

# KG help

Gina Nichols

4/13/2020

Install the R package with the data

```
#devtools::install_github("vanichols/PFIweeds2020")
library(PFIweeds2020)
```

Other libraries

```
library(tidyverse)
```

## Tranformed Data Question

We wanted to know if cover-cropping (*cc\_trt*) affects the number of seeds in the weed seedbank (*totseeds\_m2*). We measured this at 4 sites (*site\_sys*).

```
dat <- pfifun_sum_byeu(pfi_ghobsraw) %>%
  unite(site_name, sys_trt, col = "site_sys", remove = T)
```

dat

```
## # A tibble: 46 x 7
## # Groups:   field, cc_trt, rep [36]
##   site_sys field cc_trt rep blockID totseeds totseeds_m2
##   <chr>    <chr> <chr> <dbl> <chr>    <dbl>    <dbl>
## 1 Boyd_grain B42 no      1 B42_1      34      663.
## 2 Boyd_grain B42 no      2 B42_2      12      234.
## 3 Boyd_grain B42 no      3 B42_3       9      175.
## 4 Boyd_grain B42 no      4 B42_4       7      136.
## 5 Boyd_grain B42 no      5 B42_5      17      331.
## 6 Boyd_grain B42 rye      1 B42_1     116     2261.
## 7 Boyd_grain B42 rye      2 B42_2      79     1540.
## 8 Boyd_grain B42 rye      3 B42_3       6      117.
## 9 Boyd_grain B42 rye      4 B42_4      29      565.
## 10 Boyd_grain B42 rye      5 B42_5      12      234.
## # ... with 36 more rows
```

## Fitting the models

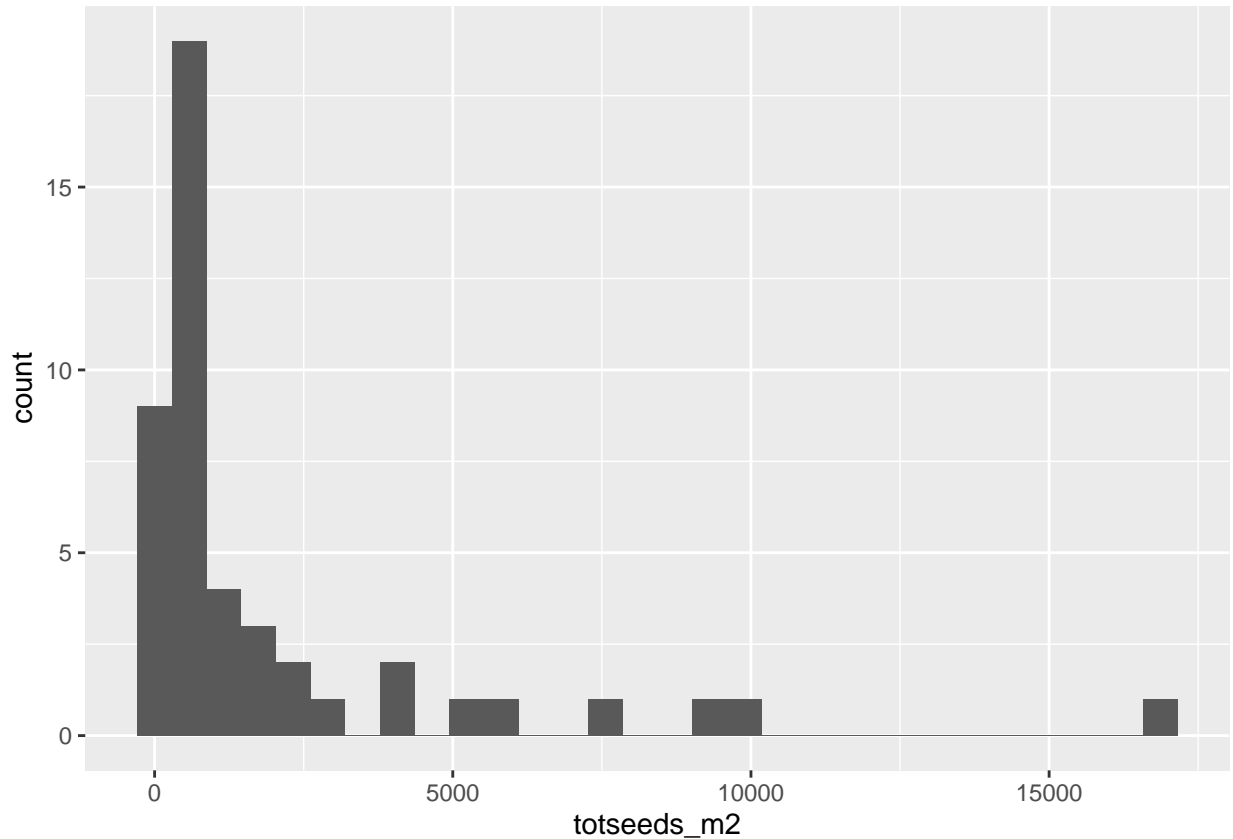
I want the *rye* treatment to appear first, so I renamed some things.

```
dstat <-
  dat %>%
  mutate(cc_trt2 = recode(cc_trt,          ##--I want rye to appear first alphabetically?
                           no = "none",
                           rye = "aryecc"))
```

There is one value that is very large. I need to look into how to assess ‘outlier-ness’, so just know I filter that value out.

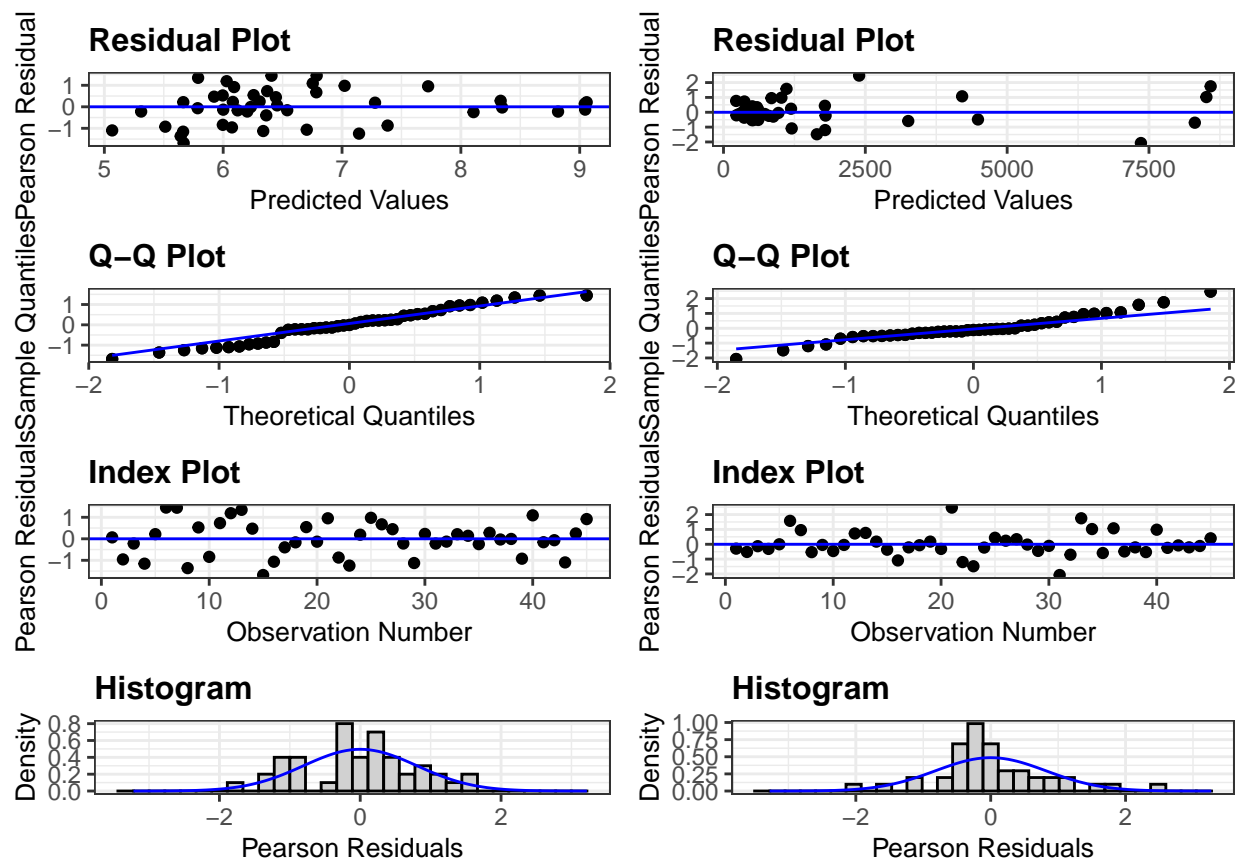
The response variable of interest, `totseeds_m2`, is skewed:

```
dat %>%  
  ggplot(aes(totseeds_m2)) +  
  geom_histogram()
```



Because of this (I think it was because it was skewed), we fit our statistical models to a log-transformed response variable. Is there a better way to justify this? Looking at the residual plots, I am not sure there is a clear answer.

```
library(ggResidpanel)  
  
m1 <- lmer(log(totseeds_m2) ~ site_sys * cc_trt2 + (1|blockID), data = filter(dstat, totseeds_m2 < 15000))  
m2 <- lmer((totseeds_m2) ~ site_sys * cc_trt2 + (1|blockID), data = filter(dstat, totseeds_m2 < 15000))  
  
ggResidpanel::resid_compare(list(m1, m2))
```



Pretending the log-transformed response variable is the best...

I fit my model:

```
m1 <- lmer(log(totseeds_m2) ~ site_sys * cc_trt2 + (1|blockID), data = filter(dstat, totseeds_m2 < 15000))
anova(m1)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##               Sum Sq Mean Sq NumDF   DenDF F value    Pr(>F)
## site_sys      19.2321   6.4107      3  21.054  15.3546 1.603e-05 ***
## cc_trt2        1.8205   1.8205      1  22.002   4.3603  0.04857 *
## site_sys:cc_trt2  3.3161   1.1054      3  21.804   2.6475  0.07445 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction between `site_sys` and `cc_trt2` is borderline significant, so I'm going to report things on a per-`site_sys` basis.

Now I use the **emmeans** package to compare the rye/none treatments in each `site_sys`, and the **broom** package to make it tidy.

```
m1em <- (emmeans(m1, pairwise ~ cc_trt2|site_sys, type = "response"))
m1em
```

```
## $emmeans
## site_sys = Boyd_grain:
##   cc_trt2 response    SE    df lower.CL upper.CL
## aryec   506    130 30.2      300      854
## none    363     93 30.2      215      612
```

```
##
## site_sys = Boyd_silage:
## cc_trt2 response SE df lower.CL upper.CL
## aryecc 549 192 36.9 270 1115
## none 1406 492 36.9 692 2856
##
## site_sys = Funcke_grain:
## cc_trt2 response SE df lower.CL upper.CL
## aryecc 3944 1831 33.5 1535 10134
## none 8033 3253 30.2 3514 18363
##
## site_sys = Stout_grain:
## cc_trt2 response SE df lower.CL upper.CL
## aryecc 334 135 30.2 146 764
## none 523 212 30.2 229 1196
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
##
## $contrasts
## site_sys = Boyd_grain:
## contrast ratio SE df t.ratio p.value
## aryecc / none 1.395 0.403 22.4 1.152 0.2614
##
## site_sys = Boyd_silage:
## contrast ratio SE df t.ratio p.value
## aryecc / none 0.390 0.160 22.4 -2.302 0.0310
##
## site_sys = Funcke_grain:
## contrast ratio SE df t.ratio p.value
## aryecc / none 0.491 0.250 24.9 -1.394 0.1755
##
## site_sys = Stout_grain:
## contrast ratio SE df t.ratio p.value
## aryecc / none 0.639 0.292 22.4 -0.979 0.3380
##
## Degrees-of-freedom method: kenward-roger
## Tests are performed on the log scale
```

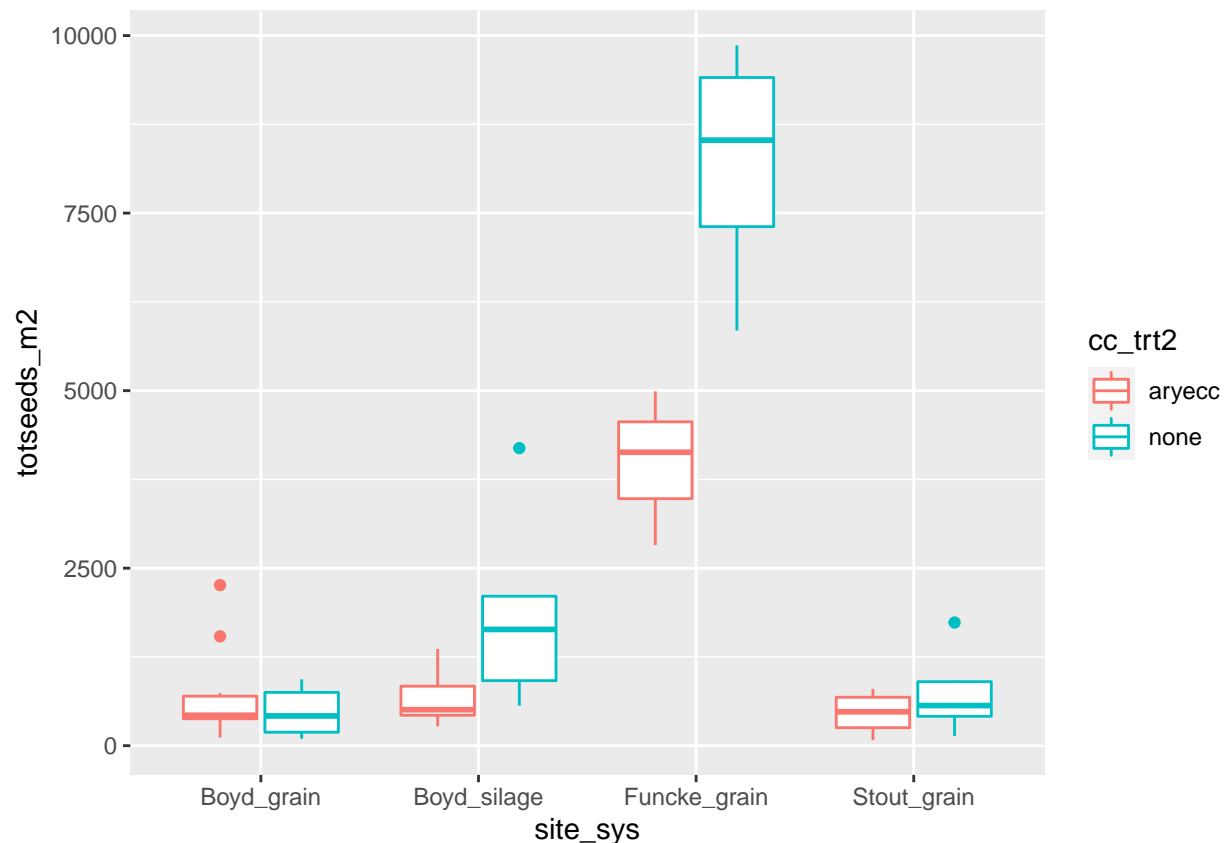
It looks like the response estimates are back-transformed (so it did that automatically?) and then the ratio is taken on the back-transformed data. E.g. for Boyd silage the ratio is 0.390 which is 549/1406.

Lydia found this on the help page: “With type =”response“, EMMs are computed by averaging together predictions on the *linear-predictor scale* and then back-transforming to the response scale”

What does that mean?

Raw data, to help think about this.

```
dstat %>% filter(totseeds_m2 < 15000) %>%
  ggplot(aes(site_sys, totseeds_m2)) +
  geom_boxplot(aes(color = cc_trt2))
```



```
mires <- broom::tidy(mlem$contrasts)
mires
```

```
## # A tibble: 4 x 7
##   contrast      site_sys    ratio std.error    df statistic p.value
##   <fct>         <fct>      <dbl>   <dbl> <dbl>   <dbl>   <dbl>
## 1 aryecc / none Boyd_grain  1.40    0.403   22.4     1.15    0.261
## 2 aryecc / none Boyd_silage  0.390    0.160   22.4    -2.30    0.0310
## 3 aryecc / none Funcke_grain 0.491    0.250   24.9    -1.39    0.176
## 4 aryecc / none Stout_grain  0.639    0.292   22.4    -0.979   0.338
```

## Question

How should I interpret/report this ratio? For example, in the **Boyd\_silage** *site\_sys*:

```
mires %>%
  filter(site_sys == "Boyd_silage") %>%
  select(contrast, site_sys, ratio, p.value)
```

```
## # A tibble: 1 x 4
##   contrast      site_sys    ratio p.value
##   <fct>         <fct>      <dbl>   <dbl>
## 1 aryecc / none Boyd_silage 0.390  0.0310
```

This means the *aryecc* treatment had 39% of the weeds as the *none* treatment? So rye reduces weed seedbanks by:

```
(1-0.39)*100
```

```
## [1] 61
```

61%?

Is it correct to use the **emmeans** values and just calculate the raw difference to present as well?

```
tidy(m1em$emmeans) %>%
  filter(site_sys == "Boyd_silage") %>%
  select(cc_trt2, site_sys, response) %>%
  pivot_wider(names_from = cc_trt2, values_from = response) %>%
  mutate(diff = none - aryec)
```

```
## # A tibble: 1 x 4
##   site_sys    aryec none diff
##   <fct>      <dbl> <dbl> <dbl>
## 1 Boyd_silage  549. 1406.  857.
```

Using *Boyd\_silage* as an example, would it be correct to say:

*At the Boyd-silage site, use of a rye cover crop reduced the number of weed seeds by 61% (SE = 16%, p = 0.03), corresponding to 857 less seeds per square meter.*

## Confusion about when response is log-transformed

Alternatively, **emmeans** offers something that seems to deal with responses that were transformed.

Instead of *type = response*, I say *type = lp*, but this doesn't seem appropriate.

What are these estimates? Is this assuming I log-transformed my data BEFORE putting it into the model, instead of doing it 'on the fly'?

```
m1em_alt2 <- emmeans(m1, pairwise ~ cc_trt2|site_sys, type = "lp") #--log scale? back-transformed? conf
broom::tidy(m1em_alt2$emmeans)
```

```
## # A tibble: 8 x 7
##   cc_trt2 site_sys    estimate std.error    df conf.low conf.high
##   <fct>   <fct>      <dbl>    <dbl> <dbl>   <dbl>    <dbl>
## 1 aryec Boyd_grain    6.23    0.256  30.2    5.70    6.75
## 2 none  Boyd_grain    5.89    0.256  30.2    5.37    6.42
## 3 aryec Boyd_silage    6.31    0.350  36.9    5.60    7.02
## 4 none  Boyd_silage    7.25    0.350  36.9    6.54    7.96
## 5 aryec Funcke_grain    8.28    0.464  33.5    7.34    9.22
## 6 none  Funcke_grain    8.99    0.405  30.2    8.16    9.82
## 7 aryec Stout_grain    5.81    0.405  30.2    4.99    6.64
## 8 none  Stout_grain    6.26    0.405  30.2    5.43    7.09
```

```
broom::tidy(m1em_alt2$contrasts)
```

```
## # A tibble: 4 x 8
##   level1 level2 site_sys    estimate std.error    df statistic p.value
##   <chr>  <chr>  <fct>      <dbl>    <dbl> <dbl>   <dbl>    <dbl>
## 1 aryec none  Boyd_grain    0.333    0.289  22.4    1.15    0.261
## 2 aryec none  Boyd_silage  -0.941    0.409  22.4   -2.30    0.0310
## 3 aryec none  Funcke_grain -0.711    0.510  24.9   -1.39    0.176
## 4 aryec none  Stout_grain  -0.447    0.457  22.4   -0.979   0.338
```

Taking exponent of the estimate, for example on *Boyd silage*, returns the 0.39 value again. So it seems like using this isn't appropriate in my situation. Even though I did use a log-transformation. This is where I'm confused.