How can I create multiple grand-mean centered or group-mean centered variables in SPSS?

Authored by

stats writer

June 30, 2024

Creating multiple grand-mean or group-mean centered variables in SPSS involves using the Compute function to subtract the overall mean or the group mean from each individual score within the variable. This process allows for the removal of the effect of the mean on the data, making the variables more comparable and reducing the impact of outliers. To do so, first, identify the variables that need to be centered, then use the Compute function to create new variables by subtracting the mean from each individual score. This can be done for multiple variables simultaneously by selecting them in the Compute function. By creating grand-mean or group-mean centered variables, researchers can improve the accuracy and reliability of their analyses.

How can I create multiple grand-mean centered or group-mean centered variables? | SPSS FAQ

NOTE: This page was created using SPSS version 17.0.2.

Group-mean centered and grand-mean centered variables are often used in multilevel models. Creating a single centered variable is simple enough to do, but creating several group-mean centered or grand-mean centered variables at once takes a little bit of programming. In the first example below, we will create an SPSS macro called group_cvars to create a series of group-mean centered variables. In the second example, we will create a macro called grand_cvars to create a series of grand-mean centered

variables. We
need to create macros for these tasks because we will be creating multiple new variables.
We will use do-loops within the macros to assign names and values to the new
variables.

For these examples, we will use the hsb2
dataset. Specifically, we will use the continuous variables read,
write and math for both examples. We will also use the categorical variable ses for the second example. The variable
ses has three levels. We will first show the syntax to be used, and then explain the
syntax.

Creating grand-mean centered variables

In the example below, we will create a macro called grand_cvars and
use it to create grand-mean centered variables for the variables read,
write and math in our data set. When we are finished, we will

**have three new variables added to our data set, read_c, write_c
and math_c. To create a grand-mean centered variable, you simply
take the mean of the variable and subtract that mean from each value of the
variable. To create a series of grand-mean centered variables, we will
need to include two pieces of information: the list of variables to be
grand-mean centered and suffix to add to the end of the name of those variables,
which is how we will name our new variables.**

```
define grand_cvars( vlist = !charend('/')
/suffix = !cmdend )

compute one_temp = 1.
exe.

!do !vname !in (!vlist)
!let !nname = !concat(!vname, !suffix)


aggregate
/outfile=* mode=addvariables overwrite = yes
```

```
/break =one_temp
/y_temp=mean(!vname).

compute !nname = !vname - y_temp.
exe.

!doend
delete variables y_temp one_temp.
!enddefine.

grand_cvars vlist = read write math
/suffix = _c.
```

Let's look at the different parts of the syntax above.

```
define grand_cvars( vlist = !charend('/')
/suffix = !cmdend )
```

This first part of the syntax is used to create and name the macro, and to pass the arguments to the macro. The command used to create a macro is define. After issuing this command, the name of the macro is provided. In our example, this is grand_cvars. Next, we need to

**define the
order of the input of the arguments. To create grand-mean centered
variables, we need to know the list of the variables that we want to center and the suffix to the name of the new variables. (In
this macro, we will use the variable names of the variables to be grand-mean
centered plus a suffix.) Because we have two pieces of information
that we need to pass to the macro, we will need to create two arguments; we
will call these arguments vlist and suffix. We
will have the user separate the arguments with a slash (*/*); this makes clear
where each argument ends. We use the !charend keyword to tell SPSS
that the vlist argument ends when the slash is encountered. The !cmdend
keyword indicates that the suffix argument ends when the end of command
marker (i.e., the period at the end of the macro call) is encountered.**

compute one_temp = 1.
exe.

The two lines of syntax above are needed only if you are using SPSS version
16 or earlier. The compute command is used to create a variable
called one_temp that is equal to 1 (in other words, it is a constant),
and the execute command (shortened to exe.) is used to create the
variable immediately. The variable one_temp is used on the break
subcommand in the aggregate command. However, as of SPSS version
17, the break subcommand is no longer a necessary subcommand with
aggregate, so this step is unnecessary. Of course, it can be included
if you are using SPSS version 17 or later with no problem.

Now that we have created, named and assigned arguments to the macro, we can

begin the process of creating the grand-mean centered variables.

!do !vname !in (!vlist)
!let !nname = !concat(!vname, !suffix)

We will begin by creating a loop to move through the list of variables to be
grand-mean centered. We use the do command for this. (The
do-loop ends later with the doend command.) Macro commands and
macro variables must begin with a !. Both !do and !let are
macro commands. In our do-loop, we will call the indexing variable !vname,
and it loops through the variables in !vlist. On the !let command, we create the macro variable !nname that will be the
concatenation of !vname and !suffix. Notice that there are
no periods at the ends of these commands.

aggregate

/outfile=* mode=addvariables overwrite = yes
/break =one_temp
/y_temp=mean(!vname).

Our next task is to create the grand-mean centered variables, and we will use
the aggregate command to do this. We need to use some subcommands
with the aggregate command. The first is the outfile subcommand, and the star (*) is used to indicate that we do not want to create a
new data set. The mode=addvariables option is used to indicate that
we want the new variables to be added to the current data set. (If we used
the outfile = * option without the mode=addvariables option, SPSS
would replace the current data file with a file containing only the newly
created variables; by including both options, we will have the newly created
variables added to the current data set.) We use the overwrite=yes option so that the next time the macro loops through, the

variable called y_temp is overwritten with the grand mean for the next
variable. On the break
subcommand, we indicate the group variable, one_temp.
Because the variable one_temp is a constant, the data are not really
broken into groups; this is done because this subcommand is necessary if you are
using SPSS version 16 or earlier. As mentioned above, if you are using
SPSS version 17 or later, you do not need this subcommand. Finally, we create the new variable (called y_temp), which contain the
grand mean, using the mean function.

```
compute !nname = !vname - y_temp.
exe.

!doend
delete variables y_temp one_temp.
!enddefine.
```

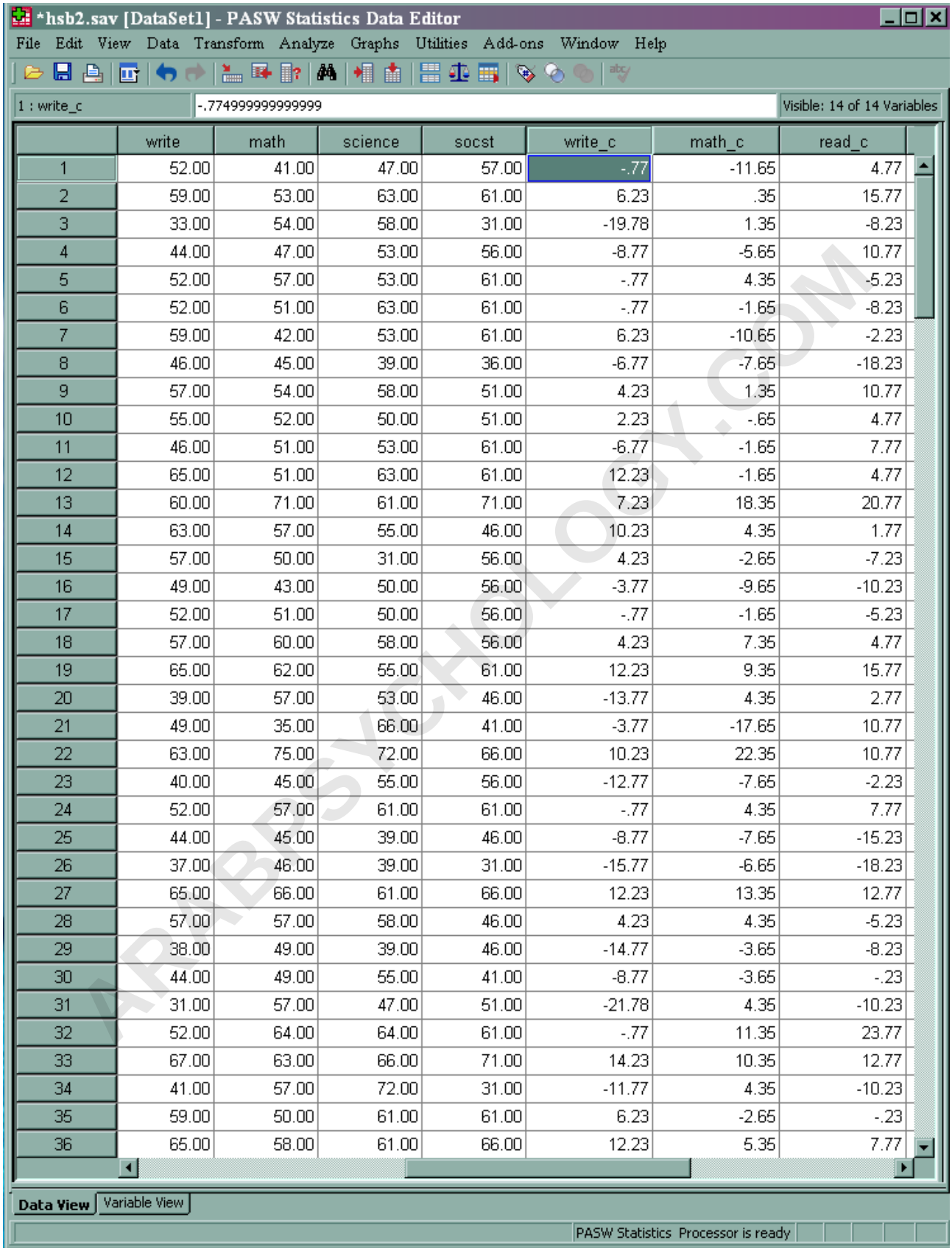
To create the grand-mean centered variable, we use the

compute
command. The macro variable !nname is the difference between the
macro variable !vname and y_temp (which was created using the
aggregate command). We include the execute command (shortened
to exe.) to have SPSS create !nname immediately, and then we end
the do-loop with !doend. Please note that there is no period at the
end of this command. We use the deleted variables command to remove the variables y_temp and one_temp from our current data set, and then we end the macro with the !enddefine command.

```
grand_cvars vlist = read write math
/suffix = _c.
```

Now we are ready to use our new macro. We start by giving the macro
name, grand_cvars. Next, we supply the list of variables to be
grand-mean centered and the suffix to include at the

end of the variable names.


*hsb2.sav [DataSet1] - PASW Statistics Data Editor

File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help

1 : write_c -.774999999999999 Visible: 14 of 14 Variables

	write	math	science	socst	write_c	math_c	read_c
1	52.00	41.00	47.00	57.00	-.77	-11.65	4.77
2	59.00	53.00	63.00	61.00	6.23	.35	15.77
3	33.00	54.00	58.00	31.00	-19.78	1.35	-8.23
4	44.00	47.00	53.00	56.00	-8.77	-5.65	10.77
5	52.00	57.00	53.00	61.00	-.77	4.35	-5.23
6	52.00	51.00	63.00	61.00	-.77	-1.65	-8.23
7	59.00	42.00	53.00	61.00	6.23	-10.65	-2.23
8	46.00	45.00	39.00	36.00	-6.77	-7.65	-18.23
9	57.00	54.00	58.00	51.00	4.23	1.35	10.77
10	55.00	52.00	50.00	51.00	2.23	-.65	4.77
11	46.00	51.00	53.00	61.00	-6.77	-1.65	7.77
12	65.00	51.00	63.00	61.00	12.23	-1.65	4.77
13	60.00	71.00	61.00	71.00	7.23	18.35	20.77
14	63.00	57.00	55.00	46.00	10.23	4.35	1.77
15	57.00	50.00	31.00	56.00	4.23	-2.65	-7.23
16	49.00	43.00	50.00	56.00	-3.77	-9.65	-10.23
17	52.00	51.00	50.00	56.00	-.77	-1.65	-5.23
18	57.00	60.00	58.00	56.00	4.23	7.35	4.77
19	65.00	62.00	55.00	61.00	12.23	9.35	15.77
20	39.00	57.00	53.00	46.00	-13.77	4.35	2.77
21	49.00	35.00	66.00	41.00	-3.77	-17.65	10.77
22	63.00	75.00	72.00	66.00	10.23	22.35	10.77
23	40.00	45.00	55.00	56.00	-12.77	-7.65	-2.23
24	52.00	57.00	61.00	61.00	-.77	4.35	7.77
25	44.00	45.00	39.00	46.00	-8.77	-7.65	-15.23
26	37.00	46.00	39.00	31.00	-15.77	-6.65	-18.23
27	65.00	66.00	61.00	66.00	12.23	13.35	12.77
28	57.00	57.00	58.00	46.00	4.23	4.35	-5.23
29	38.00	49.00	39.00	46.00	-14.77	-3.65	-8.23
30	44.00	49.00	55.00	41.00	-8.77	-3.65	-.23
31	31.00	57.00	47.00	51.00	-21.78	4.35	-10.23
32	52.00	64.00	64.00	61.00	-.77	11.35	23.77
33	67.00	63.00	66.00	71.00	14.23	10.35	12.77
34	41.00	57.00	72.00	31.00	-11.77	4.35	-10.23
35	59.00	50.00	61.00	61.00	6.23	-2.65	-.23
36	65.00	58.00	61.00	66.00	12.23	5.35	7.77

Data View Variable View

PASW Statistics Processor is ready

Creating group-mean centered variables

To create group-mean centered variables, we will again need to provide the
list of variables to be group-mean centered and the suffix. In addition,
we will also need to indicate the grouping variable.

```
define group_cvars( group = !charend('/')
/vlist = !charend('/')
/suffix = !cmdend )

!do !vname !in (!vlist)
!let !nname = !concat(!vname, !suffix)

aggregate
/outfile=* mode=addvariables overwrite = yes
/break =!group
/y_temp=mean(!vname).

compute !nname = !vname - y_temp.
exe.
!doend

delete variables y_temp.
```

```
!enddefine.

group_cvars group = ses
/vlist = read write math
/suffix = _c.
```

Let's consider the first part of the syntax above.

```
define group_cvars( group = !charend('/')
/vlist = !charend('/')
/suffix = !cmdend )
```

This first part of the syntax is used to create and name the macro, and to pass the arguments to the macro. The command used to create a macro is define. After issuing this command, the name of the macro is provided. In our example, this is group_vars. Next, we need to define the order of the input of the arguments. To create group-mean centered variables, we need to know the grouping variable, the list of the variables that

we want to center and the suffix to the name of the new variables. (In this macro, we will use the variable names of the variables to be group-mean centered plus a suffix.) Because we have three pieces of information that we need to pass to the macro, we will need to create three arguments; we will call these arguments group, vlist and suffix. We will have the user separate the arguments with a slash (/); this makes clear where each argument ends. We use the !charend keyword to tell SPSS that the group argument ends when the slash is encountered. The vlist argument ends when the next slash is encountered. The !cmdend keyword indicates that the suffix argument ends when the end of command marker (i.e., the period at the end of the macro call) is encountered.

Now that we have created, named and assigned arguments to the macro, we can

begin the process of creating the group-mean centered variables.

!do !vname !in (!vlist)
!let !nname = !concat(!vname, !suffix)

We will begin by creating a loop to move through the list of variables to be
group-mean centered. We use the do command for this. (The
do-loop ends later with the doend command.) Macro commands and
macro variables must begin with a !. Both !do and !let are
macro commands. In our do-loop, we will call the indexing variable !vname,
and it loops through the variables in !vlist. On the !let command, we create the macro variable !nname that will be the
concatenation of !vname and !suffix. Notice that there are
no periods at the ends of these commands.

aggregate

/outfile=* mode=addvariables overwrite = yes
/break =!group
/y_temp=mean(!vname).

Our next task is to create the group-mean centered variables, and we will use
the aggregate command to do this. We need to use some subcommands
with the aggregate command. The first is the outfile subcommand, and the star (*) is used to indicate that we do not want to create a
new data set. The mode=addvariables option is used to indicate that
we want the new variables to be added to the current data set. (If we used
the outfile = * option without the mode=addvariables option, SPSS
would replace the current data file with a file containing only the newly
created variables; by including both options, we will have the newly created
variables added to the current data set.) We use the overwrite=yes option so that the next time the macro loops through, the

variable called y_temp is overwritten with the group means for the next
variable. On the break
subcommand, we indicate the group variable by its macro name !group.
Finally, we create the new variable (called y_temp), which contain the
group means, using the mean function.

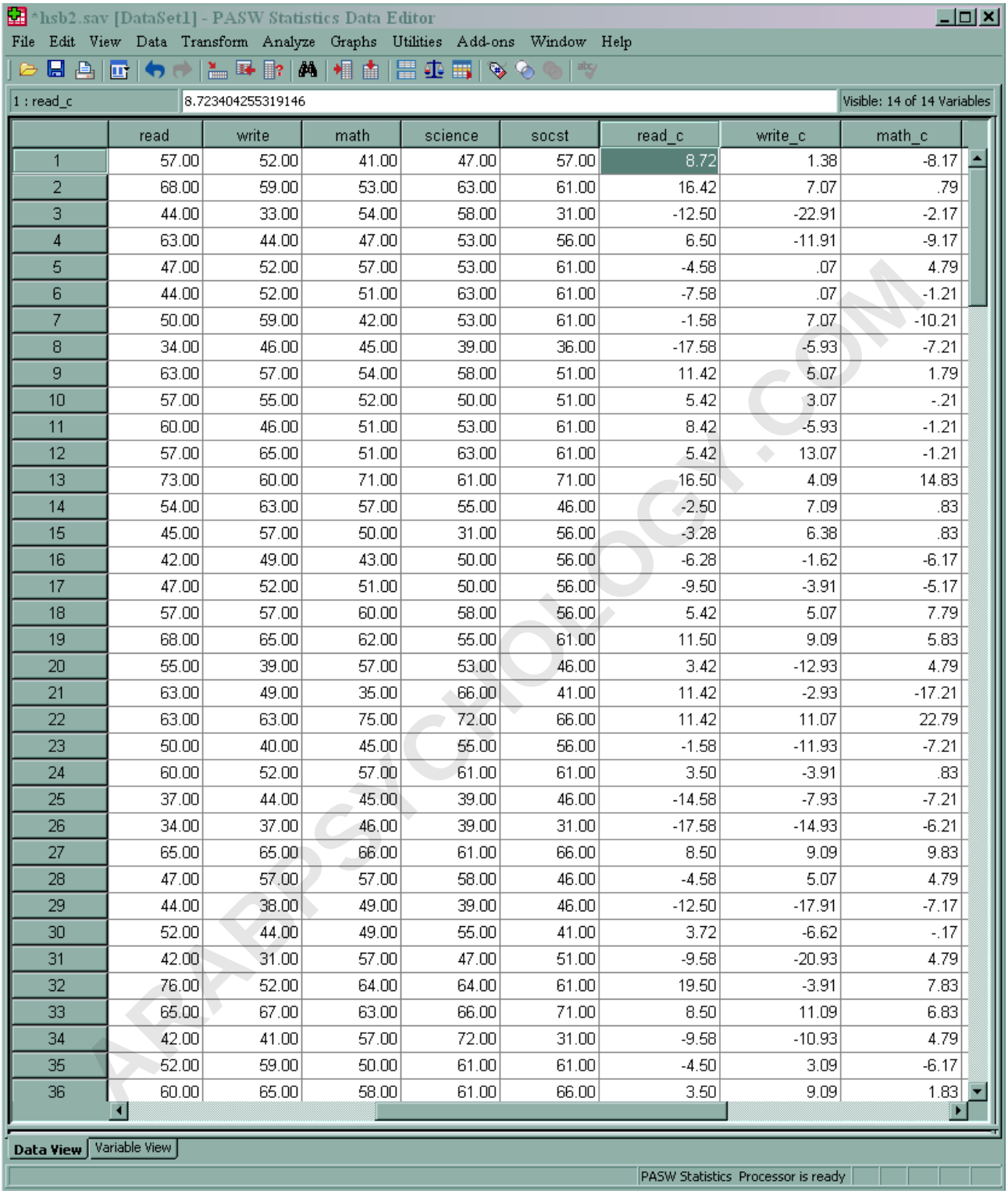
```
compute !nname = !vname - y_temp.
exe.
!doend
delete variables y_temp.
!enddefine.
```

To create the group-mean centered variable, we use the compute
command. The macro variable !nname is the difference between the
macro variable !vname and y_temp (which was created using the
aggregate command). We include the execute command (shortened
to exe.) to have SPSS create !nname immediately, and

then we end
the do-loop with !doend. Please note that there is no period at the
end of this command. We use the deleted variables command to remove the variable y_temp from our current data set, and then
we end the macro with the !enddefine command.

```
group_cvars group = ses
/vlist = read write math
/suffix = _c.
```

Now we are ready to use our new macro. We start by giving the macro
name, group_cvars. Next, we supply the three necessary arguments:
we name the group variable, the list of variables to be group-mean centered, and
the suffix to include at the end of the variable names.



	read	write	math	science	socst	read_c	write_c	math_c
1	57.00	52.00	41.00	47.00	57.00	8.72	1.38	-8.17
2	68.00	59.00	53.00	63.00	61.00	16.42	7.07	.79
3	44.00	33.00	54.00	58.00	31.00	-12.50	-22.91	-2.17
4	63.00	44.00	47.00	53.00	56.00	6.50	-11.91	-9.17
5	47.00	52.00	57.00	53.00	61.00	-4.58	.07	4.79
6	44.00	52.00	51.00	63.00	61.00	-7.58	.07	-1.21
7	50.00	59.00	42.00	53.00	61.00	-1.58	7.07	-10.21
8	34.00	46.00	45.00	39.00	36.00	-17.58	-5.93	-7.21
9	63.00	57.00	54.00	58.00	51.00	11.42	5.07	1.79
10	57.00	55.00	52.00	50.00	51.00	5.42	3.07	-.21
11	60.00	46.00	51.00	53.00	61.00	8.42	-5.93	-1.21
12	57.00	65.00	51.00	63.00	61.00	5.42	13.07	-1.21
13	73.00	60.00	71.00	61.00	71.00	16.50	4.09	14.83
14	54.00	63.00	57.00	55.00	46.00	-2.50	7.09	.83
15	45.00	57.00	50.00	31.00	56.00	-3.28	6.38	.83
16	42.00	49.00	43.00	50.00	56.00	-6.28	-1.62	-6.17
17	47.00	52.00	51.00	50.00	56.00	-9.50	-3.91	-5.17
18	57.00	57.00	60.00	58.00	56.00	5.42	5.07	7.79
19	68.00	65.00	62.00	55.00	61.00	11.50	9.09	5.83
20	55.00	39.00	57.00	53.00	46.00	3.42	-12.93	4.79
21	63.00	49.00	35.00	66.00	41.00	11.42	-2.93	-17.21
22	63.00	63.00	75.00	72.00	66.00	11.42	11.07	22.79
23	50.00	40.00	45.00	55.00	56.00	-1.58	-11.93	-7.21
24	60.00	52.00	57.00	61.00	61.00	3.50	-3.91	.83
25	37.00	44.00	45.00	39.00	46.00	-14.58	-7.93	-7.21
26	34.00	37.00	46.00	39.00	31.00	-17.58	-14.93	-6.21
27	65.00	65.00	66.00	61.00	66.00	8.50	9.09	9.83
28	47.00	57.00	57.00	58.00	46.00	-4.58	5.07	4.79
29	44.00	38.00	49.00	39.00	46.00	-12.50	-17.91	-7.17
30	52.00	44.00	49.00	55.00	41.00	3.72	-6.62	-.17
31	42.00	31.00	57.00	47.00	51.00	-9.58	-20.93	4.79
32	76.00	52.00	64.00	64.00	61.00	19.50	-3.91	7.83
33	65.00	67.00	63.00	66.00	71.00	8.50	11.09	6.83
34	42.00	41.00	57.00	72.00	31.00	-9.58	-10.93	4.79
35	52.00	59.00	50.00	61.00	61.00	-4.50	3.09	-6.17
36	60.00	65.00	58.00	61.00	66.00	3.50	9.09	1.83