Peter von Rohr
Institute of Agricultural Sciences
D-USYS
ETH Zurich

751-7602-00 V
Solutions for Exam
Applied Statistical Methods
in Animal Sciences
SS 2021

Date: 31^{st} May 2021

Name: Firstname Name

Legi-Nr: LegiNr

Problem	Maximum Number of Points	Number of Points Reached
1	16	
2	28	
3	9	
4	38	
Total	91	

Questions in German are in italics

Problem 1: Linear Regression

The same dataset is analysed with two different regression models. The R-Output of both analyses is given by Output A and Output B.

Wir haben den gleichen Datensatz mit zwei unterschiedlichen linearen Regressionsmodellen analysiert. Der R-Output dieser beiden Analysen ist nachfolgend als Output A und Output B gegeben.

Output A

```
##
## Call:
## lm(formula = y ~ X1, data = dfSimData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2899 -1.4864  0.2526  1.2982  4.6501
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8929     2.6536  -0.713    0.482
## X1            4.0680     0.8675   4.689 6.49e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.108 on 28 degrees of freedom
## Multiple R-squared:  0.4399, Adjusted R-squared:  0.4199
## F-statistic: 21.99 on 1 and 28 DF,  p-value: 6.487e-05
```

Output B

```
##
## Call:
## lm(formula = y ~ -1 + X1, data = dfSimData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0925 -1.4013 -0.0846  1.6308  4.3171
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## X1   3.4557     0.1247   27.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.09 on 29 degrees of freedom
## Multiple R-squared:  0.9636, Adjusted R-squared:  0.9623
## F-statistic: 767.6 on 1 and 29 DF,  p-value: < 2.2e-16
```

a) Give the formulas of both statistical models which belong to Output A and Output B. Where is the main difference between both models? *Geben Sie die Formeln der beiden statistischen Modelle an, welche zu Output A und Output B geführt haben. Wo liegt der hauptsächliche Unterschied zwischen den beiden Modellen?*

8

Solution

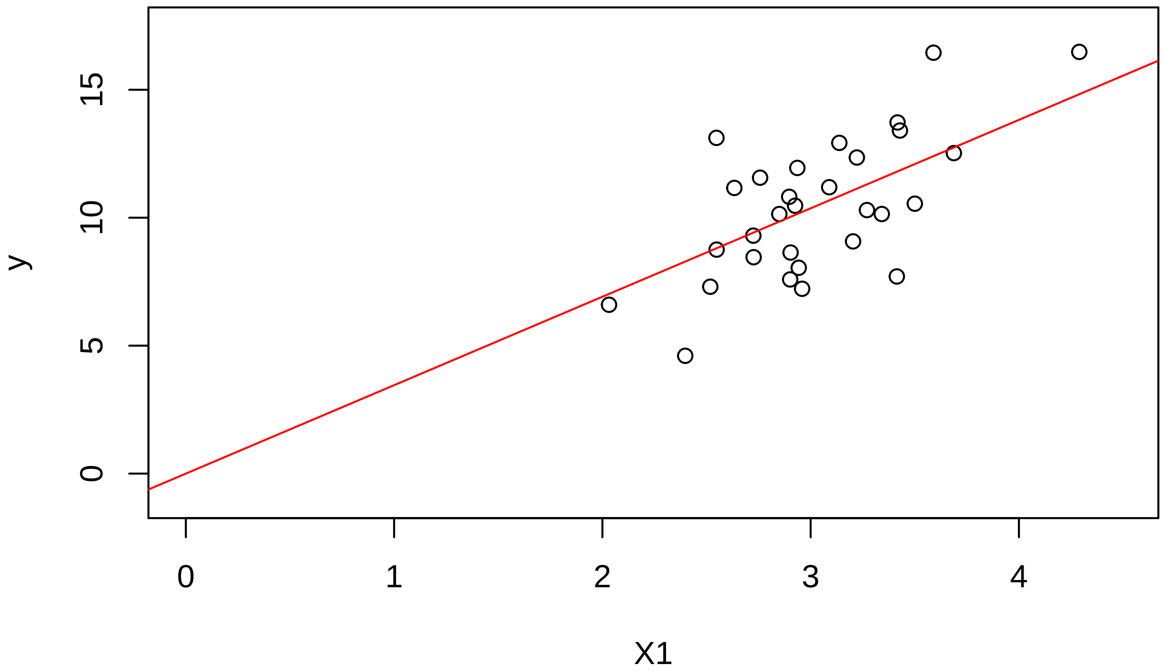
- Modell für Output A: $y_i = \beta_0 + \beta_1 X_{1i} + \epsilon_i$
- Modell für Output B: $y_i = \beta_1 X_{1i} + \epsilon_i$
- Der Hauptunterschied liegt darin, dass im Modell von Output A ein Achsenabschnitt β_0 angepasst wird und im Modell von Output B nicht.

b) For both analyses a plot was produced. Assign Plots 1 and 2 to Outputs A and B.

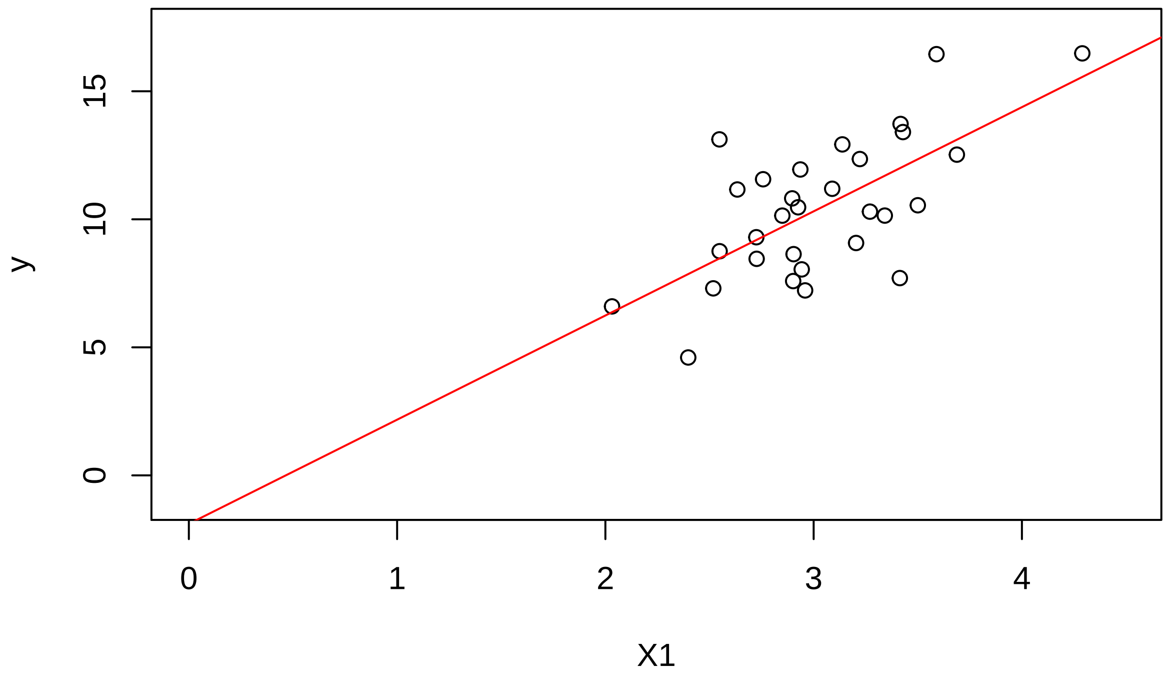
Für die zwei Analysen wurden auch zwei Plots gemacht. Ordnen Sie die Plots 1 und 2 den Outputs A und B zu.

2

Plot 1



Plot 2



Solution

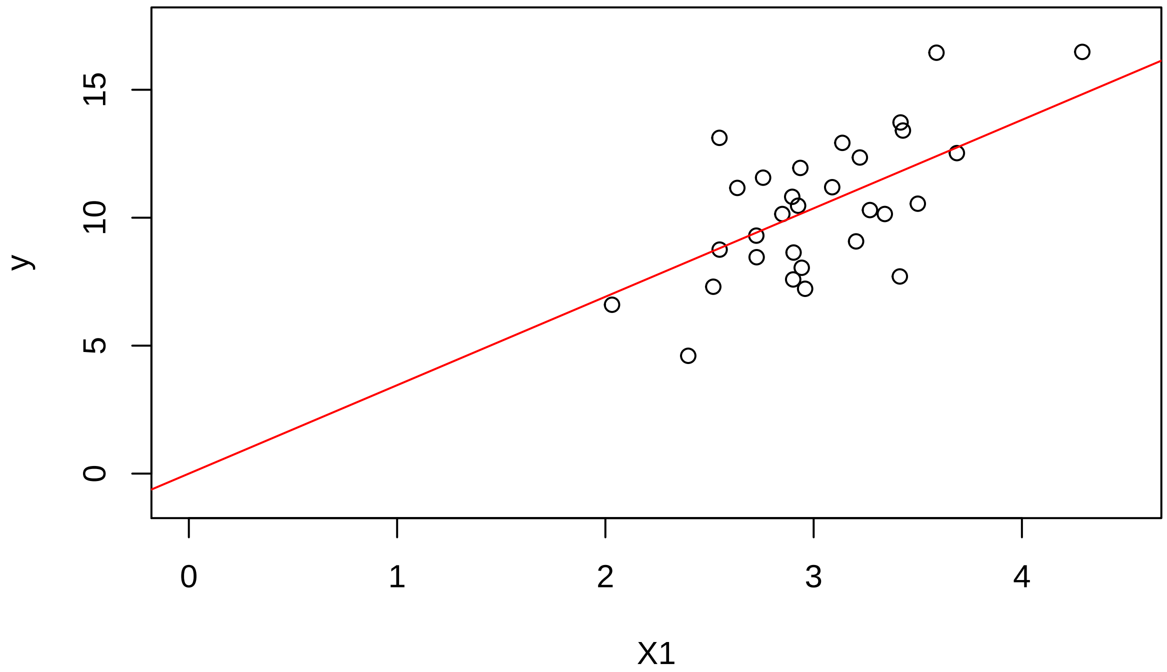
- Plot 1 gehört zu Output B
- Plot 2 gehört zu Output A

c) Enter the parameter estimates from Outputs A and B in Plots 1 and 2 by marking their lengths in the plots.

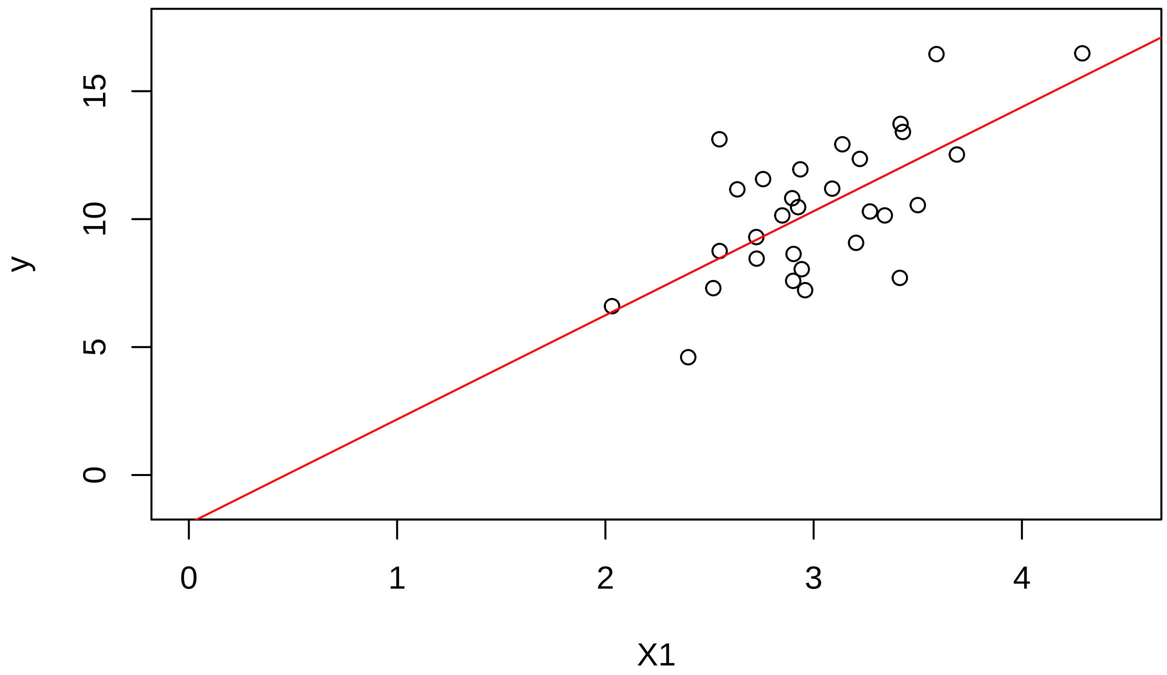
Zeichnen Sie die geschätzten Parameter (Estimate) aus den Outputs A und B in die Plots 1 und 2 ein

6

Plot 1



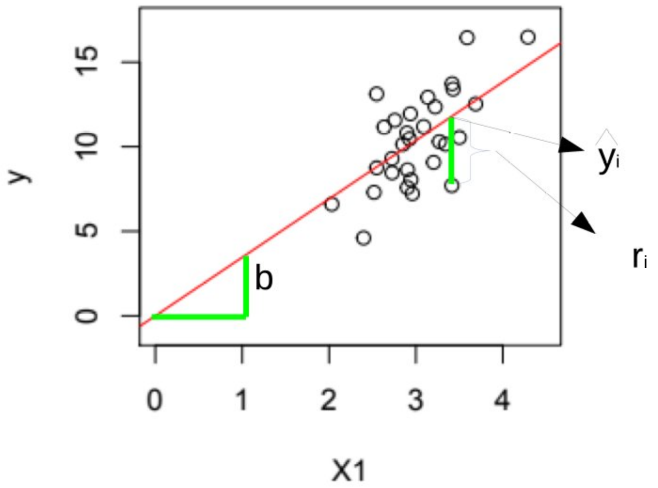
Plot 2



Solution

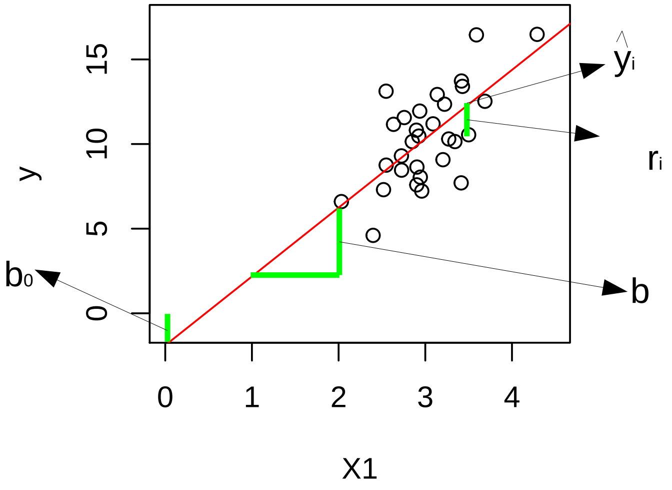
Plot 1

The following plot shows parameters from Output B



Plot 2

The following plot shows parameters from Output A



Problem 2: Bayes

The following table contains body weight and slaughter weight for 12 animals. Before the farmer sells the animal to the slaughter house, it is weighed on the farm. The slaughter weight is determined by the slaughter house.

Die folgende Tabelle enthält Lebendgewicht (‘BodyWeight‘) und Schlachtgewicht (‘SlaughterWeight‘) für 12 Tiere. Vor der Schlachtung wird das Tier auf dem Betrieb noch gewogen. Das Schlachtgewicht wird im Schlachthof bestimmt.

Animal	BodyWeight	SlaughterWeight
1	522	200
2	516	199
3	523	205
4	540	224
5	530	209
6	549	213
7	543	209
8	547	219
9	549	220
10	524	204
11	535	206
12	540	209

a) Please specify the equation that models ‘SlaughterWeight‘ (response variable) as a regression on ‘BodyWeight‘ (predictor variable). Based on the specified regression equation and based on the dataset, create a table with knowns and unknowns.

Bitte geben Sie eine Modellgleichung, welche ‘SlaughterWeight‘ (Zielgrösse) als Regression auf ‘BodyWeight‘ (Predictorvariable) modelliert. Basierend auf der spezifizierten Regressionsgleichung und basierend auf dem Datensatz, geben Sie in einer Tabelle an, welche Grössen bekannt und welche unbekannt sind.

7

Solution

- Regression Model:
- Table of knowns and unknowns

Quantity Known/Unknown ——— —————

Master Solution

- Regression Model: For a given animal i, the slaughterweight y_i can be written as

$$y_i = b_0 + b_1 * x_i + e_i$$

where b_0 is the intercept, b_1 the regression slope, e_i the random error and x_i the bodyweight of animal i. Using matrix-vector notation, we get

$$y = Xb + e$$

with y being the vector of slaughterweights, b the vector with the intercept and the regression slope, e the vector of random residuals and X is the matrix having n rows and two columns where the first column is all ones and the seond column contains the bodyweight of all animals.

- Table of knowns and unknowns

Quantity	Known/Unknown
y	known
X	known
b	unknown
e	unknown
σ_e^2	unknown

b) The following programming code in R does a Bayesian estimation of the unknowns of the regression model. Please complete the code where indicated (lines after comment starting with "TODO") such that estimates of unknowns are obtained.

Der nachfolgende Programmcode in R ergibt eine Bayes'sche Schätzung der unbekannten im Regressionsmodell. Bitte vervollständigen Sie den nachfolgenden Programmcode so, dass die Schätzungen der Unbekannten Grössen im Regressionsmodell resultieren. Die zu ergänzenden Stellen sind mit einem Kommentar, welcher mit dem Wort "TODO" beginnt, markiert.

15

Hint

- The dataset is available at: https://charlotte-ngs.github.io/gelasmss2021/data/asm_exam_p02.csv.

Solution

```r
# read the data
s_data_p02_path <- "https://charlotte-ngs.github.io/gelasmss2021/data/asm_exam_p02.csv"
tbl_reg_sw_bw <- readr::read_csv2(file = s_data_p02_path)

# take number of observations from tbl_reg_sw_bw
n_nr_obs <- nrow(tbl_reg_sw_bw)

# define Matrix X
X <- matrix(c(rep(1,n_nr_obs), tbl_reg_sw_bw$BodyWeight), ncol = 2)
# observations as vector y
y <- tbl_reg_sw_bw$SlaughterWeight
# fix constants
nuRes <- 4
varResidual <- 1
scaleRes <- varResidual * (nuRes - 2)/nuRes
mu <- mean(y)
ycorr <- y - mu
# intialise estimates for intercept, slope and residual variance
beta <- c(0,0)
meanBeta <- c(0, 0)
sigma <- 1
meanSigma <- 0
# loop over iterations of the Gibbs Sampler
niter <- 1000
for (iter in 1:niter){
  # sampling intercept
  w  <-  y - X[, 2] * beta[2]
  x <- X[, 1]
  xpxi <- 1/(t(x) %*% x)
  # TODO: compute mean of conditional distribution
  betaHat <-
  # TODO: draw sample of intercept from normal distribution
  beta[1] <-
  # sampling slope
  w <- y - X[, 1] * beta[1]
  x <- X[, 2]
  xpxi <- 1/(t(x) %*% x)
  # TODO: compute mean of conditional distribution
  betaHat <-
  # TODO: draw sample for slope from normal distribution
  beta[2] <-
  # sample residual variance
  sigma <- (crossprod(ycorr) + nuRes * scaleRes) / rchisq(1, n_nr_obs + nuRes)
 # TODO: save current sample of beta to meanBeta and sigma to meanSigma
  meanBeta <-
  meanSigma <-
  # output every 200th sample
  if (iter %% 200 == 0){
    cat(" * Iteration:         ", iter, "\n")
    cat(" * Intercept:         ", beta[1], "\n")
    cat(" * Slope:             ", beta[2], "\n")
    cat(" * Residual Variance: ", sigma, "\n")
  }
}

# output estimates
cat(" * Bayes Estimates\n")
cat(" * Intercept:         ", meanBeta[1]/niter, "\n")
cat(" * Slope:             ", meanBeta[2]/niter, "\n")
cat(" * Residual Variance: ", meanSigma/niter, "\n")
```

Master Solution

```r
# set seed
set.seed(7821)

# read the data
s_data_p02_path <- "https://charlotte-ngs.github.io/gelasmss2021/data/asm_exam_p02.csv"
tbl_reg_sw_bw <- readr::read_csv2(file = s_data_p02_path)
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Parsed with column specification:
## cols(
##   Animal = col_double(),
##   BodyWeight = col_double(),
##   SlaughterWeight = col_double()
## )
```

```r
# take number of observations from tbl_reg_sw_bw
n_nr_obs <- nrow(tbl_reg_sw_bw)

# define Matrix X
X <- matrix(c(rep(1,n_nr_obs), tbl_reg_sw_bw$BodyWeight), ncol = 2)
# observations as vector y
y <- tbl_reg_sw_bw$SlaughterWeight
# fix constants
nuRes <- 4
varResidual <- 1
scaleRes <- varResidual * (nuRes - 2)/nuRes
mu <- mean(y)
ycorr <- y - mu
# intialise estimates for intercept, slope and residual variance
beta <- c(0,0)
meanBeta <- c(0, 0)
sigma <- 1
meanSigma <- 0
# loop over iterations of the Gibbs Sampler
niter <- 1000
for (iter in 1:niter){
  # sampling intercept
  w = y - X[, 2] * beta[2]
  x = X[, 1]
  xpxi = 1/(t(x) %*% x)
  # compute mean of conditional distribution
  betaHat = t(x) %*% w * xpxi
  # draw sample of intercept from normal distribution
  beta[1] = rnorm(1, betaHat, sqrt(xpxi * sigma))
  # sampling slope
  w = y - X[, 1] * beta[1]
  x = X[, 2]
  xpxi = 1/(t(x) %*% x)
  # compute mean of conditional distribution
  betaHat = t(x) %*% w * xpxi
  # draw sample for slope from normal distribution
  beta[2] = rnorm(1, betaHat, sqrt(xpxi * sigma))
  # sample residual variance
  sigma <- (crossprod(ycorr) + nuRes * scaleRes) / rchisq(1, n_nr_obs + nuRes)
  # save current sample of beta to meanBeta and sigma to meanSigma
  # add beta samples
  meanBeta <- meanBeta + beta
  # add sample for sigma
  meanSigma <- meanSigma + sigma
  # output every 200th sample
  if (iter %% 200 == 0){
    cat(" * Iteration:         ", iter, "\n")
    cat(" * Intercept:         ", beta[1], "\n")
    cat(" * Slope:             ", beta[2], "\n")
    cat(" * Residual Variance: ", sigma, "\n")
  }
}
```

```
##  * Iteration:          200 
##  * Intercept:          175.0244 
##  * Slope:              0.06354597 
##  * Residual Variance:  25.0444 
##  * Iteration:          400 
##  * Intercept:          156.5652 
##  * Slope:              0.09664144 
##  * Residual Variance:  73.34721 
##  * Iteration:          600 
##  * Intercept:          142.4197 
##  * Slope:              0.1269605 
##  * Residual Variance:  28.03807 
##  * Iteration:          800 
##  * Intercept:          88.67871 
##  * Slope:              0.2290343 
##  * Residual Variance:  41.97097 
##  * Iteration:          1000 
##  * Intercept:          41.1039 
##  * Slope:              0.31716 
##  * Residual Variance:  67.04594
```

```
# output estimates
cat(" * Bayes Estimates\n")
```

```
##  * Bayes Estimates
```

```
cat(" * Intercept:         ", meanBeta[1]/niter, "\n")
```

```
##  * Intercept:          137.5765
```

```
cat(" * Slope:             ", meanBeta[2]/niter, "\n")
```

```
##  * Slope:              0.1352381
```

```
cat(" * Residual Variance: ", meanSigma/niter, "\n")
```

```
##  * Residual Variance:  49.37294
```

Validation with `lm()`

```
fit_sw_bw <- lm(SlaughterWeight ~ BodyWeight, data = tbl_reg_sw_bw)
summary(fit_sw_bw)
```

```
##
## Call:
## lm(formula = SlaughterWeight ~ BodyWeight, data = tbl_reg_sw_bw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2871 -3.6760 -0.0091  2.0463 11.3796
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.3864    67.3029  -1.298   0.2233
## BodyWeight    0.5556     0.1258   4.416   0.0013 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.823 on 10 degrees of freedom
## Multiple R-squared:  0.661,  Adjusted R-squared:  0.6271
## F-statistic:  19.5 on 1 and 10 DF,  p-value: 0.001302
```

c) We assume that in addition to the 12 animals shown above there is an additional animal with a body weight of 513 kg. In the slaughterhouse the slaughter weight could not be measured, hence it is missing. How are such missing observations handled in a Bayesian analysis? Please fill out the table with the knowns and unknowns once again taking into account the fact that the observation of the slaughterweight for animal 13 is missing.

Wir nehmen an, dass zusätzlich zu den 12 Tieren, welche in der Tabelle oben gezeigt wurden, noch ein zusätzliches Tier mit einem Lebendgewicht von 513 kg hinzukommt. Im Schlachthof konnte das Schlachtgewicht vom Tier 13 nicht erfasst werden und fehlt somit. Wie wird diese fehlende Beobachtung in einer Bayes'schen Analyse behandelt? Bitte ergänzen Sie die Tabelle mit den bekannten und den unbekannten Grössen unter Berücksichtigung der fehlenden Beobachtung des Schlachtgewichts von Tier 13.

6

Solution

Expanded Table with knowns and unknowns

Quantity Known/Unknown ——— —————

Master Solution

Expanded Table with knowns and unknowns

Quantity	Known/Unknown
$y[1:12]$	known
$y[13]$	unknown
X	known
b	unknown
e	unknown
σ_e^2	unknown

Problem 3: LASSO

a) LASSO is an alternative procedure to Least Squares to estimate parameters of a linear model. Which of the following equations belongs to least square and which belongs to LASSO.

LASSO ist ein alternatives Parameterschätzverfahren zu Least Squares. Ordnen Sie die nachfolgenden Gleichungen zu den beiden Verfahren Least Squares und LASSO zu.

4

$$\hat{\beta}_1 \quad = \quad argmin_\beta \left\{ ||y - X\beta||^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

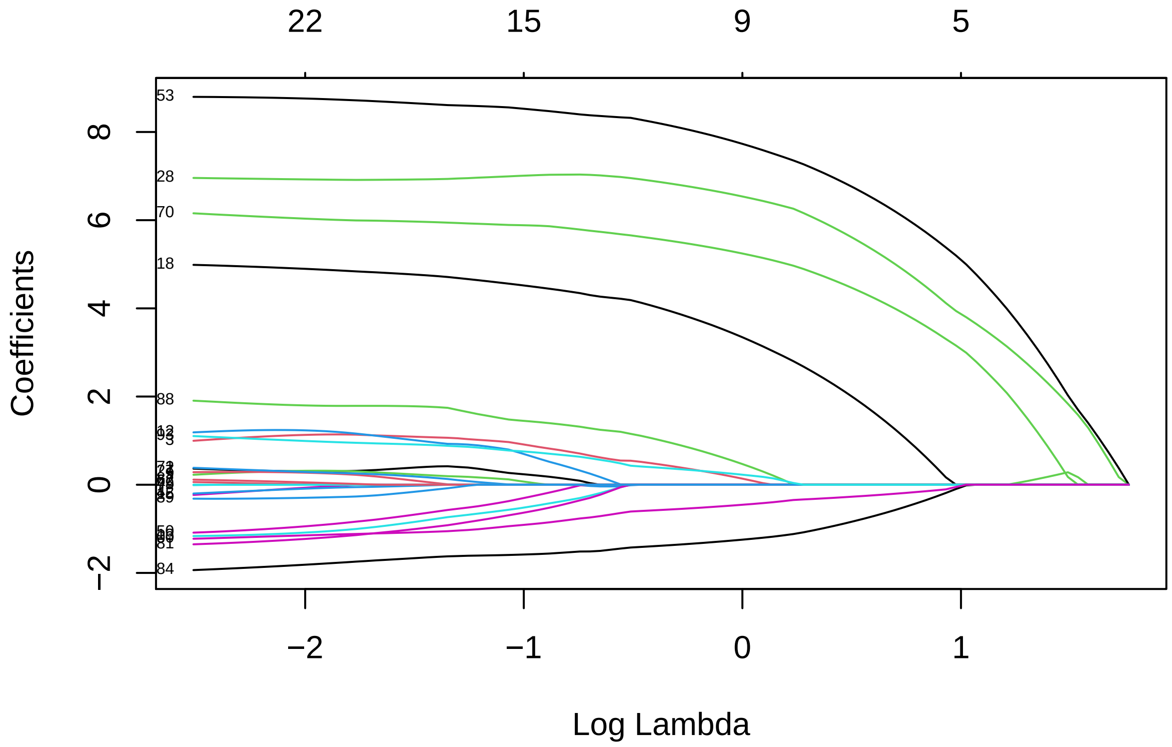
$$\hat{\beta}_2 \quad = \quad argmin_\beta ||y - X\beta||^2$$

Solution

b) We analyse a genomic dataset with 25 animals which are genotyped at 100 SNP-locations. Only 5 SNPs have an effect on the observed trait. The SNP-effects are estimated with LASSO. The results of this analysis are shown in the two plots below. The second plot can be used to determine the penalty-term (Log Lambda), such that a minimum number of SNP-effects are considered and such that the mean-squared error is still as small as possible (right dotted). Which are the 5 SNPs (indicate the numbers) with the largest absolute effects, when the penalty-term is determined based on the second plot.

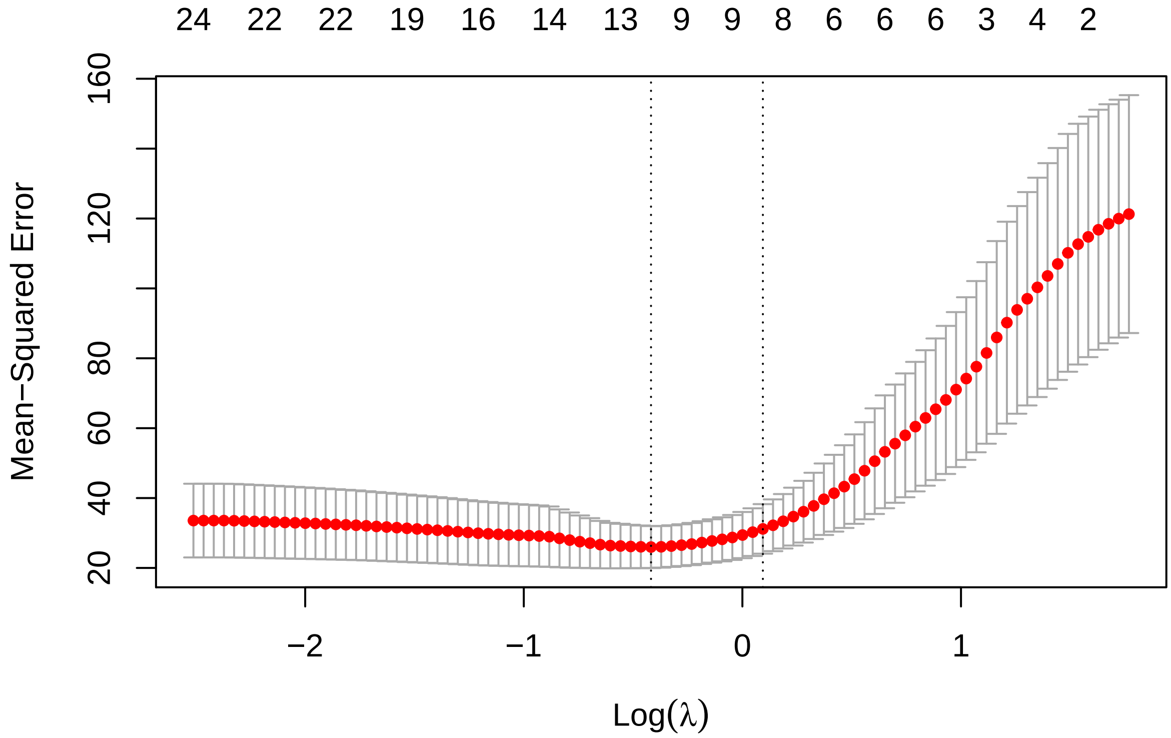
Wir analysieren einen genomischen Datensatz mit 25 Tieren, welche Daten an 100 SNP-Positionen aufweisen. Davon haben nur 5 SNP einen Effekt auf das gemessene Merkmal. Die SNP-Effekte werden mit LASSO geschätzt. Die Resultate sind in den nachfolgenden Plots gezeigt. Im zweiten Plot können wir den Strafterm (Log Lambda) so bestimmen, dass möglichst wenige SNPs berücksichtigt werden und dass gleichzeitig der mittlere quadrierte Fehler minimal bleibt (rechte gestrichelte Linie). Welches sind die 5 SNPs (bitte Nummern angeben) mit den grössten absoluten Effekten, wenn wir den Strafterm aufgrund des zweiten Plots bestimmen.

5



The penalty-term can be determined based on the right dotted line.

Der Strafterm kann aufgrund der rechten gestrichelten Linie bestimmt werden.



Solution

Master Solution

- According to the simulation, the following SNP were used

```
vec_sign_snp_idx
```

```
## [1] 71 53 70 18 28
```

- From the diagram, the marker 53, 28, 70, 84 and 18 had the highest absolute effect.
- The fit

```
summary(cvfitsnp)
```

```
##            Length Class  Mode
## lambda     93     -none- numeric
## cvm        93     -none- numeric
## cvsd       93     -none- numeric
## cvup       93     -none- numeric
## cvlo       93     -none- numeric
## nzero      93     -none- numeric
## call        4     -none- call
## name        1      -none- character
## glmnet.fit 12      elnet  list
## lambda.min  1      -none- numeric
## lambda.1se  1      -none- numeric
## index       2      -none- numeric
```

c)

0

Solution

Problem 4: Genomic BLUP

The data shown in the following table is available to predict genomic breeding values using different methods. The data is available from the URL shown below:

Die Daten in der nachfolgenden Tabelle sollen für die Schätzung von genomischen Zuchtwerten mit verschiedenen Methoden verwendet werden. Die Daten können vom folgenden URL heruntergeladen werden:

https://charlotte-ngs.github.io/gelasmss2021/data/asm_exam_p04.csv.

Animal	Sire	Dam	Sex	Observation	SNP1	SNP2
1	NA	NA	M	34.4	0	1
2	NA	NA	M	47.7	1	1
3	NA	NA	F	28.0	0	0
4	NA	NA	F	25.9	0	0
5	1	3	M	16.4	-1	0
6	1	4	F	30.6	0	1
7	2	3	F	46.0	1	1
8	2	3	F	25.9	0	0
9	5	7	F	30.6	0	1
10	5	8	F	12.6	-1	0

a) Use the two-step approach to predict genomic breeding values. Because, the number of SNP is smaller than the number of animals, marker effects can be estimated using least squares. Please indicate the type of model and specify the all the model components used to estimated marker effects. Also, describe how the genomic breeding values are computed from the marker effects.

   Verwenden Sie die Zwei-Schritt Methode zur Schätzung der genomischen Zuchtwerte. Da die Anzahl SNP kleiner ist als die Anzahl Tiere im Datensatz können die Markereffekte mit Least Squares geschätzt werden. Bitte geben Sie den Modell-Typ an und spezifizieren Sie alle Komponenten des Modells, welches zur Schätzung der Markereffekte verwendet wird. Beschreiben Sie auch, wie Sie aus den Markereffekten die genomischen Zuchtwerte berechnen.

8

Solution

Master Solution

The two steps in the two-step approach are

1. Estimate Marker Effects
2. Predict genomic breeding values

Marker Effect Marker effects are estimated using a fixed linear effect model (model type). The observation y_i of animal i can be modelled as:

$$y_i = b_0 + b_{sex,i} + \alpha_{SNP_1,i} + \alpha_{SNP_2,i} + e_i = b_0 + b_{sex,i} + \sum_{j=1}^{2} \alpha_{SNP_j,i} + e_i$$

with b_0: intercept, $b_{sex,i}$ effect of the sex of animal i, $\alpha_{SNP_j,i}$ the allele substitution effect according to the genotype of animal i and SNP j, e_i is the random residual with variance σ_e^2.

Because the number of SNP is smaller than the number of animals, the marker effects can be estimated using least squares. Hence, the function `lm()` is used.

```
# read data
tbl_data_p04 <- readr::read_csv2(file = s_data_p04_url_path)
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Parsed with column specification:
## cols(
##   Animal = col_double(),
##   Sire = col_double(),
##   Dam = col_double(),
##   Sex = col_character(),
##   Observation = col_double(),
##   SNP1 = col_double(),
##   SNP2 = col_double()
## )
```

```
lm_mrk_eff <- lm(Observation ~ Sex + SNP1 + SNP2, data = tbl_data_p04)
summary(lm_mrk_eff)
```

```
##
## Call:
## lm(formula = Observation ~ Sex + SNP1 + SNP2, data = tbl_data_p04)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8328 -0.5341 -0.1629  0.3711  1.2672
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.7328     0.4693  56.964 1.97e-09 ***
## SexM          3.3293     0.6426   5.181  0.00205 **
## SNP1         14.0966     0.5866  24.030 3.41e-07 ***
## SNP2          4.1569     0.7603   5.467  0.00156 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8935 on 6 degrees of freedom
## Multiple R-squared:  0.9957, Adjusted R-squared:  0.9936
## F-statistic: 463.3 on 3 and 6 DF,  p-value: 1.735e-07
```

Prediction of Genomic Breeding Values The genomic breeding values are predicted by the matrix vector multiplication of the genotype matrix times the marker effects.

```
mat_geno_snp <- matrix(c(tbl_data_p04$SNP1, tbl_data_p04$SNP2), ncol = 2)
coef_mrk_eff <- effects(lm_mrk_eff)
vec_mrk_eff <- c(coef_mrk_eff[["SNP1"]], coef_mrk_eff[["SNP2"]])
mat_pred_gbv <- crossprod(t(mat_geno_snp), vec_mrk_eff)
# table
tbl_res_ts_gbv <- tibble::tibble(Animal = tbl_data_p04$Animal,
                                 `Predicted Breeding Value` = round(mat_pred_gbv[,1], digits = 4))
knitr::kable(tbl_res_ts_gbv)
```

Animal	Predicted Breeding Value
1	4.8849
2	-27.4651

Animal	Predicted Breeding Value
3	0.0000
4	0.0000
5	32.3500
6	4.8849
7	-27.4651
8	0.0000
9	4.8849
10	32.3500

Ranking of the animals

```
order(mat_pred_gbv[,1])
```

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

b) Use a single-step marker-effect model to predict breeding values using the data shown above. Please specify the model type and all the components of the marker-effect model and describe how genomic breeding values are predicted. The ration (λ) between the residual variance (σ_e^2) and the QTL-variance (σ_q^2) can assumed to be 1.

Verwenden Sie ein "Single-Step" Markereffektmodell für die Schätzung der genomischen Zuchtwerte. Bitte geben Sie den Modelltyp und alle Komponenten des Markereffektmodells and und beschreiben Sie, wie die genomischen Zuchtwerte berechnet werden.

15

Solution

Master Solution

In a single-step marker effect model a mixed linear effect model is used to predict marker effects. Marker effects are treated as random effects. Genomic breeding values are predicted by summing over the appropriate marker effects. The linear mixed effect model to predict the marker effects is as follows

$$y = Xb + Wq + e$$

where y is the vector of observations, b the vector of fixed effects, q the vector of random marker effects, e the vector of random residuals, X and W are the design-matrices. The matrix W encodes the genotypes where -1 corresponds to G_2G_2, 0 to G_1G_2 and 1 to G_1G_1, assuming G_1 is the positive allele.

Inserting the values of the dataset into the model components results in

$$y = \begin{bmatrix} 34.4 \\ 47.7 \\ 28 \\ 25.9 \\ 16.4 \\ 30.6 \\ 46 \\ 25.9 \\ 30.6 \\ 12.6 \end{bmatrix}$$

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

$$W = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} b_M \\ b_F \end{bmatrix}$$

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \end{bmatrix}$$

The mixed model equations

$$\left[\begin{array}{cc} X^TX & X^TW \\ W^TX & W^TW + \lambda * I \end{array} \right] \left[\begin{array}{c} \hat{b} \\ \hat{q} \end{array} \right] = \left[\begin{array}{c} X^Ty \\ W^Ty \end{array} \right]$$

Get the solution of the mixed model equations

```
lambda <- 1
mat_XtX <- crossprod(mat_X)
mat_XtW <- crossprod(mat_X, mat_W)
mat_WtX <- crossprod(mat_W, mat_X)
mat_WtWlambdaI <- crossprod(mat_W) + lambda * diag(1, nrow = n_nr_snp)
mat_coef <- rbind(cbind(mat_XtX, mat_XtW), cbind(mat_WtX, mat_WtWlambdaI))
mat_rhs <- rbind(crossprod(mat_X, vec_y), crossprod(mat_W, vec_y))
mat_sol <- solve(mat_coef, mat_rhs)
vec_sol_fix <- mat_sol[1:2,]
```

From the marker effects

```
n_nr_sol <- nrow(mat_sol)
mrk_eff <- mat_sol[(n_nr_sol - n_nr_snp + 1):n_nr_sol,]
mrk_eff
```

```
## [1] 11.047601  4.730996
```

the genomic breeding values are computed by the product $\hat{g} = W\hat{q}$

```
mat_gen_bv <- crossprod(t(mat_W), mrk_eff)
vec_sol_gbv <- round(mat_gen_bv[,1], digits = 4)
```

The solution for the fixed effects ($\hat{b}$) and for the predicted breeding values ($\hat{g}$) are

$$\hat{b} = \begin{bmatrix} 29.6793 \\ 26.4867 \end{bmatrix}$$

$$\hat{g} = \begin{bmatrix} 4.731 \\ 15.7786 \\ 0 \\ 0 \\ -11.0476 \\ 4.731 \\ 15.7786 \\ 0 \\ 4.731 \\ -11.0476 \end{bmatrix}$$

The ranking of the animals according to the predicted breeding value

```
order(vec_sol_gbv)
```

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

Setting the predicted breeding value of the first animal to 0 leads to

```
vec_sol_gbv - vec_sol_gbv[1]
```

```
##  [1]   0.0000  11.0476  -4.7310  -4.7310 -15.7786   0.0000  11.0476  -4.7310
##  [9]   0.0000 -15.7786
```

c) Use a single-step Genomic BLUP (GBLUP) model to predict genomic breeding values. Please specify the type of model used and also list all the model components of the GBLUP model used. The ration (λ) between the residual variance (σ_e^2) and the genetic variance (σ_g^2) can assumed to be 1.

Verwenden Sie ein "Ein-Schritt" genomisches BLUP (GBLUP) Modell zur Schätzung der genomischen Zuchtwerte. Bitte geben Sie den Modelltyp und alle Modellkomponenten an, welche im GBLUP-Modell vorkommen.

15

Solution

Master Solution

The GBLUP model is a mixed linear effect model where genomic breeding values are treated as random effects. The mixed linear effect is

$$y = Xb + Zg + e$$

where y is the vector of observations, b the vector of fixed effects, g the vector of random genomic breeding values, e the vector of random residuals, X and Z are the design-matrices.

$$y = \begin{bmatrix} 34.4 \\ 47.7 \\ 28 \\ 25.9 \\ 16.4 \\ 30.6 \\ 46 \\ 25.9 \\ 30.6 \\ 12.6 \end{bmatrix}$$

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

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} b_M \\ b_F \end{bmatrix}$$

$$g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \\ g_9 \\ g_{10} \end{bmatrix}$$

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \end{bmatrix}$$

The mixed model equations

$$\left[\begin{array}{cc} X^TX & X^TZ \\ Z^TX & Z^TZ + \lambda * H^{-1} \end{array} \right] \left[\begin{array}{c} \hat{b} \\ \hat{g} \end{array} \right] = \left[\begin{array}{c} X^Ty \\ Z^Ty \end{array} \right]$$

The genomic relationship matrix G is computed using the following function

```
computeMatGrm <- function(pmatData) {
  matData <- pmatData
  # check the coding, if matData is -1, 0, 1 coded, then add 1 to get to 0, 1, 2 coding
  if (min(matData) < 0) matData <- matData + 1
  # Allele frequencies, column vector of P and sum of frequency products
  freq <- apply(matData, 2, mean) / 2
  P <- 2 * (freq - 0.5)
  sumpq <- sum(freq*(1-freq))
  # Changing the coding from (0,1,2) to (-1,0,1) and subtract matrix P
  Z <- matData - 1 - matrix(P, nrow = nrow(matData),
                            ncol = ncol(matData),
                            byrow = TRUE)
  # Z%*%Zt is replaced by tcrossprod(Z)
  return(tcrossprod(Z)/(2*sumpq))
}
# take matrix with genotypes
mat_W <- matrix(data = c(tbl_data_p04$SNP1, tbl_data_p04$SNP2), ncol = 2)
# compute the genomic relationship matrix
mat_G <- computeMatGrm(pmatData = mat_W)
# use the numerator relationship matrix A to blend G into H
ped <- pedigreemm::pedigree(sire = tbl_data_p04$Sire, dam = tbl_data_p04$Dam, label = as.character(tbl_
mat_A <- as.matrix(pedigreemm::getA(ped = ped))
n_blend_fact <- 0.05
mat_H <- (1-n_blend_fact) * mat_G + n_blend_fact * mat_A
```

Get the solution of the mixed model equations

```r
lambda <- 1
mat_XtX <- crossprod(mat_X)
mat_XtZ <- crossprod(mat_X, mat_Z)
mat_ZtX <- crossprod(mat_Z, mat_X)
mat_ZtZlambdaHinv <- crossprod(mat_Z) + lambda * solve(mat_H)
mat_coef <- rbind(cbind(mat_XtX, mat_XtZ), cbind(mat_ZtX, mat_ZtZlambdaHinv))
mat_rhs <- rbind(crossprod(mat_X, vec_y), crossprod(mat_Z, vec_y))
mat_sol <- solve(mat_coef, mat_rhs)
n_nr_sol <- nrow(mat_sol)
vec_sol_fix <- mat_sol[1:2,]
vec_sol_gbv <- mat_sol[(n_nr_sol - n_nr_obs + 1):n_nr_sol, ]
```

The solution for the fixed effects ($\hat{b}$) and for the predicted breeding values ($\hat{g}$) are

$$\hat{b} = \begin{bmatrix} 32.078 \\ 28.8637 \end{bmatrix}$$

$$\hat{g} = \begin{bmatrix} 2.2721 \\ 13.6131 \\ -2.3625 \\ -2.4172 \\ -13.6194 \\ 2.2896 \\ 13.6338 \\ -2.352 \\ 2.3674 \\ -13.6049 \end{bmatrix}$$

```r
order(vec_sol_gbv)
```

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

Setting the predicted breeding value of the first animal to 0 leads to

```r
vec_sol_gbv - vec_sol_gbv[1]
```

```
##            1            2            3            4            5            6 
##   0.00000000  11.34107481  -4.63456794  -4.68922518 -15.89142974   0.01755275 
##            7            8            9           10 
##  11.36170494  -4.62404995   0.09535781 -15.87698754
```