Chapter 5

Contrasts

1 An experiment with two factors

The following small data set was used by BHH = Box et al. (1978, Section 7.7; 8.1 in 2nd ed.). They borrowed the data from Box and Cox (1964, Section 4), a paper that contains an extensive discussion about how and why to transform data before feeding them into a standard analysis.

> *"[The data set] gives the survival times of animals in a 3 x 4 factorial experiment, the factors being (a) three poisons and (b) four treatments. Each combination of the two factors is used for four animals, the allocation to animals being completely randomized."*

BHH were using the data set to explain the virtues of data transformation. They argued that it was better to make a least squares fit for `rate = 1/time`, rather than fitting `time` itself. My purpose is different. I want to show you how different reparametrizations affect the output. The reparametrization is achieved by using different forms of ***contrasts*** for the factor variables.

2 The data set

```
BC <- read.table("boxcox.data", header=T,sep="\t")
BC$rate <- 1/BC$time # transformation suggested by BHH page 235
# poison and treatment are factors
print(BC[c(1,5,17,24:25,48),],digits=3)

##    time poison treatment rate
## 1  0.31      I         A 3.23
## 5  0.36     II         A 2.78
## 17 0.92     II         B 1.09
## 24 0.29    III         B 3.45
## 25 0.43      I         C 2.33
## 48 0.33    III         D 3.03
```

A friendlier view of the data:

```
##        A    B    C    D
## I   0.31 0.82 0.43 0.45
## I   0.45 1.10 0.45 0.71
## I   0.46 0.88 0.63 0.66
## I   0.43 0.72 0.76 0.62
## II  0.36 0.92 0.44 0.56
## II  0.29 0.61 0.35 1.02
## II  0.40 0.49 0.31 0.71
## II  0.23 1.24 0.40 0.38
## III 0.22 0.30 0.23 0.30
## III 0.21 0.37 0.25 0.36
## III 0.18 0.38 0.24 0.31
## III 0.23 0.29 0.22 0.33
```

Here are four attempts at fitting a linear model (unorthodox notation):

<5.1> $$\mathbb{E}\,\mathit{time}(\mathit{poison} = i, \mathit{treatment} = j) = m + \alpha_i + \beta_j$$

<5.2> $$\mathbb{E}\,\mathit{time}(\mathit{poison} = i, \mathit{treatment} = j) = m + \alpha_i + \beta_j + \gamma_{i,j}$$

<5.3> $$\mathbb{E}\,\mathit{rate}(\mathit{poison} = i, \mathit{treatment} = j) = m + \alpha_i + \beta_j$$

<5.4> $$\mathbb{E}\,\mathit{rate}(\mathit{poison} = i, \mathit{treatment} = j) = m + \alpha_i + \beta_j + \gamma_{i,j}$$

The $\gamma_{i,j}$ terms are called interactions.

The more tradition way to express the model would be to write $y_{i,j,k}$ for the response (such as `time`) for the kth replicate ($k = 1, \ldots, 4$) under the ith poison ($i = I, II, III$) and the jth treatment ($j = A, B, C$). For example, one would then rewrite <5.1> as

<5.5>
$$y_{i,j,k} = m + \alpha_i + \beta_j + \xi_{i,j,k},$$

where $\xi_{i,j,k}$ is an unobservable random error.

The four models correspond to the **R** commands:

```
out1 <- lm(time ~ poison + treatment, data=BC)
out2 <- lm(time ~ poison * treatment, data=BC) # interactions
out3 <- lm(rate ~ poison + treatment, data=BC)
out4 <- lm(rate ~ poison * treatment, data=BC) # interactions
```

For the moment I'll put up with whatever reparametrization **R** has used. Here is the summary for <5.1>:

```
## lm(formula = time ~ poison + treatment, data = BC)
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.25170 -0.09625 -0.01490  0.00000  0.06177  0.49830
## Rsquared:  0.65
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.452      0.056   8.088    0.000
## poisonII      -0.073      0.056  -1.308    0.198
## poisonIII     -0.341      0.056  -6.102    0.000
## treatmentB     0.363      0.065   5.614    0.000
## treatmentC     0.078      0.065   1.213    0.232
## treatmentD     0.220      0.065   3.407    0.001
```

Actually, for both factors **R** has used treatment contrasts, which seems to be the default for my copy of the program: see `options()$contrasts`. You can see from the summary that `poisonI` and `treatmentA` are missing from the list of coefficients. The constraint applied to <5.1> was $\alpha_1 = \beta_1 = 0$.

From now on, to conserve space, I'll show only an abbreviated part of the summary: the call, the coefficients, and their estimated standard errors. For example, the display for `out1` would be reduced to:

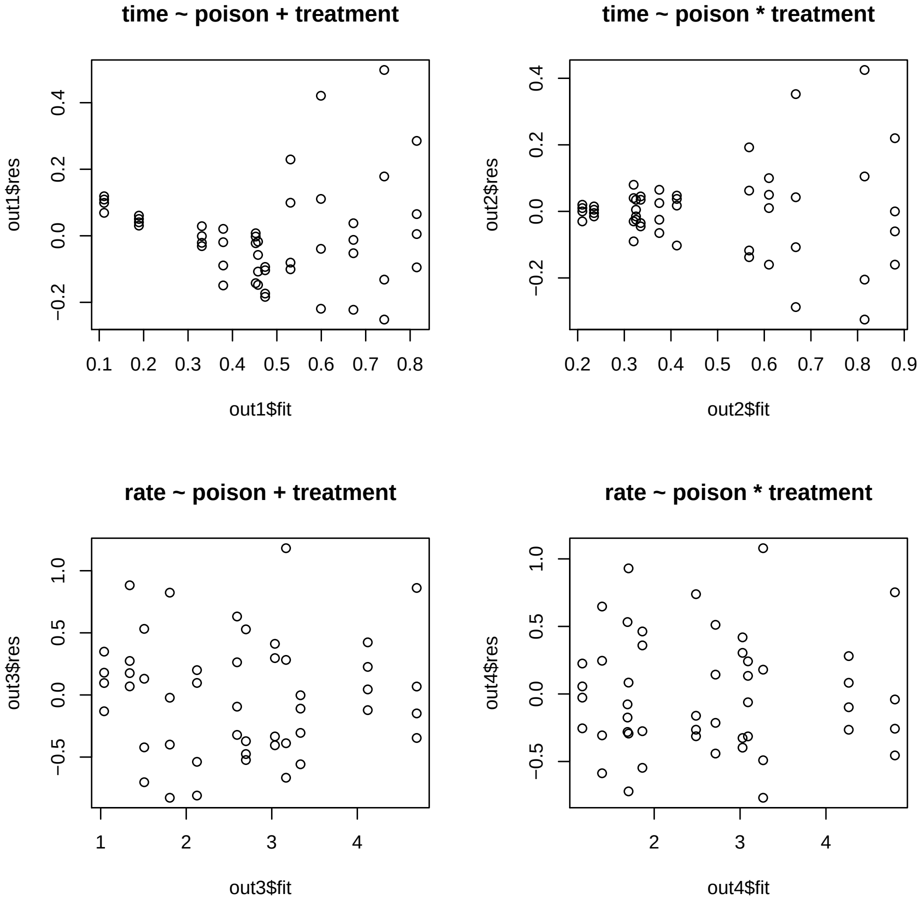
```
## lm(formula = time ~ poison + treatment, data = BC)
##        (Int)    pII   pIII    tB    tC    tD
## Est    0.452 -0.073 -0.341 0.363 0.078 0.220
## StdErr 0.056  0.056  0.056 0.065 0.065 0.065
```

The summary for `out4` would be

```
## lm(formula = rate ~ poison * treatment, data = BC)
##         (Int)  pII pIII    tB    tC    tD pII:tB pIII:tB pII:tC pIII:tC
## Est      2.49 0.78 2.32 -1.32 -0.62 -0.80  -0.55   -0.45   0.07   0.086
## StdErr   0.24 0.35 0.35  0.35  0.35  0.35   0.49    0.49   0.49   0.490
##         pII:tD pIII:tD
## Est      -0.77   -0.91
## StdErr    0.49    0.49
```

The coefficient labelled `pII:tB` corresponds to the interaction parameter $\gamma_{i,j}$ with $i = II$ and $j = B$.

The plots of residuals against fitted values certainly suggest that the assumption of constant variance is hard to believe for the `time` variable:



Notice the strange pattern in the first two residual plots.

The interaction term slightly improves the additive fit for `time`, but not by much. The fit with interaction terms for `rate` also shows only a slight improvement over the additive fit.

From now on I'll ignore the issue of how to choose a transformation and focus on the interpretation of the coefficients for `lm(rate∼)`.

3 Interpretation of coefficients for one factor

Let me start with the simpler case of just one factor, first (`out5`) with the intercept explicitly excluded and then (`out6`) with an intercept.

```
## lm(formula = rate ~ -1 + treatment, data = BC)
##         tA  tB  tC  tD
## Est    3.5 1.9 2.9 2.2
## StdErr 0.3 0.3 0.3 0.3
```

```
## lm(formula = rate ~ 1 + treatment, data = BC)
##        (Int)   tB   tC   tD
## Est      3.5 -1.7 -0.6 -1.4
## StdErr   0.3  0.4  0.4  0.4
```

The factor `BC$treatment` implicitly defines four dummy predictors, $\mathbb{1}_A$, $\mathbb{1}_B$, $\mathbb{1}_C$, and $\mathbb{1}_D$. The dummy $\mathbb{1}_A$ contains a 1 wherever `BC$treatment` contains an A and zeros elsewhere, and so on. In **R** you could manufactor `dummyT` $= \mathbb{1}_{\text{treat}} = (\mathbb{1}_A, \mathbb{1}_B, \mathbb{1}_C, \mathbb{1}_D)$ by

```
dummyT <- outer(BC$treat,levels(BC$treat),"==")+0
# the zero converts from Boolean to numeric
dimnames(dummyT)[[2]] <- levels(BC$treatment)
dummyT[c(1:2,13:14,25:26,37:38),]

##      A B C D
## [1,] 1 0 0 0
## [2,] 1 0 0 0
## [3,] 0 1 0 0
## [4,] 0 1 0 0
## [5,] 0 0 1 0
## [6,] 0 0 1 0
## [7,] 0 0 0 1
## [8,] 0 0 0 1
```

If I only showed rows $1, 13, 25, 37$ you would see I_4, the 4×4 identity matrix. Why?

Both least squares fits use the same 4-dimensional model space $\mathcal{X}$ spanned by the columns of the 48×5 matrix

<5.6>
$$X = \text{span}(\mathbb{1}, \mathbb{1}_A, \mathbb{1}_B, \mathbb{1}_C, \mathbb{1}_D).$$

The dimension is only 4 because $\mathbb{1} = \mathbb{1}_A + \mathbb{1}_B + \mathbb{1}_C + \mathbb{1}_D$.

3.1 Without intercept, contrasts not used

The `out5` model matrix X_5 is $\mathbb{1}_{\text{treat}} =$ `dummyT` itself. The generic element z of $\mathcal{X}$ is expressed as $z = b_A\mathbb{1}_A + b_B\mathbb{1}_B + b_C\mathbb{1}_C + b_D\mathbb{1}_D$. The least squares estimates $\widehat{b}_A, \dots, \widehat{b}_D$ are just the means over responses at the same level of the `BC$treatment` factor.

```
print( rbind(out6$coeff,tapply(BC$rate,BC$treatment,mean)),digits=4)

##      (Intercept) treatmentB treatmentC treatmentD
## [1,]       3.519     -1.657    -0.5721     -1.358
## [2,]       3.519      1.862     2.9472      2.161
```

3.2 With intercept, treatment contrasts

The default contrasts for the factor `BC$treatment` are treatment. (Unfortunate clash of names in there.) The model matrix for `out6` is $X_6 = (\mathbb{1}, \mathbb{1}_B, \mathbb{1}_C, \mathbb{1}_D)$. The generic z in $\mathcal{X}$ is expressed as $z = a_0\mathbb{1} + a_2\mathbb{1}_B + a_3\mathbb{1}_C + a_4\mathbb{1}_D$. There is a one-to-one correspondence between the column vector $b = [b_A, b_B, b_C, b_D]$ and the column vector $a = [a_0, a_2, a_3, a_4]$ for which

$$b_A\mathbb{1}_A + b_B\mathbb{1}_B + b_C\mathbb{1}_C + b_D\mathbb{1}_D = z = a_0\mathbb{1} + a_2\mathbb{1}_B + a_3\mathbb{1}_C + a_4\mathbb{1}_D,$$

namely $b_A = a_0$ and $b_B = a_0 + a_2$ and $b_C = a_0 + a_3$ and $b_D = a_0 + a_4$. In matrix terms $b = K_6 a$ where K_6 is the matrix

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    0    1    0
## [4,]    1    0    0    1
```

In particular, $\widehat{b} = K_6\widehat{a}$ and $\text{var}(\widehat{b}) = K_6\text{var}(\widehat{a})K_6^T$. The `summary.lm()` function, which gets called when we ask for `summary()` of an lm object, actually calculates the estimated matrix of variances and covariances for the estimated coefficients.

```
V6 <- summary(out6)$cov
ahat <- out6$coeff
newbhat <- K6 %*% ahat
V5 <- K6 %*% V6 %*% t(K6)
both <- rbind( BC.coeff(out5),t(newbhat), stderr=sqrt(diag(V5)))

## lm(formula = rate ~ -1 + treatment, data = BC)
##         tA  tB  tC  tD
## Est    3.5 1.9 2.9 2.2
## StdErr 0.3 0.3 0.3 0.3

print(both,digits=3)

##           tA    tB    tC    tD
## Est    3.519 1.862 2.947 2.161
## StdErr 0.292 0.292 0.292 0.292
##        3.519 1.862 2.947 2.161
## stderr 0.289 0.289 0.289 0.289
```

A little bit of round-off error in there.

3.3 With intercept, Helmert contrasts

Now add another column to `BC`, a copy of `BC$treatment` with its constrasts changed from the default to Helmert. You won't see any difference between `BC$treatment` and `BC$Htreatment` if you just print out `BC`, but it does make a difference to the coefficients for the least squares fit. You will also see a difference in the contrast matrices:

```
BC$Htreatment <- C(BC$treatment,helmert)
C6 <- contrasts(BC$treatment)
C7 <- contrasts(BC$Htreatment)
cbind(C6,C7)

##   B C D
## A 0 0 0 -1 -1 -1
## B 1 0 0  1 -1 -1
## C 0 1 0  0  2 -1
## D 0 0 1  0  0  3
```

Each of `out5`, `out6`, and `out7` give the same fitted vector and the same residuals. Only the parametrization for $\mathcal{X}$ changes.

```
out7 <- lm(rate ~ 1+ Htreatment, BC)
summary(cbind(out5$res,out6$res,out7$res))
```

```
##        V1                  V2                  V3          
##  Min.   :-1.6314   Min.   :-1.6314   Min.   :-1.6314  
##  1st Qu.:-0.7443   1st Qu.:-0.7443   1st Qu.:-0.7443  
##  Median :-0.2581   Median :-0.2581   Median :-0.2581  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 0.8479   3rd Qu.: 0.8479   3rd Qu.: 0.8479  
##  Max.   : 2.0362   Max.   : 2.0362   Max.   : 2.0362  
```

Write $\mathcal{L} = \{A, B, C, D\}$ for the levels of the factor `BC$treatment`. By construction, the columns of the contrast matrix C_7 form a basis for the subspace of $\mathbb{R}^4$ orthogonal to the column vector $d = [1, 1, 1, 1]$. Every member of that subspace must be of the form $C_7 h$ for some h in $\mathbb{R}^3$. Conceptually, `out7` is representing a vector z in $\mathcal{X}$ as Xf with $[f_A, f_B, f_C, f_D]$ orthogonal to d and X as in <5.6>. That is, for some h in $\mathbb{R}^3$,

$$
\begin{aligned}
z = (\mathbb{1}, \mathbb{1}_{\text{treat}})f &= f_0\mathbb{1} + \sum\nolimits_{t\in\mathcal{L}} f_t \mathbb{1}_t \qquad \text{with } \sum\nolimits_{t\in\mathcal{L}}^{4} f_t = 0 \\
&= (\mathbb{1}, \mathbb{1}_{\text{treat}}) \begin{pmatrix} 1 & 0 \\ 0 & C_7 \end{pmatrix} \begin{pmatrix} f_0 \\ h \end{pmatrix} \\
&= X_7 \begin{pmatrix} f_0 \\ h \end{pmatrix} \qquad \text{where } X_7 = (\mathbb{1}, \mathbb{1}_{\text{treat}} C_7).
\end{aligned}
$$

That is

$$
z = X_7 g \qquad \text{where } g = \begin{pmatrix} f_0 \\ h \end{pmatrix} \text{ and } f = K_7 g \qquad \text{with } K_7 = \begin{pmatrix} 1 & 0 \\ 0 & C_7 \end{pmatrix}.
$$

You can check that the `out7` least squares fit does use the 48×4 model matrix X_7, by verifying that `model.matrix(out7)` $- X_7 = 0$.

R identifies the components of $\widehat{g}$ by the names `(Intercept)`, `Htreatment1`, `Htreatment2`, and `Htreatment3`:

```
look(out7)

## lm(formula = rate ~ 1 + Htreatment, data = BC)
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -1.6310 -0.7443 -0.2581  0.0000  0.8479  2.0360 
## Rsquared:  0.312
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.622      0.146  17.947    0.000
## Htreatment1   -0.829      0.207  -4.010    0.000
## Htreatment2    0.086      0.119   0.717    0.477
## Htreatment3   -0.154      0.084  -1.823    0.075
```

Remark. There is an easier way to interpret the rows of X_7.

```
X7 <- cbind(1,dummyT %*% C7)
dimnames(X7)[[2]] <- c("int","Ht1","Ht2","Ht3")
data.frame(X7,treat=BC$treatment)[c(1:2,13:14,25:26,37:38),]

##    int Ht1 Ht2 Ht3 treat
## 1    1  -1  -1  -1     A
## 2    1  -1  -1  -1     A
## 13   1   1  -1  -1     B
## 14   1   1  -1  -1     B
## 25   1   0   2  -1     C
## 26   1   0   2  -1     C
## 37   1   0   0   3     D
## 38   1   0   0   3     D
```

Each A in `treat` is coded by the first row of C_7, each B is coded by the second row of C_7, and so on.

For the estimated coefficients we have the correspondence $\widehat{f} = K_7\widehat{g}$, so that $\text{var}(\widehat{f}) = K_7 \text{var}(\widehat{g}) K_7^T$.

```
K7 <- cbind(0,rbind(0,C7))
K7[1,1] <- 1
ghat <- out7$coeff
Fhat <- as.vector(K7 %*% ghat)
names(Fhat) <- c("int",levels(BC$Ht))
Vfhat <- K7 %*% summary(out7)$cov %*% t(K7)
fhat.stderr <- sqrt(diag(Vfhat))
est <- rbind(fhat = Fhat,std.err=fhat.stderr)
print(est,3)

##           int     A     B     C      D
## fhat    2.622 0.897 -0.76 0.325 -0.461
## std.err 0.144 0.250  0.25 0.250  0.250
```

Remember that each of `out5`, `out6`, and `out7`, together with the $\widehat{f}$ representation, give the same fitted vector $\widehat{y}$. They express $\widehat{y}$ using different estimated coefficients:

```
BC.coeff(out6,3)

## lm(formula = rate ~ 1 + treatment, data = BC)
##        (Int)     tB     tC     tD
## Est    3.519 -1.657 -0.572 -1.358
## StdErr 0.292  0.413  0.413  0.413
```

```
BC.coeff(out5,3)

## lm(formula = rate ~ -1 + treatment, data = BC)
##           tA    tB    tC    tD
## Est    3.519 1.862 2.947 2.161
## StdErr 0.292 0.292 0.292 0.292

BC.coeff(out7,3)

## lm(formula = rate ~ 1 + Htreatment, data = BC)
##        (Int)    Ht1    Ht2     Ht3
## Est    2.622 -0.829 0.0855 -0.1538
## StdErr 0.146  0.207 0.1193  0.0844

print(est,3) # fhat derived from out7

##           int     A     B     C      D
## fhat    2.622 0.897 -0.76 0.325 -0.461
## std.err 0.144 0.250  0.25 0.250  0.250
```

Which is easiest to interpret?

4 Two factors

The story gets more complicated when both the posion and treatment factors are used as predictors.

To be continued.

References

Box, G. E. and D. R. Cox (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological) 26*(2), 211–252.

Box, G. E. P., W. G. Hunter, and J. S. Hunter (1978). *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building.* New York: Wiley.