Applied Statistical Methods - Solution 6

Peter von Rohr

2022-03-30

Problem 1: Experiment Evaluation

In the paper by (Manzocchi et al. 2020) 4 different feeding treatments for dairy cows were compared. The different feeding treatments consisted of

Treatment	Feed
1	hay
2	grass-silage
3	maize silage
4	shredlage

From the results section of the paper, the values for energy corrected milk (ECM in kg/day) and coagulation time (CGT in min) of the milk are shown in the table below.

Treatment	ECM	CGT
1	24.3	11.0
2	23.6	10.6
3	25.0	10.5
4	23.8	10.3

The standard errors of the means (SEM) for the above reported target variables were

Response	SEM
ECM	1.18
CGT	0.60

The real experiment is designed according to an incomplete latin square where in two runs groups of six cows were assigned to each of the four treatments. For the purpose of this exercise, we simplify the experimental design and assume that groups of 6 cows were assigned to the treatments all at the same time. The paper mentions that besides of the treatment numerous fixed effects (experimental run and interactions between treatments and experimental runs) and covariates (lactation stage) were considered. But unfortunately, no estimates for the different effects were given. Hence we are assuming that the treatment is the major effect on our responses.

Your Tasks

- Simulate a dataset with 6 cows per treatment and assign to each of the cows a value for the two responses ECM and CGT with mean values shown in the table above and with standard deviation equal to the SEM values.

- Analyse the dataset with a fixed linear effect model
- Verify whether you can reproduce the results of the paper
- Think about what type of contrasts are ideal for this type of dataset.

Solution

- Simulate dataset similar to the one in the paper for ECM and CGT for the four treatments. The dataset for ECM and CGT contains four columns, the ID for the cow, the treatment and the two response variables. We use the function `get_treatment_data()` to generate the data for the 6 for a given treatment.

```
get_treatment_data <- function(pn_treatment, pn_nr_cow,
                               pn_mean_ecm, pn_sd_ecm,
                               pn_mean_cgt, pn_sd_cgt){
  tbl_result <- tibble::tibble(Cow = c(((pn_treatment-1)*pn_nr_cow + 1):(pn_treatment*pn_nr_cow)),
                               Treatment = rep(pn_treatment, pn_nr_cow),
                               ECM = rnorm(pn_nr_cow, mean = pn_mean_ecm, sd = pn_sd_ecm),
                               CGT = rnorm(pn_nr_cow, mean = pn_mean_cgt, sd = pn_sd_cgt))
  return(tbl_result)
}
```

The function `get_treatment_data()` can be used to generate a dataset for each treamtent. These datasets can be combined to one dataset

```
set.seed(76281)
tbl_ecm_cgt_t1 <- get_treatment_data(pn_treatment = 1,
                                     pn_nr_cow = n_nr_cow_per_group,
                                     pn_mean_ecm = tbl_ecm$ECM[1],
                                     pn_sd_ecm = tbl_sem$SEM[1],
                                     pn_mean_cgt = tbl_ecm$CGT[1],
                                     pn_sd_cgt = tbl_sem$SEM[2])
tbl_ecm_cgt_t2 <- get_treatment_data(pn_treatment = 2,
                                     pn_nr_cow = n_nr_cow_per_group,
                                     pn_mean_ecm = tbl_ecm$ECM[2],
                                     pn_sd_ecm = tbl_sem$SEM[1],
                                     pn_mean_cgt = tbl_ecm$CGT[2],
                                     pn_sd_cgt = tbl_sem$SEM[2])
tbl_ecm_cgt_t3 <- get_treatment_data(pn_treatment = 3,
                                     pn_nr_cow = n_nr_cow_per_group,
                                     pn_mean_ecm = tbl_ecm$ECM[3],
                                     pn_sd_ecm = tbl_sem$SEM[1],
                                     pn_mean_cgt = tbl_ecm$CGT[3],
                                     pn_sd_cgt = tbl_sem$SEM[2])
tbl_ecm_cgt_t4 <- get_treatment_data(pn_treatment = 4,
                                     pn_nr_cow = n_nr_cow_per_group,
                                     pn_mean_ecm = tbl_ecm$ECM[4],
                                     pn_sd_ecm = tbl_sem$SEM[1],
                                     pn_mean_cgt = tbl_ecm$CGT[4],
                                     pn_sd_cgt = tbl_sem$SEM[2])
tbl_ecm_cgt <- dplyr::bind_rows(tbl_ecm_cgt_t1,
                                tbl_ecm_cgt_t2,
                                tbl_ecm_cgt_t3,
                                tbl_ecm_cgt_t4)
```

The first three and the last three records of the genrated dataset are shown below

Cow	Treatment	ECM	CGT
1	1	24.27746	12.47980
2	1	22.82048	10.93098
3	1	26.62298	11.25025
22	4	23.28888	10.69942
23	4	22.83562	10.65387
24	4	23.47930	10.27712

- Analyse the dataset with `lm()`. For the analysis with `lm()`, it is important to convert the treatments to the datatype `factor`. This can be done via the conversion function `as.factor()`. Alternatively, it can also be specified in the formula argument to `lm()`.

```
lm_ecm_treat <- lm(ECM ~ factor(Treatment), data = tbl_ecm_cgt)
summary(lm_ecm_treat)
```

```
##
## Call:
## lm(formula = ECM ~ factor(Treatment), data = tbl_ecm_cgt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6873 -0.7700 -0.1968  0.5652  2.2981
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         24.5078     0.5002  48.998   <2e-16 ***
## factor(Treatment)2  -0.2421     0.7074  -0.342    0.736
## factor(Treatment)3  -0.3170     0.7074  -0.448    0.659
## factor(Treatment)4  -0.4837     0.7074  -0.684    0.502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.225 on 20 degrees of freedom
## Multiple R-squared:  0.02366,    Adjusted R-squared:  -0.1228
## F-statistic: 0.1615 on 3 and 20 DF,  p-value: 0.921
```

Similarly for `CGT`

```
lm_cgt_treat <- lm(CGT ~ factor(Treatment), data = tbl_ecm_cgt)
summary(lm_cgt_treat)
```

```
##
## Call:
## lm(formula = CGT ~ factor(Treatment), data = tbl_ecm_cgt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.19920 -0.31062 -0.04981  0.33162  1.23768
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         11.2421     0.2460  45.697  < 2e-16 ***
## factor(Treatment)2  -1.0181     0.3479  -2.926  0.00835 **
## factor(Treatment)3  -1.1622     0.3479  -3.340  0.00326 **
```

```
## factor(Treatment)4  -1.1895     0.3479  -3.419  0.00272 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6026 on 20 degrees of freedom
## Multiple R-squared:  0.4431, Adjusted R-squared:  0.3596
## F-statistic: 5.305 on 3 and 20 DF,  p-value: 0.007446
```

- Compare your results with the results from the paper. The results of our `lm()` - analysis did confirm the results of the paper. For ECM the comparison is shown in the following table

Treatment	ECM	ECM_LM_Results
1	24.3	24.50776
2	23.6	24.26570
3	25.0	24.19079
4	23.8	24.02401

Except for treatment 3, all levels show estimates that go in the correct direction. The same can be done with CGT

Treatment	CGT	CGT_LM_Results
1	11.0	11.24212
2	10.6	10.22405
3	10.5	10.07994
4	10.3	10.05257

For CGT, the effect sizes from the LM-analysis are larger compared to what was reported in the paper.

- What are the best contrast for our analysis. Treatment contrasts which are used by default, can be used here. Then we are assuming that treatment 1 which corresponds to "hay" feeding is assigned to be the control feeding and all other feeding strategies are interpreted as experimental treatments.

Problem 2: Significance and Size of Dataset

For some of the LM-analyses done in Problem 1, the results might not be significant. The same was also true in the paper. Their reported results were also declared to be non-significant. This might have two reasons.

1. Either the generated dataset is just a "bad" example due to the unfortunate random numbers that were drawn or
2. The size of the dataset is too small.

Check both reasons by implementing the following tasks

Your Tasks

- Repeat the simulation 30 times and check how many times a significant effect of one of the treatments can be reported.
- Increase the size of the dataset until one of the treatment effect is significant.

Solution

- Create a loop that runs over all repetitions and does the analysis as shown in the solution of Problem 1. For each of the repetitions, a new dataset is generated. This is done as shown in the solution of Problem 1 using the function `get_treatment_data()`.

```
get_treatment_data <- function(pn_treatment, pn_nr_cow,
                               pn_mean_ecm, pn_sd_ecm,
                               pn_mean_cgt, pn_sd_cgt){
  tbl_result <- tibble::tibble(Cow = c(((pn_treatment-1)*pn_nr_cow + 1):(pn_treatment*pn_nr_cow)),
                               Treatment = rep(pn_treatment, pn_nr_cow),
                               ECM = rnorm(pn_nr_cow, mean = pn_mean_ecm, sd = pn_sd_ecm),
                               CGT = rnorm(pn_nr_cow, mean = pn_mean_cgt, sd = pn_sd_cgt))
  return(tbl_result)
}
```

Instead of combining the dataset with all the treatments with sequential statements, we are doing that in a loop and are encapsulating that in a further function called `get_ecm_cgt_data()`.

```
get_ecm_cgt_data <- function(pn_nr_treatment, pn_nr_cow,
                             pvec_mean_ecm, pvec_mean_cgt,
                             pvec_sem){
  tbl_result <- NULL
  # loop over treatments
  for (i in 1:pn_nr_treatment){
    tbl_cur <- get_treatment_data(pn_treatment = i,
                                  pn_nr_cow    = pn_nr_cow,
                                  pn_mean_ecm  = pvec_mean_ecm[i],
                                  pn_sd_ecm    = pvec_sem[1],
                                  pn_mean_cgt  = pvec_mean_cgt[i],
                                  pn_sd_cgt    = pvec_sem[2])
    if (is.null(tbl_result)){
      tbl_result <- tbl_cur
    } else {
      tbl_result <- dplyr::bind_rows(tbl_result, tbl_cur)
    }
  }
  # return result
  return(tbl_result)
}
```

A single dataset can be generated with

```
tbl_ecm_cgt_data <- get_ecm_cgt_data(pn_nr_treatment = n_nr_feed,
                                     pn_nr_cow = n_nr_cow_per_group,
                                     pvec_mean_ecm = tbl_ecm$ECM,
                                     pvec_mean_cgt = tbl_ecm$CGT,
                                     pvec_sem = tbl_sem$SEM)
```

- Inside of the loop store the results of `lm()` in a new dataframe. In a loop over the number of repetitions, a new dataset is generated and analysed

```
set.seed(2443)
tbl_lm_result <- NULL
for (i in 1:n_nr_rep){
  # generate data
  tbl_ecm_cgt_data <- get_ecm_cgt_data(pn_nr_treatment = n_nr_feed,
                                       pn_nr_cow = n_nr_cow_per_group,
                                       pvec_mean_ecm = tbl_ecm$ECM,
                                       pvec_mean_cgt = tbl_ecm$CGT,
                                       pvec_sem = tbl_sem$SEM)
  # analyse data with lm
```

```
  smry_lm_ecm <- summary(lm(ECM ~ factor(Treatment), data = tbl_ecm_cgt_data))
  coef_mat <- smry_lm_ecm$coefficients
  # collect results
  tbl_coef <- dplyr::bind_cols(tibble::tibble(Repetition = c(rep(i, nrow(coef_mat)))),
                               tibble::tibble(RowNames = row.names(coef_mat)),
                               tibble::as_tibble(coef_mat))
  if (is.null(tbl_lm_result)){
    tbl_lm_result <- tbl_coef
  } else {
    tbl_lm_result <- dplyr::bind_rows(tbl_lm_result, tbl_coef)
  }

}
```

- Investigate the result dataframe and check how many times a significant result was obtained. From the result dataframe, we want to filter out all the treatment factor level estimates that have a significance level less than 0.01. This is done using the `filter()` function from the `dplyr` package. Two and more `filter()` - steps can be combined using the pipe-operator `%>%`.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
tbl_sig_result <- tbl_lm_result %>%
  filter(RowNames != "(Intercept)") %>%
  filter(`Pr(>|t|)` < n_sig_level)
tbl_sig_result
```

```
## # A tibble: 2 x 6
##   Repetition RowNames           Estimate `Std. Error` `t value` `Pr(>|t|)`
##        <int> <chr>                 <dbl>        <dbl>     <dbl>      <dbl>
## 1          6 factor(Treatment)3     1.76        0.549      3.21    0.00442
## 2         23 factor(Treatment)3     1.67        0.571      2.93    0.00828
```

Hence from a total of 30 repetitions of the analysis of the data, only 2 effect estimates were significantly different from 0 at a significance level of 0.01. This is not very frequence. Hence the chosen experimental design does not lead to a high chance to find significant results of the shown magnitude.

- Double the number of observations until significant results can be found. We start with an analysis of a given number of observations. The dataset for this initial analysis can be read from the following address

```
## https://charlotte-ngs.github.io/asmss2022/data/asm_sim_ecm_cgt.csv
```

The initial analysis is done as follows

```
tbl_ecm_cgt_base <- readr::read_csv(file = s_ecm_cgt_path)
```

```
## Rows: 24 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (4): Cow, Treatment, ECM, CGT

##
## 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.
```

```
lm_ecm_cgt_base <- lm(ECM ~ factor(Treatment), data = tbl_ecm_cgt_base)
smry_ecm_cgt_base <- summary(lm_ecm_cgt_base)
```

From the results of the lm-analysis, we are extracting the significance levels

```
mat_coef_ecm_cgt_base <- smry_ecm_cgt_base$coefficients
vec_prt_treat <- mat_coef_ecm_cgt_base[2:nrow(mat_coef_ecm_cgt_base), "Pr(>|t|)"]
vec_prt_treat
```

```
## factor(Treatment)2 factor(Treatment)3 factor(Treatment)4
##          0.7357675          0.6588973          0.5019011
```

As long as all of those levels are below a given significance level, we double the size of the data set

```
set.seed(3098)
vec_prt_treat <- rep(1, length(vec_prt_treat))
n_sig_level <- 0.01
n_nr_iter_max <- 10
n_iter_count <- 1
n_nr_cow_per_group <- 6
while (all(vec_prt_treat > n_sig_level) && n_iter_count < n_nr_iter_max){
  # generate data
  tbl_ecm_cgt_data <- get_ecm_cgt_data(pn_nr_treatment = n_nr_feed,
                                       pn_nr_cow = n_nr_cow_per_group,
                                       pvec_mean_ecm = tbl_ecm$ECM,
                                       pvec_mean_cgt = tbl_ecm$CGT,
                                       pvec_sem = tbl_sem$SEM)
  # analyse data with lm
  smry_lm_ecm <- summary(lm(ECM ~ factor(Treatment), data = tbl_ecm_cgt_data))
  mat_coef_ecm_cgt <- smry_lm_ecm$coefficients
  vec_prt_treat <- mat_coef_ecm_cgt[2:nrow(mat_coef_ecm_cgt), "Pr(>|t|)"]
  # increment count
  n_iter_count <- n_iter_count + 1
  # double the number of couws per treatment
  n_nr_cow_per_group <- n_nr_cow_per_group * 2
}
cat(" * Iteration count: ", n_iter_count, "\n")
```

```
##  * Iteration count:  3
```

```
cat(" * Size of dataset: ", n_nr_cow_per_group, "\n")
```

```
##  * Size of dataset:  24
```

```
cat(" * Summary: ")
```

```
##  * Summary:
```

```
smry_lm_ecm
```

```
##
```

```
## Call:
## lm(formula = ECM ~ factor(Treatment), data = tbl_ecm_cgt_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3945 -0.8580  0.1370  0.6896  2.3232
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          24.0917     0.3265  73.798  < 2e-16 ***
## factor(Treatment)2  -0.3694     0.4617  -0.800  0.42787
## factor(Treatment)3   1.5914     0.4617   3.447  0.00126 **
## factor(Treatment)4  -0.4162     0.4617  -0.902  0.37221
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.131 on 44 degrees of freedom
## Multiple R-squared:  0.3637, Adjusted R-squared:  0.3203
## F-statistic: 8.382 on 3 and 44 DF,  p-value: 0.0001614
```

References

Manzocchi, Elisa, Werner Hengartner, Michael Kreuzer, and Katrin Giller. 2020. "Effect of feeding hay vs. silages of various types to dairy cows on feed intake, milk composition and coagulation properties." *Journal of Dairy Research* 87 (3): 334–40. https://doi.org/10.1017/S0022029920000801.