Contrasts, Diagnostics, and Contrasts Again, for ANOVA Models

1 Contrasts

1.1 Pairwise comparisons of cell means

It is useful to get standard errors with the estimates of the coefficients of the ANOVA model. For the *cell means model* this is easy. The model says

$$Y_{ij} \sim N(\mu_i, \sigma^2), \text{ independent } \forall i, j$$

and hence

$$\bar{Y}_{i.} \sim N(\mu_i, \sigma^2/n_i), \text{ independent } \forall i$$

The independence makes it easy to compare pairs of cell means also:

$$\bar{Y}_{1.} - \bar{Y}_{2.} \sim N(\mu_1 - \mu_2, \sigma^2 \cdot [1/n_1 + 1/n_2])$$

and similarly with interesting linear combinations:

$$\sum_i w_i \bar{Y}_{i.} \sim N\left(\sum_i w_i \mu_i,\ \sigma^2 \sum_i w_i^2/n_i\right) \tag{1}$$

When we replace σ^2 with the usual regression estimator RMS, of course, all the $N(0,1)$'s get replaced by t_{n-k}'s.

Recall the blood coagulation example

```
> D1 _ ifelse(coag$diet=="A",1,0)
> D2 _ ifelse(coag$diet=="B",1,0)
> D3 _ ifelse(coag$diet=="C",1,0)
> D4 _ ifelse(coag$diet=="D",1,0)
> summary(lm(coag ~ D1 + D2 + D3 + D4 - 1, data=coag))

Call: lm(formula = coag ~ D1 + D2 + D3 + D4 - 1, data = coag)
Residuals:
 Min    1Q     Median   3Q Max
  -5 -1.25 -4.36e-016 1.25   5

Coefficients:
     Value Std. Error t value Pr(>|t|)
D1 61.0000  1.1832    51.5544  0.0000
D2 66.0000  0.9661    68.3165  0.0000
D3 68.0000  0.9661    70.3867  0.0000
D4 61.0000  0.8367    72.9089  0.0000

Residual standard error: 2.366 on 20 degrees of freedom
Multiple R-Squared: 0.9989
```



```
F-statistic: 4399 on 4 and 20 degrees of freedom, the p-value is 0

> table(coag$diet)
 A  B  C  D
 4  6  6  8
```

In the blood coagulation data it is interesting to think about whether $\mu_2 = \mu_3$, and perhaps whether $\mu_1 = \mu_3$. $(1-\alpha) \times 100\%$ confidence intervals for the differences in cell means are:

$$\begin{aligned}
\mu_2 - \mu_3 : \quad & 66 - 68 \pm t_{24-4,1-\alpha/2} \cdot \sqrt{(2.366)^2 \cdot (1/6 + 1/6)} \\
\text{or} \quad & -2.00 \pm t_{20,1-\alpha/2} \cdot 1.37 \\
\mu_1 - \mu_3 : \quad & 61 - 68 \pm t_{24-4,1-\alpha/2} \cdot \sqrt{(2.366)^2 \cdot (1/4 + 1/6)} \\
\text{or} \quad & 7 \pm t_{20,1-\alpha/2} \cdot 1.53
\end{aligned}$$

For $\alpha = 0.95$, the first interval contains zero but not the second; inverting these intervals to make hypothesis tests we would not reject $\mu_2 = \mu_3$, but we would reject $\mu_2 = \mu_3$.

There are two things to note about this example: one is nomenclature and the other is a warning.

- **Contrasts:** The parameters (and their estimates) $\mu_i - \mu_j$ are called *contrasts*: they contrast the values of two different cell means.

  As ANOVA and linear models developed, the meaning of the word *contrast* has also expanded: nowadays, a *contrast* is **any** linear combination of the cell means $L = \sum_i w_i \mu_i$ where the coefficients sum to zero, $\sum_i w_i = 0$.

  Equation (1) lets us estimate and test hypotheses about any contrast L.

  Sets of contrasts are sometimes used to reparameterize an ANOVA model to make it easier to work with numerically (that is the point of the "Helmert contrasts" that you may have read about in the SPLUS Guide to Statistics). In cases like this, the parameters or "effects" of the reparametrized model are also contrasts. This is confusing, but that's the way the nomenclature works.

- The analysis we illustrated with the contrasts $\mu_2 - \mu_3$ and $\mu_1 - \mu_3$ was flawed in two ways:

  1. **Capitalizing on chance:** We looked at the data (sample cell means) to decide which contrasts to consider. This is bad news, since the distribution theory that leads to the $N(0,1)$ and t_{n-k} distributions above depend on *not* having looked at the data yet for their validity. (Nevertheless this is a common *exploratory* device. Any results obtained this way would need to be confirmed in an independent replication of the experiment, however.)
  2. **Multiple Comparisons:** We didn't make any adjustment for the fact that we calculated two confidence intervals (or performed two hypothesis tests) instead of one. This is the *multiple inference* or *multiple comparisons* problem. Just because each interval individually has confidence 95% (say). If we are contemplating making m inferences $A_i, \ldots, A_m$ (say, $A_i = \{i^{th}$ parameter stays in its CI$\}$), then

     $$P[\cap_i A_i] = \sum_i P[A_i] - \sum_i \sum_j P[A_i \cup A_j] + \sum_i \sum_j \sum_k P[A_i \cap A_j \cap A_k] \pm \cdots \tag{2}$$

     so it would seem pretty surprising for $P[\cap_i A_i] = P[A_{i_0}],\ \forall\, i_0$.

     There are several common methods of correcting for multiple comparisons

- In the *Bonferroni* method, we take $P[A_i] = 1 - \alpha/m$ in order to get $P[\cap_i A_i] > 1 - \alpha$;
- *Scheffé's* method, which assumes the A_i's are CI's for all possible contrasts $L = \sum_i w_i \mu_i$ with $\sum_i w_i = 0$; we replace the two-sided upper-tail $t_{n-k,\alpha/2}$ cutoff with a $\sqrt{k' F_{k',n-k,\alpha}}$ cutoff.
- The *Tukey*, or *Tukey-Kramer* method, which assumes the A_i's are CI's for all pairwise comparisons $\mu_i - \mu_{i'}$. See Neter, Wasserman & Kutner, for details.
- *Fisher's least significant difference (LSD)* simply reports the uncorrected t intervals(!!!)

Naturally one wants to choose the method that leads to the narrowest intervals, but also has a defensible confidence statement. If you are just making a few comparison's, Bonferroni's method often leads to the shortest intervals. If you are making lots of comparisons, the Scheffé or Tukey methods lead to shorter intervals, with Scheffé being best for real data-snooping.

1.2 Customized Contrasts in Splus

The following example is taken from an experiment of Friendly and Franklin (1980) on the effects of presentation format on learning and memory. The subjects were asked to read a list of 40 words. Then, after performing a distracting task, they were asked to recall as many of the words as possible. This was repeated for each of five trials. Thirty subjects were randomly assigned to each of three conditions:

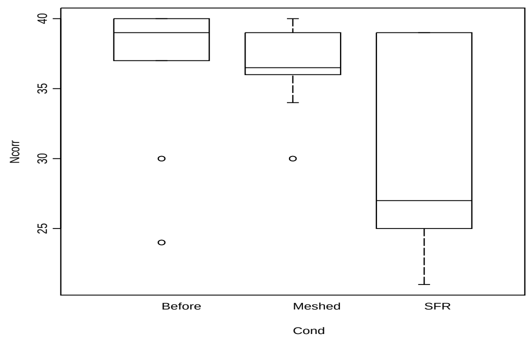
SFR, or "standard free recall", in which the order of presentation of the 40 words was randomized on each trial;

Before, in which the recalled words were presented as a group before the forgotten ones, on each trial; and

Meshed, in which the recalled and forgotten words were interspersed.

The data recorded was the number of words, out of 40, recalled on the fifth and last trial.

```
> friendly _ read.table("friendly.dat",row.names=1)
> names(friendly) _ c("Cond","Ncorr")
> cbind(friendly[1:15,],friendly[16:30,])  # just to save some space...
     Cond   Ncorr   Cond   Ncorr
 1    SFR      39 Before      24
 2    SFR      25 Before      30
 3    SFR      37 Before      39
 4    SFR      25 Before      40
 5    SFR      29 Before      40
 6    SFR      39 Meshed      40
 7    SFR      21 Meshed      39
 8    SFR      39 Meshed      34
 9    SFR      24 Meshed      37
10    SFR      25 Meshed      40
11 Before      40 Meshed      36
12 Before      38 Meshed      36
13 Before      39 Meshed      38
14 Before      37 Meshed      36
15 Before      39 Meshed      30
> plot.factor(friendly,style.bxp="old")
```

The boxplots tell us to expect some differences between the three experimental groups, but they also suggest some violations of the statistical assumptions: note the left skewing in the "Before" and "Meshed" conditions. In this case the problem is due to a "cieling effect" which is often seen in testing data: the maximum number-right score is 40 here, and in the "Before" and "Meshed" conditions, many subjects are getting all or nearly all the words right. If we had more than 40 words to test with, we might see a more symmetric number-right distribution.

Ignoring these problems[1] for now and forging ahead, a simple ANOVA fit and table tells us that the conditions matter, but this is hardly surprising given the boxplots.

```
> summary(aov(Ncorr ~ Cond,data=friendly))
          Df Sum of Sq  Mean Sq  F Value      Pr(F)
     Cond  2     264.6 132.3000 4.340868 0.02319478
Residuals 27     822.9  30.4778
```

Putting on our psychologist's hat, we might be simultaneously interested in the following two questions:

- Is mean recall the same under all three conditions?

$$\begin{aligned} H_0 &: \quad \mu_1 = \mu_2 = \mu_3 \\ H_1 &: \quad \text{not all equal} \end{aligned}$$

  For technical reasons that will be explained later, we will modify this first hypothesis test to

$$\begin{aligned} H_0 &: \quad \mu_1 = (\mu_2 + \mu_3)/2 \\ H_1 &: \quad \mu_1 \neq (\mu_2 + \mu_3)/2 \end{aligned}$$

- Is mean recall the same under the **Meshed** and **Before** conditions?

$$\begin{aligned} H_0 &: \quad \mu_2 = \mu_3 \\ H_1 &: \quad \mu_2 \neq \mu_3 \end{aligned}$$

[1] As we saw earlier in the course, we might be able to fix the problem by applying a logit or probit transformation to Ncorr/40.

Note that if the second H_0 is true, then either of the first H_0's implies the other; if the second H_0 fails, then so does the original first H_0.

A way of understanding contrasts is that they are linear combinations of parameters that sum to zero under the null hypotheses of interest. In our case,

$$\begin{aligned}\text{First (modified) } H_0 \quad &\Leftrightarrow \quad \mu_1 - \frac{1}{2}\mu_2 - \frac{1}{2}\mu_3 = 0 \\ \text{Second } H_0 \quad &\Leftrightarrow \quad 0\mu_1 + \mu_2 - \frac{1}{2}\mu_3 = 0\end{aligned}$$

We can express this as a "contrast matrix",

Group	C_1	C_2
SFR	1	0
B	$-\frac{1}{2}$	1
M	$-\frac{1}{2}$	-1

The theory of linear regression allows us to construct F-tests and t-tests for linear contrasts of cell means, directly. To make interpretation of the tests as simple as possible, use the following rules in constructing contrasts:

- For a factor with m levels, you need to construct $m-1$ contrasts (columns of the contrast matrix);
- The coefficients in each contrast must sum to zero (i.e. orthogonal to the intercept term);
- The contrasts must be orthogonal to each other (dot product of any two columns equals zero).

The contrast matrix above satisfies these conditions.

To test contrasts in Splus is fairly straightforward. We can build the contrast matrix directly

```
> Cmatrix _ matrix(scan(),ncol=2,byrow=T)
1:  1 0
3: -0.5 1
5: -0.5 -1
7:
> Cmatrix
     [,1] [,2]
[1,]  1.0    0
[2,] -0.5    1
[3,] -0.5   -1
```

Note that the rows of the contrast matrix, SFR, B, M, are not in alphabetical order. Splus always tries to deal with factor levels in sorted or alphabetical order; so we modify the contrast matrix as follows:

```
> unique(friendly$Cond)
[1] SFR    Before Meshed
> sort(.Last.value)
[1] Before Meshed SFR
```



```
> Cmatrix _ Cmatrix[c(2,3,1),] # or: Cmatrix[order(unique(friendly$Cond)),]
> Cmatrix
     [,1] [,2]
[1,] -0.5    1
[2,] -0.5   -1
[3,]  1.0    0
```

Now we can fit the model using the `contrasts=...` parameter of `aov()`. Note that I am specifically requesting a `lm()`-style summary, though, to be sure I can examine the contrasts separately.

```
> summary.lm(aov(Ncorr ~ Cond,data=friendly,contrasts=list(Cond=Cmatrix)))

Call: aov(formula = Ncorr ~ Cond, data = friendly, contrasts =
          list(Cond = Cmatrix))
Residuals:
   Min     1Q Median  3Q Max
 -12.6 -4.625    0.9 3.4 8.7

Coefficients:
               Value Std. Error  t value Pr(>|t|)
(Intercept)  34.5000   1.0079    34.2285   0.0000
      Cond1  -4.2000   1.4254    -2.9465   0.0065
      Cond2   0.0000   1.2345     0.0000   1.0000

Residual standard error: 5.521 on 27 degrees of freedom
Multiple R-Squared: 0.2433
F-statistic: 4.341 on 2 and 27 degrees of freedom, the p-value is 0.02319

Correlation of Coefficients:
      (Intercept) Cond1
Cond1 0
Cond2 0           0
```

We see that

- Our first hypothesis, $H_0: \mu_1 = (\mu_2 + \mu_3)/2$, was pretty reasonably rejected;
- Our second hypothesis, $H_0: \mu_2 = \mu_3$, was not rejected; indeed the "coefficient" estimate is zero!

What is going on with these contrasts and how can we interpret the coefficient estimates? To see this we expose a little more of the "by-hand" coding of contrasts.

```
> # Note that the dummies are created in order to match the
> # rows of Cmatrix
> D1 _ ifelse(friendly$Cond=="Before",1,0)
> D2 _ ifelse(friendly$Cond=="Meshed",1,0)
> D3 _ ifelse(friendly$Cond=="SFR",1,0)
> tmp _ cbind(D1,D2,D3)%*%Cmatrix  # matrix multiplication...
```

```
> C1 _ tmp[,1]
> C2 _ tmp[,2]
> summary(lm(Ncorr ~ C1 + C2, data=friendly))

Call: lm(formula = Ncorr ~ C1 + C2, data = friendly)
Residuals:
   Min     1Q Median  3Q Max
 -12.6 -4.625    0.9 3.4 8.7

Coefficients:
               Value Std. Error  t value Pr(>|t|)
(Intercept)  34.5000   1.0079    34.2285   0.0000
         C1  -4.2000   1.4254    -2.9465   0.0065
         C2   0.0000   1.2345     0.0000   1.0000

Residual standard error: 5.521 on 27 degrees of freedom
Multiple R-Squared: 0.2433
F-statistic: 4.341 on 2 and 27 degrees of freedom, the p-value is 0.02319

Correlation of Coefficients:
   (Intercept) C1
C1 0
C2 0           0
> anova(lm(Ncorr ~ C1 + C2, data=friendly))
Analysis of Variance Table

Response: Ncorr

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
       C1  1     264.6 264.6000 8.681735 0.0065485
       C2  1       0.0   0.0000 0.000000 1.0000000
Residuals 27     822.9  30.4778
```

Two things to note here:

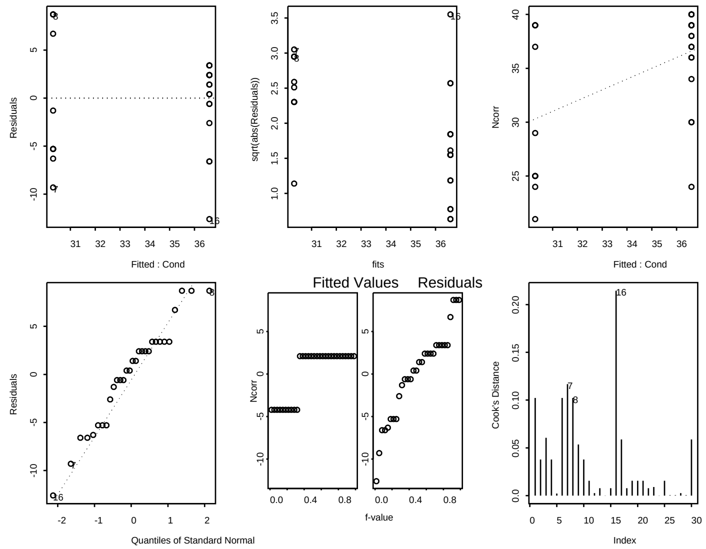
- You can see from the way I constructed these regressions that contrasts are closely related to reparametrizations of the linear regression model. The "coeffiecient estimates" that `aov()` calculated with the `contrast=...` parameter were just the coefficients of the redefined dummy variables C1 and C2.
- The new `summary()` output tells us the same thing as before. But the new `anova()` output is telling us something interesting! All of $RegSS$ is concentrated in C1, the contrast for testing $\mu_1 = (\mu_2 + \mu_3)/2$; indeed the SS for C2 (testing $\mu_2 = \mu_3$) is zero! How[2] can this be??

[2] Try fitting the cell-means model `Ncorr ~ D1 + D2 + D3 - 1`, for example.

2 Diagnostics and Remediations for ANOVA models

2.1 Graphical inspection of residuals

The diagnostic plots that one gets from plotting an aov or lm object in SPLUS give visual checks for most of the problems one could encounter in linear regression and ANOVA modeling:



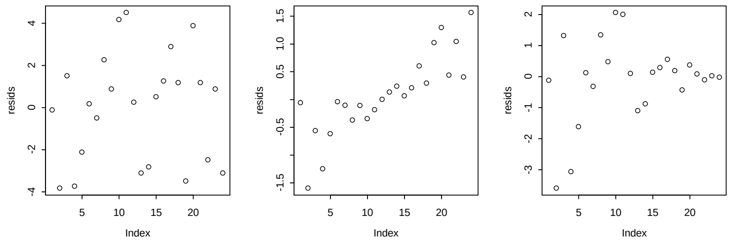
Working from top left to lower right, these plots tell us:

- *First two plots: Equal error variance in all cells?* Fitted vs. residuals and fitted vs. sqrt(|residuals|).
- *Third plot: Nonlinearity (linear models) or non-equal error variances?* Fitted vs. observed.
- *Fourth plot: Normal errors?* Normal Quantile plot of residuals.
- *Fifth (double) plot: High r^2?* Normal quantile plots of fitted values, compared with normal quantile plot of residuals.
- *Sixth plot: high leverage or influence?* Cook's distances.

An additional plot can be helpful: In the very first plot, "fitted vs residuals", it is sometimes helpful to plot residuals not against the fitted values, but rather against some potentially important aspect of the data collection process, such as

- Time order observations were taken in;
- Spatial order (e.g. in an agricultural experiment, distance from the ag. station);
- etc.

What we are looking for are trends and/or serial correlations, in these plots. For example, here are three plots of residuals in time order.



- The plot on the left is a fine-looking residual plot.
- The middle plot shows a distressing time-trend in the residuals; it is probably best to begin working with the experimenter to find out what caused the trend, and recollect data (if possible) that does not have the trend. If this is not possible, then perhaps including time as a regressor in the model, or building a time-series model for the residuals, is needed.
- The plot on the right shows great variability in the residual variance, across time. Again, working with the experimenter to find the reason and correct it is the best course of action. Second best is to try some sort of transformation to fix the problem.

2.2 Variance-stabilizing transformations and Box-Cox transformations

Referring back to a "fitted vs residuals" plot can be helpful in selecting the right transformation, when unequal variances are discovered in a residual plot:

- *If σ_i^2 is proportional to μ_i*, try
$$y^* = \sqrt{y} \quad \text{or} \quad y^* = \sqrt{y} + \sqrt{y+1}$$
- *If σ_i is proportional to μ_i*, try
$$y^* = \log y$$
- *If σ_i is proportional to μ_i^2*, try
$$y^* = 1/y$$

- *If y is a proportion*, so that $\sigma_i^2 = \pi_i(1-\pi_i)/n_i$, try
$$y^* = 2\arcsin\sqrt{y}$$

The idea behind each of these is the following. Suppose $\sigma^2_{y_{ij}} = g(\mu_i)$, and we wish to choose a "good" transformation $h(y)$. By taking variances on both sides of the Taylor approximation

$$h(y) - h(\mu) \approx h'(\mu)(y-\mu)$$

one can show that the variance after a transformation $y^* = h(y)$ is $\sigma^2_{y^*_{ij}} \approx [h'(\mu_i)]^2 g(\mu_i)$ (this is the essence of an important general technique called the *delta method*). We wish to

$$\text{choose } h(\cdot) \text{ so that } [h'(\mu_i)]^2 g(\mu_i) = k, \text{ a constant},$$

i.e. the transformed random variable y^* has constant variance k. This is a differential equation; the solutions, for $g(\mu) = \mu$, $\sqrt{(\mu)}$, etc., are listed above.

Finally,

- It might also be desirable to estimate, by hand or using Venables and Ripley's `boxcox()` function, the Box-Cox transformation
$$y^*_\lambda = (y^\lambda - 1)/\lambda$$
for the problem. The idea is to estimate λ along with the linear regression parameters, by maximum likelihood, in the model
$$y^*_\lambda = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k + \epsilon$$
A power transformation, such as the Box-Cox transformation, is also a standard way to try to correct non-normality in residuals.

After any of these transformations, the model should be re-fit and new diagnostics should be carried out, to ensure that the transformation helped and not hindered the solution of the problem.

2.3 Formal tests for equality of variances

There also exist formal tests of equality of variances in the cells of an ANOVA model. In general, visual inspection as in the previous section is adquate for many purposes.

As an example, we present the *Bartlett test*, which works for all ANOVA models, whether the number of observations per cell is constant or not; however it is fairly sensitive to non-normality in the residuals. When the number of observations per cell is constant, the *Hartley test* can be used.

The Bartlett Test. Let s_i^2 denote the sample variance of the residuals in the i^{th} cell of the data, $i = 1, \ldots, k$, each with degrees of freedom $df_i = n_i - 1$; and total degrees of freedom $df_T = \sum df_i$. Then the RMS can be expressed as an arithmetic mean

$$RMS = RSS/df_T = \frac{1}{df_T}\sum_{i=1}^k df_i s_i^2$$

On the other hand, the geometric mean is

$$GRMS = [(s_1^2)^{df_1}(s_2^2)^{df_2}\cdots(s_k^2)^{df_k}]^{1/df_T}$$

and the arithmetic-geometric mean inequality states that

$$GRMS \le RMS.$$

Bartlett showed that if the y_{ij} are all independent $N(\mu_i, \sigma^2)$ (same variance in all cells) then

$$B = \frac{df_T}{C}(\log RMS - \log GRMS) = \frac{1}{C}\left[df_T \log MSE - \sum_{i=1}^{k} df_i \log s_i^2\right]$$

is approximately χ^2_{k-1}, where

$$C = 1 + \frac{1}{3(k-1)}\left[\left(\sum_{i=1}^{k}\frac{1}{df_i}\right) - \frac{1}{df_T}\right].$$

This provides a test (large $B \Rightarrow$ reject H_0) of the hypothesis

$$H_0: \quad \sigma_1^2 = \cdots = \sigma_k^2,\ vs.$$
$$H_1: \quad \text{not all } \sigma_i^2 \text{ are equal.}$$

Here is a short program in SPLUS to implement Bartlett's test, for one-way ANOVA.

```
bartlett.test _ function(y,x) {
#
# bartlett
#
# y is the response
# x is the factor
#
        ys _ split(y,x)
        vars _ sapply(ys,var)
        df _ sapply(ys,length)-1
        dfT _ sum(df)
        RMS _ sum(df*vars)/dfT
        k _ length(unique(x))
        const _ 1 + 1/(3*(k-1)) * ( sum(1/df) - 1/dfT )
        B _ (dfT*log(RMS) - sum(df*log(vars))) / const
        return(B=B,pval=1-pchisq(B,k-1))
}
```

Bartlett's test works for more complicated ANOVA problems also, but the SPLUS code above would have to be modified somehow to do this. For the Friendly and Franklin verbal recall data that we looked at before we find

```
> bartlett.test(friendly$Ncorr,friendly$Cond)
$B:    [1] 6.021124
$pval: [1] 0.04926398
```

suggesting that there are marginally detectable differences between the variances of the three cells, SFR, Before, and Meshed.

The Hartley test The Hartley test is based on the ratio

$$H = \frac{\max(\sigma_1^2, \ldots, \sigma_k^2)}{\min(\sigma_1^2, \ldots, \sigma_k^2)}$$

and simply compares this to the distribution of H under the null hypothesis

$$H_0: \quad \sigma_1^2 = \cdots = \sigma_k^2$$

The distribution of H under H_0 has been tabulated, but it is also easy to simulate. Here is a short SPLUS program that implements the Hartley test, using a simulated p-value.

```
hartley.test _ function(y,x,reps=1000) {
#
# hartley.test
#
# y is the response
# x is the factor
#
# The simulation part of this is pretty slow!
#
        ys _ split(y,x)
        vars _ sapply(ys,var)
        ns _ sapply(ys,length)
        if (sum(abs(diff(ns)))!=0)
                stop("Cell sample sizes must be equal!")
        n _ ns[1]
        k _ length(unique(x))
        H _ max(vars)/min(vars)
        cnt _ 0
        for(i in 1:reps) {
                vars _ sapply(split(rnorm(n*k),x),var)
                if (max(vars)/min(vars)>H)
                        cnt _ cnt+1
        }
        return(H=H,pval=cnt/reps,reps=reps)
}
```

Like the SPLUS code for the Bartlett test, this is really only set up for one-way ANOVA; it would be fairly easy to adapt to other ANOVA models as well. Also, since the p-value is computed using simulation, this test could be adapted to unequal sample sizes in the cells (this adaptation would be much harder to do if we had to explicitly tabulate the distribution of H for "all" combinations of cell sample sizes).

Trying this again on the Friendly and Franklin verbal recall data we obtain

```
> table(friendly$Cond)  # verify equal number of ob's per cell
 Before Meshed SFR
     10     10  10
> hartley.test(friendly$Ncorr,friendly$Cond)
```

```
$H:    [1] 5.875
$pval: [1] 0.036
$reps: [1] 1000
```

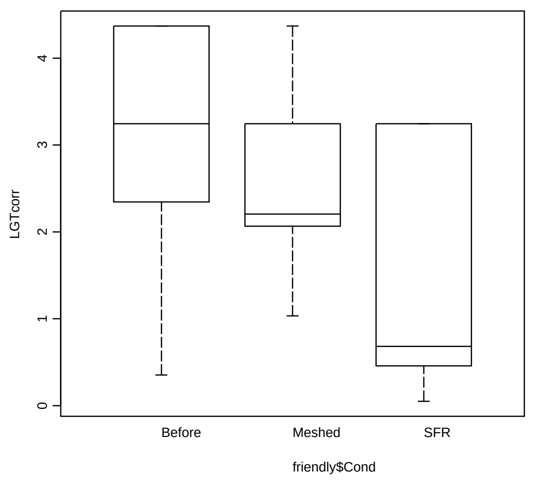
It appears that in this case the Hartley test is a bit more powerful to detect differences in cell variances than the Bartlett test. I believe this is true in general.

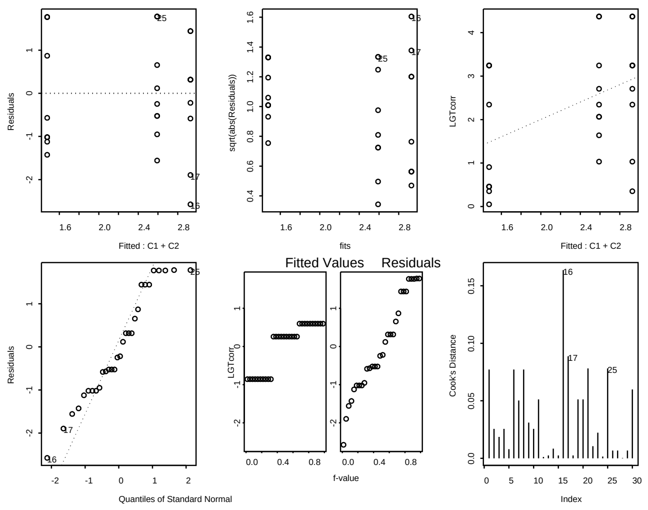
Earlier we remarked that the Friendly and Franklin verbal recall data would more nearly satisfy the statistical linear regression assumptions (primarily, normality within cells), if we performed a logit or probit transformation. A probit transformation just applies the `qnorm()` function to `friendly$Ncorr/40`. We will illustrate a `logit` transformation, which has nearly the same effect (recall that the `logit` is the inverse CDF of the *logistic distribution*, which has pdf $f(x) = e^x/(1+e^x)^2$:

```
> logit _ function(x) { log(x/(1-x)) }
> LGTcorr _ logit((friendly$Ncorr-.5)/40) # some subjects got 40/40 right...
> plot.factor(friendly$Cond,LGTcorr,style.bxp="old")
> bartlett.test(LGTcorr,friendly$Cond)
$B:    [1] 0.5168412
$pval: [1] 0.7722703

> hartley.test(LGTcorr,friendly$Cond)
$H:    [1] 1.559951
$pval: [1] 0.787
$reps: [1] 1000
```

Clearly, the remaining differences in cell variances do not matter very much. Finally, we briefly illustrate ANOVA for this data using the logit transformation

```
> summary(aov(LGTcorr ~ Cond,data=friendly))
          Df Sum of Sq  Mean Sq  F Value      Pr(F)
     Cond  2  11.54367 5.771837 3.462951 0.04583918
Residuals 27  45.00196 1.666739
> anova(lm(LGTcorr ~ C1 + C2))
Analysis of Variance Table

Response: LGTcorr

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
       C1  1  10.97334 10.97334 6.583719 0.0161595
       C2  1   0.57033  0.57033 0.342183 0.5634290
Residuals 27  45.00196  1.66674
> plot(lm(LGTcorr ~ C1 + C2))
```



3 Regression Equations for ANOVA; Reparametrization

We have seen that the "dummy variables" linear regression version of an ANOVA model is always over-parametrized. To see how ANOVA reparametrizations overcome this, we now examine the one-way ANOVA model from a linear regression point of view.

For simplicity, consider a one-way ANOVA problem for a factor with three levels, a, b and c, and two observations per cell. Let

$$X_1 = \begin{cases} 1 & \text{at level } a \\ 0 & \text{else} \end{cases} \qquad X_2 = \begin{cases} 1 & \text{at level } b \\ 0 & \text{else} \end{cases} \qquad X_3 = \begin{cases} 1 & \text{at level } c \\ 0 & \text{else} \end{cases}$$

so the linear regression model is

$$y_{ij} = \theta_0 + X_{1j}\theta_1 + X_{2j}\theta_2 + X_{3j}\theta_3 + \epsilon_{ij} \tag{3}$$

or in matrix form

$$y = \begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{bmatrix} = X\theta + \epsilon = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} + \begin{bmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{bmatrix}$$

Basically we want to know how to introduce a linear constraint, to get a new set of parameters that can be estimated uniquely (technical term: the new parameters are *identifiable*).

3.1 Matrix algebra details

We want to consider reparametrizations of the model $y = X\theta + \epsilon$, that is, we want to consider new sets of parameters

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = C \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \tag{4}$$

where

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix}.$$

Our goal is to choose a new design matrix R so that

$$y = X\theta + \epsilon = Z\beta + \epsilon$$

i.e. the two regression problems are exactly the same (more formally: X and Z have the same column space). Note that, by (4), $R\beta = ZC\theta$, so it must be that

$$X \;=\; ZC.$$

Now, we would like to solve this by "multiplying on the right by C^{-1}", but unfortunately C is not square, so it doesn't have an inverse. Instead, notice what happens if we multiply on the right by $C'(CC')^{-1}$ (this is called a "generalized inverse"):

$$\begin{aligned} X &= ZC \\ X \cdot C'(CC')^{-1} &= ZC \cdot C'(CC')^{-1} \\ X \cdot C'(CC')^{-1} &= Z \end{aligned}$$

so now we have a way of finding Z from X and C, namely $Z = XC'(CC')^{-1}$.

To summarize, our goal in understanding reparametrization is:

Step 1. Figure out what "constraint(s)" are being imposed on the θ's to get rid of the linear dependency;

Step 2. Figure out how to write that as something like $\beta = C\theta$;

Step 3. Compute the design matrix for the β's as $Z = XC'(CC')^{-1}$.

Step 4. The new regression is then $y = Z\beta + \epsilon$.

Two notes before going on:

- In many simple problems, you can intuitively get from **Step 1** to **Step 4** without fooling around with the C matrix in **Steps 2** and **3**.
- After **Step 4** it is often helpful to write the columns of Z in terms of the 0/1 dummy variables X_1, X_2, X_3, and re-expressing the cell means μ_1, μ_2 and μ_3 in terms of the new parameters.

3.2 Some illustrations

Cell means model (delete the first column). This is easy. Setting $\theta_0 = 0$ is the same as deleting the first column in the X matrix; the matrix form of the regression becomes

this becomes

$$\begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} + \begin{bmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{bmatrix}$$

and (3) becomes

$$y_{ij} = X_{1j}\theta_1 + X_{2j}\theta_2 + X_{3j}\theta_3 + \epsilon_{ij}$$

and it is easy to see in the resulting model, $\mu_1 = \theta_1$, $\mu_2 = \theta_2$ and $\mu_3 = \theta_3$, the cell means model.

Let's see how we would do this by setting up the C matrix and calculating $Z = XC'(CC')^{-1}$:

We want $\theta_0 = 0$ and

$$\mu_1 \;=\; \theta_1$$

$$\begin{aligned}\mu_2 &= \theta_2 \\ \mu_3 &= \theta_3\end{aligned}$$

or

$$\begin{bmatrix}\mu_1\\\mu_2\\\mu_3\end{bmatrix} = \begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\theta_0\\\theta_1\\\theta_2\\\theta_3\end{bmatrix} = C\theta$$

Hence

$$Z = XC'(CC')^{-1} = \begin{bmatrix}1&1&0&0\\1&1&0&0\\1&0&1&0\\1&0&1&0\\1&0&0&1\\1&0&0&1\end{bmatrix}\begin{bmatrix}0&0&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix} = \begin{bmatrix}1&0&0\\1&0&0\\0&1&0\\0&1&0\\0&0&1\\0&0&1\end{bmatrix}$$

which is the same design matrix that we got "by eye" above.

Delete the last column. Some computer packages set $\theta_3 = 0$ instead of $\theta_0 = 0$. If we set $\theta_3 = 0$, this is like deleting the last column in the matrix formulation.

$$\begin{bmatrix}y_{11}\\y_{12}\\y_{21}\\y_{22}\\y_{31}\\y_{32}\end{bmatrix} = \begin{bmatrix}1&1&0\\1&1&0\\1&0&1\\1&0&1\\1&0&0\\1&0&0\end{bmatrix}\begin{bmatrix}\theta_0\\\theta_1\\\theta_2\end{bmatrix} + \begin{bmatrix}\epsilon_{11}\\\epsilon_{12}\\\epsilon_{21}\\\epsilon_{22}\\\epsilon_{31}\\\epsilon_{32}\end{bmatrix}$$

and (3) becomes

$$y_{ij} = \theta_0 + X_{1j}\theta_1 + X_{2j}\theta_2 + \epsilon_{ij}$$

We can also do this by setting up C and computing $Z = XC'(CC')^{-1}$. We want to set $\theta_3 = 0$ and to define

$$\begin{bmatrix}\gamma_0\\\gamma_1\\\gamma_2\end{bmatrix} = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}\theta_0\\\theta_1\\\theta_2\\\theta_3\end{bmatrix} = C\theta$$

It then follows that

$$Z = XC'(CC')^{-1} = \begin{bmatrix}1&1&0&0\\1&1&0&0\\1&0&1&0\\1&0&1&0\\1&0&0&1\\1&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&0\end{bmatrix} = \begin{bmatrix}1&1&0\\1&1&0\\1&0&1\\1&0&1\\1&0&0\\1&0&0\end{bmatrix}$$

as expected. What do the parameters mean here?

$$\begin{aligned}\mu_1 &= \gamma_0 + \gamma_1 \\ \mu_2 &= \gamma_0 + \gamma_2 \\ \mu_3 &= \gamma_0.\end{aligned}$$

Hence, $\gamma_0 = \mu_3$ is the mean of the last cell, and γ_1 and γ_2 are the differences between the third cell and the first and second (respectively).

Grand mean plus treatment effects. If we set the constraint

$$\theta_1 + \theta_2 + \theta_3 = 0$$

then we can express $\theta_3 = -\theta_1 - \theta_2$ and (3) becomes

$$y_{ij} = \theta_0 + X_{1j}\theta_1 + X_{2j}\theta_2 - X_{3j}(\theta_1 + \theta_2) + \epsilon_{ij}$$

or equivalently

$$y_{ij} = \theta_0 + (X_{1j} - X_{3j})\theta_1 + (X_{2j} - X_{3j})\theta_2 + \epsilon_{ij} \tag{5}$$

and the matrix form becomes

$$\begin{bmatrix}y_{11}\\y_{12}\\y_{21}\\y_{22}\\y_{31}\\y_{32}\end{bmatrix} = \begin{bmatrix}1&1&0\\1&1&0\\1&0&1\\1&0&1\\1&-1&-1\\1&-1&-1\end{bmatrix}\begin{bmatrix}\theta_0\\\theta_1\\\theta_2\end{bmatrix} + \begin{bmatrix}\epsilon_{11}\\\epsilon_{12}\\\epsilon_{21}\\\epsilon_{22}\\\epsilon_{31}\\\epsilon_{32}\end{bmatrix}$$

Now let's try setting up C and $Z = XC'(CC')^{-1}$:

Step 1. This parametrization can be thought of as doing two things: (a) we set $\theta_0 = 0$; and (b) we define:

$$\begin{aligned}\mu &= (\theta_1 + \theta_2 + \theta_3)/3 \\ \alpha_1 &= \theta_1 - \mu = \frac{2}{3}\theta_1 - \frac{1}{3}\theta_2 - \frac{1}{3}\theta_3 \\ \alpha_2 &= \theta_2 - \mu = -\frac{1}{3}\theta_1 + \frac{2}{3}\theta_2 - \frac{1}{3}\theta_3 \\ \alpha_3 &= \theta_3 - \mu = -\frac{1}{3}\theta_1 - \frac{1}{3}\theta_2 + \frac{2}{3}\theta_3\end{aligned}$$

(as a check, note that $\alpha_1 + \alpha_2 + \alpha_3 = 0$ as desired).

Step 2. This means that $\alpha_3 = -\alpha_1 - \alpha_2$, so we can omit it from the reparametrization, and so C is the matrix in the transformation

$$\begin{bmatrix}\mu\\\alpha_1\\\alpha_2\end{bmatrix} = \begin{bmatrix}0&\frac{1}{3}&\frac{1}{3}&\frac{1}{3}\\0&\frac{2}{3}&-\frac{1}{3}&-\frac{1}{3}\\0&-\frac{1}{3}&\frac{2}{3}&-\frac{1}{3}\end{bmatrix}\begin{bmatrix}\theta_0\\\theta_1\\\theta_2\\\theta_3\end{bmatrix} = C\theta$$

Step 3. Therefore

$$Z = XC'(CC')^{-1} = \begin{bmatrix}1&1&0&0\\1&1&0&0\\1&0&1&0\\1&0&1&0\\1&0&0&1\\1&0&0&1\end{bmatrix}\begin{bmatrix}0&0&0\\1&1&0\\1&0&1\\1&-1&-1\end{bmatrix} = \begin{bmatrix}1&1&0\\1&1&0\\1&0&1\\1&0&1\\1&-1&-1\\1&-1&-1\end{bmatrix}$$

so the new model is

$$y = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \mu \\ \alpha_1 \\ \alpha_2 \end{bmatrix} + \epsilon$$

Step 4. Now if we define

$$Z_1 = X_1 - X_3 = \begin{cases} 1 & \text{at level } a \\ 0 & \text{at level } b \\ -1 & \text{at level } c \end{cases} \qquad Z_2 = X_2 - X_3 = \begin{cases} 0 & \text{at level } a \\ 1 & \text{at level } b \\ -1 & \text{at level } c \end{cases}$$

we may express the model as

$$y_{ij} = \mu + Z_{1j}\alpha_1 + Z_{2j}\alpha_2 + \epsilon_{ij} \;=\; \mu + (X_1 - X_3)\alpha_1 + (X_2 - X_3)\alpha_2 + \epsilon_{ij}$$

In this model, the cell means are

$$\begin{aligned} \mu_1 &= \mu + \alpha_1 \\ \mu_2 &= \mu + \alpha_2 \\ \mu_3 &= \mu + \alpha_3 \end{aligned}$$

so that the α's really are offsets from the grand mean μ to each cell mean.

SPLUS's default "Helmert contrasts". The default parametrization for ANOVA models in SPLUS is in terms of the "Helmert contrasts." It is easiest to explain how to set this model up directly in terms of C and $Z = XC'(CC')^{-1}$:

Step 1. Like the "grand mean plus treatment effects" parametrization we do two things: (a) we set $\theta_0 = 0$; and (b) we define:

$$\begin{aligned} \mu &= (\theta_1 + \theta_2 + \theta_3)/3 \\ \beta_1 &= \{-\theta_1 + \theta_2\}/2 \\ \beta_2 &= \{-(\theta_1 + \theta_2)/2 + \theta_3\}/3 \\ \beta_3 &= \{-(\theta_1 + \theta_2 + \theta_3)/3 + \theta_4\}/4 \end{aligned}$$

(Here, $\theta_4 \equiv 0$ [since there is no fourth category] and is just included to show you the pattern, should you ever have to construct Helmert contrasts for more than three categories).

Step 2. Again, β_3 is expressible in terms of the other parameters ($\beta_3 = 3\mu$), so we omit it from the reparametrization, and C is the matrix in the transformation

$$\begin{bmatrix} \mu \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & -\frac{1}{6} & -\frac{1}{6} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = C\theta$$

Step 3. Therefore

$$Z = XC'(CC')^{-1} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \\ 1 & 0 & 2 \\ 1 & 0 & 2 \end{bmatrix}$$

so the new model is

$$y = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \\ 1 & 0 & 2 \\ 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} \mu \\ \beta_1 \\ \beta_2 \end{bmatrix} + \epsilon$$

Step 4. Now if we define

$$Z_1 = X_2 - X_1 = \begin{cases} -1 & \text{at level } a \\ 1 & \text{at level } b \\ 0 & \text{at level } c \end{cases} \qquad Z_2 = 2X_3 - X_2 - X_1 = \begin{cases} -1 & \text{at level } a \\ -1 & \text{at level } b \\ 2 & \text{at level } c \end{cases}$$

we may express the model as

$$y_{ij} = \mu + Z_{1j}\beta_1 + Z_{2j}\beta_2 + \epsilon_{ij} \;=\; \mu + (X_2 - X_1)\beta_1 + (2X_3 - X_2 - X_1)\beta_2 + \epsilon_{ij}$$

In this model, the cell means are

$$\begin{aligned} \mu_1 &= \mu - \beta_1 - \beta_2 \\ \mu_2 &= \mu + \beta_1 - \beta_2 \\ \mu_3 &= \mu + 2\beta_2 \end{aligned}$$

Note that it is easy to read off the "treatment effects" also:

$$\begin{aligned} \alpha_1 &= -\beta_1 - \beta_2 \\ \alpha_2 &= \beta_1 - \beta_2 \\ \alpha_3 &= 2\beta_2 \end{aligned}$$