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

Chris Brien

10 June, 2025

Introduction

This vignette shows how to use `asremlPlus` (Brien, 2025a), and `dae` (Brien, 2024b), for exploring and presenting predictions (estimated marginal means: EMMs) from a linear mixed model analysis, the predictions having been produced using `lmerTest` (Kuznetsova et al., 2017), `lm` and `emmeans` (Lenth, 2023). Here, `asremlPlus`, `dae`, `lmerTest` and `emmeans` are packages for the `R` Statistical Computing environment (R Core Team, 2025) and `lm` is available from `stats` and is included in `R`.

The context is a three-factor factorial experiment on ladybirds (Welham et al., 2014, Example 8.2) that aims to 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 (-, +). These are randomized to the containers within a run so that each is replicated 3 times within a run. The response to be analysed is the logit of the proportion of live aphids that were infected.

Initialize

```r
library(knitr)
opts_chunk$set("tidy" = FALSE, comment = NA)
suppressMessages(library(lmerTest))
packageVersion("lmerTest")
```

```
## [1] '3.1.3'
```

```r
suppressMessages(library(emmeans))
packageVersion("emmeans")
```

```
## [1] '1.11.0'
```

```r
suppressMessages(library(asremlPlus))
packageVersion("asremlPlus")
```

```
## [1] '4.4.49'
```

```
suppressMessages(library(dae))
packageVersion("dae")
```

```
## [1] ’3.2.30’
```

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

Get data available in asremlPlus

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

Do an ANOVA of logits

```
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

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

```
boundary (singular) fit: see help('isSingular')
```

```
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
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('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, 2024b) 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

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

```
####  Sequence of model investigations

                   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
```

```
(chosen$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(chosen$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.

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

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

```
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

```
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 (EMMs) that conform to the chosen model. Because we would prefer error intervals based on $\pm 0.5LSD$, the `error.intervals` argument has been set to `"halfLeast"`, the `LSDtype` 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.

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

```
Joining with ‘by = join_by(fac.comb)‘
Joining with ‘by = join_by(Host)‘



#### Predictions for  transform(s) from Host:Ladybird:Cadavers

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 =  2.45e-14 2.41e-14 )
```

The above LSD values can only be used to compare pairs of EMMs for the same Host.

Explore the LSDs

To investigate the errors that would result from using the overall LSDs as opposed to the LSDs computed for each Host, the `exploreLSDs`, `pickLSDstatistics` and `plotLSDerrors` functions are used, firstly with the default value of "overall" for `LSDtype` and finally with the `LSDtype` set to `"factor.combination"` and `LSDby` to `"Host"`.

The `exploreLSDs` function produces tables of statistics for the LSDs computed for the settings of the `LSDtype` and `LSDby`; the settings of these arguments does not have to match those used in producing the `alldiffs` object. For `LSDtype` set to `"overall'`, a single LSD statistic is computed that is the based on the standard errors of all pairwise differences. To ascertain the errors that arise from using this LSD value for determining, for all pairwise comparisons, whether a comparison is significant, `exploreLSDs` compares the results using the LSD value with the *p*-values in the `p.differences` component of the `alldiffs` object. For `LSDtype` set to `"factor.combination"` and `LSDby` to `"Host"`, the LSD statistics are calculated from standard errors of the pairwise differences for each Host. Examination of the `sed` component of `diffs` reveals that there are only three different values for the standard errors of pairwise differences and, hence, only three unique values for the LSD. Of the statistics shown, there is no single value that will not result in errors.

Here are the tables of the numbers of false positive and negative error in using the values of the various LSD statistics for determining the significance of the 66 pairwise comparisons of the 12 predictions.

```
exploreLSDs(diffs, LSDtype = "overall")
```

```
####  Statistics calculated from LSD values

   c       min   quant10   quant25      mean    median   quant75   quant90       max
1 66 0.2259521 0.3913605 0.3913605 0.4087627 0.3913605 0.4519042 0.4519042 0.4519042


####  False positives resulting from the use of various LSD statistics

           c min quant10 quant25 mean median quant75 quant90 max
false.pos 66   7       2       2    2      2       0       0   0


####  False negatives resulting from the use of various LSD statistics

           c min quant10 quant25 mean median quant75 quant90 max
false.neg 66   0       0       0    2      0       4       4   4
```

The function `findLSDminerrors` can be used to find a value of the LSD that minimizes the number of false positive (negative) errors i.e. declaring a pairwise difference to be significant (not significant) when it is not significant (significant). The function has an argument `false.pos.wt` that specifies how many false negatives are equivalent to one false positive; the default for `false.pos.wt` is 10, which is a conservative approach because it greatly favours false negatives over false positives.

```
(findLSDminerrors(diffs))
```

```
              LSD false.pos false.neg false.criterion
overall 0.4486447         0         4               4
```

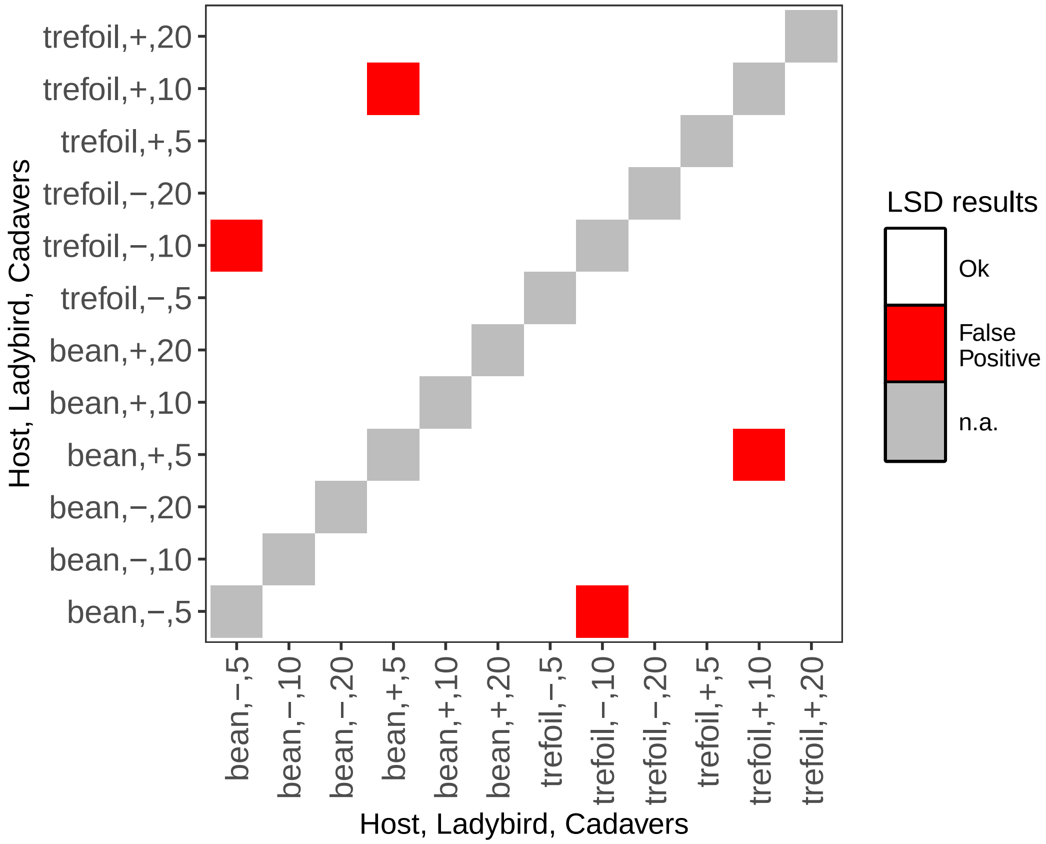
```
(minLSDpos <- findLSDminerrors(diffs, false.pos.wt = 1))
```

```
              LSD false.pos false.neg false.criterion
overall 0.3547448         2         0               2
```

The smallest value of the LSD that results in the minimum weighted sum of the false positives and negatives is 0.4489466 when the weight is 10 and is 0.354983 when the weight is one. These values result in 4 false negatives and 2 false positives respectively.

To see which of the two pairwise comparisons will be falsely identified as being significant when the LSD value is set to minimum LSD, the LSDs stored in the `alldiffs` object need to be recalculated to be based on this LSD value. Then plot the errors or save the return values obtained using the function `plotLSDerrors`. The plot below shows that the two pairs whose differences are incorrectly identified as significant have the same level of the Ladybird factor, but differ in both of the levels for the Host and Cadaver factors.

```
diffs.overall <- recalcLSD(diffs, LSDtype = "supplied", LSDsupplied = minLSDpos["LSD"])
plotLSDerrors(diffs.overall)
```



This raises the question of whether the minimum LSD should be used. There are at least four alternatives: (i) use it without restriction, on the basis that it can be concluded that using it is unlikely to result in seriously flawed conclusions; (ii) use it with the restriction that it only be applied to assess pairwise comparisons that have the same Host or the same Cadaver treatment; (i11) investigate the use of the minimum LSD with only false negative errors; and (iv) rather than use an overall LSD value, use LSD values computed from the LSDs within each Host level.

Because `LSDtype` was set to `"factor.combination"` and `LSDby` to `"Host"` in forming the object `diffs`, the LSDs for alternative (iv) are stored in the `LSD` component of the object `diffs`. Printing out the LSD component will summarize how those LSD values perform. Otherwise, the following call to `exploreLSDs` will display the properties of the LSDs for various LSD statistics:

```
exploreLSDs(diffs, LSDtype = "factor.combination", LSDby = "Host")
```

The following shows the contents of the `LSD` component of `diffs`:

```
(diffs$LSD)
```

```
         c    minLSD   meanLSD    maxLSD assignedLSD  accuracyLSD falsePos falseNeg
bean    15 0.3913605 0.3913605 0.3913605   0.3913605 1.886492e-14        0        0
trefoil 15 0.3913605 0.3913605 0.3913605   0.3913605 1.843939e-14        0        0
```

Because the minimum and maximum LSD values are equal, it follows that there is only one value of the LSD for all pairwise comparisons within each Host level and it happens that the values for the two Hosts are also equal. The table shows that zero false positive and negative errors will result from the use of the value of 0.39 for the 30 within-Host comparisons. Thus, using `findLSDminerrors(diffs, LSDtype = "factor.combinations", LSDby = "Host")` to select the minimum LSDs would only confirm what is already obvious.

Plot the predictions

```
titl <- str2expression("logit~(italic(P))")
names(titl) <- "logitP"
plotPredictions(diffs$predictions, y ="predicted.value",
                y.title = titl,
                classify = "Host:Ladybird:Cadavers",
                error.intervals = "halfLeast", interval.annotate = FALSE,
                ggplotFuncs = list(facet_grid(Ladybird ~ Host,
                                              labeller = label_both)))
```

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 statement in the legend of Figure 1 that restricts the use of the error bars to determining the significance of differences for the pairwise comparisons of EMMs for the same Host.

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

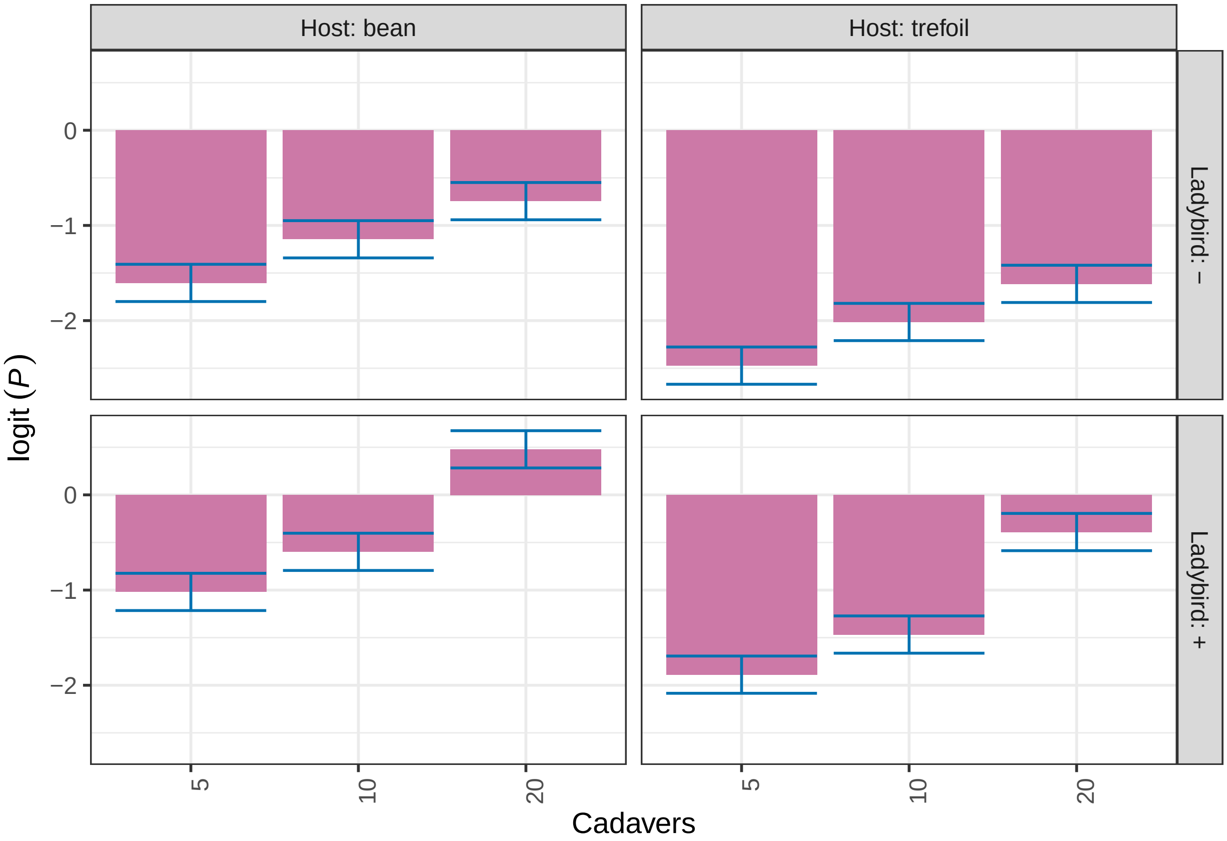
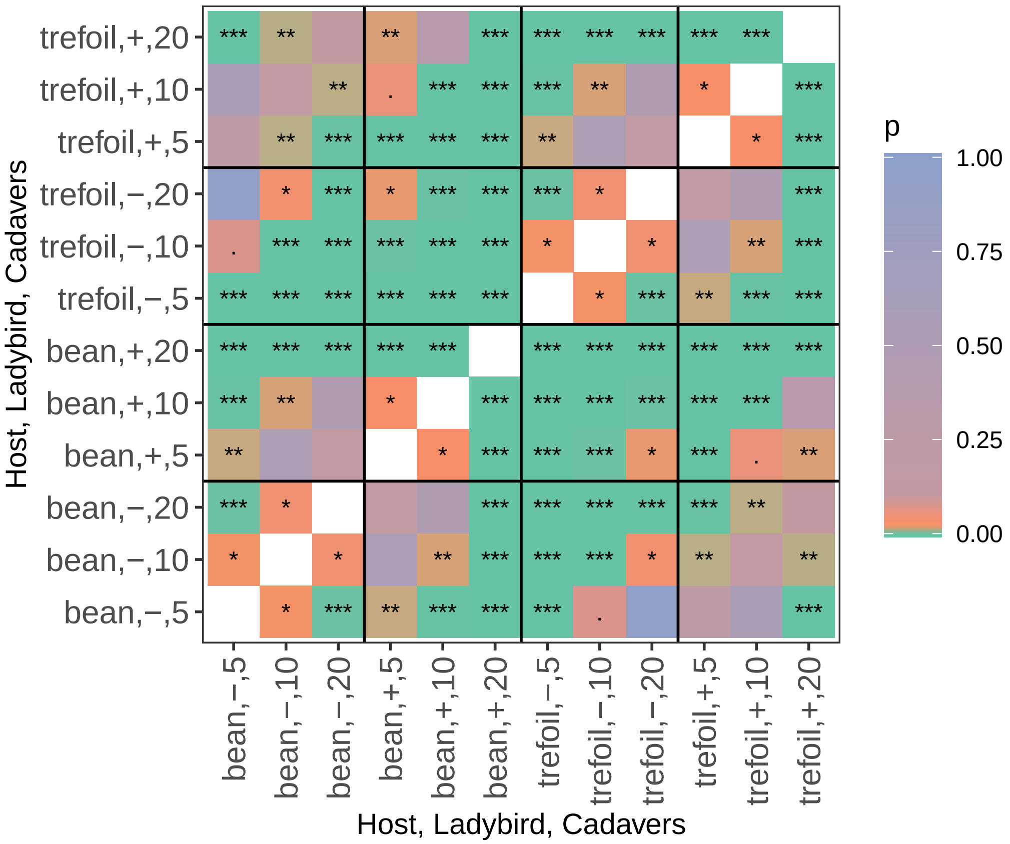


Figure 1. Estimated marginal means (EMMs) for logit(P), where P is the proportion of live aphids that were infected, for two Hosts, two Ladybird levels and three Cadaver levels. Error bars are an EMM ± half-LSD (5%). The two EMMs for the same Host are significantly different ($p \leq 0.05$) if their error bars do not overlap.



	bean,−,5	bean,−,10	bean,−,20	bean,+,5	bean,+,10	bean,+,20	trefoil,−,5	trefoil,−,10	trefoil,−,20	trefoil,+,5	trefoil,+,10	trefoil,+,20
trefoil,+,20	***	**		**		***	***	***	***	***	***	
trefoil,+,10			**	.	***	***	***	**		*		***
trefoil,+,5		**	***	***	***	***	**				*	***
trefoil,−,20		*	***	*	***	***	***	*				***
trefoil,−,10	.	***	***	***	***	***	*		*		**	***
trefoil,−,5	***	***	***	***	***	***		*	***	**	***	***
bean,+,20	***	***	***	***	***		***	***	***	***	***	***
bean,+,10	***	**		*		***	***	***	***	***	***	
bean,+,5	**				*	***	***	***	*	***	.	**
bean,−,20	***	*				***	***	***	***	***	**	
bean,−,10	*		*		**	***	***	***	*	**		**
bean,−,5		*	***	**	***	***	***	.				***

Figure 2. The p-values for each of the pairwise comparisons of the estimated marginal means for logit(P), where P is the proportion of live aphids that were infected, for two Hosts, two Ladybird levels and three Cadaver levels

```
options(width = 90)
diffs$differences
```

```
                bean,-,5  bean,-,10  bean,-,20   bean,+,5  bean,+,10  bean,+,20
bean,-,5      0.00000000 -0.4584030 -0.8590241 -0.5842863 -1.0054898 -2.0825042
bean,-,10     0.45840297  0.0000000 -0.4006211 -0.1258833 -0.5470869 -1.6241012
bean,-,20     0.85902408  0.4006211  0.0000000  0.2747378 -0.1464657 -1.2234801
bean,+,5      0.58428627  0.1258833 -0.2747378  0.0000000 -0.4212036 -1.4982179
bean,+,10     1.00548982  0.5470869  0.1464657  0.4212036  0.0000000 -1.0770144
bean,+,20     2.08250420  1.6241012  1.2234801  1.4982179  1.0770144  0.0000000
trefoil,-,5  -0.86920012 -1.3276031 -1.7282242 -1.4534864 -1.8746899 -2.9517043
trefoil,-,10 -0.41079715 -0.8692001 -1.2698212 -0.9950834 -1.4162870 -2.4933014
trefoil,-,20 -0.01017604 -0.4685790 -0.8692001 -0.5944623 -1.0156659 -2.0926802
trefoil,+,5  -0.28491385 -0.7433168 -1.1439379 -0.8692001 -1.2904037 -2.3674180
trefoil,+,10  0.13628970 -0.3221133 -0.7227344 -0.4479966 -0.8692001 -1.9462145
trefoil,+,20  1.21330408  0.7549011  0.3542800  0.6290178  0.2078143 -0.8692001
             trefoil,-,5 trefoil,-,10 trefoil,-,20 trefoil,+,5 trefoil,+,10 trefoil,+,20
bean,-,5       0.8692001    0.4107972   0.01017604   0.2849139   -0.1362897   -1.2133041
bean,-,10      1.3276031    0.8692001   0.46857901   0.7433168    0.3221133   -0.7549011
bean,-,20      1.7282242    1.2698212   0.86920012   1.1439379    0.7227344   -0.3542800
bean,+,5       1.4534864    0.9950834   0.59446231   0.8692001    0.4479966   -0.6290178
bean,+,10      1.8746899    1.4162870   1.01566586   1.2904037    0.8692001   -0.2078143
bean,+,20      2.9517043    2.4933014   2.09268024   2.3674180    1.9462145    0.8692001
trefoil,-,5    0.0000000   -0.4584030  -0.85902408  -0.5842863   -1.0054898   -2.0825042
trefoil,-,10   0.4584030    0.0000000  -0.40062111  -0.1258833   -0.5470869   -1.6241012
trefoil,-,20   0.8590241    0.4006211   0.00000000   0.2747378   -0.1464657   -1.2234801
trefoil,+,5    0.5842863    0.1258833  -0.27473781   0.0000000   -0.4212036   -1.4982179
trefoil,+,10   1.0054898    0.5470869   0.14646574   0.4212036    0.0000000   -1.0770144
trefoil,+,20   2.0825042    1.6241012   1.22348012   1.4982179    1.0770144    0.0000000
```

```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 estimated 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",
                                LSDtype = "factor.combination", LSDby = "Host")
```

```
Joining with ‘by = join_by(fac.comb)‘
Joining with ‘by = join_by(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. (2025a) `asremlPlus`: *Augments ASReml-R in fitting mixed models and packages generally in exploring prediction differences.* Version 4.4.49. https://cran.r-project.org/package=asremlPlus/ or http://chris.brien.name/rpackages/.

Brien, C. J. (2024b) `dae`: *Functions useful in the design and ANOVA of experiments.* Version 3.2.30. https://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. V., Banfa, B., Bolker, B., Buerkner, P., Giné-Vázquez, I., Herve, M. J., Jung, M., Love, J., Miguez, F., Piaskowski, J., Riebl, H., & Singmann, H. (2025) `emmeans`: *Estimated Marginal Means, aka Least-Squares Means.* Version 1.11.0. https://cran.r-project.org/package=emmeans/.

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

R Core Team (2025) `R`: *A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. https://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.