kxQ Models: Using Regression & GLM for Linear Models Including Interactions

The purpose of this study was to examine the relationship of practice and exam performance. Two aspects of practice were selected for study, the difficulty of the practice and the number of practices completed. Practice difficulty was a 3-condition variable - practice problems were either about the same difficulty as the exam problems (=1), they were easier than the exam problems (=2), or they were more difficult than the exam problems (=3). Different sections of the course were randomly assigned to receive the three difficulty levels. Students were permitted to complete as many practice problems as they liked, receiving very complete feedback after each problem. The dependent variable was performance on an examination.

Regression: Basic model

```
*recoding original grouping variable to same = 0 as comp.
* pract_dc1 compares same=1=>0 with easier = 2 => 1.
if (practgrp = 1) pract_dc1 = 0.
if (practgrp = 2) pract_dc1 = 1.
if (practgrp = 3) pract_dc1 = 0.

*pract_dc2 compare same=1=>0 with harder=3=>1.
if (practgrp = 1) pract_dc2 = 0.
if (practgrp = 2) pract_dc2 = 0.
if (practgrp = 3) pract_dc2 = 1.
```

IF statements to dummy-code the group variable:

- same is going to be the comparison group, so it is coded "0" for both dummy codes
- dc1 is going to compare easier with same, so easier is coded "1" as the target group & same is coded "0" (harder is also coded "0")
- dc2 is going to compare harder with same, so harder is coded as "1" as the target group & same is codec "0" (easier is also coded "0")

```
*centering original quant variable
compute numpract_cen = numpract - 5.792.
```

Centering the covariate requires subtracting the mean from each person's number of practices score

```
*computing interaction terms.
compute grp_pract_int1 = pract_dc1 * numpract_cen.
compute grp_pract_int2 = pract_dc2 * numpract_cen.
```

The product of each of the dummy codes with the centered quantitative are the interaction terms

```
*regression -- will get simple regression line for same(=0).
* -- will get group comparison at mean=0.
REGRESSION
  /STATISTICS COEFF R ANOVA
  /DEPENDENT testperf
  /METHOD=ENTER numpract_cen
.                    pract_dc1  pract_dc2
.                    grp_pract_int1   grp_pract_int2
```

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.867[a]	.752	.723	9.71200

a. Predictors: (Constant), grp_pract_int2, grp_pract_int1, pract_dc2, pract_dc1, numpract_cen

ANOVA[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	12036.352	5	2407.270	25.522	.000[a]
	Residual	3961.564	42	94.323		
	Total	15997.917	47			

a. Predictors: (Constant), grp_pract_int2, grp_pract_int1, pract_dc2, pract_dc1, numpract_cen
b. Dependent Variable: testperf

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	64.756	2.557		25.324	.000
	pract_dc1	-16.038	3.564	-.414	-4.500	.000
	pract_dc2	3.732	3.536	.096	1.055	.297
	numpract_cen	3.292	.963	.479	3.420	.001
	grp_pract_int1	-6.812	1.357	-.555	-5.018	.000
	grp_pract_int2	4.380	1.319	.374	3.320	.002

a. Dependent Variable: testperf

The model "works" significantly better than chance.

This model accounts for about 76% of the variance in the performance scores.

Interpreting the regression weights

constant
- The expected value of testperf when the value of all predictors (practice difficulty, number of practices and interaction) = 0
- The expected value of testperf for those in the same condition and who have 5.792 practices is 64.756

numpract_cen
- The direction and extent of the expected change in testperf for a 1-unit increase in this predictor, holding the value of the other predictors (practice & interaction) constant at 0
- The expected change in testperf as the number of practice changes for those in the same condition
- For each additional practice, those in the same condition are expected to decrease 3.292 on testperf – this effect is significant
- Note: because there is an interaction term in the model, the slopes of the three group's lines may be different – check the interaction to evaluate this.

pract_dc1

compares same & easier
- The direction and extent of this pairwise group difference, holding the other predictors (number of practices and the interaction constant at 0.
- The same vs. easier group difference controlling the number of practices and the interaction at 0 (their mean after centering)
- Those in the easier group 16.038% poorer than those in the same group, when holding for the number of practices at 5.792 – this effect is significant.
- So, the corrected mean for the same condition when practice is controlled at 5.792 is 64.756% (constant) and corrected mean difference between the groups is -16.038%, (pairwise regression weight), so the corrected mean for the easier condition is 48.718%

pract_dc2

compares same & harder
- The direction and extent of this pairwise group difference, holding the other predictors (number of practices and the interaction constant at 0.
- The same vs. harder group difference controlling the number of practices and the interaction at 0 (their mean after centering)
- Those in the harder group outperformed those in the same condition by 3.732%, when holding for the number of practices at 5.792 – this effect is not significant.
- So, the corrected mean for the easier condition when practice is controlled at 5.792 is 64.756% (constant) and corrected mean difference between the groups is 3.732% (pairwise regression weight), so the corrected mean for the same condition is 68.488%

grp_pract_int1

compare slopes of same & easier groups
- The direction and extent of the difference in the testperf-numpract slope for these two groups.
- The direction and extent of change in the practice difficulty pairwise group difference for each 1-unit increase in number of practices
- How the practice group difficulty changes as the number of practices changes
- For each additional practice, the difference between the similar difficulty practice group and the easier practice group increases by -6.812% – this effect is significant.
- So, for those in the same practice group performance increases by 3.392 for each practice, whereas for those in the easier practice group, performance decreases by 3.52 (3.292 - 6.812).

grp_pract_int2

compare slopes of same & harder groups
- The direction and extent of the difference in the testperf-numpract slope for these two groups.
- The direction and extent of change in the practice difficulty pairwise group difference for each 1-unit increase in number of practices
- How the practice group difficulty changes as the number of practices changes
- For each additional practice, the difference between the similar difficulty practice group and the harder practice group increases by 4.38 – this effect is significant.
- So, for those in the similar difficulty practice group performance decreases by 3.292 for each practice, whereas for those in the harder difficulty practice group, performance increases by 7.672 (3.292 + 4.38).

Obtaining the Plot of the Model

Difficulty	z1 wt	Z2 wt
Same	0	0
Easier	1	0
Harder	0	1

Add the labels for the groups – be sure they match the coding you used!

height z1=0 z2=0	constant	64.756
slope z1=0 z2=0	b(x)	3.292
height dif z1=1 z2=0	b(z1)	-16.038
slope dif z1=1 z2=0	b(xz1)	-6.812
height dif z1=0 z2=1	b(z2)	3.732
slope dif z1=0 z2=1	b(xz2)	4.38
	x(mean)	5.792
	x(std)	2.681

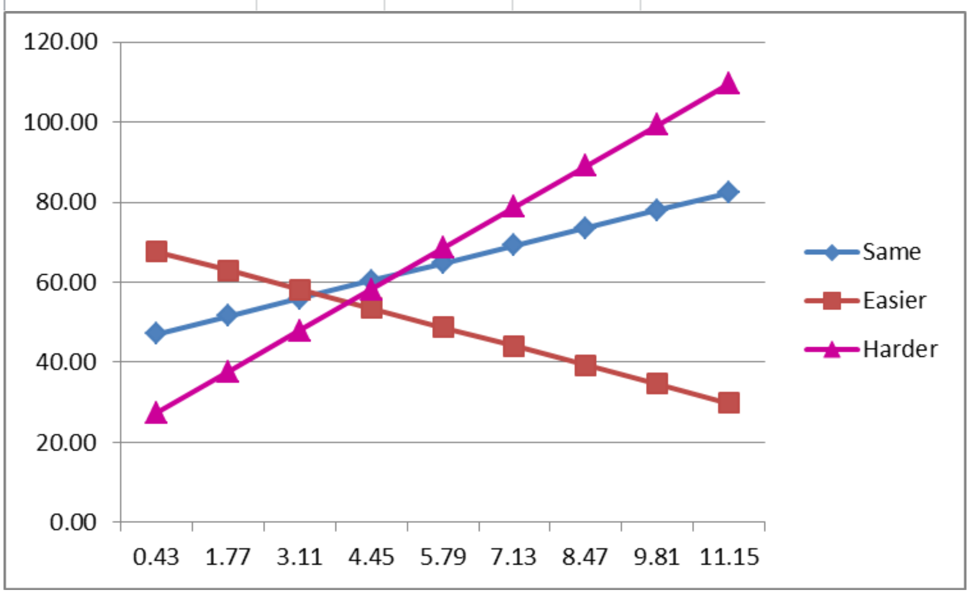
Fill in the regression weights

← notice how they are ordered! Different than SPSS output!!

Fill in the mean & std for the quant variable

	(slope * X) +		height
Same	3.292	* X +	64.756
Easier	-3.52	* X +	48.718
Intermittent FB	7.672	* X +	68.488

The program will generate the simple regression slope for each group.



Here's the plot of the model.

Remember – all the analyses shown in the following pages produce the same model!!! We may recode this or re-center that to change the specific information available from a regression weight and a significance test, but they are all the same model!

Regression: Group Differences at other Numbers of Practices

Group comparisons can be made for any value of the quantitative variable. Center the quantitative variable at the desired value, and the dummy codes will give the simple effect of group differences at that specific value.

Here is the original dummy coding of the 3 practice difficulty groups with **mean centering**.

```
* same as comparison group.
* 1=same 2=easy 3=hard.

if (practgrp = 1) pg_dc1_s0e1 = 0.
if (practgrp = 2) pg_dc1_s0e1 = 1.
if (practgrp = 3) pg_dc1_s0e1 = 0.

if (practgrp = 1) pg_dc1_s0h1 = 0.
if (practgrp = 2) pg_dc1_s0h1 = 0.
if (practgrp = 3) pg_dc1_s0h1 = 1.

Compute
 pract_meancen  = numpract - 5.792.

compute
pgs0e1_meancen_int1
 = pg_dc1_s0e1 * pract_meancen.

compute
 pgs0h1_meancen_int1
 = pg_dc1_s0h1 * pract_meancen.

Exe.

REGRESSION
  /STATISTICS COEFF R ANOVA
  /DEPENDENT testperf
  /METHOD=ENTER pract_meancen
                pg_dc1_s0e1   pg_dc1_s0h1
                pgs0e1_meancen_int1
                pgs0e1_meancen_int1.
```

Here is a set of results testing for a group difference **re-centering using 10 practices**.

```
* same as comparison group.
* 1=same 2=easy 3=hard.

if (practgrp = 1) pg_dc1_s0e1 = 0.
if (practgrp = 2) pg_dc1_s0e1 = 1.
if (practgrp = 3) pg_dc1_s0e1 = 0.

if (practgrp = 1) pg_dc1_s0h1 = 0.
if (practgrp = 2) pg_dc1_s0h1 = 0.
if (practgrp = 3) pg_dc1_s0h1 = 1.

compute
pract_10cen = numpract - 10.

compute
pgs0e1_10cen_int1
= pg_dc1_s0e1 * pract_10cen.

compute
 pgs0h1_10cen_int1
= pg_dc1_s0h1 * pract_10cen.

exe.

REGRESSION
  /STATISTICS COEFF R ANOVA
  /DEPENDENT testperf
  /METHOD=ENTER numpract_cen
                pg_dc1_s0e1   pg_dc1_s0h1
                pgs0e1_10cen_int1
                pgs0h1_10cen_int1.
```

From the original dummy coding and mean centering

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	64.757	2.557		25.325	.000
	pg_dc1_s0e1	-16.040	3.564	-.414	-4.500	.000
	pg_dc1_s0h1	3.733	3.536	.096	1.056	.297
	pract_meancen	3.292	.963	.479	3.420	.001
	pgs0e1_meancen_int1	-6.812	1.357	-.555	-5.018	.000
	pgs0h1_meancen_int1	4.380	1.319	.374	3.320	.002

a. Dependent Variable: testperf

As for the 2-group case …

The regression weight for the quantitative variable does not change

- Re-centering does not change the slope of the testperf-numpract regression line for the same group (the comparison group)

The regression weights for the interaction terms do not change

- Re-centering does not change the differences among the slopes of the testperf-numpract regression lines for the groups

Using the original dummy coding but centering at 10 practices

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	78.612	4.056		19.380	.000
	pg_dc1_s0e1	-44.704	6.558	-1.154	-6.816	.000
	pg_dc1_s0h1	22.164	6.230	.572	3.558	.001
	pract_10cen	3.292	.963	.479	3.420	.001
	pgs0e1_10cen_int1	-6.812	1.357	-.998	-5.018	.000
	pgs0h1_10cen_int1	4.380	1.319	.631	3.320	.002

a. Dependent Variable: testperf

The constant tells the mean performance of the comparison group for the recentered value of the quantitave variable

- After 10 practices, those in the Same group had an average performance of 78.612%

The regression weights for each dummy code tells the mean difference between the comparison group and the target group for that dummy code, when holding the number of practices constant at 10.

- After 10 practices, those in the Easy group scored an average of 33.908%, which is 44.704% poorer than those in the Same group ($p < .001$)

- After 10 practices, those in the Hard group scored an average of 82.992%, which is 4.380% better than those in the Same group ($p = .002$)

By selecting the dummy coding and the centering value, you can obtain test of specific group regression lines (dummy coding) and simple effect group comparisons (centering) for any model including a categorical and a quantitative predictor.

Regression: Getting the #Practice Regression Slope & Significance Test for the all 3 Practice Groups

What did we learn earlier from looking at alternative recodings of a binary variable?

- We got the same overall model from the different group codings (& interaction terms)
- We got the same simple regression models for each group from different group codings (& interaction terms)
- Different binary variable codings change the direction of the group mean difference – but test the same effect
- Different binary variable codings produce different interaction weight signs – but test the same interaction effect
- ➔ **Different binary variable codings provide H0: b=0 tests of different group's regression line slopes**
  - ➔ **We only get the test of H0: b=0 for the comparison group**

What happens when we change the coding of a multiple-category variables?

- We will get the same overall model from the different group codings (& interaction terms)
- We will get the same simple regression models for each group from different group codings (& interaction terms)
- ➔ **Different codings provide H0: b=0 tests of different group's regression line slopes**
  - ➔ **We only get the test of H0: b=0 for the comparison group used in the set of k-1 codes**
- ➔ **Different codings provide different pairwise comparisons among the groups**
  - ➔ **For each of the k-1 codes we only get tests of the mean difference between the comparison group vs each of the other k-1 groups**
- ➔ **Different codings produce different interaction codes that provide tests of different groups' regressions slopes**
  - ➔ **For each of the k-1 codes we only get tests of the slope difference between the comparison group vs each of the other k-1 groups**

So, to get a complete set of direct regression slope tests and between groups comparisons we will need to apply three different sets of dummy codes for the group variable (each with their specific interaction codes).

Keeping track of the different coding sets can get complicated, so ***be sure to create labels*** for codes that you can recover hours, days, months, way later…

```
* same as comparison group.
* 1=same 2=easy 3=hard.

if (practgrp = 1) pg_dc1_s0e1 = 0.
if (practgrp = 2) pg_dc1_s0e1 = 1.
if (practgrp = 3) pg_dc1_s0e1 = 0.

if (practgrp = 1) pg_dc1_s0h1 = 0.
if (practgrp = 2) pg_dc1_s0h1 = 0.
if (practgrp = 3) pg_dc1_s0h1 = 1.

Compute
 pract_meancen
 = numpract - 5.792.

compute
pgs0e1_meancen_int1
 = pg_dc1_s0e1 * pract_meancen.

compute
 pgs0h1_meancen_int1
 = pg_dc1_s0h1 * pract_meancen.

exe
```

```
* easy as comparison group.
* 1=same 2=easy 3=hard.

if (practgrp = 1) pg_dc2_e0s1 = 1.
if (practgrp = 2) pg_dc2_e0s1 = 0.
if (practgrp = 3) pg_dc2_e0s1 = 0.

if (practgrp = 1) pg_dc2_e0h1 = 0.
if (practgrp = 2) pg_dc2_e0h1 = 0.
if (practgrp = 3) pg_dc2_e0h1 = 1.

compute
pract_meancen
= numpract - 5.792.

compute
pge0s1_meancen_int2
= pg_dc2_e0s1 * pract_meancen.

compute
 pge0h1_meancen_int2
= pg_dc2_e0h1 * pract_meancen.

exe.
```

```
* hard as comparison group.
* 1=same 2=easy 3=hard.

if (practgrp = 1) pg_dc3_h0s1 = 1.
if (practgrp = 2) pg_dc3_h0s1 = 0.
if (practgrp = 3) pg_dc3_h0s1 = 0.

if (practgrp = 1) pg_dc3_h0e1 = 0.
if (practgrp = 2) pg_dc3_h0e1 = 1.
if (practgrp = 3) pg_dc3_h0e1 = 0.

compute
pract_meancen
= numpract - 5.792.

compute
pgh0s1_meancen_int3
= pg_dc3_h0s1 * pract_meancen.

compute
 pgh0e1_meancen_int3
= pg_dc3_h0e1 * pract_meancen.

exe.
```

All the different codings should produce the same R_2 and the same F results. All three produced the following.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.867[a]	.752	.723	9.71200

a. Predictors: (Constant), pgh0e1_meancen_int3, pgh0s1_meancen_int3, pg_dc3_h0e1, pg_dc3_h0s1, pract_meancen

ANOVA[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	12036.352	5	2407.270	25.522	.000[a]
	Residual	3961.564	42	94.323		
	Total	15997.917	47			

a. Predictors: (Constant), pgh0e1_meancen_int3, pgh0s1_meancen_int3, pg_dc3_h0e1, pg_dc3_h0s1, pract_meancen
b. Dependent Variable: testperf

Here's the output and the resulting simple regression models and plots from the 3 codings.

Coding #1 "Same" as the comparison group

Coefficients[a]

Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	64.757	2.557		25.325	.000
	pg_dc1_s0e1	-16.040	3.564	-.414	-4.500	.000
	pg_dc1_s0h1	3.733	3.536	.096	1.056	.297
	pract_meancen	3.292	.963	.479	3.420	.001
	pgs0e1_meancen_int1	-6.812	1.357	-.555	-5.018	.000
	pgs0h1_meancen_int1	4.380	1.319	.374	3.320	.002

a. Dependent Variable: testperf

Different codings will have different constants, each with the mean of the comparison group.

Different codings will have different practice regression weights, each with the slope for the comparison group.

Different codings should have different pairwise mean comparison regression weights, with "opposing sets" of the three possible comparisons across codings.

Similarly, different codings should have different pairwise regression slope comparisons, again "opposing sets" of the three possible comparisons across codings.

Regression equation for each group appears below.

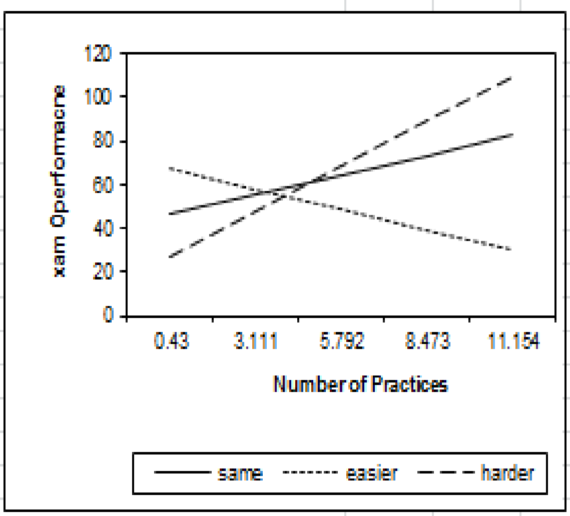
same	3.292	* X	+	64.757	
easier	-3.52	* X	+	48.717	
harder	7.672	* X	+	68.49	

Plotting points for each group apear below.

	-2 std	-1 std	mean	+1 std	+2 std
x-axis values for mean-centered scale	-5.362	-2.681	0	2.681	5.362
x-axis values for original uncentered scale	**0.43**	**3.111**	**5.792**	**8.473**	**11.154**
same	47.105	55.931	64.757	73.583	82.409
easier	67.591	58.154	48.717	39.28	29.843
harder	27.353	47.921	68.49	89.059	109.63

The different codings should all produce the same set of simple regression models for the 3 groups – the same model!

The different coding should also all produce the same set of plotting points – the same model!



The only difference in the plot of the different codings should be which groups have which line graphics (which is a consequence of how the plotting program is written, not a difference in the models).

Coding #2 "Easy" as the comparison group

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	48.717	2.483		19.622	.000
	pg_dc2_e0s1	16.040	3.564	.414	4.500	.000
	pg_dc2_e0h1	19.773	3.483	.511	5.678	.000
	pract_meancen	-3.519	.957	-.511	-3.678	.001
	pge0s1_meancen_int2	6.812	1.357	.563	5.018	.000
	pge0h1_meancen_int2	11.192	1.315	.957	8.512	.000

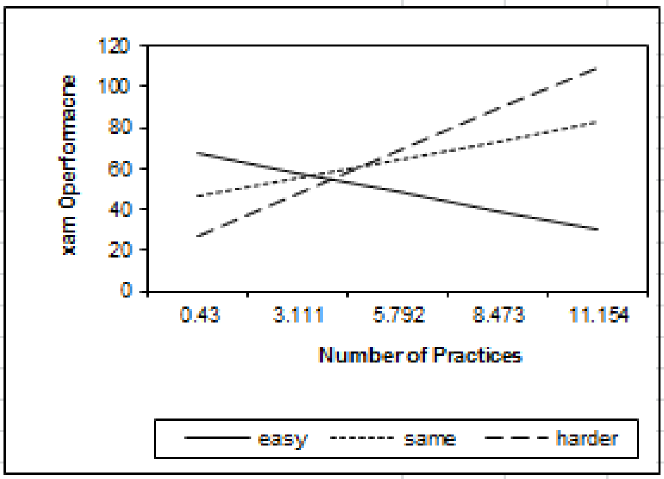
a. Dependent Variable: testperf

Regression equation for each group appears below.

same	3.292	* X	+	64.757
easier	-3.52	* X	+	48.717
harder	7.672	* X	+	68.49

Plotting points for each group apear below.

	-2 std	-1 std	mean	+1 std	+2 std
x-axis values for mean-centered scale	-5.362	-2.681	0	2.681	5.362
x-axis values for original uncentered scale	**0.43**	**3.111**	**5.792**	**8.473**	**11.154**
same	47.105	55.931	64.757	73.583	82.409
easier	67.591	58.154	48.717	39.28	29.843
harder	27.353	47.921	68.49	89.059	109.63



Coding #3 "Hard" as the comparison group

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	68.490	2.442		28.044	.000
	pg_dc3_h0s1	-3.733	3.536	-.096	-1.056	.297
	pg_dc3_h0e1	-19.773	3.483	-.511	-5.678	.000
	pract_meancen	7.672	.902	1.115	8.508	.000
	pgh0s1_meancen_int3	-4.380	1.319	-.362	-3.320	.002
	pgh0e1_meancen_int3	-11.192	1.315	-.912	-8.512	.000

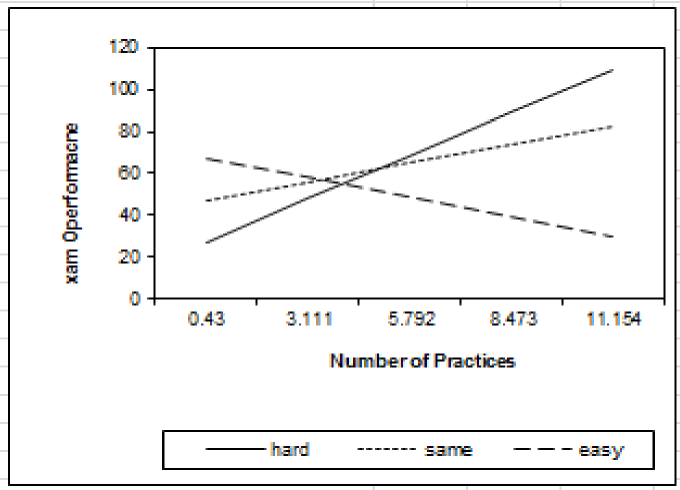
a. Dependent Variable: testperf

Regression equation for each group appears below.

hard	7.672	* X	+	68.49
same	3.292	* X	+	64.757
easy	-3.52	* X	+	48.717

Plotting points for each group apear below.

	-2 std	-1 std	mean	+1 std	+2 std
x-axis values for mean-centered scale	-5.362	-2.681	0	2.681	5.362
x-axis values for original uncentered scale	**0.43**	**3.111**	**5.792**	**8.473**	**11.154**
hard	27.353	47.921	68.49	89.059	109.63
same	47.105	55.931	64.757	73.583	82.409
easy	67.591	58.154	48.717	39.28	29.843



From all this we should have a complete set of significance tests:

Group performance difference (corrected at mean number of practices = 5.8792):

Same > Easy dif = 16.040, p < .001 — codings #1 & #2
Same "<" Hard dif = 3.733, p = .297 — codings #1 & #3
Easy < Hard dif = 19.773, p < .001 — codings #2 & #3

Performance-practice regression slope:

Same 3.292, p = .001 — coding #1
Easy -3.519, p = .001 — coding #2
Hard 7.672, p < .001 — coding #3

Performance-practice regression slope differences:

Same more positive than **Easy** dif = 6.812, p < .001 — codings #1 & #2
Hard more positive than **Same** dif = 4.380, p = .002 — codings #1 & #3
Hard more positive than **Easy** dif = 11.192, p < .001 — codings #2 & #3

GLM: Getting the Model & Comparing Practice Difficulty Groups at Several # Practices

Compute numpract_cen = numpract - 5.792.
exe.

*use numpract_cen -- will get group comparison at mean=0.
*with practice mean centered what was 5.792 (mean) is now 0
what was 2 is now (2 - 5.792) = -3.792
what was 4 is now (4 - 5.792) = -1.792
what was 6 is now (6 - 5.792) = .208
what was 8 is now (8 - 5.792) = 2.208
what was 10 is now (10 - 5.792) = 4.208.

*notice this uses a different practice difficulty variable that was recoded from the oritinal, so that "same" was coded 3 (to match the dummy coding used above).

```
UNIANOVA testperf   BY  practgrp_e1h2s3    WITH  numpract_cen
  /METHOD=SSTYPE(3)
  /EMMEANS=TABLES(practgrp_e1h2s3) WITH(numpract_cen = -3.792) COMPARE practgrp_e1h2s3)
  /EMMEANS=TABLES(practgrp_e1h2s3) WITH(numpract_cen = -1.792) COMPARE (practgrp_e1h2s3)
  /EMMEANS=TABLES(practgrp_e1h2s3) WITH(numpract_cen = .208)   COMPARE (practgrp_e1h2s3)
  /EMMEANS=TABLES(practgrp_e1h2s3) WITH(numpract_cen = 2.208)  COMPARE (practgrp_e1h2s3)
  /EMMEANS=TABLES(practgrp_e1h2s3) WITH(numpract_cen = 4.208)  COMPARE (practgrp_e1h2s3)
  /PRINT=DESCRIPTIVE PARAMETER
  /DESIGN= practgrp_e1h2s3    numpract_cen    practgrp_e1h2s3*numpract_cen.
```

Tests of Between-Subjects Effects

Dependent Variable: testperf

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	12036.352[a]	5	2407.270	25.522	.000
Intercept	167303.635	1	167303.635	1773.732	.000
practgrp_e1h2s3	3393.254	2	1696.627	17.987	.000
numpract_cen	1968.705	1	1968.705	20.872	.000
practgrp_e1h2s3 * numpract_cen	6887.914	2	3443.957	36.512	.000
Error	3961.564	42	94.323		
Total	197300.000	48			
Corrected Total	15997.917	47			

a. R Squared = .752 (Adjusted R Squared = .723)

Parameter Estimates

Dependent Variable: testperf

Parameter	B	Std. Error	t	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Intercept	64.756	2.557	25.324	.000	59.596	69.917
[practgrp_e1h2s3=1.00]	-16.038	3.564	-4.500	.000	-23.231	-8.845
[practgrp_e1h2s3=2.00]	3.732	3.536	1.055	.297	-3.404	10.868
[practgrp_e1h2s3=3.00]	0[a]	.	.	.	.	.
numpract_cen	3.292	.963	3.420	.001	1.349	5.235
[practgrp_e1h2s3=1.00] * numpract_cen	-6.812	1.357	-5.018	.000	-9.551	-4.072
[practgrp_e1h2s3=2.00] * numpract_cen	4.380	1.319	3.320	.002	1.718	7.042
[practgrp_e1h2s3=3.00] * numpract_cen	0[a]	.	.	.	.	.

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

1. practgrp_e1h2s3

Estimates

Dependent Variable: testperf

practgrp_e1h2s3	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Easy	62.064[a]	3.946	54.101	70.027
Hard	39.394[a]	3.982	31.358	47.431
Same	52.272[a]	5.072	42.036	62.508

a. Covariates appearing in the model are evaluated at the following values: numpract_cen = -3.79.

Pairwise Comparisons

Dependent Variable: testperf

(I) practgrp_e1h2s3	(J) practgrp_e1h2s3	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound	95% Confidence Interval for Difference[b] Upper Bound
Easy	Hard	22.670*	5.606	.000	11.357	33.983
	Same	9.792	6.426	.135	-3.176	22.761
Hard	Easy	-22.670*	5.606	.000	-33.983	-11.357
	Same	-12.877	6.449	.052	-25.891	.136
Same	Easy	-9.792	6.426	.135	-22.761	3.176
	Hard	12.877	6.449	.052	-.136	25.891

Based on estimated marginal means

*. The mean difference is significant at the .050 level.

b. Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).

Univariate Tests

Dependent Variable: testperf

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	1548.256	2	774.128	8.207	.001
Error	3961.564	42	94.323		

The F tests the effect of practgrp_e1h2s3. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

2. practgrp_e1h2s3

Estimates

Dependent Variable: testperf

practgrp_e1h2s3	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Easy	55.025[a]	2.707	49.563	60.488
Hard	54.739[a]	2.779	49.130	60.348
Same	58.857[a]	3.505	51.783	65.930

a. Covariates appearing in the model are evaluated at the following values: numpract_cen = -1.79.

Pairwise Comparisons

Dependent Variable: testperf

(I) practgrp_e1h2s3	(J) practgrp_e1h2s3	Mean Difference (I-J)	Std. Error	Sig.[a]	95% Confidence Interval for Difference[a] Lower Bound	95% Confidence Interval for Difference[a] Upper Bound
Easy	Hard	.286	3.880	.942	-7.543	8.116
	Same	-3.831	4.428	.392	-12.768	5.106
Hard	Easy	-.286	3.880	.942	-8.116	7.543
	Same	-4.117	4.473	.363	-13.145	4.910
Same	Easy	3.831	4.428	.392	-5.106	12.768
	Hard	4.117	4.473	.363	-4.910	13.145

Based on estimated marginal means

a. Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).

Univariate Tests

Dependent Variable: testperf

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	93.192	2	46.596	.494	.614
Error	3961.564	42	94.323		

The F tests the effect of practgrp_e1h2s3. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

3. practgrp_e1h2s3

Estimates

Dependent Variable: testperf

practgrp_e1h2s3	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Easy	47.986[a]	2.532	42.877	53.096
Hard	70.084[a]	2.469	65.100	75.067
Same	65.441[a]	2.502	60.393	70.490

a. Covariates appearing in the model are evaluated at the following values: numpract_cen = .21.

Pairwise Comparisons

Dependent Variable: testperf

(I) practgrp_e1h2s3	(J) practgrp_e1h2s3	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound	95% Confidence Interval for Difference[b] Upper Bound
Easy	Hard	-22.097*	3.537	.000	-29.235	-14.960
	Same	-17.455*	3.559	.000	-24.637	-10.272
Hard	Easy	22.097*	3.537	.000	14.960	29.235
	Same	4.643	3.515	.194	-2.451	11.736
Same	Easy	17.455*	3.559	.000	10.272	24.637
	Hard	-4.643	3.515	.194	-11.736	2.451

Based on estimated marginal means

*. The mean difference is significant at the .050 level.

b. Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).

Univariate Tests

Dependent Variable: testperf

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	4060.060	2	2030.030	21.522	.000
Error	3961.564	42	94.323		

The F tests the effect of practgrp_e1h2s3. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

4. practgrp_e1h2s3

Estimates

Dependent Variable: testperf

practgrp_e1h2s3	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Easy	40.948[a]	3.580	33.722	48.173
Hard	85.429[a]	3.313	78.743	92.115
Same	72.026[a]	2.765	66.445	77.607

a. Covariates appearing in the model are evaluated at the following values: numpract_cen = 2.21.

Pairwise Comparisons

Dependent Variable: testperf

(I) practgrp_e1h2s3	(J) practgrp_e1h2s3	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound	95% Confidence Interval for Difference[b] Upper Bound
Easy	Hard	-44.481*	4.878	.000	-54.325	-34.637
	Same	-31.078*	4.524	.000	-40.208	-21.949
Hard	Easy	44.481*	4.878	.000	34.637	54.325
	Same	13.403*	4.315	.003	4.694	22.112
Same	Easy	31.078*	4.524	.000	21.949	40.208
	Hard	-13.403*	4.315	.003	-22.112	-4.694

Based on estimated marginal means

*. The mean difference is significant at the .050 level.

b. Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).

Univariate Tests

Dependent Variable: testperf

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	8195.115	2	4097.557	43.442	.000
Error	3961.564	42	94.323		

The F tests the effect of practgrp_e1h2s3. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

5. practgrp_e1h2s3

Estimates

Dependent Variable: testperf

practgrp_e1h2s3	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Easy	33.909[a]	5.153	23.509	44.308
Hard	100.774[a]	4.729	91.231	110.316
Same	78.611[a]	4.056	70.425	86.796

a. Covariates appearing in the model are evaluated at the following values: numpract_cen = 4.21.

Pairwise Comparisons

Dependent Variable: testperf

(I) practgrp_e1h2s3	(J) practgrp_e1h2s3	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound	95% Confidence Interval for Difference[b] Upper Bound
Easy	Hard	-66.865*	6.994	.000	-80.979	-52.751
	Same	-44.702*	6.558	.000	-57.936	-31.467
Hard	Easy	66.865*	6.994	.000	52.751	80.979
	Same	22.163*	6.230	.001	9.590	34.735
Same	Easy	44.702*	6.558	.000	31.467	57.936
	Hard	-22.163*	6.230	.001	-34.735	-9.590

Based on estimated marginal means

*. The mean difference is significant at the .050 level.

b. Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).

Univariate Tests

Dependent Variable: testperf

	Sum of Squares	df	Mean Square	F	Sig.
Contrast	8854.449	2	4427.224	46.937	.000
Error	3961.564	42	94.323		

The F tests the effect of practgrp_e1h2s3. This test is based on the linearly independent pairwise comparisons among the estimated marginal means.

Together these give a pretty complete set of significance tests to help interpret the plot of the model.

Additional pairwise group comparisons could be made at any desired # practices.

Same GLM Model - but testing Performance-#practices regression for Easy & Hard Practice groups

GLM only gives the regression slope for the group coded as the comparison group -- coded=0 for all dummy codes. In order to get the full set of regression slopes for each group, and their significance tests, requires we compose additional versions of the grouping variable, each with a different group = 3.

recode practgrp_e1h2s3 (1=3) (2=2) (3=1) into practgrp_1s2h3e.

```
UNIANOVA testperf   BY  practgrp_1s2h3e    WITH  numpract_cen
  /METHOD=SSTYPE(3)
  /PRINT=DESCRIPTIVE PARAMETER
  /DESIGN= practgrp_1s2h3e    numpract_cen    practgrp_1s2h3e*numpract_cen.
```

Parameter Estimates

Dependent Variable: testperf

Parameter	B	Std. Error	t	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Intercept	48.719	2.483	19.623	.000	43.708	53.729
[practgrp_1s2h3e=1.00]	16.038	3.564	4.500	.000	8.845	23.231
[practgrp_1s2h3e=2.00]	19.770	3.483	5.677	.000	12.741	26.798
[practgrp_1s2h3e=3.00]	0[a]	.	.	.	.	.
numpract_cen	-3.519	.957	-3.678	.001	-5.451	-1.588
[practgrp_1s2h3e=1.00] * numpract_cen	6.812	1.357	5.018	.000	4.072	9.551
[practgrp_1s2h3e=2.00] * numpract_cen	11.192	1.315	8.512	.000	8.538	13.845
[practgrp_1s2h3e=3.00] * numpract_cen	0[a]	.	.	.	.	.

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

recode practgrp_e1h2s3 (1=1) (2=3) (3=2) into practgrp_e1s2h3 .

```
UNIANOVA testperf   BY  practgrp_e1s2h3    WITH  numpract_cen
  /METHOD=SSTYPE(3)
  /PRINT=DESCRIPTIVE PARAMETER
  /DESIGN= practgrp_e1s2h3    numpract_cen    practgrp_e1s2h3*numpract_cen.
```

Parameter Estimates

Dependent Variable: testperf

Parameter	B	Std. Error	t	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Intercept	68.488	2.442	28.044	.000	63.559	73.417
[practgrp_e1s2h3=1.00]	-19.770	3.483	-5.677	.000	-26.798	-12.741
[practgrp_e1s2h3=2.00]	-3.732	3.536	-1.055	.297	-10.868	3.404
[practgrp_e1s2h3=3.00]	0[a]	.	.	.	.	.
numpract_cen	7.672	.902	8.508	.000	5.853	9.492
[practgrp_e1s2h3=1.00] * numpract_cen	-11.192	1.315	-8.512	.000	-13.845	-8.538
[practgrp_e1s2h3=2.00] * numpract_cen	-4.380	1.319	-3.320	.002	-7.042	-1.718
[practgrp_e1s2h3=3.00] * numpract_cen	0[a]	.	.	.	.	.

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

using practgrp_1s2h3e gives practice regression slope for “Easy” of -3.519 ($p = .001$) ← original coding
using practgrp_e1h2s3 gives practice regression slope for “Same” of 3.292 ($p = .001$)
using practgrp_e1s2h3 gives practice regression slope for “Hard” of 7.672 ($p < .001$)