611 74158 .00
45623 2 71643 55898 73452 59642 45611 44006 .00

```

Step 4.

Aggregating the long data to one focal per row and merging back to the original data set.

```
dataset declare r_ties.
```

```
aggregate
```

```
/outfile='r_ties'
```

```
/break=focal
```

```
/nrties=sum(rtie).
```

```
dataset activate r_ties.
```

```
rename variables focal = id.
```

```
sort cases by id(A).
```

```
match files /file=*
```

```
/file='friend_wide'
```

```
/by id.
```

```
exe.
```

```
list /cases = from 1 to 5.
```

```
id nrties friend1 friend2 friend3 friend4 friend5
```

```
44006 3.00 45611 55610 74158 55898 .
```

```
45611 4.00 44006 45623 45621 74158 55898
```

```
45621 1.00 71643 45611 . . .
```

```
45623 3.00 59642 71643 45611 73452 55610
```

```
55610 3.00 45623 45611 44006 55898 71643
```

That is all to it. If you want to try it yourself, [click here](#) for the syntax file containing all the steps.