Lab 3

Analysis of Weight Gain Data

```
library("HSAUR3")
```

```
## Loading required package: tools
```

```
data("weightgain", package = "HSAUR3")
```

```
tapply(weightgain$weightgain, list(weightgain$source, weightgain$type), mean)
```

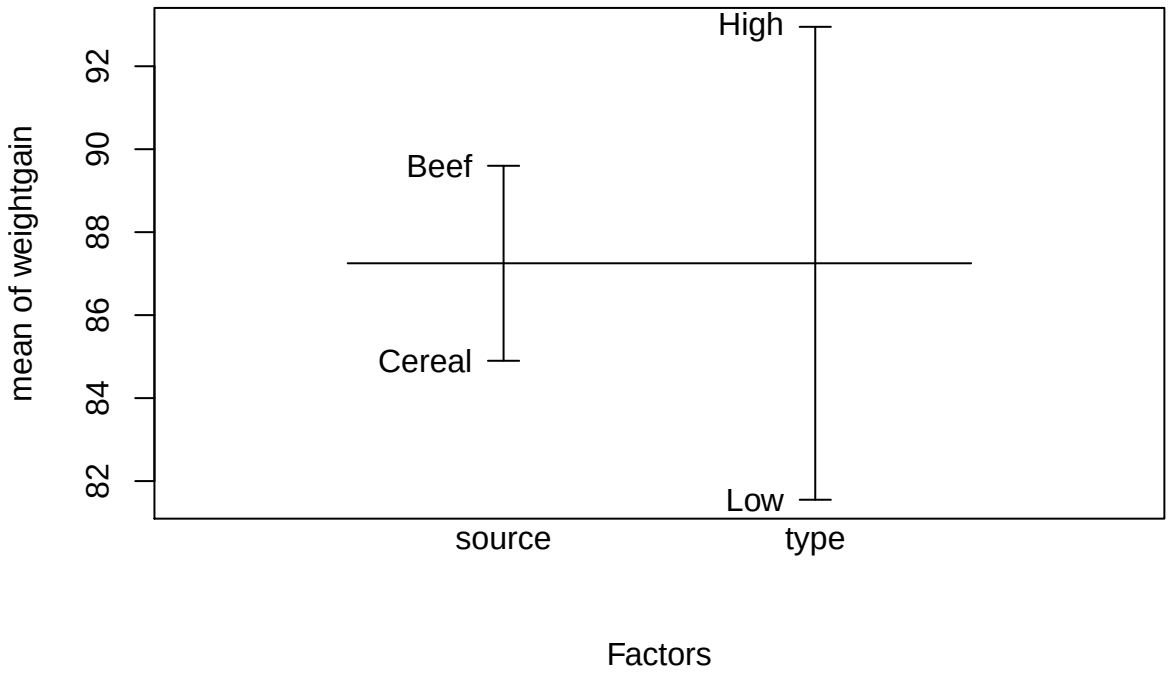
```
##         High  Low
## Beef   100.0 79.2
## Cereal  85.9 83.9
```

```
tapply(weightgain$weightgain, list(weightgain$source, weightgain$type), sd)
```

```
##            High      Low
## Beef   15.13642 13.88684
## Cereal 15.02184 15.70881
```

We see that there are only slight variations in values across different levels, so we may assume that there are no major violations of homoscedasticity.

```
plot.design(weightgain)
```



Clearly there are differences in the means of weight gain between the two levels of both covariates.

```
wg.aov <- aov(weightgain ~ source*type, data=weightgain)
summary(wg.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## source       1    221   220.9   0.988 0.3269
## type         1   1300  1299.6   5.812 0.0211 *
## source:type  1    884   883.6   3.952 0.0545 .
## Residuals   36   8049   223.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the formula for the regression model under consideration here is:

$$\text{weightgain} = \beta_0 + \beta_1 \cdot \text{sourceCereal} + \beta_2 \cdot \text{typeLow} + \beta_3 \cdot \text{sourceCereal} \cdot \text{typeLow} + \text{error}$$

where the covariates are indicator variables.

```
coef(wg.aov)
```

```
##          (Intercept)         sourceCereal              typeLow
##                100.0                -14.1                -20.8
## sourceCereal:typeLow
##                 18.8
```

Hence, $\beta_0 = 100$, $\beta_1 = -14.1$, $\beta_2 = -20.8$, $\beta_3 = 18.8$.

As for the interpretation, it makes sense to consider separate "cells" (by setting one covariate to have value 1 or 0). For instance, suppose first that sourceCereal= 0.

Then we end up with the equation

$$\text{weightgain} = \beta_0 + \beta_2 \cdot \text{typeLow} + \text{error},$$

whereas, if sourceCereal= 1, then

$$\text{weightgain} = (\beta_0 + \beta_1) + (\beta_2 + \beta_3) \cdot \text{typeLow} + \text{error}$$

We see that if sourceCereal= 1, then the intercept becomes $\beta_0 + \beta_1 = 85.9$, and the coefficient (slope) $\beta_2 + \beta_3 = -2$, compared to $\beta_0 = 100$, and $\beta_2 = -20.8$, if sourceCereal= 0. Since the covariates are indicator random variables, and the setting is balanced, their mean is 0.5, which suggests an overall increase in the mean in case sourceCereal= 0.

Similarly, if typeLow= 0, the formula becomes

$$\text{weightgain} = \beta_0 + \beta_1 \cdot \text{sourceCereal} + \text{error},$$

whereas if typeLow= 1, then

$$\text{weightgain} = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \cdot \text{sourceCereal} + \text{error}$$

Looking at the level function, we are able to see that, if typeLow=1, the interecept becomes $\beta_0 + \beta_2 = 79.2$, and the coefficient (slope) $\beta_1 + \beta_3 = 4.7$, compared to $\beta_0 = 100$ and $\beta_1 = -14.1$ otherwise, which suggests the decrease in the mean for typeLow= 1.

Regarding, the slope coefficients, note that if sourceCereal= 0, then the a unit increase in typeLow (that is if typelow=1) leads to a decrease of -20.8 in weightgain. On the other hand, if sourceCereal= 1, then a unit increase in typeLow only leads to a moderate decrease of -2. This change in the slopws also be seen in the plot below, where the change for Beef is larger than for Cereal.

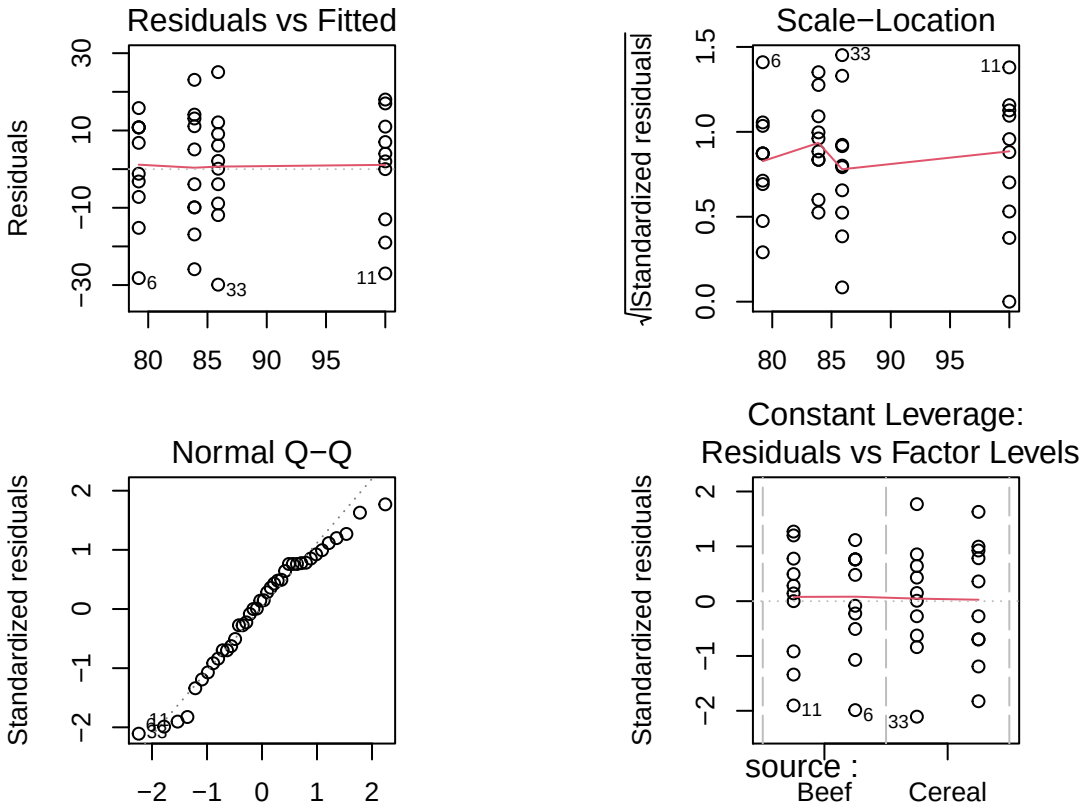
```
interaction.plot(weightgain$type, weightgain$source, weightgain$weightgain)
```

Since the lines are not parallel (different slopes), this shows the presence of interaction, which agrees with our discussion in the paragraph above.

If the interactions are significant, and the main effects are not significant, can we then drop the main effect covariates from the regression?

The answer is that you can drop the main effects, but then you would have to be careful because this completely changes the interpretation of the coefficients. Actually, droping one the main effects means that the interaction terms become effects themselves. In particular, the interaction is no longer a product of any of the effects in the model under consideration. For instance, if you were to drop, say source, from the regression, then source·type is no longer an interaction because source is not part of your main effects. In this case source·type would become an effect itself.

```
layout(matrix(1:4,ncol=2))
par(pty="s",mar=c(3,1,2,1)+0.1)
plot(wg.aov)
```



We see that there is are no major violations from the model assumptions. The mean and variance of the residuals does not seem to be varying wrt. to the fitted values, so we may say that the homoscedasticity assumption holds. Judging by the upper corner at the right of the QQ plot the residuals seem to be slightly subgaussian, but given the small sample size we may assume that the normality assumption holds.

Difference ANOVA vs Regression?

There is not really a difference between anova and regression - they are both instances of the General Linear Model. In both cases, the outcome (response) variable is continuous. In regession, the explanatory variables are continuous while in anova the explantory variables are categorical. You can also have a mix of variable types, in which you would just call the model a General Linear Model (and according to specific type, could be ancova, or analysis of covariance).

Analysis of Foster Feeding Data

We use the following code line to obtain the number of measurements per cell and then conclude that the design is not balanced.

```
tapply(foster$weight, list(foster$litgen, foster$motgen), length)
```

```
##   A B I J
## A 5 3 4 5
## B 4 5 4 2
## I 3 3 5 3
## J 4 3 3 5
```

Obviously, $2 \neq 3 \neq 4 \neq 5$, hence unbalanced design.

In a balanced (orthogonal) design, the variables are independent. Thus, no variable gives information about

the others and the order of entry into the model is irrelevant.

On the other hand, in an unbalanced (non-orthogonal) design, the sum of squares cannot be readily partitioned as in the Pythagorean theorem, leading the the issue of order of entry into the model.

And for more information than you really want, here is a response from the R FAQ: 7.18 Why does the output from anova() depend on the order of factors in the model?

In a model such as ~A+B+A:B, R will report the difference in sums of squares between the models ~1, ~A, ~A+B and ~A+B+A:B. If the model were ~B+A+A:B, R would report differences between ~1, ~B, ~A+B, and ~A+B+A:B . In the first case the sum of squares for A is comparing ~1 and ~A, in the second case it is comparing ~B and ~B+A. In a non-orthogonal design (i.e., most unbalanced designs) these comparisons are (conceptually and numerically) different.

Some packages report instead the sums of squares based on comparing the full model to the models with each factor removed one at a time (the famous 'Type III sums of squares' from SAS, for example). These do not depend on the order of factors in the model. The question of which set of sums of squares is the Right Thing provokes low-level holy wars on R-help from time to time.

There is no need to be agitated about the particular sums of squares that R reports. You can compute your favorite sums of squares quite easily. Any two models can be compared with anova(model1, model2), and drop1(model1) will show the sums of squares resulting from dropping single terms.

Now, fit the following model and obtain the summary:

```
model.1 <- aov(weight ~ litgen * motgen, data = foster)
summary(model.1)
```

```
##               Df Sum Sq Mean Sq F value  Pr(>F)
## litgen         3   60.2   20.05   0.370 0.77522
## motgen         3  775.1  258.36   4.763 0.00574 **
## litgen:motgen  9  824.1   91.56   1.688 0.12005
## Residuals     45 2440.8   54.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, variable "motgen" is significant at the level 0.01 (and hence at the level 0.05). This means there are at least two levels of variable motgen which admit significantly different means. Below, we first obtain the labels of the levels (of variable motgen) and then investigate the levels which are significantly different:

```
levels(foster$motgen)
```

```
## [1] "A" "B" "I" "J"
```

Consequently, variable motgen has four levels with labels: A, B, I ,J.

Let's move on and obtain Tukey Honest Significant Differences (TukeyHSD)

```
foster.hsd <- TukeyHSD(model.1, "motgen")
foster.hsd
```

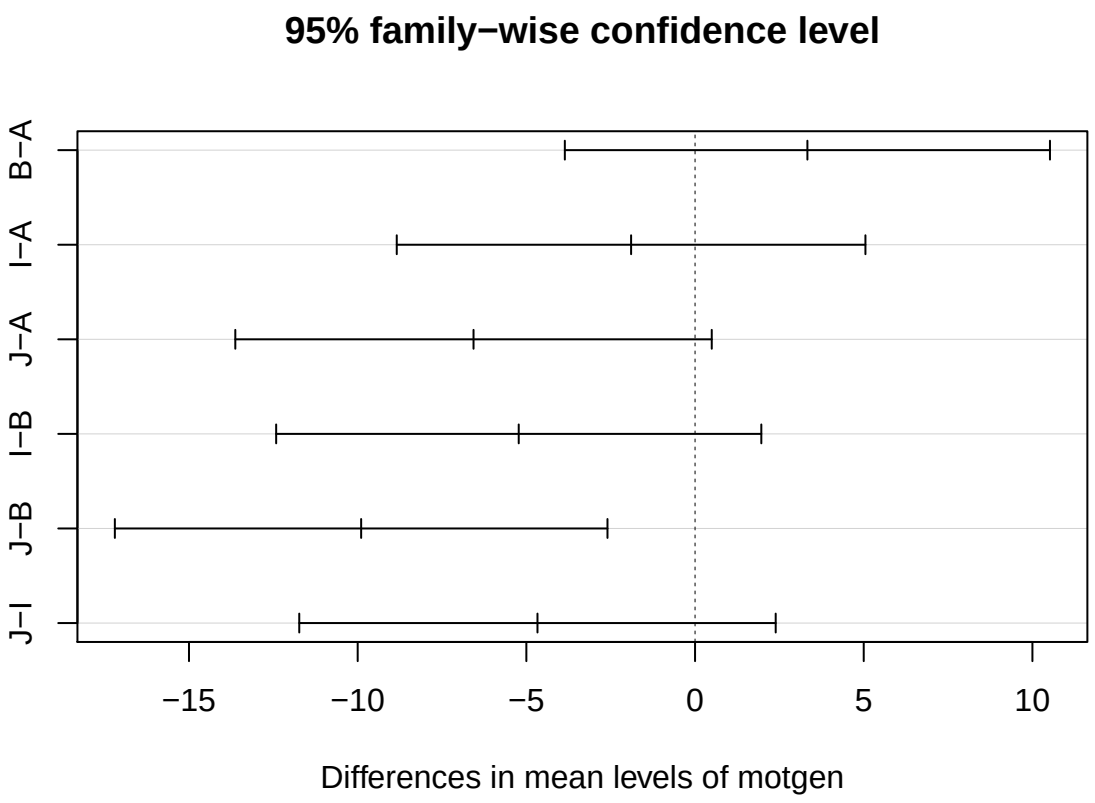
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = weight ~ litgen * motgen, data = foster)
##
## $motgen
##          diff        lwr        upr     p adj
## B-A  3.330369  -3.859729 10.5204672 0.6078581
## I-A -1.895574  -8.841869  5.0507207 0.8853702
## J-A -6.566168 -13.627285  0.4949498 0.0767540
```

```
## I-B -5.225943 -12.416041  1.9641552 0.2266493
## J-B -9.896537 -17.197624 -2.5954489 0.0040509
## J-I -4.670593 -11.731711  2.3905240 0.3035490
```

So, levels "J" and "B" are significantly different, $p - value = 0.0040509$ (< 0.05).

Following is a visualisation of the result above:

```
plot(foster.hsd)
```



Point 0 is not included in the 95% confidence interval corresponding to the difference mean "J" and "B".

Remark: Notice that the conclusion above only concerns variable "motgen". We first concluded its significancy in model.1 and then we investigated the levels of motgen which are significantly different.