Regression Models
Lecture IV: Analyses of Designed Experiments

DT9002: Postgraduate Certificate in Applied Statistics

Dr Joe Condon

School of Mathematical Sciences
Technological University Dublin

Some Terminology...

Experimental unit: The fundamental unit (person, plot, petri-dish of cells etc.) to which a treatment is applied (randomised) which is often independent of other experimental units.

Response: The primary measure on the experimental units being analysed (or modelled). Also called the dependent variable.

Treatment: A condition imposed on an experimental unit by the researcher.

Covariate: A condition (potentially) effecting an experimental unit, often not under the control of the experimenter.

Some Terminology...

Factor: Any condition that may effect the responses from the experimental units. Quite often these are assumed to have a discrete number of levels (i.e. categorical in nature). Sometimes separate factors are combined to form 'treatments', e.g. 2 mg of a drug given twice a day, compared to 1 mg of a drug given 4 times a day - the factors of daily dosage and delivery times vary to give two treatments.

Predictor variable: General term for a treatment, covariate or factor. Also called independent variables.

Replication: Imposing the same treatment or other factor on numerous experimental units is called replication.

Randomisation: The assignment of treatments(factors) to experimental units using an understood probabilistic mechanism, e.g. by rolling a dice or drawing lots etc.

One-way completely randomised design

- Characterised by a single treatment being the only factor used in the model. Therefore, we assume (rely on) that randomisation has taken care of any other latent factors.
- Apply the ANOVA method to avoid multiple testing problem initially.
- But, multiple testing re-appears in any case where ANOVA null hypothesis is rejected.
- In particular,estimates of pairwise treatment means are often of primary importance.

Example: Clover Data

The following example studies the effect of bacteria on the nitrogen content of red clover plants. The treatment is bacteria strain, and it has six levels. Five of the six levels consist of five different Rhizobium trifolii bacteria cultures combined with a composite of five Rhizobium meliloti strains. The sixth level is a composite of the five Rhizobium trifolii strains with the composite of the Rhizobium meliloti. Red clover plants are inoculated with the treatments, and nitrogen content is later measured in milligrams.

Strain					
3DOK1	19.4	32.6	27.0	32.1	33.0
3DOK13	14.3	14.4	11.8	11.6	14.2
3DOK4	17.0	19.4	9.1	11.9	15.8
3DOK5	17.7	24.8	27.9	25.2	24.3
3DOK7	20.7	21.0	20.5	18.8	18.6
COMPOS	17.3	19.4	19.1	16.9	20.8

The model is:

$$y_i = \beta_0 + \beta_1\delta_{i1} + \beta_2\delta_{i2} + \ldots + \beta_6\delta_{i6} + \epsilon_i$$

```
> clover=read.csv("Clover.csv",header=T)
> clover$Strain=factor(clover$Strain)
> fit_clover=lm(Nitrogen~Strain,data=clover)
> anovatab(fit_clover)
       df  Sum Sq  Mean Sq  F value    Pr(>F)
Model   5     847    169.4     14.4  1.48e-06
Error  24     283     11.8
Total  29    1130
```

Clearly there are treatment differences - the next question is which treatments are different from which and by how much (i.e. maybe CI's for treatment differences?)

```
> summary(fit_clover)

Call:
lm(formula = Nitrogen ~ Strain, data = clover)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    28.820      1.535  18.769 7.53e-16 ***
Strain3DOK13  -15.560      2.172  -7.166 2.09e-07 ***
Strain3DOK4   -14.180      2.172  -6.530 9.40e-07 ***
Strain3DOK5    -4.840      2.172  -2.229 0.035446 *
Strain3DOK7    -8.900      2.172  -4.099 0.000411 ***
StrainCOMPOS  -10.120      2.172  -4.660 9.85e-05 ***


Residual standard error: 3.433 on 24 degrees of freedom
Multiple R-squared:  0.7496,  Adjusted R-squared:  0.6975
F-statistic: 14.37 on 5 and 24 DF,  p-value: 1.485e-06
```


Multiple comparisons and Error Rates

The question now is: which treatments are significantly different?

There are 6 treatments which gives 15 pairwise comparisons.

For k treatments in a model, the number of pairwise comparisons (m) is given by $m = \binom{k}{2} = \frac{k(k-1)}{2}$.

If we do all 15 tests, the experimentwise error rate is 0.54!

There are many suggestions of how to deal with this. Following are some of them that generalise to all types of regression models (not specific to linear models).

There are other specific to linear regression models (e.g. Tukey's method)- but I won't discuss these as they don't generalise.

Comparisonwise error rate (CER): The probability of falsely rejecting a null hypothesis when comparing two groups, set using α.

Experimentwise error rate (EER): The probability of at least one Type I error being made during the multiple testing procedure when all the null hypotheses are true. For the Clover data, if we use GLHs to make pairwise comparisons [how would these be done?] we have EER=0.54.

NB: the EER deals with cases where All the null hypotheses are simultaneously true, i.e. the 'full null hypothesis' and does not consider 'partial null hypotheses'.

Multiple comparisons and Error Rates continued...

False Discovery Rate (FDR): The expected proportion of rejected null hypotheses that are Type I errors. If all H_0 : are true, then FDR=EER. Therefore, controlling the FDR at α also controls the EER at α. However, when some H_0 : are true, the FDR tries to control the rate of Type I errors at α compared to the overall number of rejected null hypotheses.

(Strong) Familywise Error Rate: (FER): The probability of making any false discoveries. If the FER is controlled at α, then the FDR $\leq \alpha$.

Simultaneous Confidence Intervals: (SCI) Assume that the family of tests being conducted relate to estimating a parameter (e.g. the difference between the means of two groups). This level of control requires that all CIs for such parameters cover their respective true parameter values with confidence $1 - \alpha$ simultaneously.

There is a trade-off between protecting against Type I error and increasing the risk of Type II errors. Therefore, more conservative procedures for multiple testing tend to have less statistical power.

The hierarchy would be:

Type I error Protection	Statistical Power
SCI	CER
↓	↓
FER	FDR
↓	↓
FDR	FER
↓	↓
CER	SCI

Controlling Error Rates: Methods that adjust the p-values

Level	Method	Reject H_0 :
SCI	Bonferroni	$p_j < \alpha/m$
FER	Holm	$p_{(j)} < \alpha/(m-j+1)$
FDR	Benjamini & Hochberg	$p_{(j)} < j\alpha/(m)$
CER	use α as normal	$p_j < \alpha$

P-value Adjustments

CER: Do nothing. Obviously this has the highest statistical power (technical meaning!) of the procedures, but also makes no multiple testing correction, i.e. has the highest rate of type I errors.

```
## CER for treatment comparisons clover data
> lsmeans(fit_clover,pairwise~Strain,adjust="none")
 contrast         estimate       SE df t.ratio p.value
 3DOK1 - 3DOK13      15.56 2.171513 24   7.166  <.0001
 3DOK1 - 3DOK4       14.18 2.171513 24   6.530  <.0001
 3DOK1 - 3DOK5        4.84 2.171513 24   2.229  0.0354
 3DOK1 - 3DOK7        8.90 2.171513 24   4.099  0.0004
 3DOK1 - COMPOS      10.12 2.171513 24   4.660  0.0001
 3DOK13 - 3DOK4      -1.38 2.171513 24  -0.636  0.5311
 3DOK13 - 3DOK5     -10.72 2.171513 24  -4.937  <.0001
 3DOK13 - 3DOK7      -6.66 2.171513 24  -3.067  0.0053
 3DOK13 - COMPOS     -5.44 2.171513 24  -2.505  0.0194
 3DOK4 - 3DOK5       -9.34 2.171513 24  -4.301  0.0002
 3DOK4 - 3DOK7       -5.28 2.171513 24  -2.431  0.0229
 3DOK4 - COMPOS      -4.06 2.171513 24  -1.870  0.0738
 3DOK5 - 3DOK7        4.06 2.171513 24   1.870  0.0738
 3DOK5 - COMPOS       5.28 2.171513 24   2.431  0.0229
 3DOK7 - COMPOS       1.22 2.171513 24   0.562  0.5794
```


This is very liberal - i.e. makes is relatively easy to reject a null

SCI - Bonferroni: Multiply the p-values by $m = \binom{k}{2}$

```
## SCI for treatment comparisons clover data
> lsmeans(fit_clover,pairwise~Strain,adjust="bonferroni")
 contrast         estimate       SE df t.ratio p.value
 3DOK1 - 3DOK13      15.56 2.171513 24   7.166  <.0001
 3DOK1 - 3DOK4       14.18 2.171513 24   6.530  <.0001
 3DOK1 - 3DOK5        4.84 2.171513 24   2.229  0.5317
 3DOK1 - 3DOK7        8.90 2.171513 24   4.099  0.0062
 3DOK1 - COMPOS      10.12 2.171513 24   4.660  0.0015
 3DOK13 - 3DOK4      -1.38 2.171513 24  -0.636  1.0000
 3DOK13 - 3DOK5     -10.72 2.171513 24  -4.937  0.0007
 3DOK13 - 3DOK7      -6.66 2.171513 24  -3.067  0.0793
 3DOK13 - COMPOS     -5.44 2.171513 24  -2.505  0.2914
 3DOK4 - 3DOK5       -9.34 2.171513 24  -4.301  0.0037
 3DOK4 - 3DOK7       -5.28 2.171513 24  -2.431  0.3431
 3DOK4 - COMPOS      -4.06 2.171513 24  -1.870  1.0000
 3DOK5 - 3DOK7        4.06 2.171513 24   1.870  1.0000
 3DOK5 - COMPOS       5.28 2.171513 24   2.431  0.3431
 3DOK7 - COMPOS       1.22 2.171513 24   0.562  1.0000

P value adjustment: bonferroni method for 15 tests
```

This is conservative - if no pairs of treatments are different, then is controls α well. However, when some of the pairs are different then this method is an over correction.

FER - Holm's method: Order the p-values from smallest ($j = 1$) to largest ($j = m$).

Apply the following **step-down** scheme starting at $j = 1$:

$p_{(j)} < \alpha/(m - j + 1)$ N0 $\rightarrow$ fail to reject for all p_i for $i \geq j$

Yes $\rightarrow$ reject $H_0^{(j)}$ and go to p_{j+1}

Implemented in software?:

The p-values are adjusted as follows.

$$\tilde{p}_{(j)} = \begin{cases} mp_{(1)} & \text{for } j = 1 \\ max\{\tilde{p}_{(j-1)},\ (m - j + 1) * p_{(j)}\} & \text{for } j > 1 \end{cases}$$


```
## FER for treatment comparisons clover data
> lsmeans(fit_clover,pairwise~Strain,adjust="holm")

 contrast         estimate       SE df t.ratio p.value
 3DOK1 - 3DOK13      15.56 2.171513 24   7.166  <.0001
 3DOK1 - 3DOK4       14.18 2.171513 24   6.530  <.0001
 3DOK1 - 3DOK5        4.84 2.171513 24   2.229  0.1772
 3DOK1 - 3DOK7        8.90 2.171513 24   4.099  0.0041
 3DOK1 - COMPOS      10.12 2.171513 24   4.660  0.0012
 3DOK13 - 3DOK4      -1.38 2.171513 24  -0.636  1.0000
 3DOK13 - 3DOK5     -10.72 2.171513 24  -4.937  0.0006
 3DOK13 - 3DOK7      -6.66 2.171513 24  -3.067  0.0476
 3DOK13 - COMPOS     -5.44 2.171513 24  -2.505  0.1554
 3DOK4 - 3DOK5       -9.34 2.171513 24  -4.301  0.0027
 3DOK4 - 3DOK7       -5.28 2.171513 24  -2.431  0.1601
 3DOK4 - COMPOS      -4.06 2.171513 24  -1.870  0.2951
 3DOK5 - 3DOK7        4.06 2.171513 24   1.870  0.2951
 3DOK5 - COMPOS       5.28 2.171513 24   2.431  0.1601
 3DOK7 - COMPOS       1.22 2.171513 24   0.562  1.0000

P value adjustment: holm method for 15 tests
```

FDR - Benjamini & Hochberg: Order the p-values from smallest ($j = 1$) to largest (m).

Apply the following **step-up** scheme starting at $j = m$:

$$p_{(j)} < j\alpha/m \quad \begin{array}{ll} \text{N0} \rightarrow & \text{fail to reject } H_0^{(j)}, \text{ go to } p_{(j-1)} \\ \text{Yes} \rightarrow & \text{reject } H_0^{(i)} \text{ for } \forall\, i < j \end{array}$$

Implemented in software?:

The p-values are adjusted as follows.

$$\tilde{p}_{(j)} = \begin{cases} p_{(m)} & \text{for } j = m \\ min\{\tilde{p}_{(j+1)},\ (m/j) * p_{(j)}\} & \text{for } j < m \end{cases}$$


```
## FDR for treatment comparisons clover data

> lsmeans(fit_clover,pairwise~Strain,adjust="fdr")

 contrast         estimate       SE df t.ratio p.value
 3DOK1 - 3DOK13      15.56 2.171513 24   7.166  <.0001
 3DOK1 - 3DOK4       14.18 2.171513 24   6.530  <.0001
 3DOK1 - 3DOK5        4.84 2.171513 24   2.229  0.0483
 3DOK1 - 3DOK7        8.90 2.171513 24   4.099  0.0010
 3DOK1 - COMPOS      10.12 2.171513 24   4.660  0.0004
 3DOK13 - 3DOK4      -1.38 2.171513 24  -0.636  0.5691
 3DOK13 - 3DOK5     -10.72 2.171513 24  -4.937  0.0002
 3DOK13 - 3DOK7      -6.66 2.171513 24  -3.067  0.0113
 3DOK13 - COMPOS     -5.44 2.171513 24  -2.505  0.0343
 3DOK4 - 3DOK5       -9.34 2.171513 24  -4.301  0.0007
 3DOK4 - 3DOK7       -5.28 2.171513 24  -2.431  0.0343
 3DOK4 - COMPOS      -4.06 2.171513 24  -1.870  0.0851
 3DOK5 - 3DOK7        4.06 2.171513 24   1.870  0.0851
 3DOK5 - COMPOS       5.28 2.171513 24   2.431  0.0343
 3DOK7 - COMPOS       1.22 2.171513 24   0.562  0.5794

P value adjustment: fdr method for 15 tests
```

Randomised Block Design

In a randomised block design we have a treatment effect and a block effect. The block effect is a another categorical variable that groups the observation into similar subgroups. These subgroups are inherent in the experimental units.

A classic example of blocks is from agricultural experiments where the experimental units are plots in fields.

- The plots get different fertiliser (i.e. the treatment) added to them.
- The blocking variable may well be the field itself - normally the experiment is replicated over a number of fields. Due to inherent variability between fields with regard to their underlying baseline fertility, extra variation is being introduced into the experimental data.
- Adding a blocking variable for field, accounts for the variation and prevents it from being incorrectly measured as background variation.

The important characteristic about blocks are:

- they are of no intrinsic interest to the experimenter - their are a ‘nuisance’ variable which we want to control for.
- Blocks are not assigned to experimental units randomly - they are a characteristic of the experimental unit over which the experimenter has no control, but which may be modelled as an effect.
- The idea is to account for the variability caused by variance between the blocks by including the blocking term in the model, and prevent that identifiable source of variation being incorrectly modelled as residual variation.
- A complete randomised block design is where each treatment appears in each block once.

Example: Pain Data

An experiment was conducted to assess the performance of pain relief treatments on post operative dental pain in men. A blocking variable was recorded which was a classification of the men into one of 8 'pain tolerance groups'. It was thought that the effects of self-reporting of pain relief would be related to how much pain the subject could tolerate normally - hence the use of this block. The treatments applied were,

Treatment	Treat. No.
Control	1
Codeine	2
Acupuncture	3
Codeine and Acupuncture	4

Pain Data

PainLevel	1	2	3	4	5	6	7	8
Relief	0	0.3	0.4	0.4	0.6	0.9	1	1.2
treat	1	1	1	1	1	1	1	1
PainLevel	1	2	3	4	5	6	7	8
Relief	0.6	0.7	0.8	0.9	1.5	1.6	1.7	1.6
treat	3	3	3	3	3	3	3	3
PainLevel	1	2	3	4	5	6	7	8
Relief	0.5	0.6	0.8	0.7	1	1.4	1.8	1.7
treat	2	2	2	2	2	2	2	2
PainLevel	1	2	3	4	5	6	7	8
Relief	1.2	1.3	1.6	1.5	1.9	2.3	2.1	2.4
treat	4	4	4	4	4	4	4	4

The model will be:

$$y_i = \beta_0 + t_j + b_k + \epsilon_i$$

Putting the model in terms of the required dummy variables we get:

$$y_i = \beta_0 + \beta_1\delta_{i1} + \beta_2\delta_{i2} + \beta_3\delta_{i3} + \beta_4\delta_i$$
$$+ \beta_5\gamma_{i1} + \beta_6\gamma_{i2} + \ldots + \beta_11\gamma_{i7} + \beta_12\gamma_{i8} + \epsilon_i$$

where the δ_{ij} and γ_{ij} are dummy variables for the different treatments and blocks respectively.

Fitting this model using hte `lm(.)` function is easy. The ANOVA table for this design is the usual one and is given in this example as;

```
> pain=read.table("pain.txt",header=T)
> pain$treat=factor(pain$treat);pain$PainLevel=factor(pain$
    PainLevel);
> fit_pain=lm(Relief~treat+PainLevel,data=pain)
> anovatab(fit_pain)
       df  Sum Sq  Mean Sq  F value    Pr(>F)
Model  10  11.335   1.1335     78.4  2.52e-14
Error  21   0.304   0.0145
Total  31  11.639
```


The Null hypothesis for the overall ANOVA F-test is,
$H_0 : t_j = t_k \forall \{j, k\}$ and $b_j = b_k, \forall \{j, k\}$ - i.e. so it is a joint hypothesis testing the equality of blocks and treatments.

If you think about it this test makes little sense.

- Whatever about treatment, why would you test that all blocks are equal? You have chosen the blocks because you 'know' they are not. So, the experiment is not designed to pick up significance in the blocks anyway - this would only show what is already known/strongly suspected.
- It is the treatment effect you wish to estimate and test. So, if you get a low p-value on the overall ANOVA test you still have no idea if the significance is due to blocks or treatments - i.e. you are none the wiser.

What one needs to do is to partition the model SS into a block and treatment parts.

```
> summary(fit_pain)

Call:
lm(formula = Relief ~ treat + PainLevel, data = pain)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01875    0.07051   0.266 0.792903
treat2       0.46250    0.06013   7.691 1.54e-07 ***
treat3       0.57500    0.06013   9.562 4.22e-09 ***
treat4       1.18750    0.06013  19.748 4.83e-15 ***
PainLevel2   0.15000    0.08504   1.764 0.092304 .
PainLevel3   0.32500    0.08504   3.822 0.000994 ***
PainLevel4   0.30000    0.08504   3.528 0.001998 **
PainLevel5   0.67500    0.08504   7.937 9.34e-08 ***
PainLevel6   0.97500    0.08504  11.465 1.68e-10 ***
PainLevel7   1.07500    0.08504  12.641 2.77e-11 ***
PainLevel8   1.15000    0.08504  13.523 7.80e-12 ***

Residual standard error: 0.1203 on 21 degrees of freedom
Multiple R-squared:  0.9739,  Adjusted R-squared:  0.9615
F-statistic: 78.37 on 10 and 21 DF,  p-value: 2.521e-14
```


We do this using two GLH's and we could use the multcomp library for this with the following L matrices:

$$L_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Drop1 function

Conveniently, R has a function which can be easily used here:

```
> drop1(fit_pain,test='F')
Single term deletions

Model:
Relief ~ treat + PainLevel
          Df Sum of Sq    RSS      AIC F value    Pr(>F)
<none>                  0.3038 -127.033
treat      3    5.7363 6.0400  -37.355 132.193 8.578e-14 ***
PainLevel  7    5.5987 5.9025  -46.092  55.296 4.126e-12 ***
```

Balanced designs like this yield very easy to understand estimates of averages within treatments.


```
> lsmeans(fit_pain,~treat)
 treat lsmean        SE df  lower.CL  upper.CL
 1      0.6000 0.042521 21 0.5115727 0.6884273
 2      1.0625 0.042521 21 0.9740727 1.1509273
 3      1.1750 0.042521 21 1.0865727 1.2634273
 4      1.7875 0.042521 21 1.6990727 1.8759273

Results are averaged over the levels of: PainLevel
Confidence level used: 0.95
> by(pain$Relief,pain$treat,mean)
pain$treat: 1
[1] 0.6
----------------
pain$treat: 2
[1] 1.0625
----------------
pain$treat: 3
[1] 1.175
----------------
pain$treat: 4
[1] 1.7875
```

NB: The model based (Least Squares means) are the observed treatment means. Also, the standard errors are all the same.

```
> lsmeans(fit_pain,pairwise~treat,adjust="fdr")

 contrast estimate          SE df t.ratio p.value
 1 - 2      -0.4625 0.06013378 21  -7.691  <.0001
 1 - 3      -0.5750 0.06013378 21  -9.562  <.0001
 1 - 4      -1.1875 0.06013378 21 -19.748  <.0001
 2 - 3      -0.1125 0.06013378 21  -1.871  0.0754
 2 - 4      -0.7250 0.06013378 21 -12.056  <.0001
 3 - 4      -0.6125 0.06013378 21 -10.186  <.0001

Results are averaged over the levels of: PainLevel
P value adjustment: fdr method for 6 tests
```

NB: LS treatment difference are just difference is observed treatment averages and the standard errors of the differences are all the same.

Unbalanced Randomised Block Design

The block design becomes unbalanced when not all blocks are represented in each treatment or the replication within each block is unequal.

Example: An experiment was performed to test the effect of 3 food supplements on the weight gain in pigs. The pigs were at two different developmental stages, which were to be used as block effects. The experiment was started as a balanced design by the experimenter, but due to fatalities the resultant data was unbalanced.

The Weight Gain data is given in the table below.

Weight Gain Data

Treat	Block	Gain	Treat	Block	Gain
1	1	6.8	2	1	6.4
1	1	6.6	2	2	8.2
1	2	7.1	2	2	7.5
1	2	7.3	3	1	7.8
1	2	7.1	3	1	7.1
2	1	6.5	3	2	8.8

Incorrect Analysis

Analysis 1: One-way Analysis

```
> fit_oneway=lm(gain~factor(treat),data=wg)
> anovatab(fit_oneway)
       df  Sum Sq  Mean Sq  F value  Pr(>F)
Model   2    1.67    0.834     1.89   0.207
Error   9    3.98    0.442
Total  11    5.65
```

So, do not reject the null hypothesis.

However, if the developmental stage (i.e. the block) is a genuine phenomenon - then the error SS is being inflated by that portion of the total SS that is explainable by reference to the blocks. In that case our type II error rate is increased.

Correct analysis

Analysis 2: Randomised Block Analysis. Here we include the block effect in the model:

```
> fit_block=lm(gain~factor(block)+factor(treat),data=wg)
> anovatab(fit_block)
      df Sum Sq Mean Sq F value  Pr(>F)
Model  3   4.52   1.507    10.7 0.00355
Error  8   1.12   0.141
Total 11   5.65
```

```
Single term deletions

Model:
gain ~ factor(block) + factor(treat)
               Df Sum of Sq    RSS      AIC F value    Pr(>F)
<none>                      1.1247 -20.4092
factor(block)  1    2.8533 3.9780  -7.2495 20.2968 0.001988
factor(treat)  2    2.6020 3.7267 -10.0327  9.2545 0.008294
```


Now we reject the null hypothesis of no treatment difference in the presence of the block effect. You can also say that the null hypothesis of equal treatments was rejected having corrected for blocks.

The parameter table is:

```
summary(fit_block)

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      6.3814     0.2139  29.828 1.73e-09 ***
factor(block)2   0.9977     0.2214   4.505  0.00199 **
factor(treat)2   0.2698     0.2525   1.068  0.31651
factor(treat)3   1.1860     0.2801   4.234  0.00286 **
```

Which has the same basic interpretation as before.

Comparing Treatment Means

```
> lsmeans(fit_block,~treat)
 treat    lsmean         SE df lower.CL upper.CL
     1 6.880233 0.1691351  8 6.490206 7.270259
     2 7.150000 0.1874709  8 6.717691 7.582309
     3 8.066279 0.2195966  8 7.559888 8.572670

Results are averaged over the levels of: block
Confidence level used: 0.95

> by(wg$gain,wg$treat,mean)
wg$treat: 1
[1] 6.98
----------------
wg$treat: 2
[1] 7.15
----------------
wg$treat: 3
[1] 7.9
%$
```

The LSmeans have corrected for the imbalance in the design of the experiment - and also calculated more precise standard errors.

The LSmeans give the treatment (or block) means taking account of the other factor. For example: if I declare the mean for treatment 1 to be the arithmetic mean, i.e. 6.98 then I haven't taken into account that the block effects are involved in this mean.

The LSmean on the other hand do take into account the other factor. The LSmeans also give the population marginal means - i.e. these means assume that the imbalance in the design is due to observation missing at random - and does not reflect an underlying imbalance in the population.

Thus, they estimate marginal means for a balanced population as opposed to the unbalanced design. This can only be done using a regression approach to such analyses.

Exercise: in a spreadsheet replace the 'missing values' in the weight gain example with their LS estimates and recompute the cell means.

```
> lsmeans(fit_block,pairwise~treat,adjust="fdr")

 contrast    estimate        SE df t.ratio p.value
 1 - 2     -0.2697674 0.2524916  8  -1.068  0.3165
 1 - 3     -1.1860465 0.2801143  8  -4.234  0.0086
 2 - 3     -0.9162791 0.2887352  8  -3.173  0.0197

Results are averaged over the levels of: block
P value adjustment: fdr method for 3 tests
```


Factorial Designs

Factorials are designs that examine two or more treatments at the same time. We will look at two factors - the extension to more than 2 follows directly.

Example: Go back to the Pain data again. This is really a factorial experiment. The experimenters were interested in establishing the effect of codeine on its own, the effect of acupuncture on its own and if they **interact** with each other.

Imagine this: the effect of codeine and acupuncture on post-operative dental pain are additive - i.e. codeine reduces pain on average by 'x' amount. It will reduce pain by this amount (on average) whether or not the patient also gets acupuncture. The same reasoning applies to acupuncture.

I am an experimenter who wants to research the effects of both codeine and acupuncture. How do I do this?

(a) I can do two experiments - one with codeine and one with acupuncture and analyse each as a one-way ANOVA.

(b) If the effect of acupuncture and codeine are additive I can do two experiments for the price of one (i.e. get one free).

	No Acupuncture	Acupuncture	
No Codeine	μ_{11}	μ_{12}	$\mu_{1.}$
Codeine	μ_{21}	μ_{22}	$\mu_{2.}$
	$\mu_{.1}$	$\mu_{.2}$	$\mu_{..}$

Now I can estimates of codeine effect by testing the null hypothesis;

$$H_0 : \mu_{1.} - \mu_{2.} = 0$$

And can test the acupuncture effect by testing the mull hypothesis;

$$H_0 : \mu_{.1} - \mu_{.2} = 0$$

If we call the two factors A and B we can write the model,

$$y_i = \mu + a_j + b_k + \epsilon_i$$

which is the same as for randomised blocks but with different letters.

How do you do this?

We have already done this - fit the linear model with two classification variables/factors, just as in the randomised block designs.

The difference is that both factors are of intrinsic interest to us and under the control of the experimenter - unlike block factors which are generally not of interest and not under the experimenter‘s control.

The point is this - although the interpretation of the testing will be different the mathematics remains UNCHANGED - i.e. when factor effects are additive there is nothing new here.

Pain Data (again!)

Codeine	Acupuncture	Relief	Codeine	Acupuncture	Relief
1	1	0	1	2	0.6
1	1	0.3	1	2	0.7
1	1	0.4	1	2	0.8
1	1	0.4	1	2	0.9
1	1	0.6	1	2	1.5
1	1	0.9	1	2	1.6
1	1	1	1	2	1.7
1	1	1.2	1	2	1.6
2	1	0.5	2	2	1.2
2	1	0.6	2	2	1.3
2	1	0.8	2	2	1.6
2	1	0.7	2	2	1.5
2	1	1	2	2	1.9
2	1	1.4	2	2	2.3
2	1	1.8	2	2	2.1
2	1	1.7	2	2	2.4

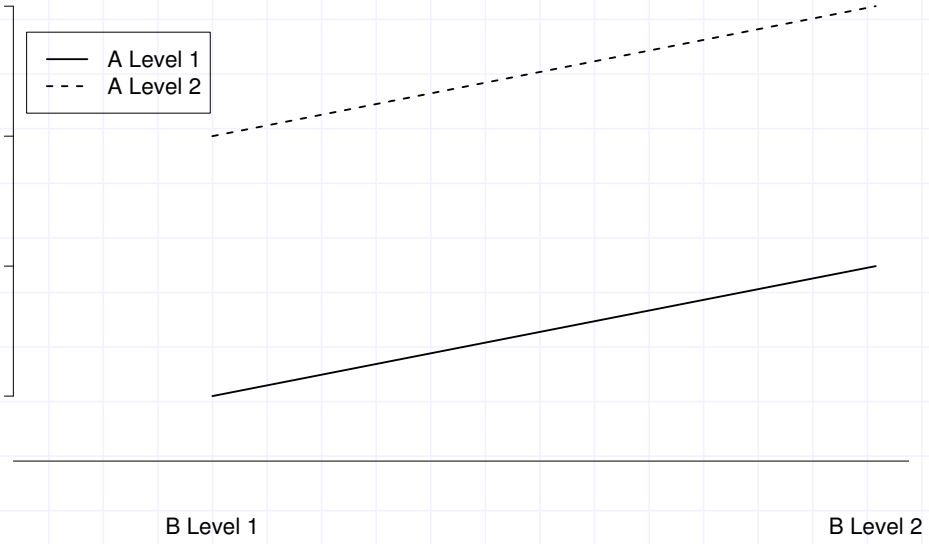
```
> pain$Codeine=factor(pain$Codeine);pain$Acupuncture=factor(
    pain$Acupuncture);
> fit_factorial1=lm(Relief~Codeine+Acupuncture,data=pain)
> anovatab(fit_factorial1)
            df Sum Sq Mean Sq F value    Pr(>F)
Model        2   5.69   2.846    13.9 5.92e-05
Error       29   5.95   0.205
Total       31  11.64

> drop1(fit_factorial1,test='F')
Single term deletions

Model:
Relief ~ Codeine + Acupuncture
            Df Sum of Sq    RSS     AIC F value    Pr(>F)
<none>                  5.9475 -47.848
Codeine      1    2.3113 8.2588 -39.343  11.270 0.0022147
Acupuncture  1    3.3800 9.3275 -35.449  16.481 0.0003401
```



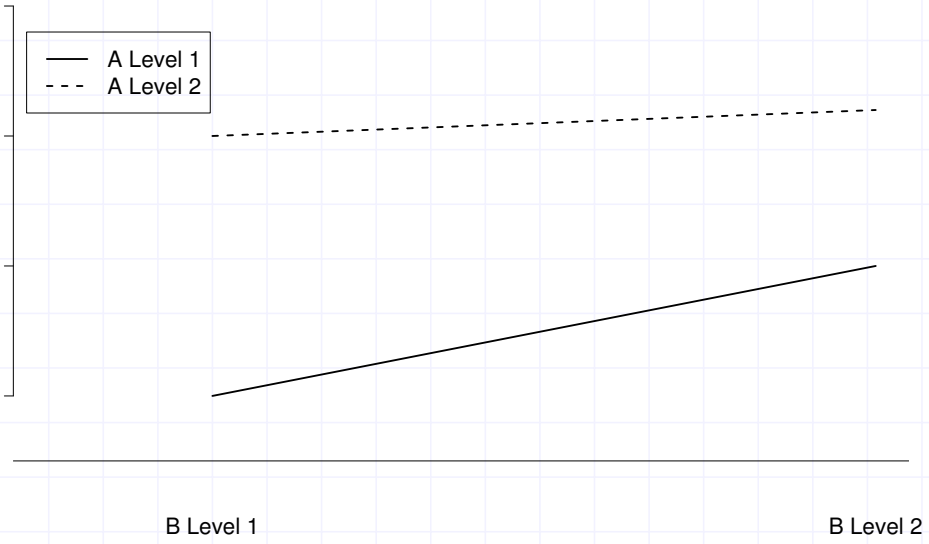
```
> summary(fit_factorial1)

Call:
lm(formula = Relief ~ Codeine + Acupuncture, data = pain)

Residuals:
    Min      1Q  Median      3Q     Max
-0.6125 -0.4031 -0.1250  0.3875  0.7000

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.5625     0.1387   4.057 0.000343 ***
Codeine2       0.5375     0.1601   3.357 0.002215 **
Acupuncture2   0.6500     0.1601   4.060 0.000340 ***


Residual standard error: 0.4529 on 29 degrees of freedom
Multiple R-squared:  0.489, Adjusted R-squared:  0.4537
F-statistic: 13.88 on 2 and 29 DF,  p-value: 5.918e-05
```

Interaction in factorial Designs

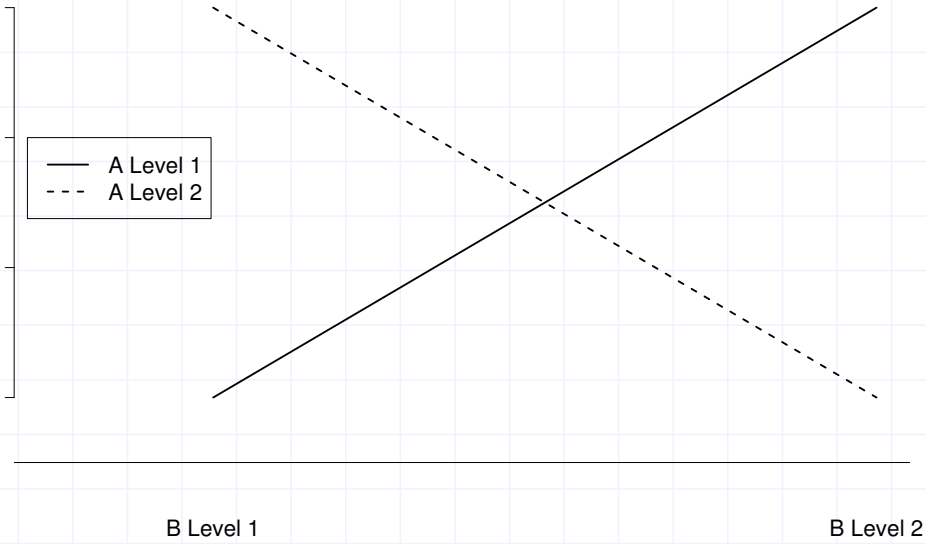
If there is no interaction we get the 'Main effects' model. The plot of the relationship gives parallel lines, e.g.:



Suppose in the above effect the assumption of additivity was not tenable. E.G. suppose that when person is getting codeine acupuncture has little or no effect - but on it's own it helps? Or maybe the reverse is true.



'Mild interaction effect'



'Strong interaction effect'

How do we test for and fit interaction?

The model becomes;

$$y_i = \mu + a_j + b_k + a_j b_k$$

Here a_j is called the **main effect** of A, b_k the main effect of B, and $a_j b_k$ is the interaction effect.

The interaction effect allows for the possibility the the main effect of one treatment may depend on the level of other treatment given.

To fit and test the interaction you need to create interaction dummy variables and test their LS parameters in the usual way.

```
> fit_factorial2=update(fit_factorial1,.~.+Codeine:
    Acupuncture)
> drop1(fit_factorial2,test='F')
Single term deletions

Model:
Relief ~ Codeine + Acupuncture + Codeine:Acupuncture
             Df Sum of Sq    RSS     AIC F value Pr(>F)
<none>                    5.9025 -46.092
Codeine
:Acupuncture  1     0.045 5.9475 -47.848  0.2135 0.6476


> summary(fit_factorial2)

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.6000     0.1623   3.696 0.000943
Codeine2                0.4625     0.2296   2.015 0.053634
Acupuncture2            0.5750     0.2296   2.505 0.018351
Codeine2:Acupuncture2   0.1500     0.3247   0.462 0.647632
```


So, the interaction effect is not significant in this case.

If there is no interaction then we get two experiments for the price of one.

If there is interaction then we need to now the nature of the interaction.

E.G. If there was an interaction effect between codeine and acupuncture then of most interest to the physician would be the nature of the interaction - does acupuncture have a greater or lesser effect in the presence of codeine? If greater then give both treatments to patients - if lesser then just give one.

Pain Data Again - What about the blocks?

```
# interaction & blocks included
> fit_factorial3=update(fit_factorial2,.~PainLevel+.)
> drop1(fit_factorial3,test='F')
Single term deletions

Model:
Relief ~ PainLevel + Codeine + Acupuncture + Codeine:
    Acupuncture
                Df Sum of Sq    RSS      AIC F value    Pr(>F)
<none>                      0.3037 -127.033
PainLevel    7    5.5988 5.9025  -46.092 55.2963 4.126e-12
Codeine
:Acupuncture 1    0.0450 0.3487 -124.612  3.1111    0.0923
```



```
> fit_factorial4=update(fit_factorial3,.~.-Codeine:
    Acupuncture)
> drop1(fit_factorial4,test='F')
Single term deletions

Model:
Relief ~ PainLevel + Codeine + Acupuncture
            Df Sum of Sq    RSS      AIC F value    Pr(>F)
<none>                    0.3487 -124.612
PainLevel    7    5.5988 5.9475  -47.848  50.455 4.273e-12
Codeine      1    2.3113 2.6600  -61.597 145.799 3.528e-11
Acupuncture  1    3.3800 3.7288  -50.789 213.219 8.428e-13
```

```
> lsmeans(fit_factorial4,pairwise~Codeine)
$lsmeans
 Codeine lsmean          SE df  lower.CL  upper.CL
 1       0.8875 0.03147645 22 0.8222218 0.9527782
 2       1.4250 0.03147645 22 1.3597218 1.4902782

Results are averaged over the levels of: PainLevel,
    Acupuncture
Confidence level used: 0.95

$contrasts
 contrast estimate          SE df t.ratio p.value
 1 - 2      -0.5375 0.04451443 22 -12.075  <.0001

Results are averaged over the levels of: PainLevel,
    Acupuncture
```



```
> lsmeans(fit_factorial4,pairwise~Acupuncture)
$lsmeans
 Acupuncture  lsmean          SE df  lower.CL  upper.CL
 1           0.83125 0.03147645 22 0.7659718 0.8965282
 2           1.48125 0.03147645 22 1.4159718 1.5465282

Results are averaged over the levels of: PainLevel, Codeine
Confidence level used: 0.95

$contrasts
 contrast estimate          SE df t.ratio p.value
 1 - 2       -0.65 0.04451443 22 -14.602  <.0001

Results are averaged over the levels of: PainLevel, Codeine
```

Interaction Example 2: Crop Data

An experiment is conducted to assess the effect of two levels of fertilisation (high and low nitrogen) and two different chemical growth promoters (1,2) on crop yield. The Crop Data is;

Yield	Fertiliser	Promoter
83	High	1
69	High	1
62	High	1
71	High	1
56	High	2
49	High	2
41	High	2
52	High	2
59	Low	1
54	Low	1
43	Low	1
50	Low	1
79	Low	2
81	Low	2
77	Low	2
77	Low	2

Analysis I: Main effects only

```
> crop=read.csv("crop.csv",header=T)
> crop$Fertiliser=factor(crop$Fertiliser);crop$Promoter=
    factor(crop$Promoter)
> fit_crop=lm(Yield~Fertiliser+Promoter,data=crop)
> drop1(fit_crop,test='F')
Single term deletions

Model:
Yield ~ Fertiliser + Promoter
           Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>                  2874.3 89.056
Fertiliser  1    85.562 2959.9 87.525  0.3870 0.5446
Promoter    1    27.563 2901.9 87.208  0.1247 0.7297
```

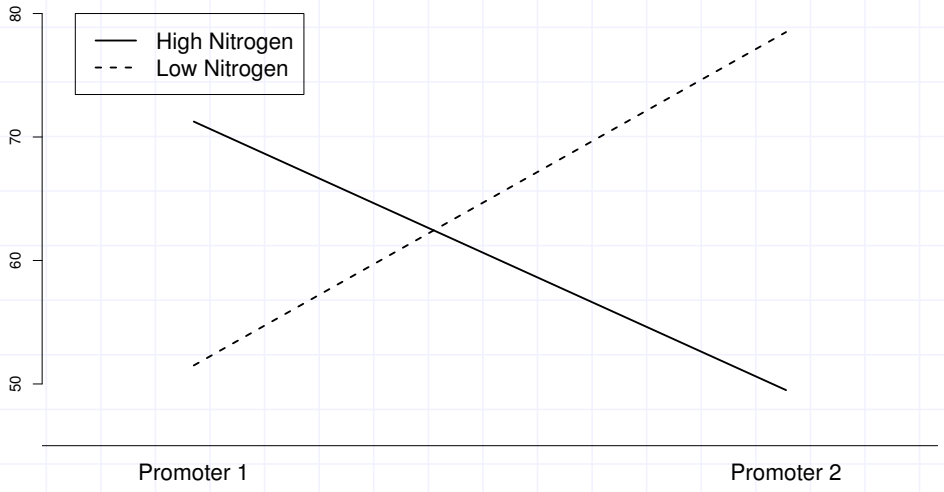
Analysis I: Interaction Model

The experimenter was interested if the growth promoters would interact with high or low levels of nitrogen.

```
> fit_crop=update(fit_crop,.~.+Fertiliser:Promoter,data=crop
    )
> drop1(fit_crop,test='F')
Single term deletions

Model:
Yield ~ Fertiliser + Promoter + Fertiliser:Promoter
             Df Sum of Sq     RSS    AIC F value    Pr(>F)
<none>                    497.75 63.000
Fertiliser
:Promoter     1    2376.6 2874.31 89.056  57.295 6.594e-06
```



```
> summary(fit_crop)

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       71.250      3.220  22.126 4.28e-11 ***
FertiliserLow    -19.750      4.554  -4.337 0.000967 ***
Promoter2        -21.750      4.554  -4.776 0.000452 ***
FertiliserLow
:Promoter2        48.750      6.440   7.569 6.59e-06 ***

Residual standard error: 6.44 on 12 degrees of freedom
Multiple R-squared:  0.8334,  Adjusted R-squared:  0.7917
F-statistic: 20.01 on 3 and 12 DF,  p-value: 5.808e-05
```



NB: There is little point in testing the main effects in the presence of such significant interaction - it would lead to very misleading results.


```
> lsmeans(fit_crop,pairwise~Fertiliser:Promoter,adjust='fdr')
$lsmeans %$
 Fertiliser Promoter lsmean       SE df lower.CL upper.CL
 High       1          71.25 3.220216 12 64.23375 78.26625
 Low        1          51.50 3.220216 12 44.48375 58.51625
 High       2          49.50 3.220216 12 42.48375 56.51625
 Low        2          78.50 3.220216 12 71.48375 85.51625

Confidence level used: 0.95

$contrasts %$
 contrast         estimate       SE df t.ratio p.value
 High,1 - Low,1      19.75 4.554073 12   4.337  0.0015
 High,1 - High,2     21.75 4.554073 12   4.776  0.0009
 High,1 - Low,2      -7.25 4.554073 12  -1.592  0.1648
 Low,1 - High,2       2.00 4.554073 12   0.439  0.6683
 Low,1 - Low,2      -27.00 4.554073 12  -5.929  0.0002
 High,2 - Low,2     -29.00 4.554073 12  -6.368  0.0002

P value adjustment: fdr method for 6 tests
```

Three factor example

The firmness of a ceramic material is possibly dependent on pressure, on temperature, and on an additive. A three factorial experiment, that includes all three factors at two levels, low/high is conducted. A randomized block design is chosen with 2 batches of ceramic material being the blocks.

		Additive 1		Additive 2	
		Block		Block	
		1	2	1	2
Pressure Low	Temp. Low	14	16	4	8
	Temp. High	7	11	24	32
Pressure High	Temp. Low	18	20	6	10
	Temp. High	9	10	26	34

Approach to three-factor analysis

- Fit full factorial with each two-way and the three-way interaction.
- If the three-way interaction is significant, then report the analysis as three two-factor experiments using the two-way interactions. In, particular, using GLHs to compare relevant treatment means is useful (if lengthy!)
- If the three-way interaction is not significant, then assess which, if any, of the two-way interaction are significant. Any significant two-way interactions are constant across the level of the third factor and can be reported as such.
- If no two-way interactions are significant, then reduce the model to the main effects model - you have three one-way analyses to report now.

Larger Factorial Models

- As the number of factors increase, there is a rapid increase in the number of treatment combinations.
- If the number of combinations stays relatively small - fit the full factorial.
- Where the number of combinations gets too big, **Fractional Factorial Models** may be considered.
- Where some factors are quantitative, the **Response Surface Models** can be used.