

Applied Statistical Methods - Solution 8

AUTHOR
Peter von Rohr

PUBLISHED
April 29, 2024

WEBR STATUS
● Ready!

Problem 1: Interactions

Use the following dataset on `Breed`, `Breast.Circumference` and `Body.Weight` and fit a fixed linear effects model with `Body.Weight` as response and `Breed` and `Breast.Circumference` as predictors and include an interaction term between the two predictors. Compute the expected difference in `Body.Weight` for two animals which differ in `Breast.Circumference` by `1cm` for every `Breed`.

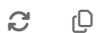
The dataset is available under

```
[1] "https://charlotte-ngs.github.io/asmasss2024/data/asm_bw_flem.csv"
```

Tasks

- Read the data for fitting the linear model

▶ Run Code



```
1 # read data
2 s_tbl_ex08_p01_path <- "https://charlotte-ngs.github.io/asmasss2024/"
3 df_bw_bc_br <- read.table(s_tbl_ex08_p01_path,
4                           header = T, sep = ",")
5 df_bw_bc_br
```

	Animal	Breast.Circumference	Body.Weight	BCS	HEI	Breed
1	1	176	471	5.0	161	Angus
2	2	177	463	4.2	121	Angus
3	3	178	481	4.9	157	Simmental
4	4	179	470	3.0	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

- Fitting the linear model

▶ Run Code



```
1 # lm with interactions
2 lm_bw_bc_br_int <- lm(Body.Weight ~ Breast.Circumference * Breed,
3                       data = df_bw_bc_br)
4 # show results with summary
5 smry_lm <- summary(lm_bw_bc_br_int)
6 smry_lm
```

Call:

```
lm(formula = Body.Weight ~ Breast.Circumference * Breed, data = df_bw_bc_br)
```

Residuals:

```
  1      2      3      4      5      6      7      8      9     10
3.286 -4.929 -3.333  1.643  6.667 -3.333  8.200 -5.600 -13.400 10.800
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	430.0000	917.1235	0.469	0.664
Breast.Circumference	0.2143	5.1716	0.041	0.969
BreedLimousin	-1151.0000	1293.2741	-0.890	0.424
BreedSimmental	-835.6667	1685.4451	-0.496	0.646
Breast.Circumference:BreedLimousin	6.5857	7.1908	0.916	0.412
Breast.Circumference:BreedSimmental	4.7857	9.4420	0.507	0.639

Residual standard error: 11.17 on 4 degrees of freedom

Multiple R-squared: 0.9103, Adjusted R-squared: 0.7981

F-statistic: 8.115 on 5 and 4 DF, p-value: 0.03212

- Expected difference in body weight for the three breeds:

Angus: The expected difference in body weight (in kg) of one centimeter increase in breast circumference corresponds to the regression coefficient of Breast.Circumference and is

▶ Run Code



```
1 # show estimate for BC
2 smry_lm$coefficients["Breast.Circumference", "Estimate"]
[1] 0.2142857
```

Limousin: Because, for the breed limousin, there is an interaction effect. We have to add the regression coefficient of Breast.Circumference to the interaction effect Breast.Circumference:BreedLimousin. From this we get

▶ Run Code



```
1 # add slope plus interaction effect for LI
2 delta_bw_li <- smry_lm$coefficients["Breast.Circumference", "Estimate"] +
3   smry_lm$coefficients["Breast.Circumference:BreedLimousin", "Estimate"]
4 delta_bw_li
[1] 6.8
```

Simmental: The same as for limousin, we have for simmental

▶ Run Code



```
1 # add slope plus interaction effect for SI
2 delta_bw_si <- smry_lm$coefficients["Breast.Circumference", "Estimate"] +
3   smry_lm$coefficients["Breast.Circumference:BreedSimmental", "Estimate"]
4 delta_bw_si
[1] 5
```

Problem 2: Simulation

Use the following values for intercept and regression slope for Body.Weight on Breast.Circumference to simulate a dataset of size N . What is the number for N that has to be chosen such that the regression analysis of the simulated data gives the same result as the true regression slope.

The true values are:

- Intercept: -1070
- Regression slope: 8.7
- Residual standard error: 12

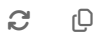
Hints

- Start with $N = 10$, simulate a dataset and analyse the data with `lm()`
- If the result (rounded to 1 digits after decimal point) is not the same then double the size of the dataset, hence use, $N = 20$
- Continue until you get close to the true value.
- Assume that the random residuals follow a normal distribution with mean zero and standard deviation equal to 12
- Take breast circumference to be normally distributed with a mean of 180 and a standard deviation of 2.6
- Use a linear regression model with an intercept to model expected body weight based on breast circumference.

Tasks

- Assign numbers given in problem description into variables

▶ Run Code



```
1 # intercept and slope
2 n_b_intercept <- -1070
3 n_b_slope <- 8.7
4 # residual standard deviation
5 n_res_std_error <- 12
6 # number of observations
7 n_nr_obs <- 10
8 # mean and sd of BC
9 n_mean_bc <- 180
10 n_sd_bc <- 2.6
```

- Start with $N = 10$ and first generate the matrix X which consists of a column of all ones and a column of breast circumference values in centimeter taken from the given normal distribution. Whenever, we generate some random numbers it is important to first set the seed with the function `set.seed()` to which an integer number is passed. This makes sure that when repeating the simulation the same results are generated.

▶ Run Code



```
1 # start by setting the seed
2 set.seed(2904)
3
4 vec_bc <- rnorm(n_nr_obs, mean = n_mean_bc, sd = n_sd_bc)
5 mat_X <- matrix(c(rep(1,n_nr_obs), vec_bc), ncol = 2)
6 mat_X
```

```
  [,1]  [,2]
[1,]  1 179.7794
[2,]  1 182.4258
[3,]  1 175.0342
[4,]  1 181.3524
[5,]  1 183.4688
[6,]  1 177.5291
[7,]  1 179.5569
[8,]  1 182.4912
```

```
[9,] 1 182.0924
[10,] 1 178.3622
```

- Simulate observations of `Body.Weight`

▶ Run Code



```
1 # vectors for intercept and slope and a vector for observations
2 vec_b <- c(n_b_intercept, n_b_slope)
3 vec_y <- crossprod(t(mat_X), vec_b) +
4   rnorm(n_nr_obs, mean=0, sd=n_res_std_error)
5 vec_y
```

```
      [,1]
[1,] 505.7876
[2,] 525.0057
[3,] 462.7660
[4,] 504.0728
[5,] 522.4016
[6,] 492.2186
[7,] 513.0021
[8,] 522.7654
[9,] 539.0176
[10,] 490.0466
```

- Analyse the simulated data with a regression model

▶ Run Code



```
1 df_bw_bc_sim <- data.frame(Body.Weight = vec_y,
2                             Breast.Circumference=vec_bc)
3 lm_bw_bc_sim <- lm(Body.Weight ~ Breast.Circumference,
4                   data = df_bw_bc_sim)
5 lm_bw_bc_sim
```

Call:

```
lm(formula = Body.Weight ~ Breast.Circumference, data = df_bw_bc_sim)
```

Coefficients:

```
(Intercept)  Breast.Circumference
      -851.239             7.541
```

- Compute absolute value of deviation between regression and simulation

▶ Run Code



```
1 # use function abs()
2 abs(lm_bw_bc_sim$coefficients[["Breast.Circumference"]] - n_b_slope)
```

```
[1] 1.15906
```

- Use a loop to iteratively increase the number of observations until the absolute deviation of the estimated slope from the true value becomes smaller than 0.1.

▶ Run Code



```
1 n_slope_tol <- 0.1
2 n_max_iter <- 10
3 n_iter_round <- 0
4 n_abs_dev <- abs(lm_bw_bc_sim$coefficients[["Breast.Circumference"]]
```

```
5 while(n_abs_dev > n_slope_tol &&
6       n_iter_round < n_max_iter){
7   # count number of iterations and
8   # determine number of observations
9   n_iter_round <- n_iter_round + 1
10  n_nr_obs <- 2 * n_nr_obs
11  # simulate breast circumference
12  vec_bc <- rnorm(n_nr_obs, mean = n_mean_bc, sd = n_sd_bc)
13  mat_X <- matrix(c(rep(1,n_nr_obs), vec_bc), ncol = 2)
14  # simulate body weight
15  vec_y <- crossprod(t(mat_X), vec_b) +
16    rnorm(n_nr_obs, mean=0, sd=n_res_std_error)
17  # analyse simulated data
18  df_bw_bc_sim <- data.frame(Body.Weight = vec_y,
19                            Breast.Circumference=vec_bc)
20  lm_bw_bc_sim <- lm(Body.Weight ~ Breast.Circumference,
21                    data = df_bw_bc_sim)
22  n_abs_dev <- abs(lm_bw_bc_sim$coefficients[["Breast.Circumference"]])
23  # results
24  cat(" * Iteration: ", n_iter_round, "\n")
25  cat(" * Number of observations: ", n_nr_obs, "\n")
26  cat(" * Regression slope: ", lm_bw_bc_sim$coefficients[["Breast.Ci
27
28 }
```

```
* Iteration: 1
* Number of observations: 20
* Regression slope: 9.505973
* Iteration: 2
* Number of observations: 40
* Regression slope: 9.100837
* Iteration: 3
* Number of observations: 80
* Regression slope: 9.777747
* Iteration: 4
* Number of observations: 160
* Regression slope: 8.878056
* Iteration: 5
* Number of observations: 320
* Regression slope: 8.601509
```