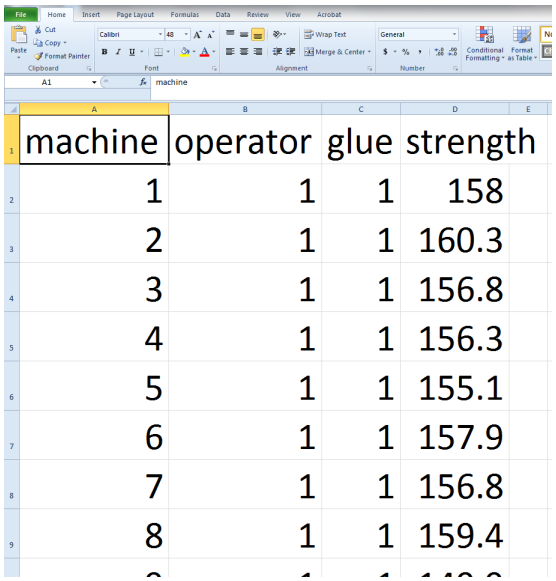
STAT 5200 Handout #14: Two-Way Random Effects Design (Ch. 11)

Example: As part of a quality control exercise, an experiment is carried out to evaluate the effects of two factors on the variability in strength of cardboard cartons. Ten Machines (for making the cartons) are randomly selected from a large number of machines, and ten Operators are randomly selected from a large number of operators. Each operator makes four carton samples using each machine. The carton samples are produced in a completely random order and a single measurement is made of the strength of each carton. This is a 10 × 10 factorial design with two random factors, Machine and Operator, and 4 replicates per factor level combination. The data are in a tab-delimited file named "carton_data.txt", which looks like this:



machine	operator	glue	strength
1	1	1	158
2	1	1	160.3
3	1	1	156.8
4	1	1	156.3
5	1	1	155.1
6	1	1	157.9
7	1	1	156.8
8	1	1	159.4

(NOTE: this is based on the textbook's Ch. 11 example; there is actually another factor Glue, with two levels. The original design is a 10 × 10 × 2 factorial design with three random factors, and 2 replicates per factor level combination. For ease of discussion, we will ignore the Glue factor.)

```
/* STAT 5200
   two-way random effects design
   (10 by 10 factorial, with both factors random effects)
   carton data
 */


/* Read in data -- data from text website (exmpl11.2) */
proc import out=work.carton
    datafile = "/home/jrstevens/STAT5200/carton_data.txt"
    dbms=tab replace;
 getnames=yes;
run;
```

```
/* Fit model (with default method=reml) */
proc mixed data=carton covtest plots=residualpanel;
  class Operator Machine;
  model Strength = ;
  random Operator | Machine;
  title1 'PROC MIXED with REML';
run;
```

PROC MIXED with REML

The Mixed Procedure

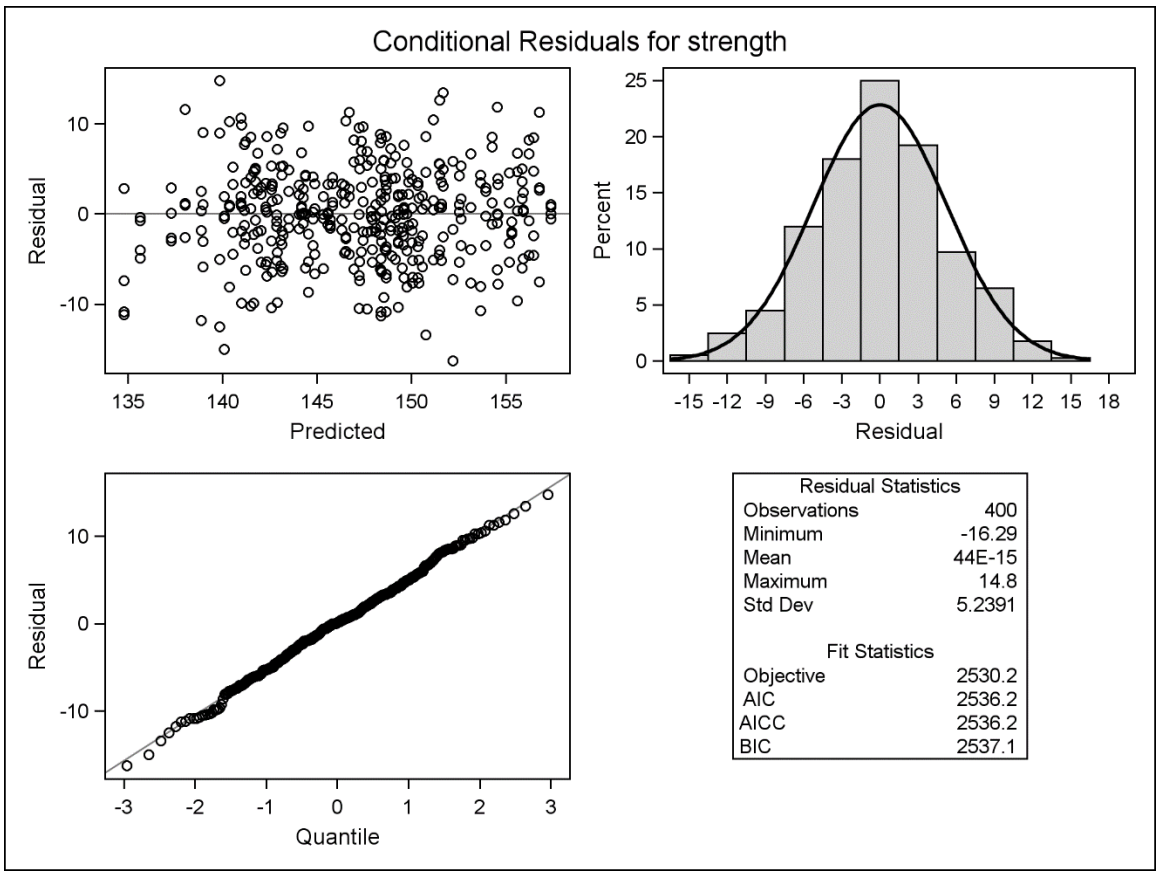
Model Information	
Data Set	WORK.CARTON
Dependent Variable	strength
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Class Level Information		
Class	**Levels**	**Values**
operator	10	1 2 3 4 5 6 7 8 9 10
machine	10	1 2 3 4 5 6 7 8 9 10

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	2747.41449304	
1	2	2530.16926202	0.00000025
2	1	2530.16903899	0.00000000

Convergence criteria met.

Covariance Parameter Estimates				
Cov Parm	**Estimate**	**Standard Error**	**Z Value**	**Pr > Z**
operator	23.9670	11.6351	2.06	0.0197
machine	6.7995	3.5434	1.92	0.0275
operator*machine	0	.	.	.
Residual	28.6607	2.0765	13.80	<.0001




```
/* Why use REML instead of TYPE3? */
proc mixed data=carton covtest method=type3;
  class Operator Machine;
  model Strength = ;
  random Operator | Machine;
  title1 'PROC MIXED with TYPE3';
  title2 '(this is what PROC GLM would do)';
run;
```

PROC MIXED with TYPE3
(this is what PROC GLM would do)

Type 3 Analysis of Variance			
Source	**DF**	**Sum of Squares**	**Mean Square**
operator	9	8886.818466	987.424274
machine	9	2705.802491	300.644721
operator*machine	81	1682.543339	20.772140
Residual	300	9237.172900	30.790576

Type 3 Analysis of Variance					
Source	**Expected Mean Square**	**Error Term**	**Error DF**	**F Value**	**Pr > F**
operator	Var(Residual) + 4 Var(operator*machine) + 40 Var(operator)	MS(operator*machine)	81	47.54	<.0001
machine	Var(Residual) + 4 Var(operator*machine) + 40 Var(machine)	MS(operator*machine)	81	14.47	<.0001
operator*machine	Var(Residual) + 4 Var(operator*machine)	MS(Residual)	300	0.67	0.9822
Residual	Var(Residual)	.	.	.	.

Covariance Parameter Estimates				
Cov Parm	**Estimate**	**Standard Error**	**Z Value**	**Pr Z**
operator	24.1663	11.6372	2.08	0.0378
machine	6.9968	3.5441	1.97	0.0484
operator*machine	-2.5046	1.0300	-2.43	0.0150
Residual	30.7906	2.5140	12.25	<.0001

2 method of moments estimates (using context of method = type3)

```
/* What to do next? */
proc mixed data=carton covtest;
  class Operator Machine;
  model Strength = ;
  random Operator Machine;
  title1 'PROC MIXED with REML';
  title2 '(and no interaction)';
run;
```

PROC MIXED with REML
(and no interaction)

Model Information	
Data Set	WORK.CARTON
Dependent Variable	strength
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Iteration History			
Iteration	**Evaluations**	**-2 Res Log Like**	**Criterion**
0	1	2747.41449304	
1	1	2530.16903896	0.00000000

Convergence criteria met.

Covariance Parameter Estimates				
Cov Parm	**Estimate**	**Standard Error**	**Z Value**	**Pr > Z**
operator	23.9691	11.6370	2.06	0.0197
machine	6.7996	3.5435	1.92	0.0275
Residual	28.6607	2.0765	13.80	<.0001

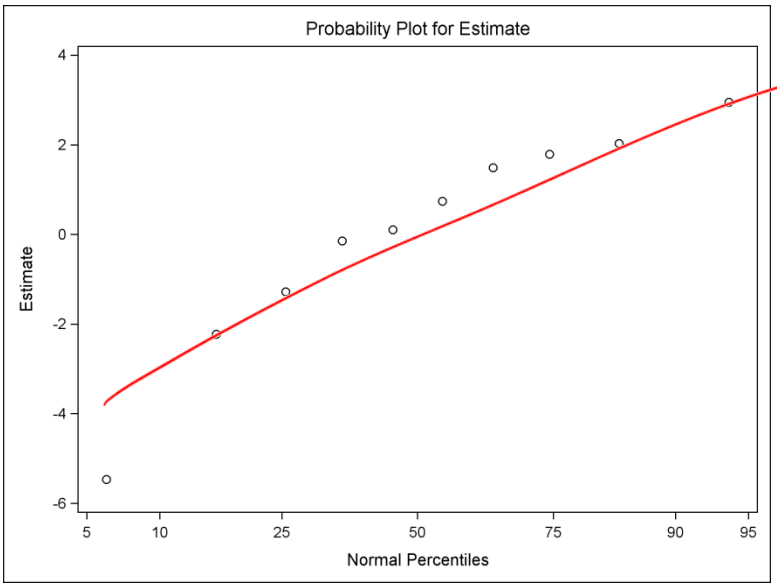
```
/* Now, what about checking normality of random effects?
   -- look at normality of random effects estimates
      (don't over-interpret,
       especially for small # of levels) */
proc mixed data=carton covtest;
  class Operator Machine;
  model Strength = ;
  random Operator Machine / solution;
  ods output solutionR=checkR;
     /* solutionR is ODS table name; send it to checkR */
run;
```

Solution for Random Effects

Effect	operator	machine	Estimate	Std Err Pred	DF	t Value	Pr > |t|
operator	1		6.6064	1.7387	381	3.80	0.0002
operator	2		0.5934	1.7387	381	0.34	0.7331
operator	3		-4.4448	1.7387	381	-2.56	0.0110
operator	4		7.2142	1.7387	381	4.15	<.0001
operator	5		-0.5954	1.7387	381	-0.34	0.7322
operator	6		-0.1043	1.7387	381	-0.06	0.9522
operator	7		-6.0940	1.7387	381	-3.50	0.0005
operator	8		-6.9567	1.7387	381	-4.00	<.0001
operator	9		2.4462	1.7387	381	1.41	0.1603
operator	10		1.3349	1.7387	381	0.77	0.4431
machine		1	-0.1469	1.1240	381	-0.13	0.8961
machine		2	1.7916	1.1240	381	1.59	0.1118
machine		3	-1.2762	1.1240	381	-1.14	0.2569
machine		4	-5.4682	1.1240	381	-4.86	<.0001
machine		5	-2.2288	1.1240	381	-1.98	0.0481
machine		6	0.7449	1.1240	381	0.66	0.5079
machine		7	0.1080	1.1240	381	0.10	0.9235
machine		8	2.9516	1.1240	381	2.63	0.0090
machine		9	2.0324	1.1240	381	1.81	0.0714
machine		10	1.4917	1.1240	381	1.33	0.1853

```
proc sort data=checkR;
  by Effect;
proc univariate data=checkR;
  by Effect;
  var Estimate;
  probplot / normal;
run;
```

Effect=machine



Effect=operator