Table 3.8: Body Weight and Breed of Beef Cattle Animals

Animal	Body Weight	Breed
1	471	Angus
2	463	Angus
3	481	Simmental
4	470	Angus
5	496	Simmental
6	491	Simmental
7	518	Limousin
8	511	Limousin
9	510	Limousin
10	541	Limousin

3.5 Contrasts

Contrasts are linear combinations of parameters. In R, contrasts are used to determine which estimable functions are used to produce results of a linear model analysis that are shown to a user. Furthermore, the user has the option to choose among different contrasts which are already available by default. It is also possible for the user to create custom made contrasts. This section introduces the basic idea of contrasts and how they are used in R.

Let us go back to our example datasets containing body weight and breed of different animals shown in Table 3.8.

3.5.1 Contrasts in R

The contrasts used in R can be seen from the function `contrasts()`. For our example dataset with body weight and breed of animals, we get

```
(mat_ctr <- contrasts(as.factor(tbl_flem_bw_breed$Breed)))
```

```
##           Limousin Simmental
## Angus            0         0
## Limousin         1         0
## Simmental        0         1
```

The information in the above shown contrasts matrix reflects the model terms in the columns of the matrix. Hence from the above matrix it can be seen that there are two terms associated with breeds in any linear model that considers breed as a factor. These two terms are Limousin and Simmental. The rows of the above shown contrasts matrix reflect the encoding of the different levels in the dataset. All animals of breed `Angus` are encoded with both zeroes for the two model terms. `Limousin` animals receive a code of 1 for the first model term and a code of 0 for the second term. Animals of breed `Simmental` receive a 0 for the first term and a 1 for the second term. The above contrasts matrix does not show the intercept. The intercept term is implicitly coded as 1 for all animals.

3.5.2 Model Matrix

The assignment of codes to the different data records can also be seen in the model matrix. In R the model matrix is obtained as a result of the function `model.matrix()`. The model matrix that goes together with the above shown contrasts for the factor `Breed` in our dataset is shown below.

```r
lm_bw_br <- lm(`Body Weight` ~ Breed, data = tbl_flem_bw_breed)
(mat_X <- model.matrix(lm_bw_br))
```

```
##    (Intercept) BreedLimousin BreedSimmental
## 1            1             0              0
## 2            1             0              0
## 3            1             0              1
## 4            1             0              0
## 5            1             0              1
## 6            1             0              1
## 7            1             1              0
## 8            1             1              0
## 9            1             1              0
## 10           1             1              0
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$Breed
## [1] "contr.treatment"
```

From the above shown model matrix, it can be seen that the encoding contained in the contrasts matrix is applied to the data records.

3.5.3 Estimable Functions

The type of estimable functions that are used in a given linear model analysis can be found by first extending the contrasts matrix by a column of all ones, reflecting the encoding of the intercept term.

```
mat_ctr_ext <- cbind(matrix(c(rep(1, nrow(mat_ctr))), ncol = 1), mat_ctr)
colnames(mat_ctr_ext)[1] <- colnames(mat_X)[1]
mat_ctr_ext
```

```
##           (Intercept) Limousin Simmental
## Angus               1        0         0
## Limousin            1        1         0
## Simmental           1        0         1
```

The matrix of estimable functions is obtained by computing the inverse of the extended contrasts matrix

```
(mat_estf <- solve(mat_ctr_ext))
```

```
##             Angus Limousin Simmental
## (Intercept)     1        0         0
## Limousin       -1        1         0
## Simmental      -1        0         1
```

Each row of the matrix of estimable functions corresponds to a model term. Each column can be seen as one component of the solution to the least squares normal equation. The estimate of the intercept term corresponds to the solution for the first breed level in the normal equations. The estimate for the model term `Limousin` corresponds to the difference beween the solution for the second breed level minus the solution of the first breed level. The estimate of the effect of the term `Simmental` is the difference between the last solution and the first breed level.

3.5.4 Validation

The results on the investigated connection between contrasts and estimable functions is validated with our example dataset. For this validation, we first need a set of solutions to the least squares normal equations. As the first step, we set up the design matrix $\mathbf{X}$ and use it to compute the crossproduct $\mathbf{X}^T\mathbf{X}$

```
mat_X <- model.matrix(lm(`Body Weight` ~ 0 + Breed, data = tbl_flem_bw_breed))
mat_X <- cbind(matrix(1, nrow = nrow(tbl_flem_bw_breed), ncol = 1), mat_X)
dimnames(mat_X) <- NULL
mat_xtx <- crossprod(mat_X)
mat_xtx
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   10    3    4    3
## [2,]    3    3    0    0
## [3,]    4    0    4    0
## [4,]    3    0    0    3
```

The generalized inverse $(\mathbf{X}^T\mathbf{X})^-$ provided by the function `MASS::ginv()` of package `MASS` is used to come up with a solution to the least squares normal equation

$$\mathbf{X}^T\mathbf{X}\mathbf{b}^0 = \mathbf{X}^T\mathbf{y}$$

A solution for $\mathbf{b}^0$ is

$$\mathbf{b}^0 = (\mathbf{X}^T\mathbf{X})^-\mathbf{X}^T\mathbf{y}$$

For our dataset we get

```
vec_y <- tbl_flem_bw_breed$`Body Weight`
mat_xty <- crossprod(mat_X, vec_y)
mat_xtx_ginv <- MASS::ginv(mat_xtx)
mat_b0 <- crossprod(mat_xtx_ginv,mat_xty)
mat_b0
```

```
##           [,1]
## [1,] 369.33333
## [2,]  98.66667
## [3,] 150.66667
## [4,] 120.00000
```

These solutions are used to construct the effect results computed by the function `lm()` in R. The summary table looks as follows

```
lm_bw_br <- lm(`Body Weight` ~ Breed, data = tbl_flem_bw_breed)
(smry_lm_bw_br <- summary(lm_bw_br))
```

```
##
## Call:
## lm(formula = `Body Weight` ~ Breed, data = tbl_flem_bw_breed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0000  -7.5000  -0.1667   2.7500  21.0000
##
```

```
## Coefficients:
##                 Estimate Std. Error t value        Pr(>|t|)
## (Intercept)      468.000      6.097  76.758 0.0000000000168 ***
## BreedLimousin     52.000      8.066   6.447        0.000351 ***
## BreedSimmental    21.333      8.623   2.474        0.042575 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.56 on 7 degrees of freedom
## Multiple R-squared:  0.8597, Adjusted R-squared:  0.8196
## F-statistic: 21.44 on 2 and 7 DF,  p-value: 0.001035
```

From the matrix of estimable functions

```
mat_estf
```

```
##             Angus Limousin Simmental
## (Intercept)     1        0         0
## Limousin       -1        1         0
## Simmental      -1        0         1
```

we can see that the intercept estimate corresponds to the mean body weight of all Angus animals. Which is

```
library(dplyr)
mean((tbl_flem_bw_breed %>% filter(Breed == "Angus"))$`Body Weight`)
```

```
## [1] 468
```

The estimate for the effect BreedLimousin is the difference between the third and the second component in the solution vector $\mathbf{b}^0$

```
mat_b0[3] - mat_b0[2]
```

```
## [1] 52
```

Similarly, the estimate for effect BreedSimmental is the difference between the last component of the solution vector and the second component of the solution vector.

```
mat_b0[4] - mat_b0[2]
```

```
## [1] 21.33333
```