Livestock Breeding and Genomics - Solution 4

Peter von Rohr

2021-10-29

Problem 1: Linear Models

Read the dataset given in https://charlotte-ngs.github.io/lbgfs2021/data/beef_data_blup.csv using the function `readr::read_csv()`.

Tasks

- Do a descriptive statistics on the given data
- Fit fixed linear models for `herd`, `sex` and both factors as fixed effects.

Solution

Read the data into R

```
tbl_data_beef <- readr::read_csv(file = s_data_url)
```

```
## Rows: 16 Columns: 5

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (5): Animal, Sire, Dam, Herd, Weaning Weight

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
tbl_data_beef
```

```
## # A tibble: 16 x 5
##    Animal  Sire   Dam  Herd `Weaning Weight`
##     <dbl> <dbl> <dbl> <dbl>            <dbl>
##  1     12     1     4     1             2.61
##  2     13     1     4     1             2.31
##  3     14     1     5     1             2.44
##  4     15     1     5     1             2.41
##  5     16     1     6     2             2.51
##  6     17     1     6     2             2.55
##  7     18     1     7     2             2.14
##  8     19     1     7     2             2.61
##  9     20     2     8     1             2.34
## 10     21     2     8     1             1.99
## 11     22     2     9     1             3.1
## 12     23     2     9     1             2.81
## 13     24     2    10     2             2.14
## 14     25     2    10     2             2.41
## 15     26     3    11     2             2.54
```

```
## 16      27      3     11      2                3.16
```

Descriptive Statistics for Weaning Weight

```r
summary(tbl_data_beef$`Weaning Weight`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.990   2.333   2.475   2.504   2.610   3.160
```

Summary is meaningful for continuous variables such as Weaning Weight. For discrete factor variables, the function table can be used

```r
table(tbl_data_beef$Herd)
```

```
##
## 1 2
## 8 8
```

The variable **sex** was not added to the dataset. If we had a variable that contains the sex of an animal, we could do the same summary with the table function as was done for herd.

Fixed Linear Effect Model

```r
lm_ww_h <- lm(`Weaning Weight` ~ Herd, data = tbl_data_beef)
summary(lm_ww_h)
```

```
##
## Call:
## lm(formula = `Weaning Weight` ~ Herd, data = tbl_data_beef)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51125 -0.16875 -0.02938  0.10406  0.65250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.49500    0.26053   9.577 1.59e-07 ***
## Herd         0.00625    0.16478   0.038     0.97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3295 on 14 degrees of freedom
## Multiple R-squared:  0.0001028,  Adjusted R-squared:  -0.07132
## F-statistic: 0.001439 on 1 and 14 DF,  p-value: 0.9703
```

The above fits the herd as a regression coefficient, but what we want is a fixed effect model. We have to change the numbers in the herd column from numbers to factors.

```r
tbl_data_beef$Herd <- as.factor(tbl_data_beef$Herd)
tbl_data_beef
```

```
## # A tibble: 16 x 5
##    Animal  Sire   Dam Herd  `Weaning Weight`
##     <dbl> <dbl> <dbl> <fct>            <dbl>
##  1     12     1     4 1                 2.61
##  2     13     1     4 1                 2.31
##  3     14     1     5 1                 2.44
```

```
##   4      15      1      5 1                   2.41
##   5      16      1      6 2                   2.51
##   6      17      1      6 2                   2.55
##   7      18      1      7 2                   2.14
##   8      19      1      7 2                   2.61
##   9      20      2      8 1                   2.34
##  10      21      2      8 1                   1.99
##  11      22      2      9 1                   3.1
##  12      23      2      9 1                   2.81
##  13      24      2     10 2                   2.14
##  14      25      2     10 2                   2.41
##  15      26      3     11 2                   2.54
##  16      27      3     11 2                   3.16
```

The function class tells us the datatype of a column

```
class(tbl_data_beef$Herd)
```

```
## [1] "factor"
```

```
lm_ww_h_fix <- lm(`Weaning Weight` ~ 0+ Herd, data = tbl_data_beef)
summary(lm_ww_h_fix)
```

```
##
## Call:
## lm(formula = `Weaning Weight` ~ 0 + Herd, data = tbl_data_beef)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51125 -0.16875 -0.02937  0.10406  0.65250
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## Herd1   2.5012     0.1165   21.47 4.11e-12 ***
## Herd2   2.5075     0.1165   21.52 3.97e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3295 on 14 degrees of freedom
## Multiple R-squared:  0.9851, Adjusted R-squared:  0.9829
## F-statistic:   462 on 2 and 14 DF,  p-value: 1.65e-13
```

Additional Problem: Mixed Linear Effect Model

Start with the Sire model

$$y = X\beta + Zu + e$$

Observations

```
vec_y <- tbl_data_beef$`Weaning Weight`
vec_y
```

```
##  [1] 2.61 2.31 2.44 2.41 2.51 2.55 2.14 2.61 2.34 1.99 3.10 2.81 2.14 2.41 2.54
## [16] 3.16
```

Design Matrices

```
mat_X <- matrix(data = c(1, 0,
                         1, 0,
                         1, 0,
                         1, 0,
                         0, 1,
                         0, 1,
                         0, 1,
                         0, 1,
                         1, 0,
                         1, 0,
                         1, 0,
                         1, 0,
                         0, 1,
                         0, 1,
                         0, 1,
                         0, 1), ncol = 2, byrow = TRUE)
mat_X
```

```
##       [,1] [,2]
##  [1,]    1    0
##  [2,]    1    0
##  [3,]    1    0
##  [4,]    1    0
##  [5,]    0    1
##  [6,]    0    1
##  [7,]    0    1
##  [8,]    0    1
##  [9,]    1    0
## [10,]    1    0
## [11,]    1    0
## [12,]    1    0
## [13,]    0    1
## [14,]    0    1
## [15,]    0    1
## [16,]    0    1
```

```
tbl_data_beef$Herd
```

```
##  [1] 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2
## Levels: 1 2
```

```
mat_Z <- matrix(data = c(1, 0, 0,
                         1, 0, 0,
                         1, 0, 0,
                         1, 0, 0,
                         1, 0, 0,
                         1, 0, 0,
                         1, 0, 0,
                         1, 0, 0,
                         0, 1, 0,
                         0, 1, 0,
                         0, 1, 0,
                         0, 1, 0,
```

```r
                                 0, 1, 0,
                                 0, 1, 0,
                                 0, 0, 1,
                                 0, 0, 1), ncol=3, byrow = TRUE)
mat_Z
```

```
##       [,1] [,2] [,3]
##  [1,]    1    0    0
##  [2,]    1    0    0
##  [3,]    1    0    0
##  [4,]    1    0    0
##  [5,]    1    0    0
##  [6,]    1    0    0
##  [7,]    1    0    0
##  [8,]    1    0    0
##  [9,]    0    1    0
## [10,]    0    1    0
## [11,]    0    1    0
## [12,]    0    1    0
## [13,]    0    1    0
## [14,]    0    1    0
## [15,]    0    0    1
## [16,]    0    0    1
```

```r
tbl_data_beef$Sire
```

```
##  [1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3
```

Solving MME

```r
lambda <- 1
mat_xtx <- t(mat_X) %*% mat_X
mat_xtx
```

```
##      [,1] [,2]
## [1,]    8    0
## [2,]    0    8
```

```r
mat_xtx <- crossprod(mat_X)
mat_xtx
```

```
##      [,1] [,2]
## [1,]    8    0
## [2,]    0    8
```

```r
mat_xtz <- crossprod(mat_X,mat_Z)
mat_xtz
```

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

```r
mat_ztx <- crossprod(mat_Z,mat_X)
mat_ztx <- t(mat_xtz)
mat_ztx
```

```
##      [,1] [,2]
```

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

```
mat_ztz <- crossprod(mat_Z)
mat_ztz
```

```
##      [,1] [,2] [,3]
## [1,]    8    0    0
## [2,]    0    6    0
## [3,]    0    0    2
```

Adding $I * \lambda$

```
mat_ilambda <- diag(1, nrow = nrow(mat_ztz), ncol = ncol(mat_ztz)) * lambda
mat_ilambda
```

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

```
mat_ztzilambda <- mat_ztz + mat_ilambda
mat_ztzilambda
```

```
##      [,1] [,2] [,3]
## [1,]    9    0    0
## [2,]    0    7    0
## [3,]    0    0    3
```

Coefficient Matrix

```
mat_coef <- rbind(cbind(mat_xtx, mat_xtz),
                  cbind(mat_ztx, mat_ztzilambda))
mat_coef
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    8    0    4    4    0
## [2,]    0    8    4    2    2
## [3,]    4    4    9    0    0
## [4,]    4    2    0    7    0
## [5,]    0    2    0    0    3
```

Right Hand Side

```
mat_xty <- crossprod(mat_X, vec_y)
mat_xty
```

```
##       [,1]
## [1,] 20.01
## [2,] 20.06
```

```
mat_zty <- crossprod(mat_Z, vec_y)
mat_zty
```

```
##       [,1]
## [1,] 19.58
## [2,] 14.79
```

```
## [3,]  5.70
```

```
mat_rhs <- rbind(mat_xty, mat_zty)
mat_rhs
```

```
##       [,1]
## [1,] 20.01
## [2,] 20.06
## [3,] 19.58
## [4,] 14.79
## [5,]  5.70
```

The Solution

```
vec_unknown <- solve(mat_coef, mat_rhs)
vec_unknown
```

```
##          [,1]
## [1,]  2.60630
## [2,]  2.53485
## [3,] -0.10940
## [4,] -0.10070
## [5,]  0.21010
```

The first two components of the solution vector `vec_unknown` corresponds to estimates of the effects of the two herds. The remaining three components of vector `vec_unknown` correspond to the three sire breeding values.