SAS for a 3x2 factorial with analysis of covariance

```
data factorial_cov;
input block A B x y;
cards;
...
;
proc glm;
	title 'Model to test individual ANOVAS';
		class block A B;
		model x y= block A B A*B;
proc glm;
	title 'Model to test general regression';
		model y= x;
proc glm;
	title 'Model to test ANCOVA';
		class block A B;
		model y= block A B A*B x / solution;
		output=check p=predi r=resi;
		lsmeans A;
		lsmeans B;
		contrast 'A lineal' A  -1  0 1;
proc glm;
	title 'Homogeneity regressions in A';
		class A B block;
		model y= A B A*B block x A*x;
proc glm;
	title 'Homogeneity regressions in B';
		class A B block;
		model y= A B A*B block x B*x;
proc univariate data check normal;
	title 'normality of residuals';
	var resi; ;
run; quit;
```

For a partition of interaction (Lec. 9)

create ID for A-B combination

model y= block ID x;

contrast 'A lineal' ID -1 0 1 -1   0 1;

contrast 'A quad'   ID -1 2 -1 -1 2 -1;

contrast 'B'   ID 1 1 1 -1 -1 -1;

contrast 'A li. X B' ID -1 0 1 1   0 -1;

For a partition of unequally spaced trt

class block B ~~A~~;

model y= block B x A A*A;

For a mixed model (A random, Lec. 10.7)

random block A A*B;

test h= A e= A*B;

contrast 'A lineal' A -1 0 1 / e= A*B;

If no interaction (main effects)

Homogeneity of regressions in main

Levene's test for main effects adjusted Z

If interaction (simple effects)

Homogeneity of regressions in ID

Levene's test for ID adjusted Z

$$Z = Y - b*(X - Xmean)$$

from solutio

from X ANOVA

INTERACTIONS in Three-way factorial

Example: SAS for a 4x3x2 factorial

```
proc glm;
        title 'ANOVAS';
                class block A B C;
                model  y= block A B C A*B A*C B*C A*B*C;
```

---

2 interactions significant: e.g. A*B and A*C significant, B*C NS:

```
proc sort;
  by A;
proc glm;
        class block B C;
        model  y= block B C B*C;
  by A;
```

---

1 interaction significant: e.g. A*B significant, A*C and B*C NS:

OK to study **main effect on C**, e.g contrast 'C lineal'   C -1 0 1;

But for A and B

1) Study the effect of A within each level of B (for all C combined)
2) Study the effect of B within each level of A (for all C combined)

```
proc sort;
  by B;
proc glm;
        class block A;
        model  y= block A;      If C is random you need to include
        means A;                C & A*C to use the A*C as error term
  by B;
```

---

3 interactions significant: If all three double interactions are significant:

1) Study A within each combination B*C
2) Study B within each combination A*C
3) Study C within each combination A*B

To do this you need replications within A*B*C combinations

SPLIT PLOT

CRD		RCBD		Latin Square	
A Error A	a-1 a(r-1)	Blocks A Error A	r-1 a-1 (r-1)(a-1)	Rows Columns A Error A	a-1 a-1 a-1 (a-1)(a-2)
Total	ra-1	Total	ra-1	Total	ra-1
Factor B **A x B** **Error B** **Total**	b-1 (a-1)(b-1) a(r-1)(b-1) rab-1	**Factor B** **A x B** **Error B** **Total**	b-1 (a-1)(b-1) a(r-1)(b-1) rab-1	**Factor B** **A x B** **Error B** **Total**	b-1 (a-1)(b-1) a(r-1)(b-1) rab-1

CRD

```
proc glm;
class rep A B;
model y= A rep*A
          B A*B;
test h=A e=rep*A;
```

RCBD

```
proc glm;
class bl A B;
     model y=Bl A bl*A
              B A*B;
   test h=A e=bl*A;
```

LS

```
proc glm;
class row col A B;
     model y=col row A col*row*A
                B A*B;
     test h=A e= col*row*A;
```

Replicated LS (= rows & columns)

```
proc glm;
class sqr row col A B;
model y=sqr col row A sqr*col*row*A
        B A*B;
test h=A e= sqr*col*row*A;
```

STRIP PLOT

```
proc GLM;
  class block A B;
  model yield = block
                A A*block
                B B*block
                B*A;
  test h=A e=A*block;
  test h=B e=B*block;
```

SPLIT SPLIT PLOT

```
proc glm;
    class Block a b c;
    model response= Block a Block*a
                         b a*b Block*b*a
                         c a*c b*c a*b*c;
    test h=a   e=Block*a;
    test h=b   e=Block*b*a;
    test h=a*b e=Block*b*a;
```

Strip Plot EMS with Random statement

```
proc glm;
  class block A B;
  model yield=block A block*A
              B block*B A*B;
  random block block*A block*B /Test;
```

Source	Type III Expected Mean Square
block	Var(Error) + 4 Var(block*B) + 4 Var(block*A)+ 8 Var(block)
A	**Var(Error) + 4 Var(block*A)** + Q(A,A*B)
block*A	**Var(Error) + 4 Var(block*A)**
B	**Var(Error) + 4 Var(block*B)** + Q(B,A*B)
block*B	**Var(Error) + 4 Var(block*B)**
A*B	Var(Error) + Q(A*B)

```
  test h=A e=block*A;
  test h=B e=block*B;
```

Split Plot EMS with Random statement

```
proc glm;
  class block A B;
  model yield=block A block*A
              B A*B;
  random block block*A /Test;
```

Source	Type III Expected Mean Square
block	Var(Error) + 4 Var(block*A) + 8 Var(block)
A	**Var(Error) + 4 Var(block*A)** + Q(A,A*B)
block*A	**Var(Error) + 4 Var(block*A)**
B	Var(Error) + Q(B,A*B)
A*B	Var(Error) + Q(A*B)

```
  test h=A e=block*A;
```

Three way factorial with one split (NOT a split-split-Plot)

1. RCBD Split-Plot. Main-Plot Treatment structure is an A*B factorial**.**
Main plot is split by factor C on the sub-plot

Block 1

A2-B1	A1-B2	A1-B1	A2-B2
C3	C1	C2	C2
C2	C3	C1	C3
C1	C2	C3	C1

Block 2

A1-B1	A2-B2	A2-B1	A1-B2
C1	C2	C1	C3
C3	C1	C2	C2
C2	C3	C3	C1

```
proc glm;
  class block A B C;
  model y = block A B A*B block*A*B C A*C B*C A*B*C;
        test h= a      e= block*A*B;
        test h= b      e= block*A*B;
        test h=a*b     e= block*A*B;
```

block*A*B= block*A*B + block*A + block*B. Remember that SAS includes all the interactions of lower level not included in the model

Alternatively, open the A B factorial as a single treatment Z with all factor combination levels. Error is Z*Block as before.

2. RCBD Split-Plot. A= Main-Plot is split by a B*C factorial on the sub-plot

Block 1

A2	A1	A3
B2C1	B2C2	B1C1
B1C1	B1C1	B1C2
B2C2	B1C2	B2C2
B1C2	B2C1	B2C1

Block 2

A1	A3	A2
B1C2	B2C1	B2C1
B1C1	B1C2	B1C2
B2C2	B1C1	B2C2
B2C1	B2C2	B1C1

```
proc glm;
  class block A B C;
  model y = block A block*A B A*B C A*C B*C A*B*C;
  test h=a e=block*A;
```

Alternatively, open the B*C factorial as a single treatment Z with all factor combination levels.

Expected Mean Squares for a mixed model with a nested factor

Five varieties of barley are tested in three locations selected at random from the Central Valley. In each location, the experiment is organized as a RCBD with four blocks.

Variety is a fixed factor
Locality is a random factor
Blocks are nested within Location (Block 1 in Location 1 is not more similar to Block 1 in location 2 than to the other blocks in the same location)

Table of expected mean squares and correct F tests

Fixed or Random Number of levels Factor	**F** **5** **i**	**R** **3** **j**	**R** **4** **k**	**Expected MS**	**F**
$\alpha_{i\ (variety)}$	0	3	4	$\sigma^2_\varepsilon + 4\sigma^2_{\alpha\gamma} + 12\Sigma\alpha^2/4$	$MS_\alpha / MS_{\alpha\beta}$
$\beta_{j\ (location)}$	5	1	4	$\sigma^2_\varepsilon + 4\sigma^2_{\alpha\gamma} + 5\sigma^2_{\gamma(j)} + 20\sigma^2_\beta$	$(MS_\beta + MS_\varepsilon)/(MS_{\alpha\beta} + MS_{\gamma(j)})$
$\gamma_{k(j)}$ (Block within location)	5	1	1	$\sigma^2_\varepsilon + 5\sigma^2_{\gamma(j)}$	$MS_{\gamma(j)} / MS_\varepsilon$
$(\alpha\beta)_{ij}$	**1**	1	4	$\sigma^2_\varepsilon + 4\sigma^2_{\alpha\gamma}$	$MS_{\alpha\beta}/MS_\varepsilon$
$\varepsilon_{k(ij)}$	1	1	1	σ^2_ε	

To test A or B we use the interaction $MS_{\alpha\beta}$ because we are trying to extend our conclusions across the universe of all locations in the Central Valley. Then the effect or A or B needs to be significantly **larger than the interactions**, which represent the differences in responses for A in the different B levels.

```
proc GLM;
   class Block Loc Var;
   model yield = Loc Var Loc*Var block(Loc);
   random block(Loc), Loc, Var*Loc /Test ;
    test h= Var  e= Loc*Var
```

RCBD with 1 observation

The error is automatically the **Trt*Block**

If Trt NS -> Tukey test to test multiplicative effects: if yes → transform data

RCBD with more than 1 observation

	Trt1	Trt2	Trt3
Block 1	**X X**	**X X**	**X X**
Block 2	**X X**	**X X**	**X X**

To test the **block*Trt** we include it in the model

```
proc GLM;
  class block Trt ;
  model yield = block Trt block*Trt;
  random block Trt*block ;
run; quit;
```

Source	Type III Expected Mean Square
block	Var(Error) + 4 Var(block*Trt) + 12 Var(block)
Trt	Var(Error) + 4 Var(block*Trt) + Q(Trt)
block*Trt	Var(Error) + 4 Var(block*Trt)

A) If **Block*Trt** is NOT significant: **eliminate it from the model**

- This is equivalent to declare block and block*trt as random)

B) If **Block*Trt** interaction is significant: we have a problem.

- If it becomes NS by transformation, it was a multiplicative effect, if not->
- The fact that the interaction is significant indicates that the effect of the Trt is different in each block!
- It does not make much sense to describe the Trt effects by block (this is a **general situation for any random factor**).
- If you still want to make a conclusion for Trt that is valid across all blocks, the correct error term is the interaction. See the result of the random statement.
- So the advice will be the same: **eliminate Block*Trt from the model**

Remember: the significance of the interaction may be pointing to an interesting biological or ecological problem!

Two nested and 1 crossed factors

Objective: extend conclusions about fire vs. not fire in Central Valley Grasslands.

Variable: diversity of nematodes in the soil

2 types of Environments

A: Fires between 1950-1990

B: No Fires

4 random locations with previous Fire and 4 within no Fire.

1 2 3 4

2 blocks

1 2

2 treatments T: Grazed and Not Grazed

1 2

```
data ex99_3;
input Fire $ L block N Diversity@@;
cards;
YES 1 1 1 3.7   YES 1 1 2 3.9     YES 1 2 1 3.9   YES 1 2 2 3.7
YES 2 1 1 3.3   YES 2 1 2 3.1     YES 2 2 1 3.5   YES 2 2 2 3.7
YES 3 1 1 3.4   YES 3 1 2 3.2     YES 3 2 1 4.0   YES 3 2 2 4.3
YES 4 1 1 3.6   YES 4 1 2 3.5     YES 4 2 1 3.8   YES 4 2 2 3.5

NO  1 1 1 4.1   NO  1 1 2 4.4     NO  1 2 1 4.0   NO  1 2 2 4.2
NO  2 1 1 3.6   NO  2 1 2 3.7     NO  2 2 1 3.9   NO  2 2 2 3.4
NO  3 1 1 3.8   NO  3 1 2 3.9     NO  3 2 1 3.7   NO  3 2 2 3.1
NO  4 1 1 3.5   NO  4 1 2 3.2     NO  4 2 1 3.9   NO  4 2 2 3.8
;
Proc GLM;
class FIRE L T Block;
model Diversity= FIRE L(FIRE) T T*FIRE T*L(FIRE) Block(L FIRE);
   random L(FIRE) Block(L FIRE) T*L(FIRE);
   test h= T      e= T*L(FIRE);
   test h= T*FIRE  e= T*L(FIRE);
   test h= FIRE e= L(FIRE);
run; quit;
```

Source	Type III Expected Mean Square
Fire	Var(Error) + 2 Var(block(L Fire)) + 2 Var(T*L(Fire)) + 4 Var(L(Fire)) + Q(Fire)
L(Fire)	Var(Error) + 2 Var(block(L Fire)) + 2 Var(T*L(Fire)) + 4 Var(L(Fire))
T	Var(Error) + 2 Var(T*L(Fire)) + Q(T)
Fire*T	Var(Error) + 2 Var(T*L(Fire)) + Q(Fire*T)
T*L(Fire)	Var(Error) + 2 Var(T*L(Fire))
block(L Fire)	Var(Error) + 2 Var(block(L