Using asreml and asremlPlus for the Ladybird example from Welham et al. (2014)

Chris Brien

10 December, 2024

Introduction

This vignette shows how to use `asremlPlus` (Brien, 2024a), 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 `asreml` (Butler et al., 2023). Here, `asremlPlus`, `dae` and `asreml` are packages for the `R` Statistical Computing environment (R Core Team, 2024).

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

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

```
## Offline License checked out Tue Dec 10 17:00:54 2024
```

```
packageVersion("asreml")
```

```
## [1] '4.2.0.332'
```

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

```
## [1] '4.4.43'
```

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

```
## [1] '3.2.30'
```

```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 asreml to analyse the logits

Mixed model analysis of logits

```r
m <- asreml(logitP ~ Host*Cadavers*Ladybird,
            random = ~ Run,
            residual = ~ Run:Plant,
            data = Ladybird.dat)
```

```
ASReml Version 4.2 10/12/2024 17:00:54
          LogLik        Sigma2     DF     wall
 1      3.071301     0.2262391     60   17:00:54  (  1 restrained)
```

```
 2      3.642605     0.2269125     60    17:00:54  (   1 restrained)
 3      3.712498     0.2271056     60    17:00:54  (   1 restrained)
 4      3.717213     0.2271206     60    17:00:54  (   1 restrained)
 5      3.717512     0.2271216     60    17:00:54  (   1 restrained)
```

```r
summary(m)$varcomp
```

```
               component  std.error  z.ratio bound %ch
Run         2.298309e-08         NA       NA     B  NA
Run:Plant!R 2.271216e-01 0.04156985 5.463612     P   0
```

As expected the Run component is bound (`B`) at approximately zero. This results in a change in the estimate of the residual variance to 0.227. To allow for a negative estimate we will unconstrain the Run component. 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.

Unconstrain Reps to make the analysis equivalent to ANOVA

```r
m <- setvarianceterms(m$call, terms = "Run", bounds = "U")
```

```
ASReml Version 4.2 10/12/2024 17:00:55
          LogLik        Sigma2     DF     wall
 1      3.071301     0.2262391     60    17:00:55  (   1 restrained)
 2      3.642605     0.2269125     60    17:00:55  (   1 restrained)
 3      3.802834     0.2274541     60    17:00:55  (   1 restrained)
 4      3.839487     0.2334176     60    17:00:55
 5      3.955683     0.2309710     60    17:00:55
 6      3.973334     0.2300353     60    17:00:55
 7      3.974051     0.2298346     60    17:00:55
```

```r
summary(m)$varcomp
```

```
               component   std.error   z.ratio bound %ch
Run         -0.004504789 0.002896281 -1.555370     U 0.1
Run:Plant!R  0.229834648 0.042316936  5.431269     P 0.0
```

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

Set up an asrtests object

```r
current.asrt <- as.asrtests(m)
print(current.asrt, which = "pseudoanova")
```

```
####  Pseudo-anova table for fixed terms


Wald tests for fixed effects.
Response: logitP

                      Df denDF   F.inc     Pr
(Intercept)            1     1 1550.00 0.0162
Host                   1    59   59.17 0.0000
Cadavers               2    59   37.04 0.0000
Ladybird               1    59   48.26 0.0000
Host:Cadavers          2    59    0.67 0.5158
Host:Ladybird          1    59    0.99 0.3234
Cadavers:Ladybird      2    59    3.77 0.0287
Host:Cadavers:Ladybird 2    59    0.44 0.6493
```

The `asrtests` object contains a `wald.tab` component which can be printed by specifying that the `pseudoanova` is printed. The F-values for the fixed terms in this table are the same as those in 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)
print(HCL.marg)
```

```
                       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(current.asrt, terms.marginality = HCL.marg)
current.asrt <- chosen$asrtests.obj
print(current.asrt, which = "test", omit.columns = c("AIC", "BIC"))
```

```
####  Sequence of model investigations for logitP

                   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 `asrtests.obj`, the `asrtests` object resulting from the model selection. In particular, the `asrtests` object contains a `test.summary` that details the tests performed in choosing the model; the `AIC` and `BIC` columns are omitted from `test.summary` because their inclusion has not been requested. 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
```

Obtain predictions under the chosen model and form an alldiffs object

```
diffs <- predictPlus(current.asrt$asreml.obj,
                     classify = "Host:Ladybird:Cadavers",
                     linear.transformation = ~Cadavers:Ladybird + Host,
                     wald.tab = current.asrt$wald.tab,
                     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 logitP transform(s) from Host:Ladybird:Cadavers


Notes:
The original predictions, obtained as described below, have
been linearly transformed to form estimated marginal means.
- The predictions are obtained by averaging across the hypertable
  calculated from model terms constructed solely from factors in
  the averaging and classify sets.
- Use 'average' to move ignored factors into the averaging set.
- The ignored set: Run

      Host Ladybird Cadavers predicted.value standard.error upper.halfLeastSignificant.limit
1     bean        -        5      -1.6038338      0.1417454                       -1.4080222
2     bean        -       10      -1.1454308      0.1417454                       -0.9496192
3     bean        -       20      -0.7448097      0.1417454                       -0.5489981
4     bean        +        5      -1.0195475      0.1417454                       -0.8237359
5     bean        +       10      -0.5983440      0.1417454                       -0.4025323
6     bean        +       20       0.4786704      0.1417454                        0.6744820
7  trefoil        -        5      -2.4730339      0.1417454                       -2.2772223
8  trefoil        -       10      -2.0146309      0.1417454                       -1.8188193
9  trefoil        -       20      -1.6140098      0.1417454                       -1.4181982
10 trefoil        +        5      -1.8887476      0.1417454                       -1.6929360
11 trefoil        +       10      -1.4675441      0.1417454                       -1.2717325
12 trefoil        +       20      -0.3905297      0.1417454                       -0.1947181
   lower.halfLeastSignificant.limit est.status
1                        -1.7996454  Estimable
2                        -1.3412425  Estimable
3                        -0.9406214  Estimable
4                        -1.2153592  Estimable
5                        -0.7941556  Estimable
6                         0.2828588  Estimable
7                        -2.6688455  Estimable
8                        -2.2104426  Estimable
9                        -1.8098215  Estimable
10                       -2.0845593  Estimable
11                       -1.6633557  Estimable
12                       -0.5863414  Estimable


LSD values

minimum LSD =  0.3916233 0.3916233

mean LSD =  0.3916233 0.3916233

maximum LSD =  0.3916233 0.3916233

(sed range / mean sed =  9.92e-16 8.5e-16 )
```

Setting the `terms` argument to `Host:Ladybird:Cadavers` requests predictions for all combinations of the three factors and the `linear.transformation` argument is used to obtain estimated marginal means (EMMs) that conform to the chosen model. The `wald.tab` is supplied so that it can be used to get the degrees of freedom for the *t*-value to be used in calculating the LSD; the degrees of freedom to the source for the `terms` argument will be used. 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` being added to the `predictions` component of the `alldiffs` object.

Or, caclulate predictions to check first and then transform to conform to chosen model

```
diffs.full <- predictPlus(current.asrt$asreml.obj,
                          classify = "Host:Ladybird:Cadavers",
                          wald.tab = current.asrt$wald.tab,
                          tables = "none", Vmatrix = TRUE)

diffs <- linTransform(diffs.full, linear.transformation = ~Cadavers:Ladybird + Host,
                      wald.tab = current.asrt$wald.tab,
                      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 logitP transform(s) from Host:Ladybird:Cadavers


Notes:
The original predictions, obtained as described below, have
been linearly transformed to form estimated marginal means.
- The predictions are obtained by averaging across the hypertable
  calculated from model terms constructed solely from factors in
  the averaging and classify sets.
- Use 'average' to move ignored factors into the averaging set.
- The ignored set: Run

      Host Ladybird Cadavers predicted.value standard.error upper.halfLeastSignificant.limit
1     bean        -        5      -1.6038338      0.1417454                       -1.4080222
2     bean        -       10      -1.1454308      0.1417454                       -0.9496192
3     bean        -       20      -0.7448097      0.1417454                       -0.5489981
4     bean        +        5      -1.0195475      0.1417454                       -0.8237359
5     bean        +       10      -0.5983440      0.1417454                       -0.4025323
6     bean        +       20       0.4786704      0.1417454                        0.6744820
7  trefoil        -        5      -2.4730339      0.1417454                       -2.2772223
8  trefoil        -       10      -2.0146309      0.1417454                       -1.8188193
9  trefoil        -       20      -1.6140098      0.1417454                       -1.4181982
10 trefoil        +        5      -1.8887476      0.1417454                       -1.6929360
11 trefoil        +       10      -1.4675441      0.1417454                       -1.2717325
```

```
12 trefoil        +       20      -0.3905297      0.1417454                        -0.1947181
   lower.halfLeastSignificant.limit est.status
1                         -1.7996454  Estimable
2                         -1.3412425  Estimable
3                         -0.9406214  Estimable
4                         -1.2153592  Estimable
5                         -0.7941556  Estimable
6                          0.2828588  Estimable
7                         -2.6688455  Estimable
8                         -2.2104426  Estimable
9                         -1.8098215  Estimable
10                        -2.0845593  Estimable
11                        -1.6633557  Estimable
12                        -0.5863414  Estimable


LSD values

minimum LSD =  0.3916233 0.3916233

mean LSD =  0.3916233 0.3916233

maximum LSD =  0.3916233 0.3916233

(sed range / mean sed =  9.92e-16 8.5e-16 )
```

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.

The function `pickLSDstatistics` can be used to pick a statistic that minimizes the number of false negative errors i.e. declaring a pairwise difference to be nonsignificant when it is significant. The function has an argument `false.pos.wt` that specifies how many false negatives are equivalent to one false positive, a false positive occurring when a pairwise difference that is nonsignificant is declared to be significant; it allows the choice of an LSD statistic that balances the number of false positives and negatives. 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.

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

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

   c       min   quant10   quant25      mean    median   quant75   quant90       max
1 66 0.2261038 0.3916233 0.3916233 0.4090372 0.3916233 0.4522076 0.4522076 0.4522076


####  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
```

```r
(pickLSDstatistics(diffs))
```

```
[1] "q75"
```

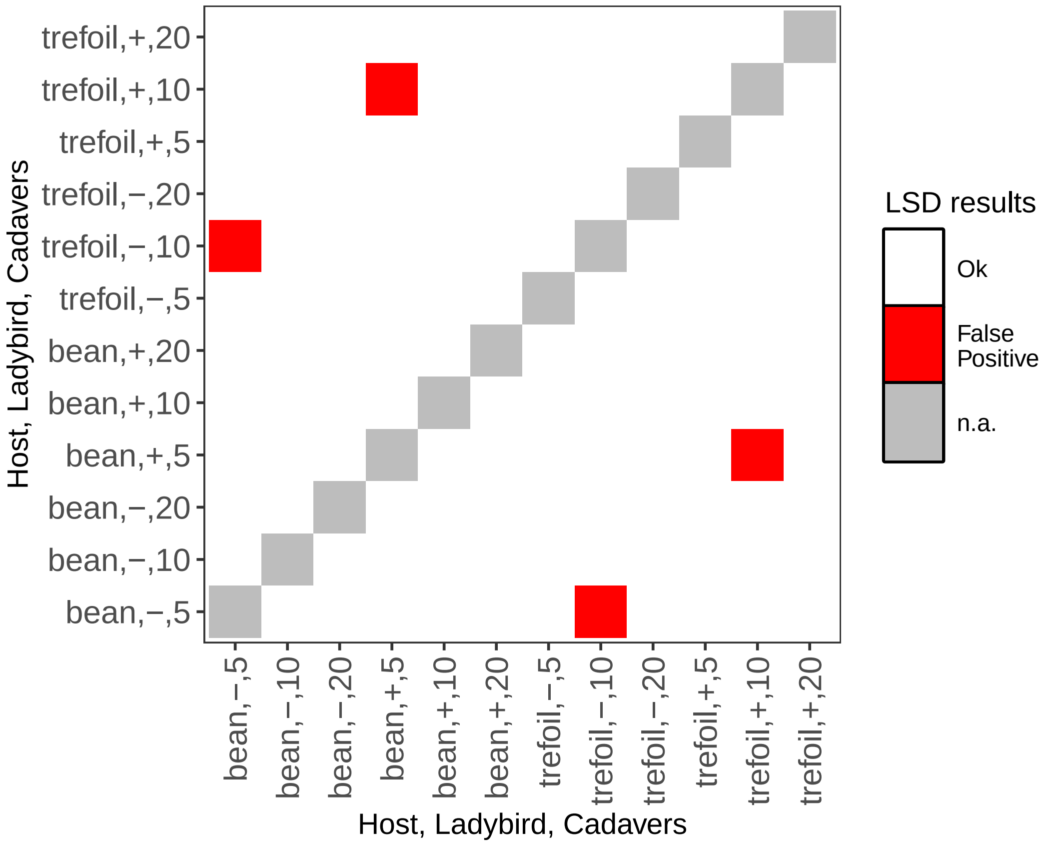
```r
(pickLSDstatistics(diffs, false.pos.wt = 1))
```

```
[1] "q10"
```

Of the statistics shown, there is no choice that will not result in errors. Using the 75th quantile of the all LSD values for all pairwise comparisons (`quant75`) will result in minimum number of false negatives, without there being any false positives; in this case, there will be four out of the 66 comparisons whose conclusions will be false negatives. If one is prepared to consider false negatives and positives as being equally bad, then set **`false.pos.wt`** to one and the 10th quantile of the LSD values (`quant10`') will be chosen as the smallest value of the LSD statistics that has the minimum weighted sum of the errors; using it will result in two of the 66 comparisons yielding false positive results.

To see which of the two pairwise comparisons will be falsely identified as being significant when the LSD value is set to `quant10`, the LSDs stored in the **`alldiffs`** object need to be recalculated to be based on the value for this statistic. 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.

```r
diffs.overall <- recalcLSD(diffs, LSDtype = "overall", LSDstatistic = "q10")
plotLSDerrors(diffs.overall)
```



This raises the question of whether the 10th quantile of all of the LSDs 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 an overall LSD based on `quant75`; 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 0.3916233 0.3916233 0.3916233   0.3916233 5.669852e-16        0        0
trefoil 15 0.3916233 0.3916233 0.3916233   0.3916233 5.669852e-16        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 positives and negatives will result from the use of the value of 0.39 for the 30 within-Host comparisons.

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)))
```

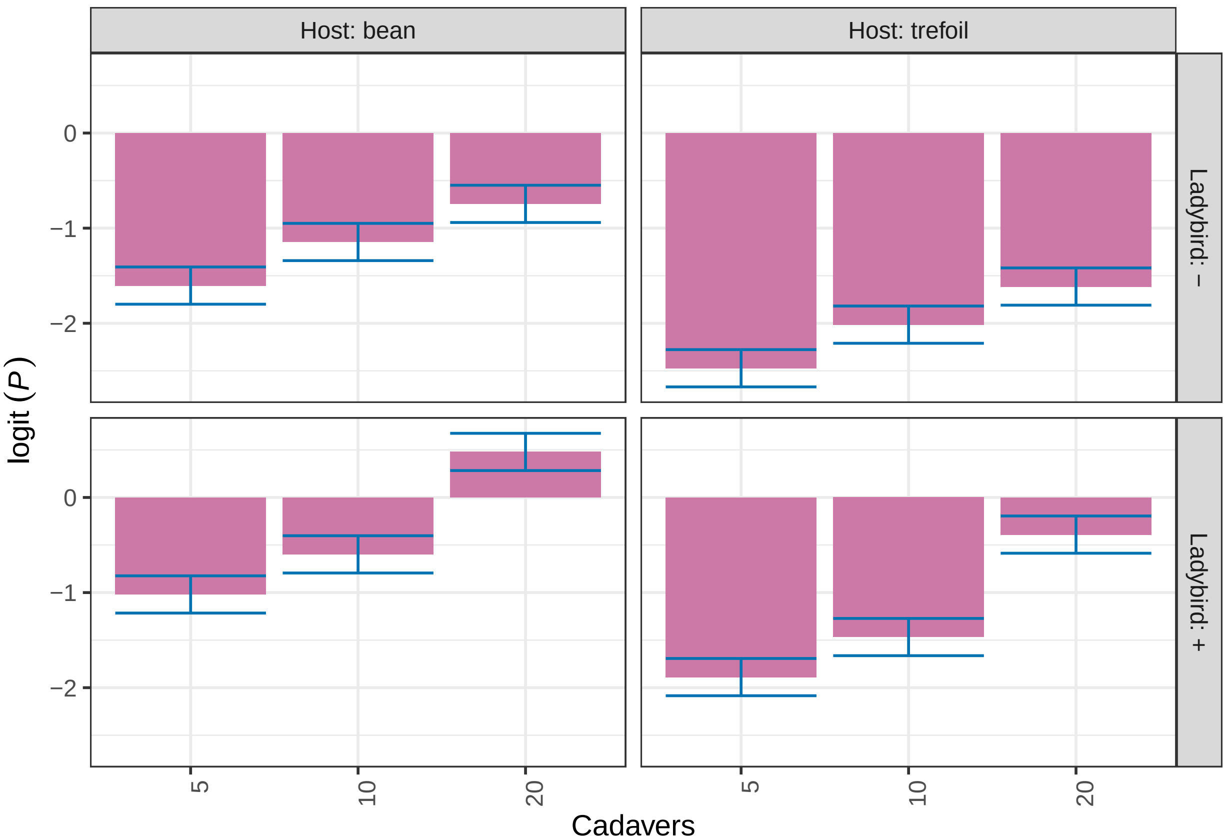


Figure 1. Estimated marginal means (EMMs) for logit(P), where P is the proportion of live aphids that were infected, for two Hosts, two Ladybird levels and three Cadaver levels. Error bars are an EMM ± half-LSD (5%). The two EMMs for the same Host are significantly different ($p \leq 0.05$) if their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

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.

Get and plot the predictions with a single function call

The `predictPresent` function combines the functionality of `predictPlus` and `plotPredictions`, as demonstrated now. Also, the use of `plotPvalues` to plot the pairwise *p*-values is displayed. The `predictPresent` function has the capability of producing `alldiffs` objects for multiple `terms` and these are stored in a list each of which is named for the term whose `alldiffs` object it stores. Thus, the term has to be specified in referencing components of `diffs`.

```
titl <- "logit(P)"
names(titl) <- "logitP"
diffs <- predictPresent(current.asrt$asreml.obj,
                        terms = "Host:Ladybird:Cadavers",
                        linear.transformation = ~Cadavers:Ladybird + Host,
                        titles = titl,
                        wald.tab = current.asrt$wald.tab,
                        error.intervals = "halfLeast",
                        LSDtype = "factor.combination", LSDby = "Host",
                        tables = "none",
                        ggplotFuncs = list(facet_grid(Ladybird ~ Host,
                                                      labeller = label_both)))
```

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

```
plotPvalues(diffs$Host.Ladybird.Cadavers, factors.per.grid = 1, show.sig = TRUE)
```

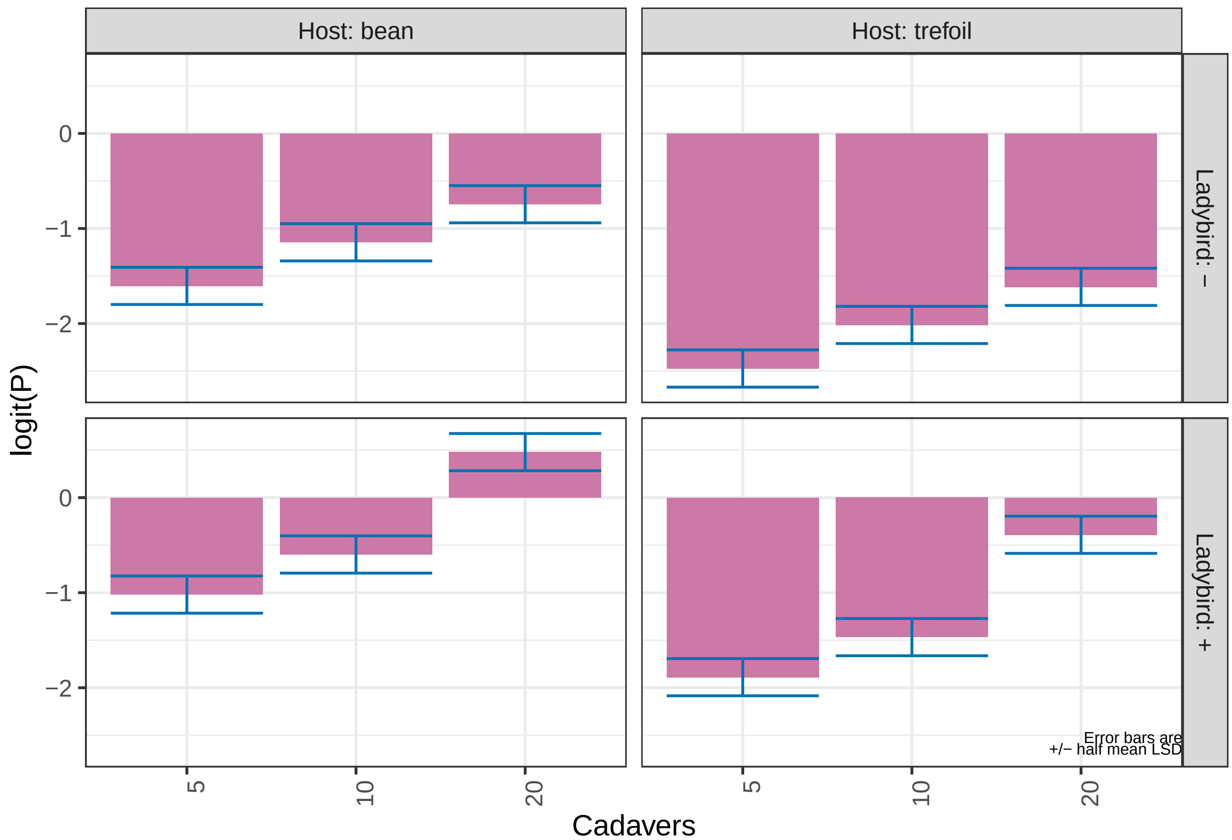


Figure 2. Estimated marginal means (EMMs) for logit(P), where P is the proportion of live aphids that were infected, for two Hosts, two Ladybird levels and three Cadaver levels. Error bars are an EMM $\pm$ half-LSD (5%). The two EMMs for the same Host are significantly different ($p \leq 0.05$) if their error bars do not overlap.

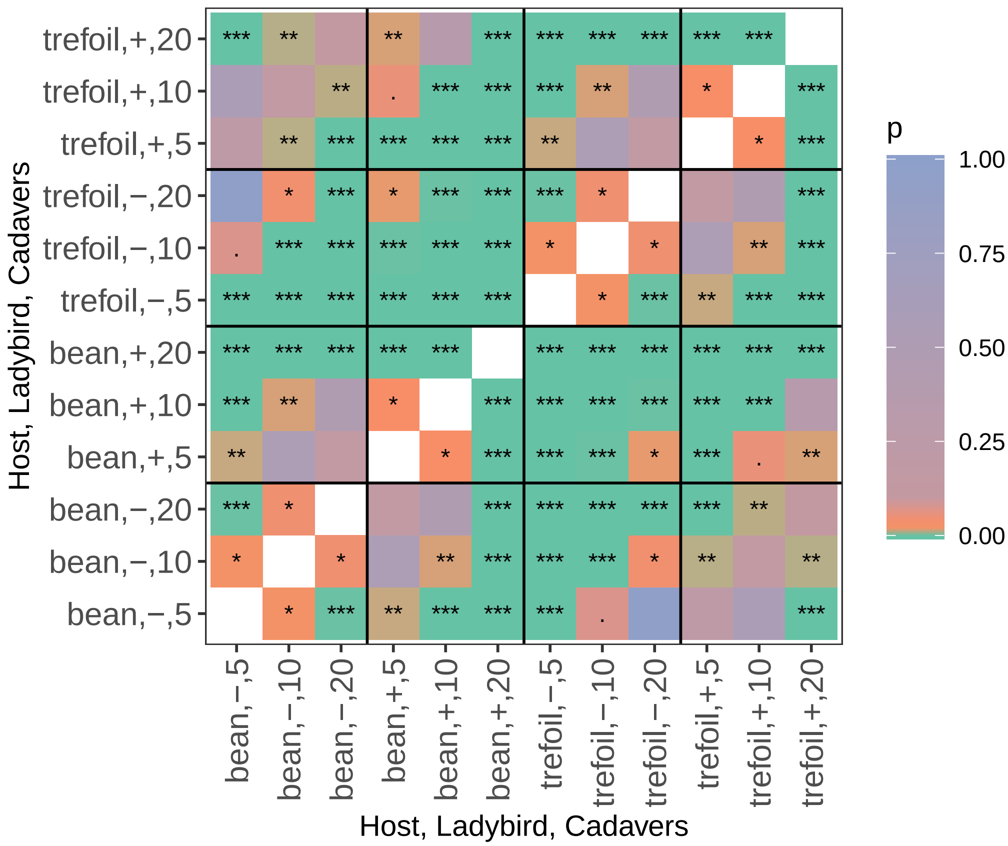


Figure 3. The p-values for each of the pairwise comparisons of the estimated marginal means for logit(P), where P is the proportion of live aphids that were infected, for two Hosts, two Ladybird levels and three Cadaver levels

```r
options(width = 90)
diffs$Host.Ladybird.Cadavers$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$Host.Ladybird.Cadavers$sed)
```

```
              bean,-,5 bean,-,10 bean,-,20  bean,+,5 bean,+,10 bean,+,20 trefoil,-,5
bean,-,5            NA 0.1957142 0.1957142 0.1957142 0.1957142 0.1957142   0.1129957
bean,-,10    0.1957142        NA 0.1957142 0.1957142 0.1957142 0.1957142   0.2259913
bean,-,20    0.1957142 0.1957142        NA 0.1957142 0.1957142 0.1957142   0.2259913
bean,+,5     0.1957142 0.1957142 0.1957142        NA 0.1957142 0.1957142   0.2259913
bean,+,10    0.1957142 0.1957142 0.1957142 0.1957142        NA 0.1957142   0.2259913
bean,+,20    0.1957142 0.1957142 0.1957142 0.1957142 0.1957142        NA   0.2259913
trefoil,-,5  0.1129957 0.2259913 0.2259913 0.2259913 0.2259913 0.2259913          NA
trefoil,-,10 0.2259913 0.1129957 0.2259913 0.2259913 0.2259913 0.2259913   0.1957142
trefoil,-,20 0.2259913 0.2259913 0.1129957 0.2259913 0.2259913 0.2259913   0.1957142
trefoil,+,5  0.2259913 0.2259913 0.2259913 0.1129957 0.2259913 0.2259913   0.1957142
trefoil,+,10 0.2259913 0.2259913 0.2259913 0.2259913 0.1129957 0.2259913   0.1957142
trefoil,+,20 0.2259913 0.2259913 0.2259913 0.2259913 0.2259913 0.1129957   0.1957142
             trefoil,-,10 trefoil,-,20 trefoil,+,5 trefoil,+,10 trefoil,+,20
bean,-,5        0.2259913    0.2259913   0.2259913    0.2259913    0.2259913
bean,-,10       0.1129957    0.2259913   0.2259913    0.2259913    0.2259913
bean,-,20       0.2259913    0.1129957   0.2259913    0.2259913    0.2259913
bean,+,5        0.2259913    0.2259913   0.1129957    0.2259913    0.2259913
bean,+,10       0.2259913    0.2259913   0.2259913    0.1129957    0.2259913
bean,+,20       0.2259913    0.2259913   0.2259913    0.2259913    0.1129957
trefoil,-,5     0.1957142    0.1957142   0.1957142    0.1957142    0.1957142
trefoil,-,10           NA    0.1957142   0.1957142    0.1957142    0.1957142
trefoil,-,20    0.1957142           NA   0.1957142    0.1957142    0.1957142
trefoil,+,5     0.1957142    0.1957142          NA    0.1957142    0.1957142
trefoil,+,10    0.1957142    0.1957142   0.1957142           NA    0.1957142
trefoil,+,20    0.1957142    0.1957142   0.1957142    0.1957142           NA
```

Perform the analysis with just 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.

```
ns.terms <- current.asrt$test.summary$terms[current.asrt$test.summary$action == "Nonsignificant"]
red.asrt <- changeTerms(current.asrt, dropFixed = paste(ns.terms, collapse = "+"))
summary(red.asrt$asreml.obj)$varcomp
```

```
               component   std.error   z.ratio bound %ch
Run         -0.004327122 0.002802876 -1.543815     U 0.8
Run:Plant!R  0.223431497 0.039503516  5.655990     P 0.0
```

```
print(red.asrt, which = "pseudoanova")
```

```
####  Pseudo-anova table for fixed terms


Wald tests for fixed effects.
Response: logitP

                  Df denDF   F.inc     Pr
(Intercept)        1     1 1550.00 0.0162
Host               1    64   60.88 0.0000
Cadavers           2    64   38.12 0.0000
Ladybird           1    64   49.65 0.0000
Cadavers:Ladybird  2    64    3.88 0.0256
```

```
diffs.red <- predictPlus(red.asrt$asreml.obj,
                         classify = "Host:Ladybird:Cadavers",
                         wald.tab = current.asrt$wald.tab,
                         error.intervals = "halfLeast", interval.annotate = FALSE,
                         LSDtype = "factor.combination", LSDby = "Host",
                         tables = "predictions")
```

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



#### Predictions for logitP from Host:Ladybird:Cadavers


Notes:
- The predictions are obtained by averaging across the hypertable
  calculated from model terms constructed solely from factors in
  the averaging and classify sets.
- Use 'average' to move ignored factors into the averaging set.
- The ignored set: Run
```

```
      Host Ladybird Cadavers predicted.value standard.error
1     bean        -        5      -1.6038338      0.1398332
2     bean        -       10      -1.1454308      0.1398332
3     bean        -       20      -0.7448097      0.1398332
4     bean        +        5      -1.0195475      0.1398332
5     bean        +       10      -0.5983440      0.1398332
6     bean        +       20       0.4786704      0.1398332
7  trefoil        -        5      -2.4730339      0.1398332
8  trefoil        -       10      -2.0146309      0.1398332
9  trefoil        -       20      -1.6140098      0.1398332
10 trefoil        +        5      -1.8887476      0.1398332
11 trefoil        +       10      -1.4675441      0.1398332
12 trefoil        +       20      -0.3905297      0.1398332
   upper.halfLeastSignificant.limit lower.halfLeastSignificant.limit est.status
1                        -1.4107942                       -1.7968734  Estimable
2                        -0.9523913                       -1.3384704  Estimable
3                        -0.5517702                       -0.9378493  Estimable
4                        -0.8265080                       -1.2125871  Estimable
5                        -0.4053044                       -0.7913835  Estimable
6                         0.6717100                        0.2856308  Estimable
7                        -2.2799944                       -2.6660735  Estimable
8                        -1.8215914                       -2.2076705  Estimable
9                        -1.4209703                       -1.8070494  Estimable
10                       -1.6957081                       -2.0817872  Estimable
11                       -1.2745045                       -1.6605837  Estimable
12                       -0.1974902                       -0.5835693  Estimable


LSD values

minimum LSD =  0.3860791 0.3860791

mean LSD =  0.3860791 0.3860791

maximum LSD =  0.3860791 0.3860791

(sed range / mean sed =  1.44e-16 1.44e-16 )
```

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

```
             bean,-,5 bean,-,10 bean,-,20  bean,+,5 bean,+,10 bean,+,20 trefoil,-,5
bean,-,5           NA 0.1929435 0.1929435 0.1929435 0.1929435 0.1929435   0.1113960
bean,-,10   0.1929435        NA 0.1929435 0.1929435 0.1929435 0.1929435   0.2227920
bean,-,20   0.1929435 0.1929435        NA 0.1929435 0.1929435 0.1929435   0.2227920
bean,+,5    0.1929435 0.1929435 0.1929435        NA 0.1929435 0.1929435   0.2227920
bean,+,10   0.1929435 0.1929435 0.1929435 0.1929435        NA 0.1929435   0.2227920
bean,+,20   0.1929435 0.1929435 0.1929435 0.1929435 0.1929435        NA   0.2227920
trefoil,-,5  0.1113960 0.2227920 0.2227920 0.2227920 0.2227920 0.2227920          NA
trefoil,-,10 0.2227920 0.1113960 0.2227920 0.2227920 0.2227920 0.2227920   0.1929435
trefoil,-,20 0.2227920 0.2227920 0.1113960 0.2227920 0.2227920 0.2227920   0.1929435
trefoil,+,5  0.2227920 0.2227920 0.2227920 0.1113960 0.2227920 0.2227920   0.1929435
trefoil,+,10 0.2227920 0.2227920 0.2227920 0.2227920 0.1113960 0.2227920   0.1929435
```

```
trefoil,+,20 0.2227920 0.2227920 0.2227920 0.2227920 0.2227920 0.1113960   0.1929435
             trefoil,-,10 trefoil,-,20 trefoil,+,5 trefoil,+,10 trefoil,+,20
bean,-,5        0.2227920    0.2227920   0.2227920    0.2227920    0.2227920
bean,-,10       0.1113960    0.2227920   0.2227920    0.2227920    0.2227920
bean,-,20       0.2227920    0.1113960   0.2227920    0.2227920    0.2227920
bean,+,5        0.2227920    0.2227920   0.1113960    0.2227920    0.2227920
bean,+,10       0.2227920    0.2227920   0.2227920    0.1113960    0.2227920
bean,+,20       0.2227920    0.2227920   0.2227920    0.2227920    0.1113960
trefoil,-,5     0.1929435    0.1929435   0.1929435    0.1929435    0.1929435
trefoil,-,10           NA    0.1929435   0.1929435    0.1929435    0.1929435
trefoil,-,20    0.1929435           NA   0.1929435    0.1929435    0.1929435
trefoil,+,5     0.1929435    0.1929435          NA    0.1929435    0.1929435
trefoil,+,10    0.1929435    0.1929435   0.1929435           NA    0.1929435
trefoil,+,20    0.1929435    0.1929435   0.1929435    0.1929435           NA
```

References

Brien, C. J. (2024a) `asremlPlus`: *Augments ASReml-R in fitting mixed models and packages generally in exploring prediction differences.* Version 4.4.43. 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/.

Butler, D. G., Cullis, B. R., Gilmour, A. R., Gogel, B. J. and Thompson, R. (2023). *ASReml-R Reference Manual Version 4.2.* VSN International Ltd, https://https://asreml.kb.vsni.co.uk/.

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 (2024) `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.