Using lm with asremlPlus for the Ladybird example from Welham et al. (2014)

Chris Brien

20 January, 2020

Introduction

Welham et al. (2014, Example 8.2) is an experiment tot answer the question "Will ladybirds transfer fungus to aphids on plants?" The experiment consists of 2 runs of 36 containers, each with a plant and aphids. There are three factors that results in 12 treatments: Host plant (beans, trefoil), infected Cadavers (5, 10, 20), Ladybird (-, +). Ther are randomized to the containers within a run so that each is replicated 3 times within a run. The respose to be analysed is the logit of the proportion of live aphids that were infected.

In this vignette the packages `lmerTest` (Kuznetsova et al., 2017), `emmeans` (Lenth, 2019), `asremlPlus` (Brien, 2020a) and `dae` (Brien, 2020b), packages for the R Statistical Computing environment (R Core Team, 2020), are used in analyzing the ladybird experiment.

Initialize

```
library(lmerTest)
```

```
## Loading required package: lme4

## Loading required package: Matrix

##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step
```

```
library(emmeans)
```

```
## Welcome to emmeans.
## NOTE -- Important change from versions <= 1.41:
##     Indicator predictors are now treated as 2-level factors by default.
##     To revert to old behavior, use emm_options(cov.keep = character(0))
```

```
library(asremlPlus)
```

```
## ASReml-R needs to be loaded if the mixed-model functions are to be used.
##
```

```
## ASReml-R is available from VSNi. Please visit http://www.vsni.co.uk/ for more information.
```

```r
library(dae)
```

```
## Loading required package: ggplot2
```

```r
options(width = 95, show.signif.stars = FALSE)
```

Get data available in asremlPlus

```r
data("Ladybird.dat")
```

Do an ANOVA of logits

```r
Ladybird.aov <- aov(logitP ~ Host*Cadavers*Ladybird + Error(Run/Plant),
                    data=Ladybird.dat)
summary(Ladybird.aov)
```

```
##
## Error: Run
##           Df  Sum Sq Mean Sq F value Pr(>F)
## Residuals  1 0.06766 0.06766
##
## Error: Run:Plant
##                        Df Sum Sq Mean Sq F value   Pr(>F)
## Host                    1 13.599  13.599  59.172 1.82e-10
## Cadavers                2 17.027   8.514  37.044 3.78e-11
## Ladybird                1 11.091  11.091  48.257 3.33e-09
## Host:Cadavers           2  0.308   0.154   0.670   0.5158
## Host:Ladybird           1  0.228   0.228   0.992   0.3234
## Cadavers:Ladybird       2  1.735   0.867   3.774   0.0287
## Host:Cadavers:Ladybird  2  0.200   0.100   0.435   0.6493
## Residuals              59 13.560   0.230
```

The anova table gives the F-tests for the three-factor effects and interactions. Note the `Residuals Mean Sq` value for `Run:Plant` of 0.230. Also, it is clear that the Run component is negative, given that the `Residuals Mean Sq` value for `Run` is less than that for `Run:Plant`; it is (0.06766 – 0.230) / 36). From the table it is seen that the only significant interaction is Cadavers:Ladybird and that the Host main effect is significant.

Use lmerTest and lm to analyse the logits

Mixed model analysis of logits

```r
m1.lmer <- lmerTest::lmer(logitP ~ Host*Cadavers*Ladybird + (1|Run),
                          data=Ladybird.dat)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
summary(m1.lmer)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
```

```
## Formula: logitP ~ Host * Cadavers * Ladybird + (1 | Run)
##    Data: Ladybird.dat
##
## REML criterion at convergence: 102.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9633 -0.5217  0.1360  0.5789  2.1896
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Run      (Intercept) 0.0000   0.0000
##  Residual             0.2271   0.4766
## Number of obs: 72, groups:  Run, 2
##
## Fixed effects:
##                                  Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                     -1.603097   0.194560 60.000000  -8.240 1.91e-11
## Hosttrefoil                     -0.870675   0.275149 60.000000  -3.164  0.00244
## Cadavers10                       0.564771   0.275149 60.000000   2.053  0.04448
## Cadavers20                       0.919229   0.275149 60.000000   3.341  0.00144
## Ladybird+                        0.547710   0.275149 60.000000   1.991  0.05109
## Hosttrefoil:Cadavers10          -0.212735   0.389120 60.000000  -0.547  0.58661
## Hosttrefoil:Cadavers20          -0.120410   0.389120 60.000000  -0.309  0.75806
## Hosttrefoil:Ladybird+            0.073153   0.389120 60.000000   0.188  0.85151
## Cadavers10:Ladybird+            -0.040048   0.389120 60.000000  -0.103  0.91837
## Cadavers20:Ladybird+             0.414204   0.389120 60.000000   1.064  0.29138
## Hosttrefoil:Cadavers10:Ladybird+  0.005698   0.550299 60.000000   0.010  0.99177
## Hosttrefoil:Cadavers20:Ladybird+  0.449979   0.550299 60.000000   0.818  0.41676
##
## Correlation of Fixed Effects:
##             (Intr) Hsttrf Cdvr10 Cdvr20 Ldybr+ Hs:C10 Hs:C20 Hst:L+ C10:L+ C20:L+ H:C10:
## Hosttrefoil -0.707
## Cadavers10  -0.707  0.500
## Cadavers20  -0.707  0.500  0.500
## Ladybird+   -0.707  0.500  0.500  0.500
## Hsttrfl:C10  0.500 -0.707 -0.707 -0.354 -0.354
## Hsttrfl:C20  0.500 -0.707 -0.354 -0.707 -0.354  0.500
## Hsttrfl:Ld+  0.500 -0.707 -0.354 -0.354 -0.707  0.500  0.500
## Cdvrs10:Ld+  0.500 -0.354 -0.707 -0.354 -0.707  0.500  0.250  0.500
## Cdvrs20:Ld+  0.500 -0.354 -0.354 -0.707 -0.707  0.250  0.500  0.500  0.500
## Hstt:C10:L+ -0.354  0.500  0.500  0.250  0.500 -0.707 -0.354 -0.707 -0.707 -0.354
## Hstt:C20:L+ -0.354  0.500  0.250  0.500  0.500 -0.354 -0.707 -0.707 -0.354 -0.707  0.500
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

As expected the Run component is bound at zero, leading to a singular model. This results in a change in the estimate of the residual variance to 0.227. To allow for a negative estimate we will redo the analysis with `Run` fixed, because with `lme4` (`lmerTest`) one cannot unconstrain the `Run` component to allow it to be negative. As Littell et al. (2006, p.150) say

> if you do not set the negative variance component estimate to zero, but allow it to remain negative, you get better control over Type I error and, for cases of negative wholeplot error variance estimates, greater power. Therefore, this is the recommended procedure.

Analyse with Reps fixed using `lm` to make the analysis equivalent to ANOVA

The function `lm` has to be used because there are no random terms; `lme4` cannot be used because it requires at least one random term.

```
m.lm <- lm(logitP ~ Run + Host*Cadavers*Ladybird,
                         data=Ladybird.dat)
(aov.tab <- anova(m.lm))
```

```
## Analysis of Variance Table
##
## Response: logitP
##                        Df  Sum Sq Mean Sq F value    Pr(>F)
## Run                     1  0.0677  0.0677  0.2944   0.58946
## Host                    1 13.5992 13.5992 59.1720 1.815e-10
## Cadavers                2 17.0274  8.5137 37.0444 3.784e-11
## Ladybird                1 11.0907 11.0907 48.2571 3.329e-09
## Host:Cadavers           2  0.3078  0.1539  0.6695   0.51579
## Host:Ladybird           1  0.2279  0.2279  0.9916   0.32341
## Cadavers:Ladybird       2  1.7349  0.8675  3.7744   0.02867
## Host:Cadavers:Ladybird  2  0.1999  0.1000  0.4350   0.64932
## Residuals              59 13.5596  0.2298
```

Now the `Run:Plant` variance estimate is equal to that for the `Residuals Mean Sq` for `Run:Plant` from the anova table.

Obtain the marginality matrix for the fixed terms

The `pstructure` function from the `dae` package (Brien, 2020) produce the marginality matrix for a formula as a side effect and we take advantage of that to obtain the matrix required here.

```
Ladybird.pstr <- pstructure(formula = ~ Host*Cadavers*Ladybird,
                            data = Ladybird.dat)
(HCL.marg <- marginality(Ladybird.pstr))
```

```
##                        Host Cadavers Host:Cadavers Ladybird Host:Ladybird Cadavers:Ladybird
## Host                      1        0             1        0             1                 0
## Cadavers                  0        1             1        0             0                 1
## Host:Cadavers             0        0             1        0             0                 0
## Ladybird                  0        0             0        1             1                 1
## Host:Ladybird             0        0             0        0             1                 0
## Cadavers:Ladybird         0        0             0        0             0                 1
## Host:Cadavers:Ladybird    0        0             0        0             0                 0
##                        Host:Cadavers:Ladybird
## Host                                        1
## Cadavers                                    1
## Host:Cadavers                               1
## Ladybird                                    1
## Host:Ladybird                               1
## Cadavers:Ladybird                           1
## Host:Cadavers:Ladybird                      1
```

This marginality matrix is interpreted by taking a row term and noting that it is marginal to any column term with a one in this row.

Choose marginality-compliant model

```
choosing <- chooseModel(aov.tab, DF = "Df", denDF = 59, p.values = "Pr(>F)" ,
                        terms.marginality =HCL.marg)
(choosing$choose.summary)
```

```
##
##
## ####  Table of hypothesis tests performed
##
##                       terms DF denDF      p         action
## 1 Host:Cadavers:Ladybird  2    59 0.6493 Nonsignificant
## 2      Cadavers:Ladybird  2    59 0.0287    Significant
## 3          Host:Ladybird  1    59 0.3234 Nonsignificant
## 4          Host:Cadavers  2    59 0.5158 Nonsignificant
## 5                   Host  1    59 0.0000    Significant
```

```
(choosing$sig.terms)
```

```
## [[1]]
## [1] "Cadavers:Ladybird"
##
## [[2]]
## [1] "Host"
```

The `chooseModel` function produces a list with components `sig.terms`, a list with the terms in the marginality-compliant model, and `choose.summary`, a data.frame that details the tests performed in choosing the model. Note that `chooseModel` does not test the main effects for Cadavers or Ladybird, because these are marginal to the significant two-factor interaction Cadavers:Ladybird.

Form formula for selected model

```
chosen.mod <- paste(unlist(choosing$sig.terms), collapse = " + ")
(chosen.mod <- as.formula(paste("~", chosen.mod)))
```

```
## ~Cadavers:Ladybird + Host
```

Form predictions that conform to the chosen model

Use `emmeans` to get the predictions and associated statistics for the full model.

```
HCL.emm <- emmeans::emmeans(m1.lmer, specs = ~ Host:Cadavers:Ladybird)
HCL.preds <- summary(HCL.emm)
den.df <- min(HCL.preds$df)
HCL.vcov <- vcov(HCL.emm)
```

Setting the `specs` argument to `Host:Ladybird:Cadavers` requests predictions for all combinations of the three factors.

Modify HCL.preds to be compatible with a predictions.frame

Basically, this is an exercise in renaming the columns in the `data.frame` containing the predictions.

```r
names(HCL.preds)
```

```
## [1] "Host"     "Cadavers" "Ladybird" "emmean"   "SE"       "df"       "lower.CL" "upper.CL"
```

```r
HCL.preds <- as.predictions.frame(HCL.preds, predictions = "emmean",
                                  se = "SE", interval.type = "CI",
                                  interval.names = c("lower.CL", "upper.CL"))
names(HCL.preds)
```

```
## [1] "Host"                     "Cadavers"                 "Ladybird"
## [4] "predicted.value"          "standard.error"           "df"
## [7] "lower.Confidence.limit"   "upper.Confidence.limit"   "est.status"
```

Form an alldiffs object with predictions obtained with `emmeans`

```r
HCL.diffs <- allDifferences(predictions = HCL.preds, classify = "Host:Ladybird:Cadavers",
                            vcov = HCL.vcov, tdf = den.df)
```

The functions `allDifferences` is used to form the `alldiffs.obj` that contains a `predictions` component, along with components related to pairwise comparisons. The `predictions` component contains upper and lower confidence limits produced by `emmeans`. The `tdf` is supplied so that it can be used to get the degrees of freedom for the t-value to be used in calculating the error intervals.

Transform the prediction to conform to chosen model

The `linTransform` function is used to obtain estimated marginal means (emm) that conform to the chosen model. Because we would prefer error intervals based on $\pm 0.5 LSD$, the `error.intervals` argument has been set to `"halfLeast"`, the `meanLSD.type` argument to `"factor.combination"` and the `LSDby` argument to `"Host"` so that the average LSD will be calculated for each Host. This necessary because, under the chosen model, the LSDs differ between Hosts. It results in `lower.halfLeastSignificant.limit` and `upper.halfLeastSignificant.limit` replacing the limits based on the confidence intervals in the `predictions` component of the resulting `alldiffs` object.

```r
diffs <- linTransform(HCL.diffs, linear.transformation = ~Cadavers:Ladybird + Host,
                      error.intervals = "halfLeast",
                      meanLSD.type = "factor.combination", LSDby = "Host",
                      tables = "predictions")
```

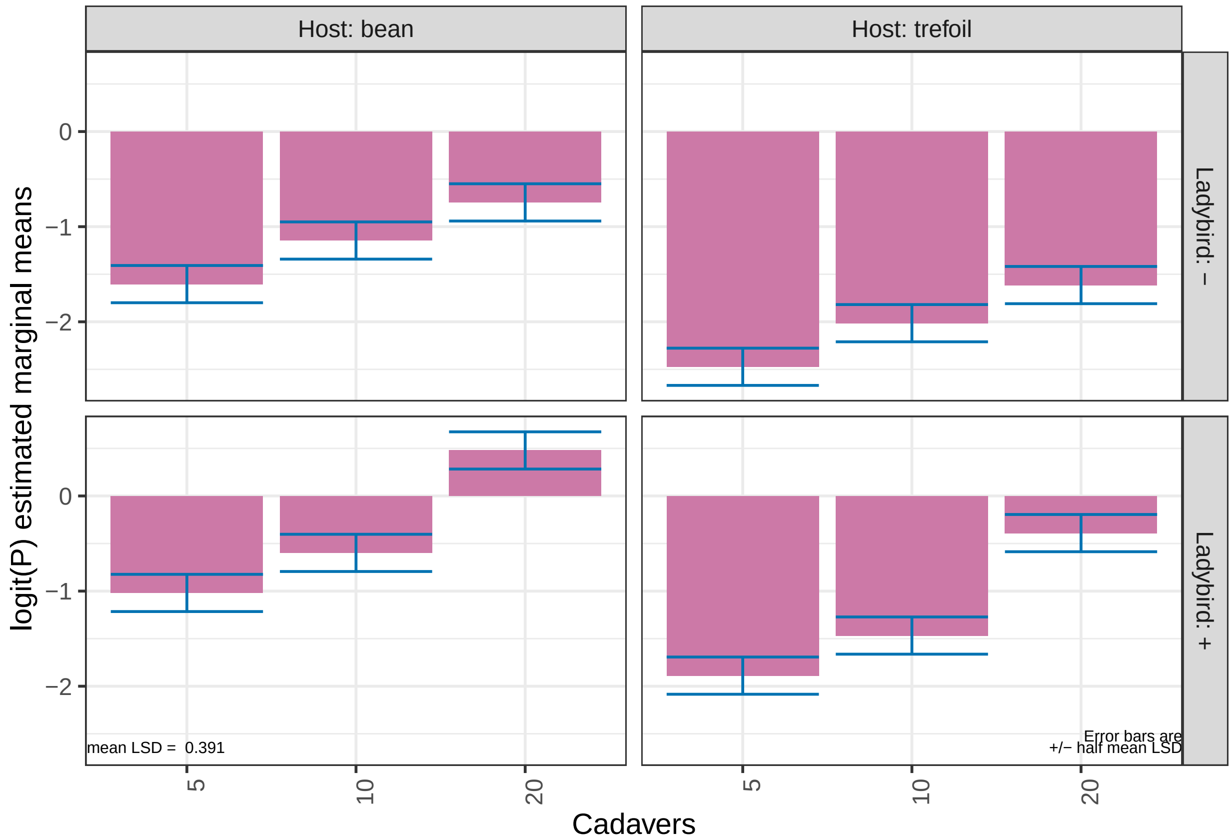
```
##
##
## #### Predictions for  transform(s)
##
## The original predictions, obtained as described below, have
## been linearly transformed to form estimated marginal means.
##       Host Ladybird Cadavers predicted.value standard.error   df
## 1     bean        -        5      -1.6038338      0.1485977 47.2
## 2     bean        -       10      -1.1454308      0.1485977 47.2
## 3     bean        -       20      -0.7448097      0.1485977 47.2
## 4     bean        +        5      -1.0195475      0.1485977 47.2
## 5     bean        +       10      -0.5983440      0.1485977 47.2
## 6     bean        +       20       0.4786704      0.1485977 47.2
## 7  trefoil        -        5      -2.4730339      0.1485977 47.2
## 8  trefoil        -       10      -2.0146309      0.1485977 47.2
## 9  trefoil        -       20      -1.6140098      0.1485977 47.2
```

```
## 10 trefoil        +        5      -1.8887476      0.1485977 47.2
## 11 trefoil        +       10      -1.4675441      0.1485977 47.2
## 12 trefoil        +       20      -0.3905297      0.1485977 47.2
##    upper.halfLeastSignificant.limit lower.halfLeastSignificant.limit est.status
## 1                        -1.4081535                       -1.7995140  Estimable
## 2                        -0.9497506                       -1.3411111  Estimable
## 3                        -0.5491295                       -0.9404900  Estimable
## 4                        -0.8238673                       -1.2152278  Estimable
## 5                        -0.4026637                       -0.7940242  Estimable
## 6                         0.6743507                        0.2829901  Estimable
## 7                        -2.2773537                       -2.6687142  Estimable
## 8                        -1.8189507                       -2.2103112  Estimable
## 9                        -1.4183296                       -1.8096901  Estimable
## 10                       -1.6930674                       -2.0844279  Estimable
## 11                       -1.2718638                       -1.6632243  Estimable
## 12                       -0.1948495                       -0.5862100  Estimable
##
##
## LSD values
##
## minimum LSD =  0.3913605 0.3913605
##
## mean LSD =  0.3913605 0.3913605
##
## maximum LSD =  0.3913605 0.3913605
##
## (sed range / mean sed =  3.45e-14 3.45e-14 )
```

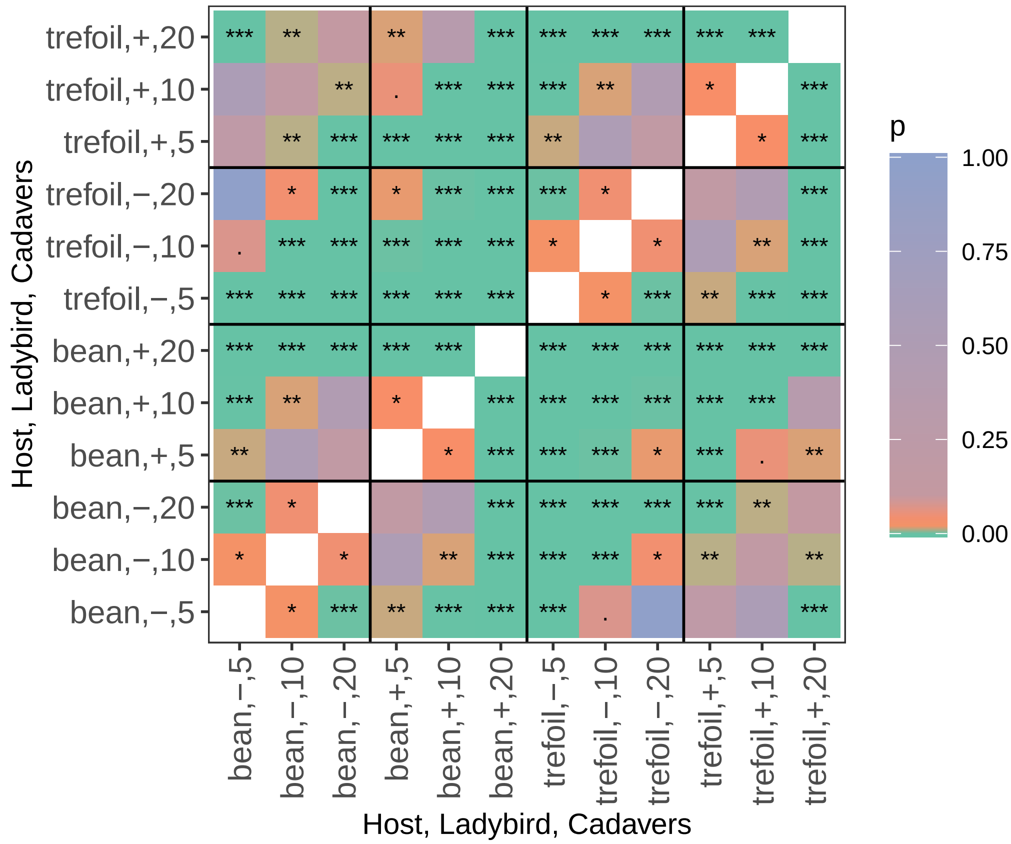
Plot the predictions

The function `plotPredictions` uses `ggplot` to produce the plot and the `ggplotFuncs` argument allows the addition of `ggplot` functions to modify the plot. In this case, the `facet.grid` function is respecified to include `prepender` functions that modify the labels of the facets to include the factor names. Note the the error bars in the plots are of $\pm 0.5LSD$ so that pairs of prediction with nonoverlapping bars are significantly different (Snee, 1981).

```
plotPredictions(diffs$predictions, y ="predicted.value",
                y.title = "logit(P) estimated marginal means",
                classify = "Host:Ladybird:Cadavers",
                error.intervals = "halfLeast",
                ggplotFuncs = list(facet_grid(Ladybird ~ Host,
                                              labeller = label_both)))
```




```r
plotPvalues(diffs, factors.per.grid = 1, show.sig = TRUE)
```




```r
options(width = 90)
print(diffs$sed)
```

```
##                bean,-,5 bean,-,10 bean,-,20  bean,+,5 bean,+,10 bean,+,20 trefoil,-,5
## bean,-,5             NA 0.1945600 0.1945600 0.1945600 0.1945600 0.1945600   0.1123293
## bean,-,10     0.1945600        NA 0.1945600 0.1945600 0.1945600 0.1945600   0.2246586
## bean,-,20     0.1945600 0.1945600        NA 0.1945600 0.1945600 0.1945600   0.2246586
## bean,+,5      0.1945600 0.1945600 0.1945600        NA 0.1945600 0.1945600   0.2246586
## bean,+,10     0.1945600 0.1945600 0.1945600 0.1945600        NA 0.1945600   0.2246586
## bean,+,20     0.1945600 0.1945600 0.1945600 0.1945600 0.1945600        NA   0.2246586
## trefoil,-,5   0.1123293 0.2246586 0.2246586 0.2246586 0.2246586 0.2246586          NA
## trefoil,-,10  0.2246586 0.1123293 0.2246586 0.2246586 0.2246586 0.2246586   0.1945600
## trefoil,-,20  0.2246586 0.2246586 0.1123293 0.2246586 0.2246586 0.2246586   0.1945600
## trefoil,+,5   0.2246586 0.2246586 0.2246586 0.1123293 0.2246586 0.2246586   0.1945600
## trefoil,+,10  0.2246586 0.2246586 0.2246586 0.2246586 0.1123293 0.2246586   0.1945600
## trefoil,+,20  0.2246586 0.2246586 0.2246586 0.2246586 0.2246586 0.1123293   0.1945600
##               trefoil,-,10 trefoil,-,20 trefoil,+,5 trefoil,+,10 trefoil,+,20
## bean,-,5         0.2246586    0.2246586   0.2246586    0.2246586    0.2246586
## bean,-,10        0.1123293    0.2246586   0.2246586    0.2246586    0.2246586
## bean,-,20        0.2246586    0.1123293   0.2246586    0.2246586    0.2246586
## bean,+,5         0.2246586    0.2246586   0.1123293    0.2246586    0.2246586
## bean,+,10        0.2246586    0.2246586   0.2246586    0.1123293    0.2246586
## bean,+,20        0.2246586    0.2246586   0.2246586    0.2246586    0.1123293
## trefoil,-,5      0.1945600    0.1945600   0.1945600    0.1945600    0.1945600
## trefoil,-,10            NA    0.1945600   0.1945600    0.1945600    0.1945600
## trefoil,-,20     0.1945600           NA   0.1945600    0.1945600    0.1945600
```

```
## trefoil,+,5     0.1945600    0.1945600          NA    0.1945600    0.1945600
## trefoil,+,10    0.1945600    0.1945600   0.1945600           NA    0.1945600
## trefoil,+,20    0.1945600    0.1945600   0.1945600    0.1945600           NA
```

Perform the analysis with just the selected model fitted

The model with nonsignificant fixed terms dropped is obtained in order to compare it with the fit when they are retained and the etimated marginal means for the chosen model are obtained.

```
m.sig.lm <- lm(logitP ~ Run + Cadavers*Ladybird + Host,
                         data=Ladybird.dat)
(aov.tab <- anova(m.sig.lm))
```

```
## Analysis of Variance Table
##
## Response: logitP
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## Run                1  0.0677  0.0677  0.3029   0.58398
## Cadavers           2 17.0274  8.5137 38.1160 1.255e-11
## Ladybird           1 11.0907 11.0907 49.6531 1.542e-09
## Host               1 13.5992 13.5992 60.8836 7.179e-11
## Cadavers:Ladybird  2  1.7349  0.8675  3.8836   0.02559
## Residuals         64 14.2952  0.2234
```

```
HCL.emm <- emmeans::emmeans(m.sig.lm, specs = ~ Host:Cadavers:Ladybird)
HCL.preds <- summary(HCL.emm)
den.df <- min(HCL.preds$df)
HCL.vcov <- vcov(HCL.emm)
HCL.preds <- as.predictions.frame(HCL.preds, predictions = "emmean",
                                  se = "SE", interval.type = "CI",
                                  interval.names = c("lower.CL", "upper.CL"))
diffs.red <- allDifferences(predictions = HCL.preds, classify = "Host:Ladybird:Cadavers",
                            vcov = HCL.vcov, tdf = den.df)
diffs.red <- redoErrorIntervals(diffs, error.intervals = "halfLeast",
                                meanLSD.type = "factor.combination", LSDby = "Host")

options(width = 90)
print(diffs.red$sed)
```

```
##               bean,-,5 bean,-,10 bean,-,20  bean,+,5 bean,+,10 bean,+,20 trefoil,-,5
## bean,-,5            NA 0.1945600 0.1945600 0.1945600 0.1945600 0.1945600   0.1123293
## bean,-,10    0.1945600        NA 0.1945600 0.1945600 0.1945600 0.1945600   0.2246586
## bean,-,20    0.1945600 0.1945600        NA 0.1945600 0.1945600 0.1945600   0.2246586
## bean,+,5     0.1945600 0.1945600 0.1945600        NA 0.1945600 0.1945600   0.2246586
## bean,+,10    0.1945600 0.1945600 0.1945600 0.1945600        NA 0.1945600   0.2246586
## bean,+,20    0.1945600 0.1945600 0.1945600 0.1945600 0.1945600        NA   0.2246586
## trefoil,-,5  0.1123293 0.2246586 0.2246586 0.2246586 0.2246586 0.2246586          NA
## trefoil,-,10 0.2246586 0.1123293 0.2246586 0.2246586 0.2246586 0.2246586   0.1945600
## trefoil,-,20 0.2246586 0.2246586 0.1123293 0.2246586 0.2246586 0.2246586   0.1945600
## trefoil,+,5  0.2246586 0.2246586 0.2246586 0.1123293 0.2246586 0.2246586   0.1945600
## trefoil,+,10 0.2246586 0.2246586 0.2246586 0.2246586 0.1123293 0.2246586   0.1945600
## trefoil,+,20 0.2246586 0.2246586 0.2246586 0.2246586 0.2246586 0.1123293   0.1945600
##              trefoil,-,10 trefoil,-,20 trefoil,+,5 trefoil,+,10 trefoil,+,20
## bean,-,5        0.2246586    0.2246586   0.2246586    0.2246586    0.2246586
## bean,-,10       0.1123293    0.2246586   0.2246586    0.2246586    0.2246586
```

```
## bean,-,20        0.2246586    0.1123293   0.2246586    0.2246586    0.2246586
## bean,+,5         0.2246586    0.2246586   0.1123293    0.2246586    0.2246586
## bean,+,10        0.2246586    0.2246586   0.2246586    0.1123293    0.2246586
## bean,+,20        0.2246586    0.2246586   0.2246586    0.2246586    0.1123293
## trefoil,-,5      0.1945600    0.1945600   0.1945600    0.1945600    0.1945600
## trefoil,-,10            NA    0.1945600   0.1945600    0.1945600    0.1945600
## trefoil,-,20     0.1945600           NA   0.1945600    0.1945600    0.1945600
## trefoil,+,5      0.1945600    0.1945600          NA    0.1945600    0.1945600
## trefoil,+,10     0.1945600    0.1945600   0.1945600           NA    0.1945600
## trefoil,+,20     0.1945600    0.1945600   0.1945600    0.1945600           NA
```

References

Brien, C. J. (2020a) `asremlPlus`: *Augments ASReml-R in fitting mixed models and packages generally in exploring prediction differences*. Version 4.1-37. http://cran.r-project.org/package=asremlPlus/ or http://chris.brien.name/rpackages/.

Brien, C. J. (2020b) `dae`: *Functions useful in the design and ANOVA of experiments.* Version 3.1-22. http://cran.r-project.org/package=dae/ or http://chris.brien.name/rpackages/.

Kuznetsova, A., Brockhoff, P. B. and Christensen, R. H. B. (2017) `lmerTest Package`: Tests in Linear Mixed Effects Models. *Journal of Statistical Software*, **82**, 1–26.

Lenth, R. (2019) `emmeans`: *Estimated Marginal Means, aka Least-Squares Means.* Version 1.4.3.01. http://cran.r-project.org/package=emmeans/.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., & Schabenberger, O. (2006). *SAS for Mixed Models* (2nd ed.). Cary, N.C.: SAS Press.

R Core Team (2020) *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. http://www.r-project.org.

Snee, R. D. (1981). Graphical Display and Assessment of Means. *Biometrics*, **37**, 835–836.

Welham, S. J., Gezan, S. A., Clark, S. J., & Mead, A. (2014). *Statistical Methods in Biology: Design and Analysis of Experiments and Regression.* Boca Raton: Chapman and Hall/CRC.