Factorial treatment structures

- A word about factorials
- Specifying interactions among factorial effects in R
- The relationship between factors and treatment
- Interpreting results of an experiment with a factorial treatment strauctures
- Visualizing simple and main effects

A word about factorials

A factorial is *not* an experimental design. Why? Because the term "factorial" merely describes the structure of the treatment effects (i.e. the factors), *not* how they are randomized. Specifically, a factorial treatment structure is one in which all levels of every factor are present in all possible combinations with all the levels of every other factor in the experiment (i.e. the crossing of factors is complete and orthogonal). It is this complete, orthogonal structure that allows an experimenter to neatly partition the treatment sums of squares and *gain insight into interactions among factors*. Seen in this way, it becomes clear that any of the true experimental designs (i.e. randomization strategies) we have discussed so far (CRD, RCBD, Latin Square) can be factorials, provided the treatments are structured correctly.

A factorial is a complete, orthogonal structure of treatment effects intended to provide insight into their interactions.

Specifying Interactions Among Factorial Effects in R

Specifications about designs with factorial treatment structures are entered through the model statement within the lm() function, and this syntax can assume several forms:

Colons (:) are used to partition out *specific* interactions from the Treatment SS and are useful when certain interactions must be used as error terms in custom F tests. Examples:

lm(Y ~ A + B + A:B)	*specifies partitioning of SST into main effect A, me B and interaction AxB*
lm(Y ~ A + A:B + A:B:C)	*specifies me A and interactions AxB and AxBxC*

Stars (*) are used as a nice shortcut to partition the Treatment SS into *all* possible combinations of the included factors. Examples:

lm(Y ~ A*B)	*is equivalent to lm(Y ~ A + B + A*B)*
lm(Y ~ A*B*C)	*is equivalent to lm(Y ~ A + B + C + A:B + A:C + B:C + A:B:C)*

An additional nice trick to know is the use of the **carat (^)** symbol in factorial model statements. The carat (^) allows you to specify all possible combinations of model factors up to a certain level (e.g. two-way effects), saving you some typing. An example:

lm(Y ~ Block + (A + B + C)^2)	*is equivalent to lm(Y ~ Block + A + B + C + A:B + A:C + B:C); notice this excludes the three-way effect A:B:C*

The Relationship Between Factors and Treatment

Until now, we have had only a single 'treatment' variable (the effect of which we are trying to understand) with zero (CRD) or one (RCBD) blocking variables (the effects of which we are trying to account for in terms of error control but not really investigate). With factorials, we now have two or more 'factors' that are experimentally equivalent to the single 'treatment' variable from the first half of the course. To illustrate this equivalence, reconsider Example 1 from Lab 1.3.1: An experiment with 6 treatments (L08, L12, L16, H08, H12, H16), where L/H refers to Low/High temperatures and 8/12/16 refers to hours of light. This is exactly equivalent to having temperature as one factor and light as another, organized as a factorial:

lm (Growth = **Treatment**)	df Treatment = 5
lm (Growth = **Temp + Light + Temp:Light**)	df Temp = 1
	df Light = 2
	df Temp:Light = 2
	Sum = 5

What this is meant to show is that the old classification variable 'Treatment' is simply a combination of two factors (light and temperature). Rewriting the model in terms of the factors does not affect the Model df at all; it simply expands the class variable 'Treatment' into 'Temp + Light + Temp:Light'. Before, we accomplished this partitioning of the treatment SS through orthogonal contrasts. *The insights gained through each approach are equivalent.*

Example 1 *Two-Way ANOVA with interactions [Lab7ex1.R]*

In a study comparing the relative growth of five varieties of mustard (VAR) in three experimental soil mixtures (SOIL), six pots were prepared with each VAR-SOIL combination. The 90 pots were randomly allocated to six greenhouse benches (BLOCKS) and cumulative yields were measured. In this experiment, the researchers are interested only in these five varieties and three soil mixtures; so VARIETY and SOIL can be regarded as fixed factors.

```
#read in, re-classify, and inspect the data
must_dat<-as.data.frame(must_dat)
must_dat$Soil<-as.factor(must_dat$Soil)
must_dat$Var<-as.factor(must_dat$Var)
must_dat$Block<-as.factor(must_dat$Block)
str(must_dat, give.attr=F)

#The ANOVA
must_mod<-lm(Yield ~ Block + Soil*Var, must_dat)
anova(must_mod)
```

NOTE: This initial analysis enables us to see if the interaction is significant and decide: Main or simple effects?

Take a look at the resulting ANOVA table:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Block	5	22.13	4.43	0.2134	0.95573	
Soil	2	953.16	476.58	22.9813	**2.125e-08**	***
Var	4	11.38	2.85	0.1372	0.96799	
Soil:Var	8	374.49	46.81	2.2573	**0.03301**	*
Residuals	**70**	1451.64	20.74			

This is an RCBD with 6 blocks. Even though there are six replications per Soil-Variety combination (which allows us to include their interaction in the model), there is only *one* replication per Soil-Variety-Block combination. The upshot of this is that the Block*Factor interactions are inside the experimental error for this ANOVA. In other words, if the model statement had been:

```
must_mod<-lm(Yield ~ Block*Soil*Var, must_dat)
```

there would have been no variation left to estimate the error ($df_e = 0$), because:

Block * Soil =	10 df
Block * Var =	20 df
Block * Soil * Var =	40 df
	70 df = df_e

We exclude the two-way Block interactions from the model because, in general, we don't care about them (remember, we block to reduce the error term, not to gain understanding of the effect of blocking). In other words, this is a choice we make. Excluding the three-way Block interaction is *not* a choice, however; it cannot be a part of the model because it is the only term we have for our error. Of course, to justifiably relegate all these interactions to the error term requires you to show that they are NS. This can be accomplished for the two-way interactions by simply putting them into an exploratory model and looking at them:

```
#Exploratory model to look at 2-way Block interactions
must.explor_mod<-lm(Yield ~ (Block + Soil + Var)^2, must_dat)
anova(must.explor_mod)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
~~Block~~	~~5~~	~~22.13~~	~~4.43~~	~~0.2292~~	~~0.94758~~	
~~Soil~~	~~2~~	~~953.16~~	~~476.58~~	~~24.6806~~	~~1.043e-07~~	~~***~~
~~Var~~	~~4~~	~~11.38~~	~~2.85~~	~~0.1473~~	~~0.96312~~	
Block:Soil	10	242.29	24.23	1.2548	**0.28805**	**NS**
Block:Var	20	436.95	21.85	1.1314	**0.35892**	**NS**
~~Soil:Var~~	~~8~~	~~374.49~~	~~46.81~~	~~2.4242~~	~~0.03078~~	~~*~~
~~Residuals~~	~~40~~	~~772.39~~	~~19.31~~			

A little side note about what interactions to include in your model

Although one can do exploratory work with different interactions in the model and then merge the ones that are not significant into the error, you should always keep the *treatment* interactions in the model, whether significant or not.

Interpreting Results of an Experiment with a Factorial Treatment Structure

The first ANOVA table above indicates that there are significant differences among soil mixtures but not among varieties. More importantly, however, it shows that the **interaction between these two factors is significant** (i.e. the effects of soil are different for the different varieties, and vice versa).

Because the interaction is significant, it is not appropriate to analyze the main effects. One must compare the soil means separately for each variety (simple effects), and vice versa.

The continuing script...

```
#Analyze the simple effects of Soil by subsetting the data...
must_Var1_dat<-subset(must_dat, must_dat$Var == "1")
must_Var2_dat<-subset(must_dat, must_dat$Var == "2")
must_Var3_dat<-subset(must_dat, must_dat$Var == "3")
must_Var4_dat<-subset(must_dat, must_dat$Var == "4")
must_Var5_dat<-subset(must_dat, must_dat$Var == "5")

#...and then performing multiple ANOVAs
anova(lm(Yield ~ Block + Soil, must_Var1_dat))
anova(lm(Yield ~ Block + Soil, must_Var2_dat))
anova(lm(Yield ~ Block + Soil, must_Var3_dat))
anova(lm(Yield ~ Block + Soil, must_Var4_dat))
anova(lm(Yield ~ Block + Soil, must_Var5_dat))

#Tukey mean separations of Soil, within each level of Var
tukey_v1 <- HSD.test(lm(Yield ~ Block + Soil, must_Var1_dat), "Soil")
tukey_v2 <- HSD.test(lm(Yield ~ Block + Soil, must_Var2_dat), "Soil")
tukey_v3 <- HSD.test(lm(Yield ~ Block + Soil, must_Var3_dat), "Soil")
tukey_v4 <- HSD.test(lm(Yield ~ Block + Soil, must_Var4_dat), "Soil")
tukey_v5 <- HSD.test(lm(Yield ~ Block + Soil, must_Var5_dat), "Soil")
```

The above code tells R to generate five different ANOVAs, one for each variety. The results:

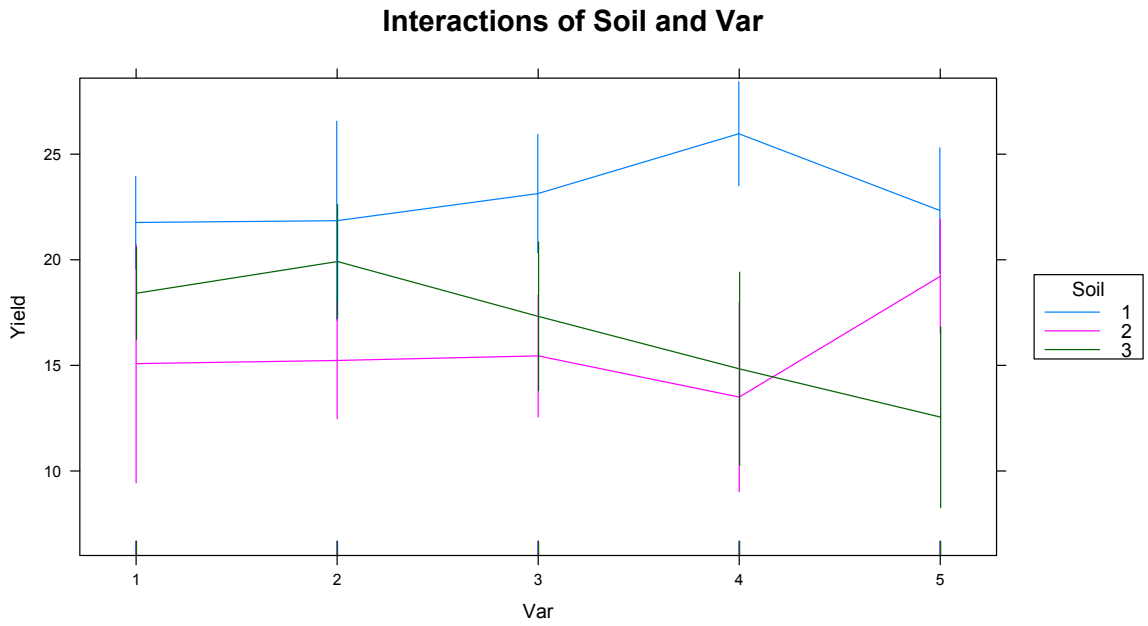
Variety	Treatment	Block	MSD	Tukey
1	0.0519 NS	0.1822 NS	6.45	1 = 3 3 = 2
2	0.0746 NS	0.5530 NS	7.15	1 = 3 = 2
3	**0.0130 ****	0.3708 NS	5.90	1 = 3 3 = 2
4	**0.0041 *****	0.6843 NS	8.38	1 3=2
5	**0.0144 ****	0.8428 NS	7.50	1 = 2 2 = 3

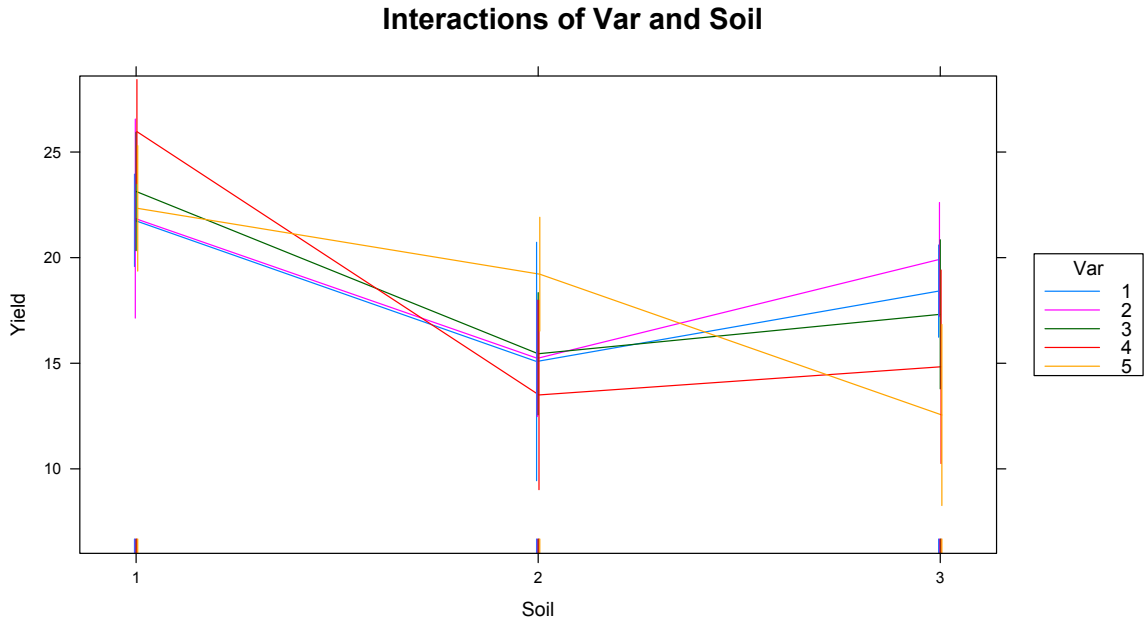
By investigating the simple effects, we see that only some varieties are significantly affected by the soil mixtures. The MSE and means separation tests vary across varieties.

Visualizing Simple Effects

The interaction plots

```
#Generate interaction plots
#install.packages("HH")
library(HH)
intxplot(Yield ~ Var, groups = Soil, data=must_dat, se=TRUE, ylim=range(must_dat$Yield),
 offset.scale=500)
intxplot(Yield ~ Soil, groups = Var, data=must_dat, se=TRUE, ylim=range(must_dat$Yield),
 offset.scale=500)
```





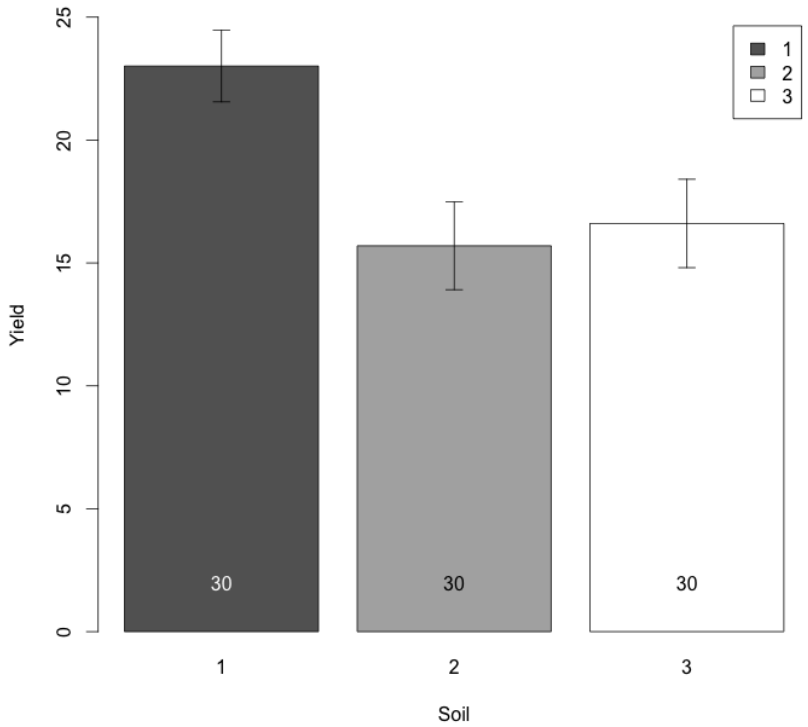
The non-parallel nature of the lines in these interaction plots demonstrate visually the significant interaction we found in the ANOVA.

DON'T FORGET: As always, we need to test assumptions in these analyses. In this particular example, there are *eight* different ANOVAs (one for variety for each of the *three* soil mixtures and one for each soil mixture for each of the *five* varieties), the assumptions of each of which must be met.

An added thorn: This analysis of simple effects involves an *enormous* number of independent questions: 8 Shapiro-Wilk tests, 8 Levene's tests, 8 non-additivity tests, 45 Tukey pairwise comparisons! This has major implications in terms of the experiment-wise error rate, so be aware!

Some additional code

```
#TEMPLATE For the interested: making a barplot of main effects using tapply

#first, get the means, by Soil type
mean.Yield <- tapply(must_dat$Yield, list(must_dat$Soil), mean)
#then, get the standard deviations, by Soil type
sd.Yield <- tapply(must_dat$Yield, list(must_dat$Soil), sd)
#finally, get the sample sizes, again by Soil type
n.Yield <- tapply(must_dat$Yield, list(must_dat$Soil), length)
se.Yield <- sd.Yield/(n.Yield)**(1/2)

barplot(mean.Yield) #makes a simple barplot!

#Pimp your barplot
mids <- barplot(mean.Yield,
 beside = TRUE, legend = TRUE,
 xlab = "Soil",
 ylab = "Yield",
 ylim = c(0,25),
 col=grey(c(0.4,0.7,1)))

#now, to add error bars, we assign the barplot above to an object called "mids"
arrows(mids, mean.Yield - 2*se.Yield, mids, mean.Yield + 2*se.Yield, code = 3, angle = 90, length
 = 0.1)

#now add text, labeling the bars
text(mids, 2, paste(n.Yield), col=c("white", rep("black", 3)))
```



Example 2 *Three-Way ANOVA with one replication [Lab7ex2.R]*

The following is the code for a generic CRD with a 3x5x2 factorial treatment structure. First, the data:

A	B	C	Y
1	1	1	61
1	2	1	39
1	3	1	121
…	…	…	…
3	3	2	63
3	4	2	167
3	5	2	128

Now the code:

```
fact_mod1<-lm(Y ~ A*B*C, fact_dat)
anova(fact_mod1)
```

Running the program like this will make you sad because there are zero degrees of freedom for the error term and thus no estimation of the error SS. The result? No F or p-values, and a cranky R:

```
          Df Sum Sq Mean Sq F value Pr(>F)
A          2 3599.3  1799.6
B          4 6423.1  1605.8
C          1 5333.3  5333.3
A:B        8 9675.1  1209.4
A:C        2 5692.5  2846.2
B:C        4 7987.0  1996.8
A:B:C      8   23.2     2.9
Residuals  0    0.0
Warning message:
ANOVA F-tests on an essentially perfect fit are unreliable
```

The solution to this problem is to assume that there is no three-way interaction, allowing us to then use the three-way interaction as an estimate of the experimental error. To do this, modify the model statement above as follows, and re-run:

```
fact_mod2<-lm(Y ~ (A + B + C)^2, fact_dat)
anova(fact_mod2)
```

```
          Df Sum Sq Mean Sq F value    Pr(>F)
A          2 3599.3  1799.6  620.56 1.682e-09 ***
B          4 6423.1  1605.8  553.72 8.364e-10 ***
C          1 5333.3  5333.3 1839.08 9.639e-11 ***
A:B        8 9675.1  1209.4  417.03 1.140e-09 ***
A:C        2 5692.5  2846.2  981.46 2.714e-10 ***
B:C        4 7987.0  1996.8  688.53 3.510e-10 ***
Residuals  8   23.2     2.9
```

You should also be able to determine which assumptions to test here and how to do them.

Visualizing Three-Way Interactions

Can't we do better than just *assume* a three-way interaction to be NS? What is a three-way interaction anyway? Though words may only confuse the issue here, one way to think about it might be:

> **A three-way interaction exists if the character of the interaction between two factors differs depending upon the level of a third factor.**

Difficult to articulate but easy to visualize. Walk through the following steps to see how one can cleverly visualize three-way interactions in a two-dimensional plot:

1. The first thing to do is go back to the original data and create a new variable [Lab7ex2b.csv]:

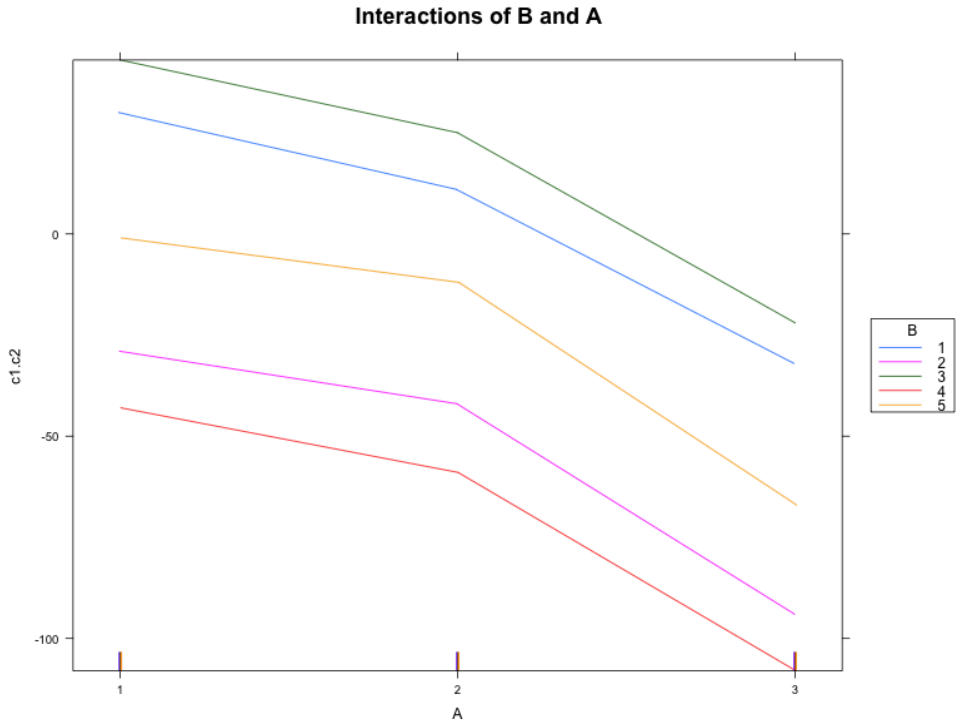
A	B	C	c1-c2	Y
1	1	1	30	61
1	2	1	-29	39
1	3	1	43	121
…	…	…	…	…
3	3	2	-22	63
3	4	2	-108	167
3	5	2	-67	128

This new variable (c1-c2) is simply the effect of c1 relative to c2 for any given combination of levels of Factors A and B. [*Side note: If C had three levels (c1, c2, c3) instead of just two, the procedure outlined here would have to be carried out for three new variables (c1-c2, c1-c3, and c2-c3) instead of just one.*]

2. Generate interaction plots [intxplot()] using (c1-c2) as the response variable:

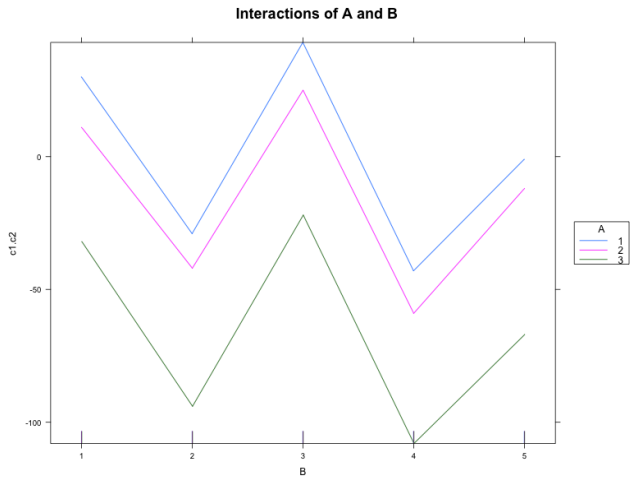
```
#Create the c1-c2 variable and re-import the data
fact_dat<-as.data.frame(fact_dat)
fact_dat$A<-as.factor(fact_dat$A)
fact_dat$B<-as.factor(fact_dat$B)
str(fact_dat, give.attr=F)

#Generate 3-way interaction plot
#install.packages("HH")
library(HH)
intxplot(c1_c2 ~ A, groups = B,
      data=fact_dat, se=TRUE, ylim=range(fact_dat$c1_c2),
      offset.scale=500)
intxplot(c1_c2 ~ B, groups = A,
      data=fact_dat, se=TRUE, ylim=range(fact_dat$c1_c2),
      offset.scale=500)
```

The output (it's like seeing in four dimensions!)



One way to think about this: Each line represents one level of B, and the average of each line represents the effect of C for each level of B. While these averages differ among lines (i.e. B:C is significant), their differences are fairly constant across all levels of A.

In other words, the roughly parallel nature of the lines in this interaction plot shows us that the difference in the effects of C at the different levels of B do not vary significantly across the levels of A. We see this same lack of a 3-way interaction in the other plot as well:



[*Translation*: No significant three-way interaction, so we are justified in using A:B:C as our error term.]

phew!