SPSS FAQ

How can I compare regression coefficients between two groups?

Sometimes your research hypothesis may predict that the size of a regression coefficient should be bigger for one group than for another. For example, you might believe that the regression coefficient of **height** predicting **weight** would be higher for men than for women. Below, we have a data file with 10 fictional females and 10 fictional males, along with their **height** in inches and their **weight** in pounds.

```
data list free
 / id * gender (A8) height * weight.
begin data.
 1   F  56 117
 2   F  60 125
 3   F  64 133
 4   F  68 141
 5   F  72 149
 6   F  54 109
 7   F  62 128
 8   F  65 131
 9   F  65 131
10   F  70 145
11   M  64 211
12   M  68 223
13   M  72 235
14   M  76 247
15   M  80 259
16   M  62 201
17   M  69 228
18   M  74 245
19   M  75 241
20   M  82 269
end data.
execute.
```

We analyzed their data separately using the **regression** commands below. Note that we have to do two regressions, one with the data for females only and one with the data for males only. We can use the **split file** command to split the data file by **gender** and then run the regression. The parameter estimates (coefficients) for females and males are shown below, and the results do seem to suggest that **height** is a stronger predictor of **weight** for males (3.18) than for females (2.09).

```
sort cases by gender.
split file by gender.
regression
 /dep weight
 /method = enter height.
split file off.
```

Variables Entered/Removed[b]

gender	Model	Variables Entered	Variables Removed	Method
F	1	height[a]	.	Enter
M	1	height[a]	.	Enter

a. All requested variables entered.

b. Dependent Variable: weight

Model Summary

gender	Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
F	1	.989[a]	.978	.976	1.91504
M	1	.994[a]	.988	.987	2.40738

a. Predictors: (Constant), height

ANOVA[b]

gender	Model		Sum of Squares	df	Mean Square	F	Sig.
F	1	Regression	1319.561	1	1319.561	359.812	.000[a]
		Residual	29.339	8	3.667		
		Total	1348.900	9			
M	1	Regression	3882.536	1	3882.536	669.926	.000[a]
		Residual	46.364	8	5.795		
		Total	3928.900	9			

a. Predictors: (Constant), height

b. Dependent Variable: weight

Coefficients[a]

			Unstandardized Coefficients		Standardized Coefficients		
gender	Model		B	Std. Error	Beta	t	Sig.
F	1	(Constant)	-2.397	7.053		-.340	.743
		height	2.096	.110	.989	18.969	.000
M	1	(Constant)	5.602	8.930		.627	.548
		height	3.190	.123	.994	25.883	.000

a. Dependent Variable: weight

We can compare the regression coefficients of males with females to test the null hypothesis Ho: $B_f = B_m$, where B_f is the regression coefficient for females, and B_m is the regression coefficient for males. To do this analysis, we first make a dummy variable called **female** that is coded 1 for female and 0 for male, and a variable **femht** that is the product of **female** and **height**. We then use **female**, **height** and **femht** as predictors in the regression equation.

```
split file off.

compute female = 0.
if gender = "F" female = 1.
compute femht = female*height.
execute.

regression
 /dep weight
 /method = enter female height femht.
```

The output is shown below.

Variables Entered/Removed[b]

Model	Variables Entered	Variables Removed	Method
1	femht, height, female[a]	.	Enter

a. All requested variables entered.

b. Dependent Variable: weight

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.999[a]	.999	.999	2.17518

a. Predictors: (Constant), femht, height, female

ANOVA[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	60327.097	3	20109.032	4250.111	.000[a]
	Residual	75.703	16	4.731		
	Total	60402.800	19			

a. Predictors: (Constant), femht, height, female

b. Dependent Variable: weight

Coefficients[a]

		Unstandardized Coefficients		Standardized Coefficients		
Model		B	Std. Error	Beta	t	Sig.
1	(Constant)	5.602	8.069		.694	.497
	female	-7.999	11.371	-.073	-.703	.492
	height	3.190	.111	.421	28.646	.000
	femht	-1.094	.168	-.638	-6.520	.000

a. Dependent Variable: weight

The term **femht** tests the null hypothesis Ho: $B_f = B_m$. The T value is -6.52 and is significant, indicating that the regression coefficient B_f is significantly different from B_m.

Let's look at the parameter estimates to get a better understanding of what they mean and how they are interpreted.
First, recall that our dummy variable **female** is 1 if female and 0 if male; therefore, males are the omitted group. This is needed for proper interpretation of the estimates.

```
          Parameter
Variable  Estimate
INTERCEP  5.601677 : This is the intercept for the males (omitted group)
                     This corresponds to the intercept for males in
                     the separate groups analysis.
FEMALE   -7.999147 : Intercept Females - Intercept males
                     This corresponds to differences of the
                     intercepts from the separate groups analysis.
                     and is indeed -2.397470040 - 5.601677149
HEIGHT    3.189727 : Slope for males (omitted group), i.e. Bm.
FEMHT    -1.093855 : Slope for females - Slope for males
                     (i.e. Bf - Bm).
                     From the separate groups, this is indeed
                     2.095872170 - 3.189727463 .
```

It is also possible to run such an analysis using **glm**, using syntax like that below. Note that other statistical packages, such as SAS and Stata, omit the group of the dummy variable that is coded as zero. However, SPSS omits the group coded as one. Therefore, when you compare the output from the different packages, the results seem to be different. To make the SPSS results match those from other packages, you need to create a new variable that has the opposite coding (i.e., switching the zeros and ones). We do this with the **male** variable. We do not know of an option in SPSS **glm** to easily change which group is the omitted group. (Please note that you can use the **contrast** subcommand to get the contrast coefficient for **female** using 0 as the reference group; however, the coding of **female** in the interaction is such that 1 is used as the reference group, so the use of the **contrast** subcommand is not very helpful in this situation.)

```
compute male = not female.

glm weight by male with height
 /design = male height male by height
 /print = parameter.
```

Between-Subjects Factors

		N
male	.00	10
	1.00	10

Tests of Between-Subjects Effects

Dependent Variable: weight

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	60327.097[a]	3	20109.032	4250.111	.000
Intercept	.376	1	.376	.079	.782
male	2.342	1	2.342	.495	.492
height	4695.831	1	4695.831	992.480	.000
male * height	201.115	1	201.115	42.506	.000
Error	75.703	16	4.731		
Total	733114.000	20			
Corrected Total	60402.800	19			

a. R Squared = .999 (Adjusted R Squared = .999)

Parameter Estimates

Dependent Variable: weight

					95% Confidence Interval	
Parameter	B	Std. Error	t	Sig.	Lower Bound	Upper Bound
Intercept	5.602	8.069	.694	.497	-11.504	22.707
[male=.00]	-7.999	11.371	-.703	.492	-32.104	16.105
[male=1.00]	0[a]	.	.	.	.	.
height	3.190	.111	28.646	.000	2.954	3.426
[male=.00] * height	-1.094	.168	-6.520	.000	-1.450	-.738
[male=1.00] * height	0[a]	.	.	.	.	.

a. This parameter is set to zero because it is redundant.

SPSS FAQ

How can I compare regression coefficients across three (or more) groups?

Sometimes your research hypothesis may predict that the size of a regression coefficient may vary across groups. For example, you might believe that the regression coefficient of **height** predicting **weight** would differ across three age groups (young, middle age, senior citizen). Below, we have a data file with 10 fictional young people, 10 fictional middle age people, and 10 fictional senior citizens, along with their **height** in inches and their **weight** in pounds. The variable **age** indicates the age group and is coded 1 for young people, 2 for middle aged, and 3 for senior citizens. Below we show two ways that you can get this data file into SPSS. One way is to cut and paste the following code into an SPSS syntax window and run it.

```
data list list / id age height weight.
begin data.
1  1    56     140
2  1    60     155
3  1    64     143
4  1    68     161
5  1    72     139
6  1    54     159
7  1    62     138
8  1    65     121
9  1    65     161
10  1    70     145
11  2    56     117
12  2    60     125
13  2    64     133
14  2    68     141
15  2    72     149
16  2    54     109
17  2    62     128
18  2    65     131
19  2    65     131
20  2    70     145
21  3    64     211
22  3    68     223
23  3    72     235
24  3    76     247
25  3    80     259
26  3    62     201
27  3    69     228
28  3    74     245
29  3    75     241
30  3    82     269
end data.
execute.
```

Another way is to click on compreg3.sav and then use the **get file** command (insert the proper drive letter if you did not place the file in your current directory):

```
get file 'c:\compreg3.sav'.
```

After first sorting by age, we analyze the data for each age group separately using the **regression** command. In order to use just the data for a specific age group, we need to use a filter to "filter out" the other data. Remember that when you have completed the analysis, you need to turn the filter off.

```
sort cases by age.
split file by age.
regression
 /dep weight
 /method=enter height.
split file off.
exe.
```

The parameter estimates (coefficients) for the young, middle age, and senior citizens are shown below, and the results do seem to suggest that **height** is a stronger predictor of **weight** for seniors (3.18) than for the middle aged (2.09). The results also seem to suggest that **height** does not predict **weight** as strongly for the young (-.37) as for the middle aged and seniors. However, we would need to perform specific significance tests to be able to make claims about the differences among these regression coefficients.

```
< some output omitted to save space >
```

Model Summary

age	Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1.00	1	.170[a]	.029	-.093	13.42080
2.00	1	.989[a]	.978	.976	1.91504
3.00	1	.994[a]	.988	.987	2.40738

a. Predictors: (Constant), height

ANOVA[b]

age	Model		Sum of Squares	df	Mean Square	F	Sig.
1.00	1	Regression	42.657	1	42.657	.237	.640[a]
		Residual	1440.943	8	180.118		
		Total	1483.600	9			
2.00	1	Regression	1319.561	1	1319.561	359.812	.000[a]
		Residual	29.339	8	3.667		
		Total	1348.900	9			
3.00	1	Regression	3882.536	1	3882.536	669.926	.000[a]
		Residual	46.364	8	5.795		
		Total	3928.900	9			

a. Predictors: (Constant), height

b. Dependent Variable: weight

Coefficients[a]

age	Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
			B	Std. Error	Beta		
1.00	1	(Constant)	170.166	49.430		3.443	.009
		height	-.377	.774	-.170	-.487	.640
2.00	1	(Constant)	-2.397	7.053		-.340	.743
		height	2.096	.110	.989	18.969	.000
3.00	1	(Constant)	5.602	8.930		.627	.548
		height	3.190	.123	.994	25.883	.000

a. Dependent Variable: weight

We can compare the regression coefficients among these three age groups to test the null hypothesis

Ho: $B_1 = B_2 = B_3$

where B_1 is the regression for the young, B_2 is the regression for the middle aged, and B_3 is the regression for senior citizens. To do this analysis, we first make a dummy variable called **age1** that is coded 1 if young (age=1), 0 otherwise, and **age2** that is coded 1 if middle aged (age=2), 0 otherwise. We also create **age1ht** that is **age1** times **height**, and **age2ht** that is **age2** times **height**.

```
compute age1 = 0.
compute age2 = 0.
if age = 1 age1 = 1.
if age = 2 age2 = 1.
compute age1ht = age1*height.
compute age2ht = age2*height.
execute.
```

We can now use **age1 age2 height**, **age1ht** and **age2ht** as predictors in the regression equation in the **regress** command below. The **regression** command will be followed by

```
/method = test(age1 age2)
```

and

```
/method = test(age1ht age2ht)
```

The first one provides a 2 degree of freedom to determine if, taken together, the variable **age** is statistically significant. We have included this for the sake of completeness, because this is a standard part of the analysis. The second subcommand tests the null hypothesis

Ho: $B_1 = B_2 = B_3$

This test will also have 2 degrees of freedom because it compares among three regression coefficients.

```
regression
 /dep weight
 /method = enter height
 /method=test(age1 age2)
 /method = test(age1ht age2ht).
```

```
< some output omitted to save space >
```

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.714[a]	.510	.492	35.27749
2	.974[b]	.948	.942	11.93281
3	.989[c]	.979	.974	7.94944

a. Predictors: (Constant), height

b. Predictors: (Constant), height, age2, age1

c. Predictors: (Constant), height, age2, age1, age2ht, age1ht

ANOVA[e]

Model			Sum of Squares	df	Mean Square	F	Sig.	R Square Change
1	Regression		36265.967	1	36265.967	29.141	.000[a]	
	Residual		34846.033	28	1244.501			
	Total		71112.000	29				
2	Subset Tests	age1, age2	31143.844	2	15571.922	109.360	.000[b]	.438
	Regression		67409.811	3	22469.937	157.803	.000[c]	
	Residual		3702.189	26	142.392			
	Total		71112.000	29				
3	Subset Tests	age1ht, age2ht	2185.544	2	1092.772	17.292	.000[b]	.031
	Regression		69595.355	5	13919.071	220.261	.000[d]	
	Residual		1516.645	24	63.194			
	Total		71112.000	29				

a. Predictors: (Constant), height

b. Tested against the full model.

c. Predictors in the Full Model: (Constant), height, age2, age1.

d. Predictors in the Full Model: (Constant), height, age2, age1, age2ht, age1ht.

e. Dependent Variable: weight

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-158.525	61.382		-2.583	.015
	height	4.958	.918	.714	5.398	.000
2	(Constant)	108.492	27.745		3.910	.001
	height	1.765	.381	.254	4.635	.000
	age1	-74.524	6.261	-.722	-11.903	.000
	age2	-89.824	6.261	-.870	-14.347	.000
3	(Constant)	5.602	29.489		.190	.851
	height	3.190	.407	.459	7.838	.000
	age1	164.565	41.555	1.593	3.960	.001
	age2	-7.999	41.555	-.077	-.192	.849
	age1ht	-3.567	.613	-2.208	-5.817	.000
	age2ht	-1.094	.613	-.677	-1.784	.087

a. Dependent Variable: weight

The analysis below shows that the null hypothesis

Ho: $B_1 = B_2 = B_3$

SPSS FAQ

How do I interpret the parameter estimates for dummy variables in regression or glm?

Consider this simple data file that has nine subjects (**sub**) in three groups (**iv**) with a score on the outcome or dependent variable (**dv**).

```
data list list / sub iv dv.
begin data
1 1 48
2 1 49
3 1 50
4 2 17
5 2 20
6 2 23
7 3 28
8 3 30
9 3 32
end data.
```

Below we use the **means** command to find the overall mean and the means for the three groups.

```
means tables = dv by iv.
```

As we see below, the overall mean is 33, and the means for groups 1, 2 and 3 are 49, 20 and 30 respectively.

Case Processing Summary

	Cases					
	Included		Excluded		Total	
	N	Percent	N	Percent	N	Percent
DV * IV	9	100.0%	0	.0%	9	100.0%

Report
DV

IV	Mean	N	Std. Deviation
1.00	49.0000	3	1.00000
2.00	20.0000	3	3.00000
3.00	30.0000	3	2.00000
Total	33.0000	9	12.89380

Let's run a standard ANOVA on these data using **glm**.

```
glm dv by iv.
```

The results of the ANOVA are shown below.

Between-Subjects Factors

		N
IV	1.00	3
	2.00	3
	3.00	3

Tests of Between-Subjects Effects
Dependent Variable: DV

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	1302.000(a)	2	651.000	139.500	.000
Intercept	9801.000	1	9801.000	2100.214	.000
IV	1302.000	2	651.000	139.500	.000
Error	28.000	6	4.667		
Total	11131.000	9			
Corrected Total	1330.000	8			
a R Squared = .979 (Adjusted R Squared = .972)					

Now, let's take this information we have found and relate it to the results that we get when we run a similar analysis using dummy coding. Let's make a data file called **dummy2** that has dummy variables called **iv1** (1 if iv=1), **iv2** (1 if iv=2) and **iv3** (1 if iv=3). Note that **iv3** is not really necessary, but it could be useful for further exploring the meaning of dummy variables. We will then use the **regression** command to predict **dv** from **iv1** and **iv2**.

```
compute iv1 = 0.
if iv = 1 iv1 = 1.
compute iv2 = 0.
if iv = 2 iv2 = 1.
compute iv3 = 0.
if iv = 3 iv3 = 1.
execute.

regression
 /dependent = dv
 /method = enter iv1 iv2.
```

The output is shown below.

Variables Entered/Removed(b)

Model	Variables Entered	Variables Removed	Method
1	IV2, IV1(a)	.	Enter
a All requested variables entered.			
b Dependent Variable: DV			

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.989(a)	.979	.972	2.16025
a Predictors: (Constant), IV2, IV1				

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1302.000	2	651.000	139.500	.000(a)
	Residual	28.000	6	4.667		
	Total	1330.000	8			
a Predictors: (Constant), IV2, IV1						
b Dependent Variable: DV						

Coefficients(a)

		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
Model		B	Std. Error	Beta		
1	(Constant)	30.000	1.247		24.054	.000
	IV1	19.000	1.764	.737	10.772	.000
	IV2	-10.000	1.764	-.388	-5.669	.001
a Dependent Variable: DV						

First, note that from the ANOVA using the **glm** command that the F-value was 139.5 and for the regression using the **regression** command the F-value (for the model) is also 139.5. This illustrates that the overall test of the model using regression is really the same as doing an ANOVA.

After the **ANOVA** table, there is a table entitled **Coefficients**. What is the interpretation of the values listed there, the 30, 19 and -10? Notice how we have **iv1** and **iv2** that refer to group 1 and group 2, but we did not include any dummy variable referring to group 3. Group 3 is often called the **omitted group** or **reference group**. Recall that the means of the 3 groups were 49, 20 and 30 respectively. The **intercept** term is the mean of the dependent variable, which we called **dv**, for the **omitted group**, and indeed the parameter estimate (in the column **B**) from the output is the mean of group 3, 30. The parameter estimate for **iv1** is the mean of the dependent variable, **dv**, for group 1 minus the mean of the dependent variable for group 3, 49 - 30 = 19, and indeed that is the parameter estimate for **iv1**. Likewise, the parameter estimate for **iv2** is the mean of the dependent variable for group 2 minus the mean of the dependent variable for group 3, 20 - 30 = -10, the parameter estimate for **iv2**.

So, in summary:

Intercept	mean of group 3 (mean of omitted group)
iv1	mean of group 1 - group 3 (omitted group)
iv2	mean of group 2 - group 3 (omitted group)

Try running this example, but use **iv2** and **iv3** using **regression** (making group 1 the omitted group) and see what happens.

Finally, consider how the parameter estimates can be used in the regression model to obtain the means for the groups (the predicted values).

The regression model is

```
Ypredicted = 30 + iv1*19 + iv2*-10

For group 1: Ypredicted = 30 + 1 * 19 + 0 * -10 = 49
For group 2: Ypredicted = 30 + 0 * 19 + 1 * -10 = 20
For group 3: Ypredicted = 30 + 0 * 19 + 0 * -10 = 30
```

As you see, the regression formula predicts that each group will have the mean value of its group.

You can also perform the same analysis using **glm**. The **print = parameter** subcommand tells SPSS to print the regression coefficients.

```
glm dv with iv1 iv2
  /print = parameter.
```

Tests of Between-Subjects Effects
Dependent Variable: DV

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	1302.000(a)	2	651.000	139.500	.000
Intercept	2700.000	1	2700.000	578.571	.000
IV1	541.500	1	541.500	116.036	.000
IV2	150.000	1	150.000	32.143	.001
Error	28.000	6	4.667		
Total	11131.000	9			
Corrected Total	1330.000	8			
a R Squared = .979 (Adjusted R Squared = .972)					

Parameter Estimates
Dependent Variable: DV

Parameter	B	Std. Error	t	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Intercept	30.000	1.247	24.054	.000	26.948	33.052
IV1	19.000	1.764	10.772	.000	14.684	23.316
IV2	-10.000	1.764	-5.669	.001	-14.316	-5.684

SPSS FAQ

How can I perform hypothesis tests in glm?

Sometimes you may want to test hypotheses about the parameters after a linear regression analysis. On this page, we show a couple of examples of how to perform these hypothesis tests using the **lmatrix** and **kmatrix** subcommands in the **glm** procedure. These examples will use data set hsb2.sav. Let's say that we have run a linear regression model as follows.

```
glm write with female read math
/print=parameter
/design=female read math.
```

Parameter Estimates

Dependent Variable:writing score

Parameter	B	Std. Error	t	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Intercept	11.896	2.863	4.155	.000	6.250	17.542
female	5.443	.935	5.822	.000	3.599	7.287
read	.325	.061	5.355	.000	.205	.445
math	.397	.066	5.986	.000	.267	.528

Written as a regression equation, we have the following:

write = b_0 + b_1* **female** + b_2***read** + b_3***math**, where b_0 = 11.896, b_1 = 5.443, b_2 = .325 and b_3 = .397.

Example 1

Let's say that we want to test if the coefficient for **read** is equal to the coefficient for **math**. The **lmatrix** subcommand allows us to specify our hypothesis test in terms of the linear combination of the regression coefficients. In our case, our null hypothesis is that b_2 = b_3, or equivalently, b_2-b_3 = 0. This leads to our **lmatrix** subcommand with 1 following the variable **read** and -1 following the variable **math**.

```
glm write with female read math
/print=parameter
/design=female read math
/lmatrix = 'math = read' read 1 math -1.
```

Custom Hypothesis Tests

Contrast Results (K Matrix)[a]

Contrast			Dependen... writing score
L1	Contrast Estimate		-.072
	Hypothesized Value		0
	Difference (Estimate - Hypothesized)		-.072
	Std. Error		.116
	Sig.		.534
	95% Confidence Interval for Difference	Lower Bound	-.301
		Upper Bound	.156

a. Based on the user-specified contrast coefficients (L') matrix: math = read

Test Results

Dependent Variable:writing score

Source	Sum of Squares	df	Mean Square	F	Sig.
Contrast	16.792	1	16.792	.388	.534
Error	8473.526	196	43.232		

In the output, we see the difference between the two parameters is -.072 = (.325 - .397), as we expected. What the output also gives is the standard error for the difference and the confidence interval. The Test Results table shows the F-value and the p-value.

Example 2

Let's say that we want to test if the coefficient for **female** is equal to 4.2. In order to do this, we need to use the **kmatrix** subcommand, because we are testing if the value is something other than 0. You might want to do this, if, for example, you had regression coefficients from a previous model and you wanted to see if they were equal to the coefficients obtained with your current model. To keep the example simple, we will test only one variable (**female**) in this example.

```
glm write with female read math
/print=parameter
/design=female read math
/lmatrix = 'female' female 1
/kmatrix  4.2.
```

Contrast Results (K Matrix)[a]

Contrast			Dependen... writing score
L1	Contrast Estimate		5.443
	Hypothesized Value		4.200
	Difference (Estimate - Hypothesized)		1.243
	Std. Error		.935
	Sig.		.185
	95% Confidence Interval for Difference	Lower Bound	-.601
		Upper Bound	3.087

a. Based on the user-specified contrast coefficients (L') matrix: female

Test Results

Dependent Variable:writing score

Source	Sum of Squares	df	Mean Square	F	Sig.
Contrast	76.452	1	76.452	1.768	.185
Error	8473.526	196	43.232		

Example 3

Let's say that we want to test if the coefficient for **female** is equal to 4.2 and that the coefficient for **read** is equal to the coefficient for **math**. This will be a two degree-of-freedom test since there are two hypotheses that we want to test simultaneously. Notice that the values specified on the **kmatrix** subcommand are listed in the same order as the tests listed on the **lmatrix** subcommand.

```
glm write with female read math
/print=parameter
/design=female read math
/lmatrix = 'test' female 1; read 1 math -1
/kmatrix  4.2; 0.
```

Contrast Results (K Matrix)[a]

Contrast			Dependen... writing score
L1	Contrast Estimate		5.443
	Hypothesized Value		4.200
	Difference (Estimate - Hypothesized)		1.243
	Std. Error		.935
	Sig.		.185
	95% Confidence Interval for Difference	Lower Bound	-.601
		Upper Bound	3.087
L2	Contrast Estimate		-.072
	Hypothesized Value		.000
	Difference (Estimate - Hypothesized)		-.072
	Std. Error		.116
	Sig.		.534
	95% Confidence Interval for Difference	Lower Bound	-.301
		Upper Bound	.156

a. Based on the user-specified contrast coefficients (L') matrix: test

Test Results

Dependent Variable:writing score

Source	Sum of Squares	df	Mean Square	F	Sig.
Contrast	95.324	2	47.662	1.102	.334
Error	8473.526	196	43.232		

SPSS FAQ

How can I perform a Sobel test on a single mediation effect in SPSS?

This page uses code provided by William Dudley PhD and Jose Benuzillo MA, University of Utah College of Nursing, 2/11/04. It estimates the percentage of the total effect that is mediated and the ratio of the indirect to the direct effect by using the Sobel test. It is for the simplest model in which there is only one mediator, one independent variable and one outcome variable, all of which are continuous variables.

We thank the authors of this program for very kindly sharing it with us and for allowing us to post it on our site. You can download a word document explaining their program here.

All of the code on this page can be found in this SPSS syntax file. We use this example SPSS data file for running their program.

To begin, we indicate which of our variables are the dependent, independent, and mediator variables. We then run three regression models representing the paths of interest for the Sobel test. From each regression, we create and save a dataset. We will only show the output generated by the **Report** sections of code.

NOTE: When using this code, you can use the first three lines to indicate which of your variables play which role in the model and then run the rest of the code without making any more changes. We have included all of the code in this page to illustrate the steps in the program. But you can certainly just download the syntax, change the first three lines, and be ready to run all the analysis code!

```
compute indepvar        = Pain .    /*your IV*/
compute depenvar        = Depress.  /*your DV*/
compute mediator        = Function. /*you MEDIATOR*/
exe.

REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT mediator
  /METHOD=ENTER indepvar
  /OUTFILE=COVB('reg1.sav').

REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT depenvar
  /METHOD=ENTER indepvar mediator
  /OUTFILE=COVB('reg2.sav').

REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10)
  /NOORIGIN
  /DEPENDENT depenvar
  /METHOD=ENTER  indepvar
  /OUTFILE=COVB('reg3.sav').
```

Next, these three regression datasets are combined after indicating which is which. A subset of observations are kept corresponding to estimates, standard errors, degrees of freedom, and p-values

```
get file='reg1.sav'.
compute reg=1.
exe.
SAVE OUTFILE='reg1.sav'
  /COMPRESSED.

get file='reg2.sav'.
compute reg=2.
exe.
SAVE OUTFILE='reg2.sav'
  /COMPRESSED.

get file='reg3.sav'.
compute reg=3.
exe.
SAVE OUTFILE='reg3.sav'
  /COMPRESSED.

add files file=='reg1.sav'
         /file=='reg2.sav'
         /file=='reg3.sav'
/keep=  reg DEPVAR_ ROWTYPE_ VARNAME_  CONST_ indepvar mediator.
exe.

sel if rowtype_ = 'EST' or rowtype_ = 'SE' or rowtype_ = 'DFE'
                 or rowtype_ = 'SIG'.
exe.
```

Next, the data is reshaped from wide to long.

```
numeric a.
numeric sa.
numeric siga.
numeric b.
numeric sb.
numeric sigb.
numeric df.
numeric t.
numeric st.
numeric sig.
        if reg = 1 and rowtype_ = 'EST' a              = indepvar.
        if reg = 1 and rowtype_ = 'SE'  sa             = indepvar.
        if reg = 1 and rowtype_ = 'SIG' siga    = indepvar.
        if reg = 2 and rowtype_ = 'EST' b              = mediator.
        if reg = 2 and rowtype_ = 'SE'  sb             = mediator.
        if reg = 2 and rowtype_ = 'SIG' sigb    = mediator.
        if reg = 2 and rowtype_ = 'DFE' df             = mediator.
        if reg = 2 and rowtype_ = 'EST' tprime  = indepvar.
        if reg = 3 and rowtype_ = 'EST' t              = indepvar.
        if reg = 3 and rowtype_ = 'SE'  st             = indepvar.
        if reg = 3 and rowtype_ = 'SIG' sig            = indepvar.
exe.

compute regx=1.
exe.


aggregate outfile 'sobel.sav'
 /break regx
 /a = max(a)
 /sa=max(sa)
 /siga=max(siga)
 /b=max(b)
 /sb=max(sb)
 /sigb=max(sigb)
 /df=max(df)
 /sig=max(sig)
 /tprime = max(tprime)
 /t=max(t)
 /st=max(st).
get file 'sobel.sav'.
format a b tprime t (F8.4).

compute ab = a*b.
compute ttprime = t-tprime.

format ab ttprime  (F8.4).

exe.
```

From these values, our statistics of interest are calculated, formatted, and labeled.

```
compute sobel= ttprime/(sqrt (( (b*b)*(sa*sa) ) + ( (a*a)*(sb*sb) ))).
compute absobel= abs(sobel).
compute p_val=2*(1-cdfnorm(absobel)).
compute t1=(t-(a*b)).
compute toteff=(a*b/((a*b)+t1)).
compute ratio=((a*b)/t1).
compute toteff = 100* toteff.
compute goodman = ttprime/sqrt(((b*b)*(sa*sa))+((a*a)*(sb*sb))+((sa*sa)*(sb*sb))).
compute absgood = abs(goodman).
compute goodman2 = ttprime/sqrt(((b*b)*(sa*sa))+((a*a)*(sb*sb))-((sa*sa)*(sb*sb))).
compute absgood2 = abs(goodman2).
compute p_val2 = 2*(1-cdfnorm(absgood)).
compute p_val3 = 2*(1-cdfnorm(absgood2)).
exe.


format  p_val p_val2 p_val3
           sig siga sigb
           sobel goodman goodman2
        toteff ratio st sb sa(F8.6).
exe.

variable label
sig    'P value of c'
siga   'P value of a'
sigb   'P value of b'
a      'Reg coeff for the association between IV and MEDIATOR'
sa     'Standard error of a'
b      'Reg coeff for the association between the MEDIATOR and IV on DV'
sb     'Standard error of b'
df     'Degrees of freedom'
t      'Reg coeff for the association between IV and DV'
st     'Standard error of c'
sobel  'Sobel'
p_val  'P value'
goodman 'Goodman test'
p_val2  'P value'
goodman2 'GoodmanII test'
p_val3 'P Value'
toteff 'Percentage of the total effect that is mediated'
ratio  'Ratio of the indirect to the direct effect'.
exe.
```

Results are presented with **Report**.

```
Report
  /FORMAT= CHWRAP(ON) PREVIEW(OFF) CHALIGN(BOTTOM)  UNDERSCORE(ON)
  ONEBREAKCOLUMN(OFF) CHDSPACE(1)  SUMSPACE(0)  AUTOMATIC  NOLIST
   BRKSPACE(0)
  PAGE(1) MISSING'.' LENGTH(1, 59) ALIGN(LEFT) TSPACE(1) FTSPACE(1)
 MARGINS(1,110)
 /TITLE=
  LEFT 'Regression Analysis Results'
  RIGHT 'Page )PAGE'
   /VARIABLES
 t      (VALUES)  (RIGHT)  (OFFSET(0)) (9)
 st     (VALUES)  (RIGHT)  (OFFSET(0)) (7)
 sig    (VALUES)  (RIGHT)  (OFFSET(0)) (7)
 a      (VALUES)  (RIGHT)  (OFFSET(0)) (9)
 sa     (VALUES)  (RIGHT)  (OFFSET(0)) (7)
 siga   (VALUES)  (RIGHT)  (OFFSET(0)) (7)
 b      (VALUES)  (RIGHT)  (OFFSET(0)) (9)
 sb     (VALUES)  (RIGHT)  (OFFSET(0)) (7)
 sigb   (VALUES)  (RIGHT)  (OFFSET(0)) (7)
 /BREAK (TOTAL)  (SKIP(1)) /SUMMARY
 SUM( t) SKIP(1) SUM( st ) SUM( sig ) SUM( a ) SUM( sa ) SUM( siga ) SUM( b ) SUM( sb ) SUM( sigb )   '' .
```

Regression Analysis Results — Page 1

Reg coeff for the association between IV and DV	Standard error of c	P value of c	Reg coeff for the association between IV and MEDIATOR	Standard error of a	P value of a	Reg coeff for the association between the MEDIATOR and IV on DV	Standard error of b	P value of b
.1117	.025721	.000026	-.0973	.015686	.000000	-.5227	.128626	.000079

```
Report
  /FORMAT= CHWRAP(ON) PREVIEW(OFF) CHALIGN(BOTTOM)  UNDERSCORE(ON)
  ONEBREAKCOLUMN(OFF) CHDSPACE(1)  SUMSPACE(0)  AUTOMATIC  NOLIST
   BRKSPACE(0)
  PAGE(2) MISSING'.' LENGTH(1, 59) ALIGN(LEFT) TSPACE(1) FTSPACE(1)
 MARGINS(1,100)
 /TITLE=
  LEFT 'Mediation Analysis Results'
  RIGHT 'Page )PAGE'
 /VARIABLES
 sobel  (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 p_val  (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 toteff (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 ratio  (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 goodman        (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 p_val2 (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 goodman2       (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 p_val3 (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 /BREAK (TOTAL) (SKIP(1)) /SUMMARY
 SUM( sobel) SKIP(1) SUM( p_val ) SUM( toteff ) SUM( ratio) SUM( goodman ) SUM( p_val2 ) SUM( goodman2 ) SUM( p_val3 ) ''
```

Mediation Analysis Results — Page 2

Sobel	P value	Percentage of the total effect that is mediated	Ratio of the indirect to the direct effect	Goodman test	P value	GoodmanII test	P Value
3.399121	.000676	45.524883	.835701	3.368622	.000755	3.430464	.000603

```
Report
  /FORMAT= CHWRAP(ON) PREVIEW(OFF) CHALIGN(BOTTOM)  UNDERSCORE(ON)
  ONEBREAKCOLUMN(OFF) CHDSPACE(1)  SUMSPACE(0)  AUTOMATIC  NOLIST
   BRKSPACE(0)
  PAGE(3) MISSING'.' LENGTH(1, 59) ALIGN(LEFT) TSPACE(1) FTSPACE(1)
 MARGINS(1,45)
 /TITLE=
  LEFT 'Percent Mediated'
  RIGHT 'Page )PAGE'
 /VARIABLES
 toteff (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 ratio  (VALUES)  (RIGHT)  (OFFSET(0)) (10)
 /BREAK (TOTAL) (SKIP(1)) /SUMMARY
 SUM( toteff ) SKIP(1) SUM( ratio) ''.
```

Percent Mediated — Page 3

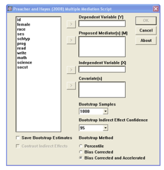
Percentage of the total effect that is mediated	Ratio of the indirect to the direct effect
45.524883	.835701

SPSS FAQ
How can I analyze multiple mediators in SPSS?

This page uses code by Kristopher J. Preacher of the University of Kansas and Andrew F. Hayes of Ohio State University. The code can be found on the webpage accompanying their paper:

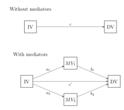
Preacher, K. J., & Hayes, A. F. (2008). Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models.*Behavior Research Methods, 40*, 879-891.

To use the Preacher and Hayes's script, first download the .sbs file. The link for the documentation describing how to download and install the .sbs file is here. Then open it in SPSS and run it by clicking on the green arrow or choosing "Run" from the Macro menu. This will open an SPSS dialog window.

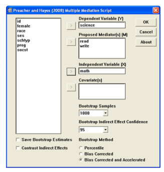


Example 1: Multiple mediators

For this example, we will use the hsb2 dataset with **science** as the dependent variable, **math** as the independent variable and **read** and **write** as the two mediator variables. The paths in such a model are depicted below. In our analysis, we are interested in finding these paths to calculate the direct and indirect effects of our variables.



To begin, we indicate which of our variables are the dependent, independent, and mediator variables in the dialog window.



This generates the output below.

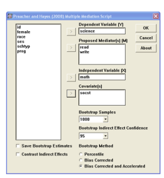
```
Run MATRIX procedure:

Dependent, Independent, and Proposed Mediator Variables:
DV =   science
IV =   math
MEDS = read
       write

Sample size
        200

IV to Mediators (a paths)
          Coeff        se         t         p
read      .7248     .0583   12.4378     .0000
write     .6247     .0566   11.0452     .0000

Direct Effects of Mediators on DV (b paths)
          Coeff        se         t         p
read      .3015     .0687    4.3903     .0000
write     .2065     .0708    2.9185     .0039

Total Effect of IV on DV (c path)
         Coeff        se         t         p
math     .6666     .0583   11.4371     .0000

Direct Effect of IV on DV (c-prime path)
         Coeff        se         t         p
math     .3190     .0767    4.1605     .0000

Model Summary for DV Model
      R-sq  Adj R-sq         F       df1       df2         p
     .4999     .4923   65.3187    3.0000  196.0000     .0000

*****************************************************************

           NORMAL THEORY TESTS FOR INDIRECT EFFECTS

Indirect Effects of IV on DV through Proposed Mediators (ab paths)
         Effect        se         Z         p
TOTAL     .3476     .0596    5.8277     .0000
read      .2186     .0524    4.1692     .0000
write     .1290     .0454    2.8422     .0045

****************************************************************

           BOOTSTRAP RESULTS FOR INDIRECT EFFECTS

Indirect Effects of IV on DV through Proposed Mediators (ab paths)
           Data      boot      Bias        SE
TOTAL     .3476     .3449    -.0027     .0645
read      .2186     .2164    -.0022     .0537
write     .1290     .1285    -.0005     .0496

Bias Corrected and Accelerated Confidence Intervals
          Lower     Upper
TOTAL     .2230     .4700
read      .1125     .3245
write     .0294     .2235

****************************************************************

Level of Confidence for Confidence Intervals:
  95

Number of Bootstrap Resamples:
  1000

------ END MATRIX -----
```

The results above assuming normality suggest that each of the separate indirect effects as well as the total indirect effect are significant. From the above results it is also possible to compute the ratio of indirect to direct effect (.3476/.3190 = 1.09) and the proportion of the total effect due to the indirect effect (.3476/(.3476 + .3190) = .52).

The normal theory tests for indirect effects compute the standard errors using the delta method which assumes that the estimates of the indirect effect are normally distributed. For many situations this is acceptable, but it does not work well for the indirect effects which are usually positively skewed and kurtotic. Thus the z-test and p-values for these indirect effects generally cannot be trusted. Therefore, it is recommended that bootstrap standard errors and confidence intervals be used. Additonally, if your outcome is binary, a proportion, or a percent, bootstrap estimates should be used. These can be found in the next block of output. These standard errors are slightly larger than those calculated assuming normality and the overall interpretation remains the same.

Example 2: Multiple mediators with control variable(s)

What do you do if you also have control variables? You add them as covariates to the model. Let's say that **socst** is a covariate. We can see the new results below.




```
Run MATRIX procedure:

Dependent, Independent, and Proposed Mediator Variables:
DV =   science
IV =   math
MEDS = read
       write

Statistical Controls:
CONTROL= socst

Sample size
        200

IV to Mediators (a paths)
          Coeff        se         t         p
read      .5038     .0634    7.9512     .0000
write     .4144     .0617    6.7112     .0000

Direct Effects of Mediators on DV (b paths)
          Coeff        se         t         p
read      .3100     .0729    4.2517     .0000
write     .2149     .0748    2.8723     .0045

Total Effect of IV on DV (c path)
         Coeff        se         t         p
math     .5671     .0684    8.2863     .0000

Direct Effect of IV on DV (c-prime path)
         Coeff        se         t         p
math     .3218     .0773    4.1654     .0000

Partial Effect of Control Variables on DV
          Coeff        se         t         p
socst    -.0227     .0645    -.3519     .7253

Model Summary for DV Model
      R-sq  Adj R-sq         F       df1       df2         p
     .5003     .4900   48.8010    4.0000  195.0000     .0000

****************************************************************

           BOOTSTRAP RESULTS FOR INDIRECT EFFECTS

Indirect Effects of IV on DV through Proposed Mediators (ab paths)
           Data      boot      Bias        SE
TOTAL     .2452     .2439    -.0014     .0478
read      .1562     .1539    -.0023     .0413
write     .0891     .0900     .0009     .0342

Bias Corrected and Accelerated Confidence Intervals
          Lower     Upper
TOTAL     .1597     .3522
read      .0863     .2476
write     .0311     .1671

****************************************************************

Level of Confidence for Confidence Intervals:
  95

Number of Bootstrap Resamples:
  1000

------ END MATRIX -----
```

SPSS FAQ

How can I get out-of-sample predicted values?

Sometimes it is useful to get predicted values for cases that were not used in the regression analysis. There are two ways to do this in SPSS. Let's use the hsb2 dataset and create some missing values in a variable. Specifically, we will set the first nine values in the variable **write** to be missing. Then we will use **write** as our outcome variable in an OLS regression analysis. Of course, the cases with missing values will not be used in the analysis, but we can still get the predicted values for those cases.

```
get file ='d:/data/hsb2.sav'.

sort cases by id.
if id lt 10 write = $sysmis.
list write read math
/cases=from 1 to 12.
```

```
   write      read      math

      .       34.00     40.00
      .       39.00     33.00
      .       63.00     48.00
      .       44.00     41.00
      .       47.00     43.00
      .       47.00     46.00
      .       57.00     59.00
      .       39.00     52.00
      .       48.00     52.00
    54.00     47.00     49.00
    46.00     34.00     45.00
    44.00     37.00     45.00

Number of cases read:  12    Number of cases listed:  12
```

Method 1

When running the **regression** command, we can use the **save** subcommand to save the predicted values to the current data file. We have supplied the name for the new variable in parentheses after the SPSS keyword **pred**. After running the regression, we will list the first 12 cases in the data set for the variables **write** and **pred_1**.

```
regression
  /dependent write
  /method = enter read math
  /save pred(pred_1).

<output omitted>

list write pred_1
/cases from 1 to 12.
```

```
    write      pred_1

      .      42.24554
      .      40.81015
      .      54.03857
      .      45.58411
      .      47.28941
      .      48.53128
      .      56.83733
      .      48.67533
      .      51.30748
    54.00    49.77315
    46.00    44.31532
    44.00    45.19271

 Number of cases read:  12    Number of cases listed:  12
```

Method 2

Another way to get out-of-sample predictions is to save the model information to an .xml file, use the **model handle** command to name the .xml file, and then use the **ApplyModel** function of the **compute** command to create the predicted values. We will list the first 12 cases in the data file for the variables **write** and **yhat**.

```
regression
  /dependent  write
  /method = enter read math
  /outfile=model('d:/data/working/hsb_m1.xml').

<output omitted>

model handle name = m1 file='d:/data/working/hsb_m1.xml'.

compute yhat = ApplyModel(m1,'predict').

list write yhat
/cases from 1 to 12.
```

```
    write     yhat

      .      42.25
      .      40.81
      .      54.04
      .      45.58
      .      47.29
      .      48.53
      .      56.84
      .      48.68
      .      51.31
    54.00    49.77
    46.00    44.32
    44.00    45.19

 Number of cases read:  12    Number of cases listed:  12
```

Now let's look at **pred_1** and **yhat** side by side; as you can see, they are the same.

```
formats pred_1 yhat (f8.5).

list write pred_1 yhat
/cases from 1 to 12.
```

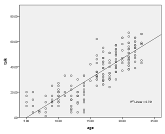
```
    write   pred_1     yhat

      .   42.24554 42.24554
      .   40.81015 40.81015
      .   54.03857 54.03857
      .   45.58411 45.58411
      .   47.28941 47.28941
      .   48.53128 48.53128
      .   56.83733 56.83733
      .   48.67533 48.67533
      .   51.30748 51.30748
    54.00 49.77315 49.77315
    46.00 44.31532 44.31532
    44.00 45.19271 45.19271

 Number of cases read:  12    Number of cases listed:  12
```

SPSS FAQ

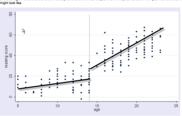
How can I run a piecewise regression in SPSS?

Say that you want to look at the relationship between how much a child talks on the phone and the age of the child. You get a random sample of 200 kids. You ask them how old they are and how many minutes they spend talking on the phone. Finally, you save the data as talk.sav. You then make a scatterplot of the data as shown below.

```
GGRAPH
  /GRAPHDATASET NAME="iGraphDataset" VARIABLES= talk age
  /GRAPHSPEC SOURCE=INLINE INLINETEMPLATE=["<addFitLine  type='linear' target='pair'/> "].
BEGIN GPL
SOURCE: s=userSource( id( "iGraphDataset" ) )
DATA:talk=col( source(s), name( "talk" ) )
DATA: age=col( source(s), name( "age" ) )
GUIDE: axis( dim( 1 ), label( "age" ) )
GUIDE: axis( dim( 2 ), label( "talk" ) )
SCALE: linear( dim( 1 ), min( 5 ), max( 25 ) )
SCALE: linear( dim( 2 ), min( 0 ), max( 80 ) )
ELEMENT: point( position( ( age * talk ) ) )
END GPL.
```



Thinking about this more, you decide that you think that the amount of time that kids talk on the phone changes dramatically at age 14, and that the slope might change at that age as well. You think that a piecewise regression might make more sense, where before age 14 there is an intercept and linear slope, and after age 14, there is a different intercept and different linear slope, kind of like pictured below with just freehand drawing of what the two regression lines might look like.



Try 1: Separate regressions

To investigate this, we can run two separate regressions, one for before age 14, and one for after age 14. We can compare the results of these two models.

```
* Before age 14.
compute before14 = (age < 14).
filter by before14.
regression /dep=talk /method=enter age.
filter off.
```

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	8.075	3.981		2.028	.047
	age	.682	.382	.225	1.786	.079

a. Dependent Variable: talking on the phone

```
* At age 14 and after.
compute after14 = (age >= 14).
filter by after14.
regression /dep=talk /method=enter age.
filter off.
```

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	-24.973	5.709		-4.374	.000
	age	3.629	.301	.718	12.037	.000

a. Dependent Variable: talking on the phone

Note how the slopes do seem quite different for the two groups. However, the intercepts don't make much sense, since they are the predicted time talking on the phone when one is 0 years old.

Try 2: Separate regression with age centered at 14

Let's rescale (center) **age** by subtracting 14. Then, when **age** is 0, that really refers to being 14 years old.

```
* age14 subtracts 14 from age, so age is 0 when child is 14.
compute age14 = (age - 14).

* Now, rerun regression when child is below 14.
compute before14 = (age < 14).
filter by before14.
regression /dep=talk /method=enter age14.
filter off.
```

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	17.624	1.752		10.057	.000
	age14	.682	.382	.225	1.786	.079

a. Dependent Variable: talk talking on the phone

```
* Now, rerun regression when child is 14 years of age or older.
compute after14 = (age >= 14).
filter by after14.
regression /dep=talk /method=enter age14.
filter off.
```

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	25.834	1.626		15.884	.000
	age14	3.629	.301	.718	12.037	.000

a. Dependent Variable: talk talking on the phone

Note how the slopes for the two groups stayed the same, but now the intercepts (**Constant**) are the predicted talking time at age 14 for the two groups. We can see that at age 14 there seems to be not only a change in the slope (from .682 to 3.62) but also a jump in the intercept (from 17.6 to 25.8). This suggest that at age 14, there is discontinuous jump in time talking on the phone as well as a change in the slope as well. However, this is merely suggestive; we should really test this in a combined model.

Try 3: Combined model, coding for separate slope and intercept

We now combine the two models into a single model. To do this, we need to create some new variables.

- **age1** is the age centered around age 14 but converted to 0s after age 14 (representing the effect of age for before 14 year olds).
- **age2** is the age centered around age 14 but converted to 0s before 14 (representing the effect of age for after 14 year olds).
- **int1** is 1 before age 14 (representing the intercept for before 14 year olds).
- **int2** is 1 after age 14 (representing the intercept for after 14 year olds).

```
compute  age1 = (age - 14).
if (age >= 14) age1 = 0 .
compute  age2 = (age - 14).
if (age < 14) age2 = 0 .

compute  int1 = 1.
if (age >= 14) int1 = 0.
compute  int2 = 1.
if (age < 14)  int2 = 0.
execute.
```

That might have been confusing, so let us show what these variables look like in a table below. Note that we have a strange person who is 13.99 years old (very, very close to being 14, but not quite). This person will be helpful for seeing the effect of the jump from going from being under 14 to being 14.

```
* Check the coding.

* Save our data file so far.
save outfile = "c:\data\talk2.sav".
execute.

* Collapse the data to make the coding easier to see.
aggregate
  /outfile=* mode = addvaraibles
  /break=age int1 int2 age1 age2
  /count=N.
list cases.
```

age	int1	int2	age1	age2	count
5.00	1.00	.00	-9.00	.00	4
6.00	1.00	.00	-8.00	.00	4
7.00	1.00	.00	-7.00	.00	2
8.00	1.00	.00	-6.00	.00	5
9.00	1.00	.00	-5.00	.00	6
10.00	1.00	.00	-4.00	.00	13
11.00	1.00	.00	-3.00	.00	3
12.00	1.00	.00	-2.00	.00	13
13.00	1.00	.00	-1.00	.00	11
13.99	1.00	.00	-.01	.00	1
14.00	.00	1.00	.00	.00	11
15.00	.00	1.00	.00	1.00	2
16.00	.00	1.00	.00	2.00	15
17.00	.00	1.00	.00	3.00	20
18.00	.00	1.00	.00	4.00	12
19.00	.00	1.00	.00	5.00	25
20.00	.00	1.00	.00	6.00	8
21.00	.00	1.00	.00	7.00	22
22.00	.00	1.00	.00	8.00	16
23.00	.00	1.00	.00	9.00	7

Now we can go back to the **talk2.sav** data file before we did this collapsing.

```
get file = "c:\data\talk2.sav".
```

Now we are ready to run our combined regression. We will put in the intercept for both groups, so we don't need an intercept from SPSS so we use the **origin** option to put the regression through the origin (i.e., no intercept). This is necessary because our model has an implied constant, **int1** plus **int2** adds up to 1. Note that the r-square is not valid for this model and should not be reported.

```
* Run the regression, compare to try 2.
regression  /origin /dependent=talk
  /method=enter int1 int2 age1 age2
  /save=pred(yhat).
```

Coefficients[a,b]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	int1	17.624	2.009	.255	8.771	.000
	int2	25.834	1.547	.557	16.701	.000
	age1	.682	.438	.045	1.558	.121
	age2	3.629	.287	.422	12.656	.000

a. Dependent Variable: talk talking on the phone

b. Linear Regression through the Origin

Now let's obtain the predicted values (shown in the table below) and relate those to the meaning of the coefficients above.

- **age1** is the slope when age is less than 14. For example, as you go from being age 5 to 6, phone talking goes from 11.48 to 12.17, and that equals .68 (with rounding).
- **age2** is the slope when age is 14 or higher. For example, as you go from being age 15 to 16, phone talking goes from 29.46 to 33.09, and that equals 3.62 (with rounding).
- **int1** is the predicted mean for someone who is just infinitely close to being 14 years old (but not quite 14). Note that when someone is 13.99 their predicted mean is very close to **int1** (17.616 versus 17.624).
- **int2** is the predicted mean for someone who just turned 14 years old, and note that 25.83 is the value for **int2** and is the value for the predicted value at exactly age 14.

```
means tables=yhat by age.
```

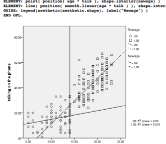
yhat Unstandardized Predicted Value * age

yhat Unstandardized Predicted Value

age	Mean	N	Std. Deviation
5.00	11.48639	4	.00000000
6.00	12.16831	4	.00000000
7.00	12.85023	2	.00000000
8.00	13.53215	5	.00000000
9.00	14.21407	6	.00000000
10.00	14.89599	13	.00000000
11.00	15.57791	3	.00000000
12.00	16.25983	13	.00000000
13.00	16.94175	11	.00000000
13.99	17.61685	1	.
14.00	25.83397	11	.00000000
15.00	29.46302	2	.00000000
16.00	33.09206	15	.00000000
17.00	36.72111	20	.00000000
18.00	40.36015	12	.00000000
19.00	43.97920	25	.00000000
20.00	47.60824	8	.00000000
21.00	51.23729	22	.00000000
22.00	54.86634	16	.00000000
23.00	58.49538	7	.00000000
Total	34.41000	200	15.14670989

Below we show a graph of the results.

```
compute newage = 0.
if age ge 14 newage = 1.
exe.

GGRAPH
  /GRAPHDATASET NAME="iGraphDataset" VARIABLES= talk age newage
  /GRAPHSPEC SOURCE=INLINE EDITABLE=YES
     INLINETEMPLATE=["<addFitLine  type='linear' target='subgroup'/>" ].
BEGIN GPL
SOURCE: s=userSource( id( "iGraphDataset" ) )
DATA: talk=col( source(s), name( "talk" ) )
DATA: age=col( source(s), name( "age" ) )
DATA: newage=col( source(s), name( "newage" ), unit.category()  )
GUIDE: axis( dim( 1 ), label( "age" ) )
GUIDE: axis( dim( 2 ), label( "talking on the phone" ) )
GUIDE: legend( aesthetic( aesthetic.shape.interior ), label( "newage" ) )
SCALE: linear( dim( 1 ), min( 5 ), max( 25 ) )
SCALE: linear( dim( 2 ), min( 0 ), max( 80 ) )
GUIDE: form.line(position(14, *))
ELEMENT: point( position( age * talk ), shape.interior(newage) )
ELEMENT: line( position( smooth.linear(age * talk ) ), shape.interior( newage) )
GUIDE: legend(aesthetic(aesthetic.shape), label("Newage") )
END GPL.
```



You might want to test whether the difference in the intercepts is 0 or whether the change in the slopes is different from 0. The next section shows how we can do this.

Try 4: Alternate coding, coding to compare intercepts and slopes

This is another way you can code this model. Note that we include **age14** and **age2** for the two terms for age, and include the intercept (by not excluding it) and **int2** to represent the intercept values. With this coding, **age2** and **int2** represent the change in slope and intercept from being less than 14 to being 14 and older.

```
regression
/dependent=talk
/method=enter age14 age2 int2
/save=pred(yhat2).
```

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	17.624	2.009		8.771	.000
	age14	.682	.438	.186	1.558	.121
	age2	2.947	.523	.518	5.631	.000
	int2	8.210	2.536	.220	3.238	.001

a. Dependent Variable: talk talking on the phone

Using this coding scheme, here is the meaning of the coefficients.

- **age14** is the slope when age is less than 14.
- **age2** is the **change** in the slope as a result of becoming age 14 or higher (as compared to being less than 14).
- **(Constant)** is the predicted mean for someone who is just infinitely close to being 14 years old (but not quite 14).
- **int2** is the predicted mean for someone who just turned 14 years old minus the predicted mean for someone who is infinitely close to being 14 years old (the jump that occurs at age 14).

As you can see, the coefficients for **age2** and **int2** now focus on the **change** that results from becoming 14 years old.

Below we compute the predicted values calling them **yhat2**. Note how the predicted values are the same for this model and the prior model, because the models are essentially the same, they are just parameterized differently.

```
means tables=yhat2 by age.
```

Report

yhat2 Unstandardized Predicted Value

age	Mean	N	Std. Deviation
5.00	11.48639	4	.00000000
6.00	12.16831	4	.00000000
7.00	12.85023	2	.00000000
8.00	13.53215	5	.00000000
9.00	14.21407	6	.00000000
10.00	14.89599	13	.00000000
11.00	15.57791	3	.00000000
12.00	16.25983	13	.00000000
13.00	16.94175	11	.00000000
13.99	17.61685	1	.
14.00	25.83397	11	.00000000
15.00	29.46302	2	.00000000
16.00	33.09206	15	.00000000
17.00	36.72111	20	.00000000
18.00	40.36015	12	.00000000
19.00	43.97920	25	.00000000
20.00	47.60824	8	.00000000
21.00	51.23729	22	.00000000
22.00	54.86634	16	.00000000
23.00	58.49538	7	.00000000
Total	34.41000	200	15.14670989

Summary

This brief FAQ compared different ways of creating piecewise regression models. All of these models are equivalent, just parameterized differently. They all generate the exact predicted values. The differences in parameterization are merely a rescrambling of the intercepts and slopes for the two segments of the regression model. You can choose the coding strategy that you like best, although it is often useful to use both schemes.

SPSS FAQ

How can I output the results of my regression to an SPSS data file?

Sometimes it is useful to output the results of a regression analysis to a data file for further analyses. To do this in SPSS, you can use the **output** subcommand of the **regression** command. You have two choices of what to save using this subcommand: you can save the covariance matrix of the coefficients (with the **covb** option) or you can save the correlation matrix of the coefficients (with the **covr** option). Let us use a data set called hsb2 as an example. We will save the results and the covariance matrix of the coefficients in a file called **out1.sav**.

```
get file 'd:\hsb2.sav'.
regression
 /dep = write
 /method = enter read female
 /outfile = covb('d:\out1.sav').
```

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	20.228	2.714		7.454	.000
	READ	.566	.049	.612	11.459	.000
	FEMALE	5.487	1.014	.289	5.410	.000

a. Dependent Variable: WRITE

```
get file 'd:\out1.sav'.
list.

DEPVAR_  ROWTYPE_ VARNAME_   CONST_      READ   FEMALE

WRITE    COV      CONST_       7.36      -.13     -.70
WRITE    COV      READ         -.13       .00      .00
WRITE    COV      FEMALE       -.70       .00     1.03
WRITE    EST                  20.23       .57     5.49
WRITE    SE                    2.71       .05     1.01
WRITE    SIG                    .00       .00      .00
WRITE    DFE                 197.00    197.00   197.00

Number of cases read:  7    Number of cases listed:  7
```

As you can see above, the covariances between the estimates have been saved to this file, as well as the estimates, their standard errors, the significance and the error degrees of freedom. Note that the precision of the values saved in **out1.sav** is greater than the two decimal places shown here. Two decimal places are shown because that is the default number of decimal places to display in an SPSS data set. You can easily increase the number of decimal places shown by going to the "Variable View" of the SPSS Data Editor and increasing the value in the column labeled "Decimals" (you may have to increase the column width first).

Now let's run the same regression and this time use the **corb** option instead.

```
get file 'd:\hsb2.sav'.
regression
 /dep = write
 /method = enter read female
 /outfile = corb('d:\out2.sav').
```

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	20.228	2.714		7.454	.000
	READ	.566	.049	.612	11.459	.000
	FEMALE	5.487	1.014	.289	5.410	.000

a. Dependent Variable: WRITE

```
get file 'd:\out2.sav'.
list.

DEPVAR_  ROWTYPE_ VARNAME_   CONST_      READ   FEMALE

WRITE    COR      CONST_       1.00      -.96     -.25
WRITE    COR      READ         -.96      1.00      .05
WRITE    COR      FEMALE       -.25       .05     1.00
WRITE    EST                  20.23       .57     5.49
WRITE    SE                    2.71       .05     1.01
WRITE    SIG                    .00       .00      .00
WRITE    DFE                 197.00    197.00   197.00

Number of cases read:  7    Number of cases listed:  7
```

As we can see, the correlations between the coefficients have been saved to the data set, as well as the estimates, their standard errors, the significance and the error degrees of freedom.

Results can also be saved into data sets using the Output Management System (OMS). For more information on how to use OMS to output results to data sets, please see How can I output my results to a data file in SPSS? .

SPSS FAQ

How can I test a group of variables in SPSS regression?

Suppose that you want to run a regression model and to test the statistical significance of a group of variables. For example, let's say that you want to predict students' writing score from their reading, math and science scores. The data set with these variables in it can be downloaded by following this link: hsb2.sav .

The SPSS syntax for this would be:

```
regression
 /dependent = write
 /method = enter read math science.
```

Variables Entered/Removed(b)

Model	Variables Entered	Variables Removed	Method
1	science score, reading score, math score(a)	.	Enter

a All requested variables entered.
b Dependent Variable: writing score

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.684(a)	.467	.459	6.97111

a Predictors: (Constant), science score, reading score, math score

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	8353.990	3	2784.663	57.302	.000(a)
	Residual	9524.885	196	48.596		
	Total	17878.875	199			

a Predictors: (Constant), science score, reading score, math score
b Dependent Variable: writing score

Coefficients(a)

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	13.192	3.069		4.299	.000
	reading score	.236	.069	.255	3.410	.001
	math score	.319	.076	.316	4.222	.000
	science score	.202	.069	.211	2.918	.004

a Dependent Variable: writing score

Now let's suppose that you wanted to test the combined effect of math and science on writing. The SPSS syntax for doing that is below. Note that the variables listed in the **method = test()** subcommand are not listed on the **method = enter** subcommand. In other words, the independent variables are listed only once. Also note that, unlike other SPSS subcommands, you can have multiple **method =** subcommands within the **regression** command.

```
regression
 /dependent = write
 /method = enter read
 /method = test(math science).
```

Variables Entered/Removed(b)

Model	Variables Entered	Variables Removed	Method
1	reading score(a)	.	Enter
2	science score, math score	.	Test

a All requested variables entered.
b Dependent Variable: writing score

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.597(a)	.356	.353	7.62487
2	.684(b)	.467	.459	6.97111

a Predictors: (Constant), reading score
b Predictors: (Constant), reading score, science score, math score

ANOVA(d)

Model			Sum of Squares	df	Mean Square	F	Sig.	R Square Change
1	Regression		6367.421	1	6367.421	109.521	.000(a)	
	Residual		11511.454	198	58.139			
	Total		17878.875	199				
2	Subset Tests	math score, science score	1986.569	2	993.284	20.439	.000(b)	.111
	Regression		8353.990	3	2784.663	57.302	.000(c)	
	Residual		9524.885	196	48.596			
	Total		17878.875	199				

a Predictors: (Constant), reading score
b Tested against the full model.
c Predictors in the Full Model: (Constant), reading score, science score, math score.
d Dependent Variable: writing score

Coefficients(a)

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	23.959	2.806		8.539	.000
	reading score	.552	.053	.597	10.465	.000
2	(Constant)	13.192	3.069		4.299	.000
	reading score	.236	.069	.255	3.410	.001
	math score	.319	.076	.316	4.222	.000
	science score	.202	.069	.211	2.918	.004

a Dependent Variable: writing score

Excluded Variables(b)

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	math score	.396(a)	5.583	.000	.370	.561
	science score	.322(a)	4.609	.000	.312	.603

a Predictors in the Model: (Constant), reading score
b Dependent Variable: writing score

If you wanted to test all three variables together, the syntax would be:

```
regression
 /dependent = write
 /method = test(read math science).
```

Variables Entered/Removed(a)

Model	Variables Entered	Variables Removed	Method
1	science score, reading score, math score	.	Test

a Dependent Variable: writing score

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.684(a)	.467	.459	6.97111

a Predictors: (Constant), science score, reading score, math score

ANOVA(c)

Model			Sum of Squares	df	Mean Square	F	Sig.	R Square Change
1	Subset Tests	reading score, math score, science score	8353.990	3	2784.663	57.302	.000(a)	.467
	Regression		8353.990	3	2784.663	57.302	.000(b)	
	Residual		9524.885	196	48.596			
	Total		17878.875	199				

a Tested against the full model.
b Predictors in the Full Model: (Constant), science score, reading score, math score.
c Dependent Variable: writing score

Coefficients(a)

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	13.192	3.069		4.299	.000
	reading score	.236	.069	.255	3.410	.001
	math score	.319	.076	.316	4.222	.000
	science score	.202	.069	.211	2.918	.004

a Dependent Variable: writing score

You will notice that the output from the first example with the three independent variables on the **method = enter** subcommand and the output from this example with the three independent variables on the **method = test()** subcommand are virtually identical. The only difference between them is the line in the ANOVA table that gives the test of the subset, which in this case is all of the variables. The point of this example is that you can put all of the independent variables in the regression on the **method = test()** subcommand and not use a **method = enter** subcommand if you like.

SPSS FAQ
How can I create a scatterplot with a regression line in SPSS?

There are at least two ways to make a scatterplot with a regression line in SPSS. One way is to use the **graph** command, and another way is to use the **ggraph** command. Both are illustrated below.

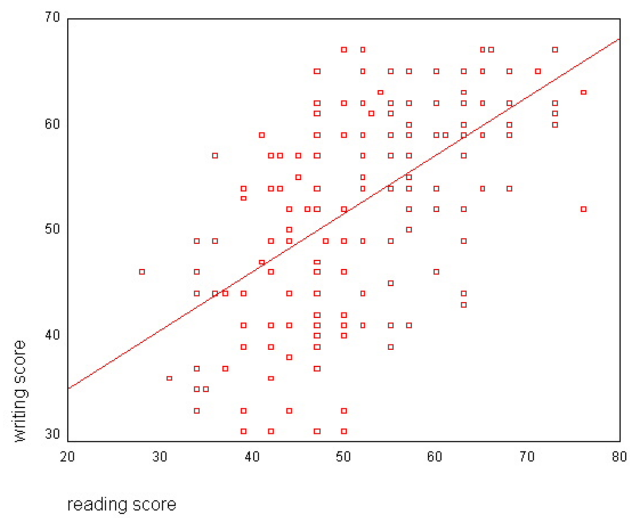
Let's read in an example dataset, hsb2, which contains data from the High School and Beyond study.

```
get file 'c:\hsb2.sav'.
```

Let's do a scatterplot of the variables **write** with **read**. Note that after running the code below, you need to double click on the graph, which will open up the chart editor window. Select "chart" from the menu at the top and then "options" (which is the first item in the menu). On the right of the dialogue box is a check box called "Total" under the heading "Fit Line". Click on the box to put in the check, or click on "Fit Options" to select a different type of fit method, such as lowess, quadratic or cubic. Close the chart editor so that the changes to take effect on your graph.

Using the graph command

```
graph
  /scatterplot(bivar)=read with write.
```



Using the ggraph command

The **ggraph** command was introduced in version 14 of SPSS. This command can be used to create and edit scatterplots. Below is the syntax for creating a scatterplot with the regression line.

```
GGRAPH
  /GRAPHDATASET NAME="graphdataset" VARIABLES=read write
  /GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
 SOURCE: s=userSource(id("graphdataset"))
 DATA: read=col(source(s), name("read"))
 DATA: write=col(source(s), name("write"))
 GUIDE: axis(dim(1), label("reading score"))
 GUIDE: axis(dim(2), label("writing score"))
 ELEMENT: line( position( smooth.linear(read*write ) ) )
 ELEMENT: point(position(read*write))
END GPL.
```