Soilpad analysis

Jarad Niemi

2022-03-20

Contents

```
knitr::opts_chunk$set(echo = TRUE,
                      cache = TRUE,
                      fig.width = 12,
                      fig.height = 12)

library("lme4")
```

```
## Loading required package: Matrix
```

```
library("lmerTest")
```

```
##
## Attaching package: 'lmerTest'

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

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

```
library("tidyverse"); theme_set(theme_bw())
```

```
## -- Attaching packages ------------------------------------ tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1


## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
```

```r
library("emmeans")
library("ggResidpanel")
library("data.table")
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
library("stringr")
#install.packages("rstatix")
library("ggpubr")
```

```
## Warning: package 'ggpubr' was built under R version 4.1.3
```

```r
library("rstatix")
```

```
## Warning: package 'rstatix' was built under R version 4.1.3

##
## Attaching package: 'rstatix'

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

```r
library("broom")
```

```
## Warning: package 'broom' was built under R version 4.1.3
```

```r
options(width = 120)

dir.create("fig", showWarnings = FALSE)

sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252  LC_CTYPE=English_United States.1252    LC_MONETARY=Englis
## [4] LC_NUMERIC=C                           LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] broom_0.7.12       rstatix_0.7.0      ggpubr_0.4.0       data.table_1.14.0 ggResidpanel_0.3.0
##  [7] forcats_0.5.1      stringr_1.4.0      dplyr_1.0.7        purrr_0.3.4       readr_2.0.1
## [13] tibble_3.1.4       ggplot2_3.3.5      tidyverse_1.3.1    lmerTest_3.1-3    lme4_1.1-27.1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-152       fs_1.5.0           lubridate_1.7.10   httr_1.4.2         numDeriv_2016.8-
##  [6] tools_4.1.1        backports_1.2.1    utf8_1.2.2         R6_2.5.1           DBI_1.1.1
## [11] lazyeval_0.2.2     colorspace_2.0-2   withr_2.4.2        tidyselect_1.1.1   curl_4.3.2
## [16] compiler_4.1.1     cli_3.0.1          rvest_1.0.1        xml2_1.3.2         plotly_4.9.4.1
## [21] scales_1.1.1       DEoptimR_1.0-9     mvtnorm_1.1-2      robustbase_0.93-8  digest_0.6.27
## [26] foreign_0.8-81     minqa_1.2.4        rmarkdown_2.10     qqplotr_0.0.5      rio_0.5.27
## [31] pkgconfig_2.0.3    htmltools_0.5.1.1  dbplyr_2.1.1       htmlwidgets_1.5.3  rlang_0.4.11
## [36] readxl_1.3.1       rstudioapi_0.13    generics_0.1.0     jsonlite_1.7.2     zip_2.2.0
## [41] car_3.0-11         magrittr_2.0.1     Rcpp_1.0.7         munsell_0.5.0      fansi_0.5.0
## [46] abind_1.4-5        lifecycle_1.0.0    stringi_1.7.3      yaml_2.2.1         carData_3.0-4
## [51] MASS_7.3-54        grid_4.1.1         crayon_1.4.1       lattice_0.20-44    haven_2.4.3
## [56] cowplot_1.1.1      splines_4.1.1      hms_1.1.0          knitr_1.33         pillar_1.6.2
## [61] boot_1.3-28        estimability_1.3   ggsignif_0.6.3     reprex_2.0.1       glue_1.4.2
## [66] evaluate_0.14      modelr_0.1.8       vctrs_0.3.8        nloptr_1.2.2.2     tzdb_0.1.2
## [71] cellranger_1.1.0   gtable_0.3.0       assertthat_0.2.1   openxlsx_4.2.4     xfun_0.25
## [76] xtable_1.8-4       viridisLite_0.4.0  ellipsis_0.3.2
```

Read in data

```r
library("tidyverse")

flume <- read_csv("../data/tidy/flume_event_data612_UPDATE.csv") %>%
  mutate(Year = factor(Year)) %>%
  subset(subtreatment != 'grass strip') %>%
  subset(SiteID != 'MCN') %>%
```

```
  subset(subset=!(SiteID=="RHO" & Year == 2016)) %>%
  subset(subset=!(SiteID=="RHO" & Year == 2017))
```

```
## Rows: 432 Columns: 15
```

```
## -- Column specification ------------------------------------------------------
## Delimiter: ","
## chr  (5): SiteID, subtreatment, Treatment, sampleID, random
## dbl (10): precipitation, rain_time, rf_event, sample_event, ro_event, Year, flow_time, flow, tss_sum
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
flume_sum <- flume %>%
  group_by(Treatment, Year, SiteID, sample_event, tss_sum) %>%
  summarize(tss_load = tss_sum,
            ln_tss_load = log(tss_load+0.000198)) %>%
  distinct()
```

```
## 'summarise()' has grouped output by 'Treatment', 'Year', 'SiteID', 'sample_event', 'tss_sum'. You ca
```

```
ppt_sum <- flume %>%
  group_by(Treatment, Year, SiteID, sample_event) %>%
  summarize(ppt_sum = sum(precipitation)) %>%
  ungroup() %>%
  filter(!duplicated(cbind(Year, SiteID, sample_event)))
```

```
## 'summarise()' has grouped output by 'Treatment', 'Year', 'SiteID'. You can override using the '.grou
```

```
sample_anova <- flume_sum %>%
  filter(!is.na(tss_sum)) %>%
  select(Year, SiteID, Treatment, sample_event, tss_sum) %>%
  group_by(SiteID, Year, Treatment, sample_event) %>%
    summarize(tss_load = sum(tss_sum)) %>%
  ungroup() %>%
  select(Year, SiteID, Treatment, sample_event, tss_load) %>%
  pivot_wider(names_from = Treatment, values_from = tss_load)
```

```
## 'summarise()' has grouped output by 'SiteID', 'Year', 'Treatment'. You can override using the '.grou
```

```
pivot_sample <- sample_anova %>%
  inner_join(ppt_sum,by=c("SiteID", "Year", "sample_event")) %>%
  mutate(ln_ppt = log(ppt_sum),
         ln_ctl = log(control+0.005322915),
         ln_trt = log(strips+0.0104)) %>%
  subset(select = -c(Treatment))

long_load <- sample_anova %>%
  gather(Treatment, tss_load, control:strips) %>%
  arrange(Treatment, tss_load) %>%
  select(SiteID, Treatment, Year, sample_event, tss_load)
```

```
rf_ro_pivot <- long_load %>%
  mutate(random = (ifelse(SiteID == 'ARM', 'NR',
  ifelse(SiteID == 'EIA', 'R',
  ifelse(SiteID == 'MCN', 'R',
  ifelse(SiteID == 'HOE', 'NR',
  ifelse(SiteID == 'MAR', 'NR',
  ifelse(SiteID == 'RHO', 'R',
  ifelse(SiteID == 'WHI', 'NR',
  ifelse(SiteID == 'WOR', 'R', 0)))))))))
```

```
full_df <- rf_ro_pivot %>%
  inner_join(ppt_sum,by=c("SiteID", "Year", "sample_event")) %>%
  mutate(ln_ppt = log(ppt_sum),
         #ln_tss_load = log(tss_load+0.005322915),
         Treatment = Treatment.x) %>%
  subset(select = -c(Treatment.y, Treatment.x)) %>%
  arrange(Year, SiteID, Treatment, sample_event)
```
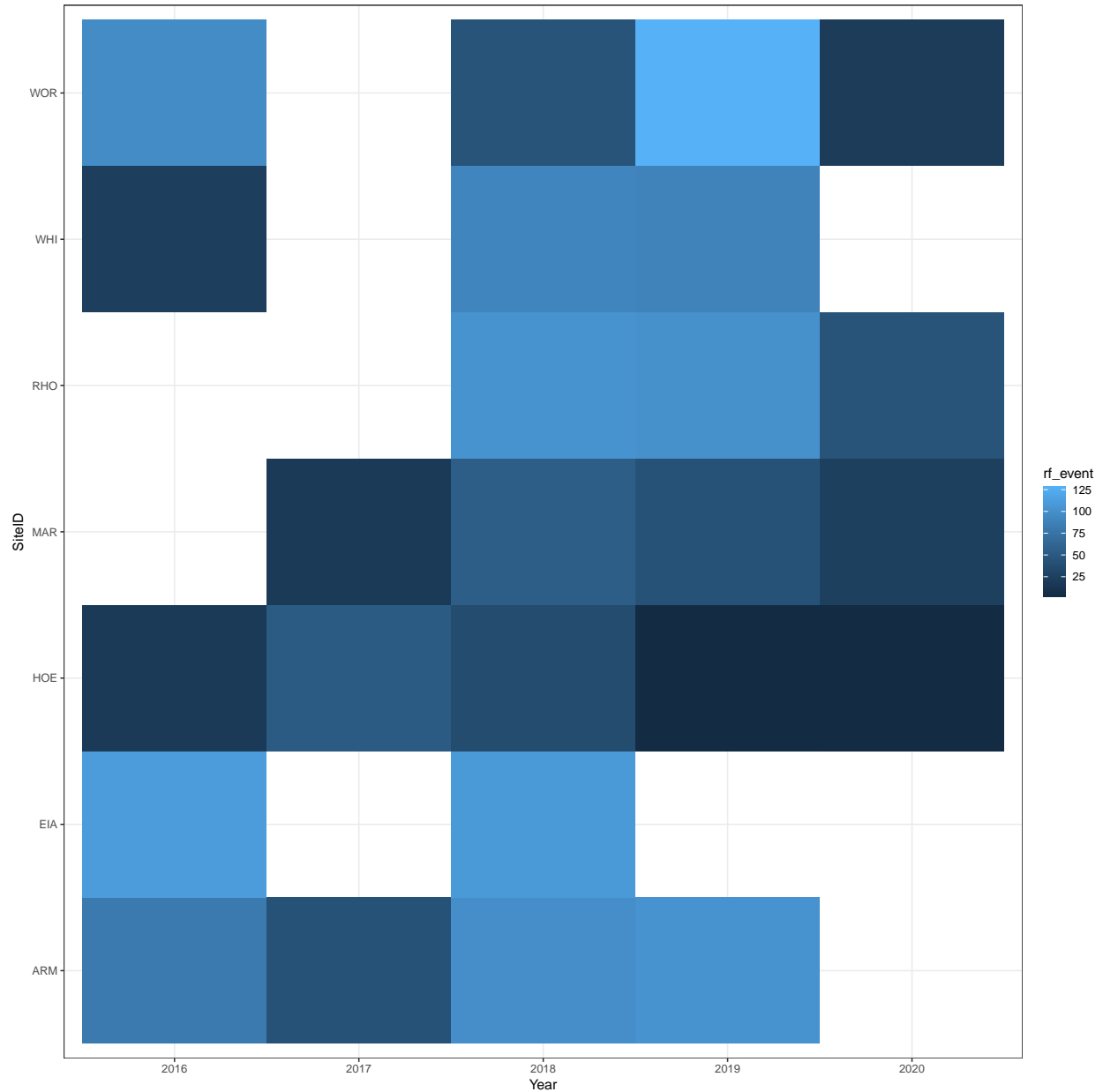
Exploratory analysis

Site-year with rainfall event

```
site_year_rfevent <- flume %>%
  select(SiteID, Year, rf_event) %>%
  unique()
```

```
ggplot(site_year_rfevent, aes(Year, SiteID, fill=rf_event)) +
  geom_tile()
```

Number of samples

Calculate the number of observations for each treatment-position-year-site-time combination.

```
TSS_counts <- flume %>%
  group_by(Year, SiteID, rf_event) %>%
  distinct() %>%
  summarize(n = n(), .groups = "drop")
```

Plot the number of observations for each combination.

```
g <- ggplot(TSS_counts, aes(x = Year, y= SiteID, fill = n)) +
  geom_tile() +
  geom_text(aes(label = n), color = "white") +
  scale_fill_gradient(low = "blue", high = "red") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

g
```
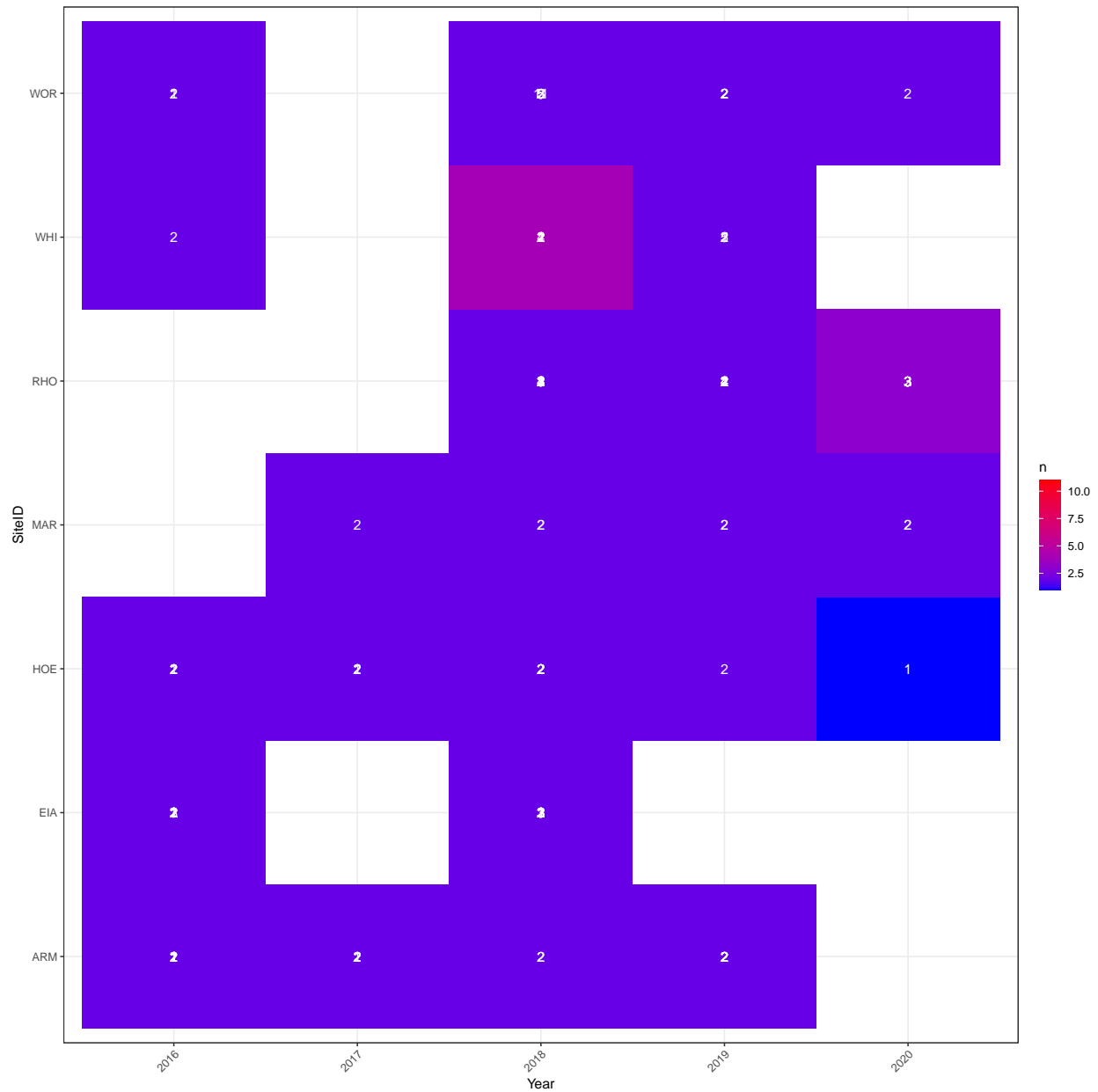


```
##ggsave("fig/soilpad_counts_no_diversion.png", g, width = 12, height = 12)
```

Data visualization

```
#h <- ggplot(full_df, aes(x=ln_ppt, y=ln_tss_load, color=Treatment), inherit.aes = FALSE) +
#  geom_point() +
#  geom_smooth(method=lm, se=FALSE, fullrange=TRUE) +
#  ggtitle("Log-log relationship between TSS load and rainfall accumulation \n(full dataset)") +
#  theme(plot.title = element_text(size=14, face="bold",hjust = 0.5))


#ggsave("fig/randReg_ppt_load.png", h, width = 12, height = 12)


#pivot_sample %>%
#  anova_test(ln_trt ~ ln_ppt*ln_ctl)
## purr https://stackoverflow.com/questions/50702152/compare-models-via-anova-with-purrr-or-dplyr
## anova() and may need an linear model built up.
```

Main Analyses

There are three main analyses of interest:

- confirmatory, design-based analysis
- exploratory, covariate analysis
- relationship of sediment flow to sediment loss

Confirmatory, design-based analysis

Treatment effect

```
m_flume <- lmerTest::lmer(log(tss_load+0.005322915) ~
                          #(1 | SiteID) +
                          (1 | SiteID:Treatment) +
                          Treatment*ln_ppt +
                          Year,
                      data = full_df)

summary(m_flume)


## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: log(tss_load + 0.005322915) ~ (1 | SiteID:Treatment) + Treatment *      ln_ppt + Year
##    Data: full_df
##
## REML criterion at convergence: 976.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3160 -0.5896  0.0503  0.6579  2.4023
##
## Random effects:
```

```
##   Groups            Name           Variance Std.Dev.
##  SiteID:Treatment (Intercept) 2.758    1.661
##  Residual                     6.388    2.527
## Number of obs: 205, groups:  SiteID:Treatment, 14
##
## Fixed effects:
##                       Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)             4.4278     1.2677  95.8559   3.493 0.000725 ***
## Treatmentstrips        -4.2363     1.7151  87.7991  -2.470 0.015445 *
## ln_ppt                  1.6637     0.3164 187.4190   5.259 3.94e-07 ***
## Year2017                1.3674     0.7734 193.2561   1.768 0.078636 .
## Year2018                1.2971     0.5425 194.8437   2.391 0.017762 *
## Year2019                2.2507     0.5949 196.5246   3.784 0.000205 ***
## Year2020               -0.0821     0.8955 196.1304  -0.092 0.927046
## Treatmentstrips:ln_ppt -0.6535     0.4458 187.5339  -1.466 0.144360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) Trtmnt ln_ppt Yr2017 Yr2018 Yr2019 Yr2020
## Trtmntstrps -0.677
## ln_ppt       0.779 -0.583
## Year2017    -0.162  0.000  0.048
## Year2018    -0.280  0.000  0.020  0.456
## Year2019    -0.228  0.001  0.071  0.451  0.676
## Year2020    -0.194  0.004  0.019  0.309  0.481  0.478
## Trtmntstr:_ -0.560  0.828 -0.705  0.000  0.000  0.000  0.000
```

```
m_flume_step <- step(m_flume, reduce.random = FALSE, alpha.fixed = 0.1)
m_flume_model <- get_model(m_flume_step)
summary(m_flume_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: log(tss_load + 0.005322915) ~ (1 | SiteID:Treatment) + Treatment +      ln_ppt + Year
##    Data: full_df
##
## REML criterion at convergence: 979.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1653 -0.5918  0.0390  0.6712  2.3583
##
## Random effects:
##  Groups            Name           Variance Std.Dev.
##  SiteID:Treatment (Intercept) 2.734    1.654
##  Residual                     6.430    2.536
## Number of obs: 205, groups:  SiteID:Treatment, 14
##
## Fixed effects:
##               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)    3.38849    1.05065  56.49578   3.225 0.002095 **
## Treatmentstrips -2.15443    0.95847  11.62356  -2.248 0.044847 *
## ln_ppt          1.33724    0.22525 188.35424   5.937 1.37e-08 ***
## Year2017        1.36510    0.77580 194.36456   1.760 0.080048 .
```

```
## Year2018          1.29717    0.54417 195.94883    2.384 0.018091 *
## Year2019          2.25025    0.59662 197.59061    3.772 0.000214 ***
## Year2020         -0.08321    0.89818 197.20859   -0.093 0.926284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Trtmnt ln_ppt Yr2017 Yr2018 Yr2019
## Trtmntstrps -0.457
## ln_ppt       0.656  0.000
## Year2017    -0.196 -0.001  0.067
## Year2018    -0.339  0.001  0.029  0.455
## Year2019    -0.276  0.001  0.100  0.451  0.675
## Year2020    -0.235  0.007  0.027  0.309  0.481  0.478
```

##https://campus.datacamp.com/courses/hierarchical-and-mixed-effects-models-in-r/linear-mixed-effect-mo

```
treatment = emmeans(m_flume, pairwise ~ Treatment,
                 type = "response", # calculated log ahead of time instead of in model; the minus in
                 lmer.df = "asymptotic")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
confint(treatment)$contrasts
```

```
##  contrast         ratio   SE  df asymp.LCL asymp.UCL
##  control / strips  8.78 8.44 Inf      1.33      57.8
##
## Results are averaged over the levels of: Year
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
```

```
year = emmeans(m_flume, ~ Year,
                 type = "response",
                 lmer.df = "asymptotic")
confint(year)
```

```
##  Year response      SE  df asymp.LCL asymp.UCL
##  2016     0.142 0.0933 Inf    0.0374     0.504
##  2017     0.574 0.4508 Inf    0.1206     2.658
##  2018     0.534 0.2892 Inf    0.1835     1.537
##  2019     1.395 0.7992 Inf    0.4525     4.280
##  2020     0.131 0.1164 Inf    0.0200     0.723
##
## Results are averaged over the levels of: Treatment
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log(mu + 0.005) scale
```

```
treatment = emmeans(m_flume, pairwise ~ Treatment|ln_ppt,
                    at=list(ln_ppt=c(-4,-3,-2)),
                    type = "response",
                    lmer.df = "asymptotic")

confint(treatment)$contrasts ## exp. the values
```

```
## ln_ppt = -4:
##  contrast          ratio     SE  df asymp.LCL asymp.UCL
##  control / strips  5.06   5.21 Inf     0.676      38.0
##
## ln_ppt = -3:
##  contrast          ratio     SE  df asymp.LCL asymp.UCL
##  control / strips  9.74   9.40 Inf     1.468      64.5
##
## ln_ppt = -2:
##  contrast          ratio     SE  df asymp.LCL asymp.UCL
##  control / strips 18.71  20.54 Inf     2.179     160.8
##
## Results are averaged over the levels of: Year
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
```
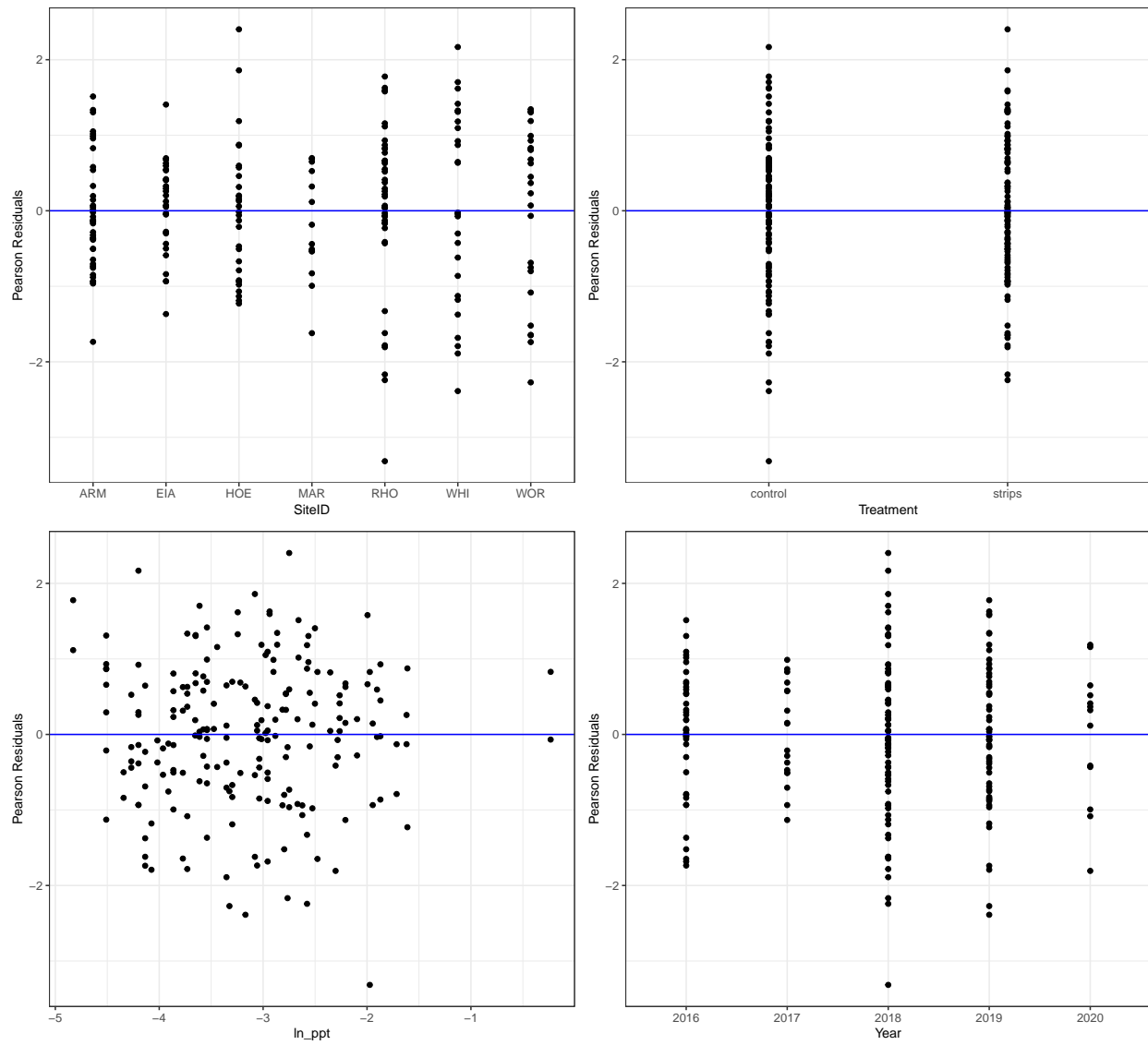
```
resid_panel(m_flume,
            plots = c("resid","index","yvp","qq"),
            smoother = TRUE, qqbands = TRUE)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```
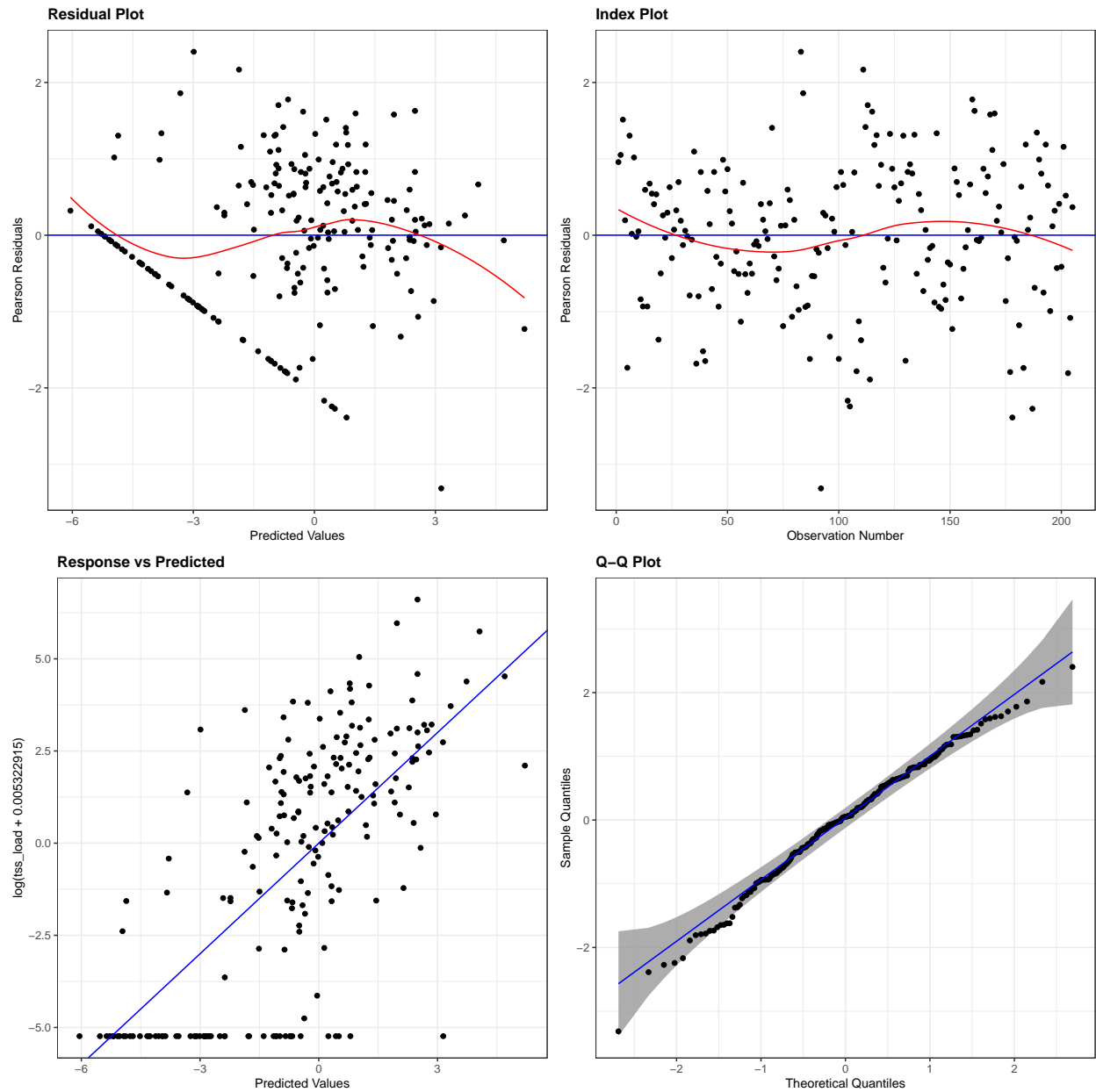
```
resid_xpanel(m_flume)
```

Plots of Residuals vs Predictor Variables

Possibly heavy-tailed residuals.

Non-constant variance amongst Site and Diversion.

Ratio comparison

List of PadIDs that had a diversion pair.