Applied Statistical Methods - Solution 3

Peter von Rohr

2022-03-12

Problem 1: Linear Regression on Genomic Information

Use the following dataset which is also given in:

https://charlotte-ngs.github.io/asmss2022/data/asm_flem_genomic_data.csv

to estimate marker effects for the single loci using a linear regression model.

Animal	SNP G	SNP H	Observation
1	G_1G_1	H_1H_2	510
2	G_1G_2	H_1H_1	528
3	G_1G_2	H_1H_1	505
4	G_1G_1	H_2H_2	539
5	G_1G_1	H_1H_1	530
6	G_1G_2	H_1H_2	489
7	G_1G_2	H_2H_2	486
8	G_2G_2	H_1H_1	485
9	G_1G_2	H_2H_2	478
10	G_2G_2	H_1H_2	479
11	G_1G_1	H_1H_2	520
12	G_1G_1	H_1H_1	521
13	G_2G_2	H_1H_2	473
14	G_2G_2	H_1H_2	457
15	G_1G_2	H_1H_1	497
16	G_1G_2	H_1H_2	516
17	G_1G_1	H_1H_2	524
18	G_1G_1	H_1H_2	502
19	G_1G_1	H_2H_2	508
20	G_1G_2	H_1H_2	506

Solution

- Read the data using `read.csv()`

```r
s_ex03p01_data_path <- "https://charlotte-ngs.github.io/asmss2022/data/asm_flem_genomic_data.csv"
tbl_ex03p01_data <- readr::read_csv(file = s_ex03p01_data_path)
```

```
## Rows: 20 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): SNP G, SNP H
## dbl (2): Animal, Observation
```

```
##
## 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_ex03p01_data
```

```
## # A tibble: 20 x 4
##    Animal `SNP G`  `SNP H`  Observation
##     <dbl> <chr>    <chr>          <dbl>
##  1      1 $G_1G_1$ $H_1H_2$         510
##  2      2 $G_1G_2$ $H_1H_1$         528
##  3      3 $G_1G_2$ $H_1H_1$         505
##  4      4 $G_1G_1$ $H_2H_2$         539
##  5      5 $G_1G_1$ $H_1H_1$         530
##  6      6 $G_1G_2$ $H_1H_2$         489
##  7      7 $G_1G_2$ $H_2H_2$         486
##  8      8 $G_2G_2$ $H_1H_1$         485
##  9      9 $G_1G_2$ $H_2H_2$         478
## 10     10 $G_2G_2$ $H_1H_2$         479
## 11     11 $G_1G_1$ $H_1H_2$         520
## 12     12 $G_1G_1$ $H_1H_1$         521
## 13     13 $G_2G_2$ $H_1H_2$         473
## 14     14 $G_2G_2$ $H_1H_2$         457
## 15     15 $G_1G_2$ $H_1H_1$         497
## 16     16 $G_1G_2$ $H_1H_2$         516
## 17     17 $G_1G_1$ $H_1H_2$         524
## 18     18 $G_1G_1$ $H_1H_2$         502
## 19     19 $G_1G_1$ $H_2H_2$         508
## 20     20 $G_1G_2$ $H_1H_2$         506
```

- Re-code the genotypes to numeric values

```
tbl_ex03p01_data <- dplyr::mutate(tbl_ex03p01_data,
                                  `SNP G` = dplyr::recode(`SNP G`,
                                                          "$G_2G_2$" = 0,
                                                          "$G_1G_2$" = 1,
                                                          "$G_1G_1$" = 2))
tbl_ex03p01_data <- dplyr::mutate(tbl_ex03p01_data,
                                  `SNP H` = dplyr::recode(`SNP H`,
                                                          "$H_2H_2$" = 0,
                                                          "$H_1H_2$" = 1,
                                                          "$H_1H_1$" = 2))

tbl_ex03p01_data
```

```
## # A tibble: 20 x 4
##    Animal `SNP G` `SNP H` Observation
##     <dbl>   <dbl>   <dbl>       <dbl>
##  1      1       2       1         510
##  2      2       1       2         528
##  3      3       1       2         505
##  4      4       2       0         539
##  5      5       2       2         530
##  6      6       1       1         489
##  7      7       1       0         486
##  8      8       0       2         485
##  9      9       1       0         478
## 10     10       0       1         479
## 11     11       2       1         520
## 12     12       2       2         521
## 13     13       0       1         473
## 14     14       0       1         457
## 15     15       1       2         497
## 16     16       1       1         516
## 17     17       2       1         524
## 18     18       2       1         502
## 19     19       2       0         508
## 20     20       1       1         506
```

- Fit the multiple regression to the data

```
lm_mult_reg_genomic <- lm(formula = Observation ~ `SNP G` + `SNP H`, data = tbl_ex03p01_data)
summary(lm_mult_reg_genomic)
```

```
##
## Call:
## lm(formula = Observation ~ `SNP G` + `SNP H`, data = tbl_ex03p01_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4643  -8.2468  -0.6883   3.9448  26.9383
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  465.425      7.476  62.252  < 2e-16 ***
## `SNP G`       23.318      3.861   6.040 1.33e-05 ***
## `SNP H`        8.403      4.127   2.036   0.0577 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.8 on 17 degrees of freedom
## Multiple R-squared:  0.691,  Adjusted R-squared:  0.6546
## F-statistic: 19.01 on 2 and 17 DF,  p-value: 4.621e-05
```

The marker-effects for the two loci correspond to

```
vec_coeff <- coefficients(lm_mult_reg_genomic)
cat("\nMarker Effect for locus G: ", vec_coeff["`SNP G`"], "\n")
```

```
##
## Marker Effect for locus G:  23.31818
```

```
cat("Marker Effect for locus H: ", vec_coeff["`SNP H`"], "\n")
```

```
## Marker Effect for locus H:  8.402597
```

Problem 2: Regression On Dummy Variables

Use the dataset with the breeds assigned to every animal and find out the influence of the breed on the response variable `body weight`. The data is available from

```
## [1] "https://charlotte-ngs.github.io/asmss2022/data/asm_bw_flem.csv"
```

Start by fitting a linear model with `Breed` as the only factor in the model, hence ignore the independent variables such as `Breast Circumference`, `BCS` and `HEI`.

Solution

- Read the data

```
s_ex03p02_data_path <- "https://charlotte-ngs.github.io/asmss2022/data/asm_bw_flem.csv"
tbl_ex03p02_data <- readr::read_csv(file = s_ex03p02_data_path)
```

```
## Rows: 10 Columns: 6

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Breed
## dbl (5): Animal, Breast Circumference, Body Weight, BCS, HEI

##
## 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_ex03p02_data
```

```
## # A tibble: 10 x 6
##    Animal `Breast Circumference` `Body Weight`   BCS   HEI Breed
##     <dbl>                  <dbl>         <dbl> <dbl> <dbl> <chr>
##  1      1                    176           471   5     161 Angus
##  2      2                    177           463   4.2   121 Angus
##  3      3                    178           481   4.9   157 Simmental
##  4      4                    179           470   3     165 Angus
##  5      5                    179           496   6.8   136 Simmental
##  6      6                    180           491   4.9   123 Simmental
##  7      7                    181           518   4.4   163 Limousin
##  8      8                    182           511   4.4   149 Limousin
##  9      9                    183           510   3.5   143 Limousin
## 10     10                    184           541   4.7   130 Limousin
```

- Fit a linear model including breed as a factor

```
lm_reg_dummy_bw_breed <- lm(formula = `Body Weight` ~ Breed, data = tbl_ex03p02_data)
summary(lm_reg_dummy_bw_breed)
```

```
##
## Call:
## lm(formula = `Body Weight` ~ Breed, data = tbl_ex03p02_data)
##
## 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 1.68e-11 ***
## 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
```

Problem 3: Estimable Function

Use the matrix vector-notation to setup the model for a regression on dummy variable with the data on breeds and body weight as used in Problem 2. The aim of this problem is to find the estimable functions used in the output of `lm()`.

The model is given by

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

Setup the least squares normal equations. Find a solution for $\mathbf{b}^0$ and construct the estimable function that is used in the output `lm()`.

Solution

- Define elements of least squares normal equations

The required elements are the vector $\mathbf{y}$ and the matrix $\mathbf{X}$. The are defined as follows

```
s_ex03p03_data_path <- "https://charlotte-ngs.github.io/asmss2022/data/asm_bw_flem.csv"
tbl_ex03p03_data <- readr::read_csv(file = s_ex03p03_data_path)
```

```
## Rows: 10 Columns: 6

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Breed
## dbl (5): Animal, Breast Circumference, Body Weight, BCS, HEI

##
## 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.
```

In a first step, the records in our dataframe are sorted according to their breed.

```
tbl_ex03p03_data <- tbl_ex03p03_data[order(tbl_ex03p03_data$Breed),]
tbl_ex03p03_data
```

```
## # A tibble: 10 x 6
##    Animal `Breast Circumference` `Body Weight`   BCS   HEI Breed
##     <dbl>                  <dbl>         <dbl> <dbl> <dbl> <chr>
##  1      1                    176           471   5     161 Angus
##  2      2                    177           463   4.2   121 Angus
##  3      4                    179           470   3     165 Angus
##  4      7                    181           518   4.4   163 Limousin
##  5      8                    182           511   4.4   149 Limousin
##  6      9                    183           510   3.5   143 Limousin
##  7     10                    184           541   4.7   130 Limousin
##  8      3                    178           481   4.9   157 Simmental
##  9      5                    179           496   6.8   136 Simmental
## 10      6                    180           491   4.9   123 Simmental
```

After the sorting process, the elements of the least squares normal equation can be extracted.

```
vec_y <- tbl_ex03p03_data$`Body Weight`
mat_X <- model.matrix(lm(`Body Weight` ~ 0 + Breed, data = tbl_ex03p03_data))
attr(mat_X, "assign") <- NULL
attr(mat_X, "contrasts") <- NULL
colnames(mat_X) <- NULL
rownames(mat_X) <- NULL
mat_X <- cbind(matrix(1,nrow = nrow(mat_X), ncol = 1), mat_X)
mat_X
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]    1    1    0    0
##  [2,]    1    1    0    0
##  [3,]    1    1    0    0
##  [4,]    1    0    1    0
##  [5,]    1    0    1    0
##  [6,]    1    0    1    0
##  [7,]    1    0    1    0
##  [8,]    1    0    0    1
##  [9,]    1    0    0    1
## [10,]    1    0    0    1
```

Which corresponds to

$$\mathbf{y} = \begin{bmatrix} 471 \\ 463 \\ 470 \\ 518 \\ 511 \\ 510 \\ 541 \\ 481 \\ 496 \\ 491 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

- Find a solution for $\mathbf{b}^0$

A solution for $\mathbf{b}^0$ can be found using a generalzed inverse. In R this can be done with

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

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

A generalized inverse is obtained by

```
(mat_xtx_ginv <- MASS::ginv(mat_xtx))
```

```
##             [,1]        [,2]         [,3]        [,4]
## [1,] 0.057291667  0.02604167  0.005208333  0.02604167
## [2,] 0.026041667  0.22395833 -0.088541667 -0.10937500
## [3,] 0.005208333 -0.08854167  0.182291667 -0.08854167
## [4,] 0.026041667 -0.10937500 -0.088541667  0.22395833
```

The solution for $\mathbf{b}^0$

```
mat_xty <- crossprod(mat_X, vec_y)
(mat_b0 <- crossprod(mat_xtx_ginv, mat_xty))
```

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

The first question is whether the elements in `mat_b0` are a solution to the least squares normal equation. This can be verified by inserting the solution back into the normal equations.

```
crossprod(mat_xtx, mat_b0) - mat_xty
```

```
##               [,1]
## [1,] -9.094947e-13
## [2,] -2.273737e-13
## [3,] -4.547474e-13
## [4,] -4.547474e-13
```

- Construct the estimable function. As a hint, assume the missing factor level in the output of `lm()` to be zero.

The second question is what type of estimable function is used by the function `lm()` in R. Because there is no solution given for `BreedAngus`, it seams reasonable to assume that the effect for that level is set to 0. The levels of the other breeds are just differences to the effect for the level `BreedAngus`. Such an estimable functions for the two breed effects for Limousin and Simmental would be expressed in terms of the following two vectors q_L and q_S

$$q_L = \begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}$$

We first have to very whether q_L is an estimable function. This can be done by verifying whether

$$q_L^t H = q_L^T$$

where $\mathbf{H} = \mathbf{G}\mathbf{X}^T\mathbf{X}$ with G being a generalized inverse of (X^TX).

```
mat_H <- crossprod(mat_xtx_ginv, mat_xtx)
mat_H
```

```
##      [,1]  [,2]  [,3]  [,4]
## [1,] 0.75  0.25  0.25  0.25
## [2,] 0.25  0.75 -0.25 -0.25
## [3,] 0.25 -0.25  0.75 -0.25
## [4,] 0.25 -0.25 -0.25  0.75
```

```
crossprod(vec_q_l, mat_H)
```

```
##              [,1] [,2] [,3]         [,4]
## [1,] 5.551115e-17   -1    1 1.665335e-16
```

Similarly for q_S

```
vec_q_s <- c(0, -1, 0, 1)
crossprod(vec_q_s, mat_H)
```

```
##               [,1] [,2]          [,3] [,4]
## [1,] -5.551115e-17   -1 -1.110223e-16    1
```

Since both vectors q_L and q_S are valid estimable functions, their estimates are computed as follows. First for the effect of the breed Limousin

```
crossprod(vec_q_l, mat_b0)
```

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

Next, the effect for the breed Simmental

```
crossprod(vec_q_s, mat_b0)
```

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