analysis_random

Jnelly

3/16/2022

```
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")

options(width = 120)

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

```r
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] data.table_1.14.0  ggResidpanel_0.3.0 emmeans_1.7.0      forcats_0.5.1      stringr_1.4.0
##  [7] purrr_0.3.4        readr_2.0.1        tidyr_1.1.3        tibble_3.1.4       ggplot2_3.3.5
## [13] lmerTest_3.1-3     lme4_1.1-27.1      Matrix_1.3-4
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2          viridisLite_0.4.0   jsonlite_1.7.2      splines_4.1.1       modelr_0.1.8
##  [6] assertthat_0.2.1    cellranger_1.1.0    robustbase_0.93-8   yaml_2.2.1          numDeriv_2016.8
## [11] pillar_1.6.2        backports_1.2.1     lattice_0.20-44     glue_1.4.2          digest_0.6.27
## [16] rvest_1.0.1         minqa_1.2.4         colorspace_2.0-2    cowplot_1.1.1       htmltools_0.5.1
## [21] pkgconfig_2.0.3     broom_0.7.12        haven_2.4.3         xtable_1.8-4        mvtnorm_1.1-2
## [26] scales_1.1.1        tzdb_0.1.2          generics_0.1.0      ellipsis_0.3.2      withr_2.4.2
## [31] lazyeval_0.2.2      cli_3.0.1           magrittr_2.0.1      crayon_1.4.1        readxl_1.3.1
## [36] estimability_1.3    evaluate_0.14       fs_1.5.0            fansi_0.5.0         nlme_3.1-152
## [41] MASS_7.3-54         xml2_1.3.2          tools_4.1.1         hms_1.1.0           lifecycle_1.0.0
## [46] plotly_4.9.4.1      munsell_0.5.0       reprex_2.0.1        qqplotr_0.0.5       compiler_4.1.1
## [51] rlang_0.4.11       grid_4.1.1         nloptr_1.2.2.2     rstudioapi_0.13    htmlwidgets_1.5
## [56] rmarkdown_2.10     boot_1.3-28        gtable_0.3.0       DBI_1.1.1          R6_2.5.1
## [61] lubridate_1.7.10   knitr_1.33         utf8_1.2.2         stringi_1.7.3      Rcpp_1.0.7
## [66] vctrs_0.3.8        DEoptimR_1.0-9     dbplyr_2.1.1       tidyselect_1.1.1   xfun_0.25
```

Read in data

```
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 ‘.group
```

```
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)
```

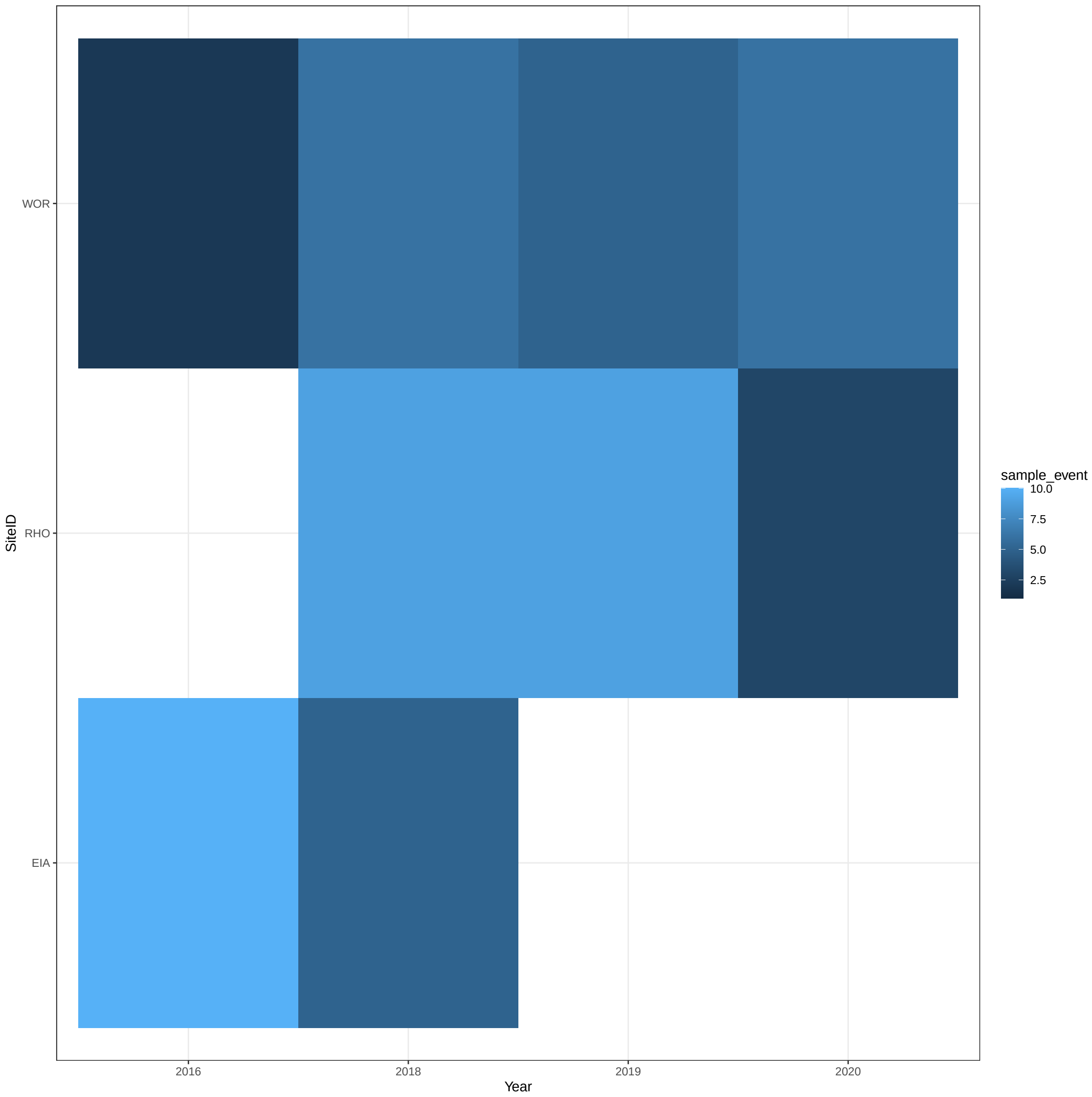
```
flumeR <- full_df %>%
  #filter(!is.na(ro_event)) %>%
    subset(random == 'R')
```

Exploratory analysis

Site-year with rainfall event

```
site_year_rfeventR <- flumeR %>%
  select(SiteID, Year, sample_event) %>%
  unique()

ggplot(site_year_rfeventR, aes(Year, SiteID, fill=sample_event)) +
  geom_tile()
```



Number of samples

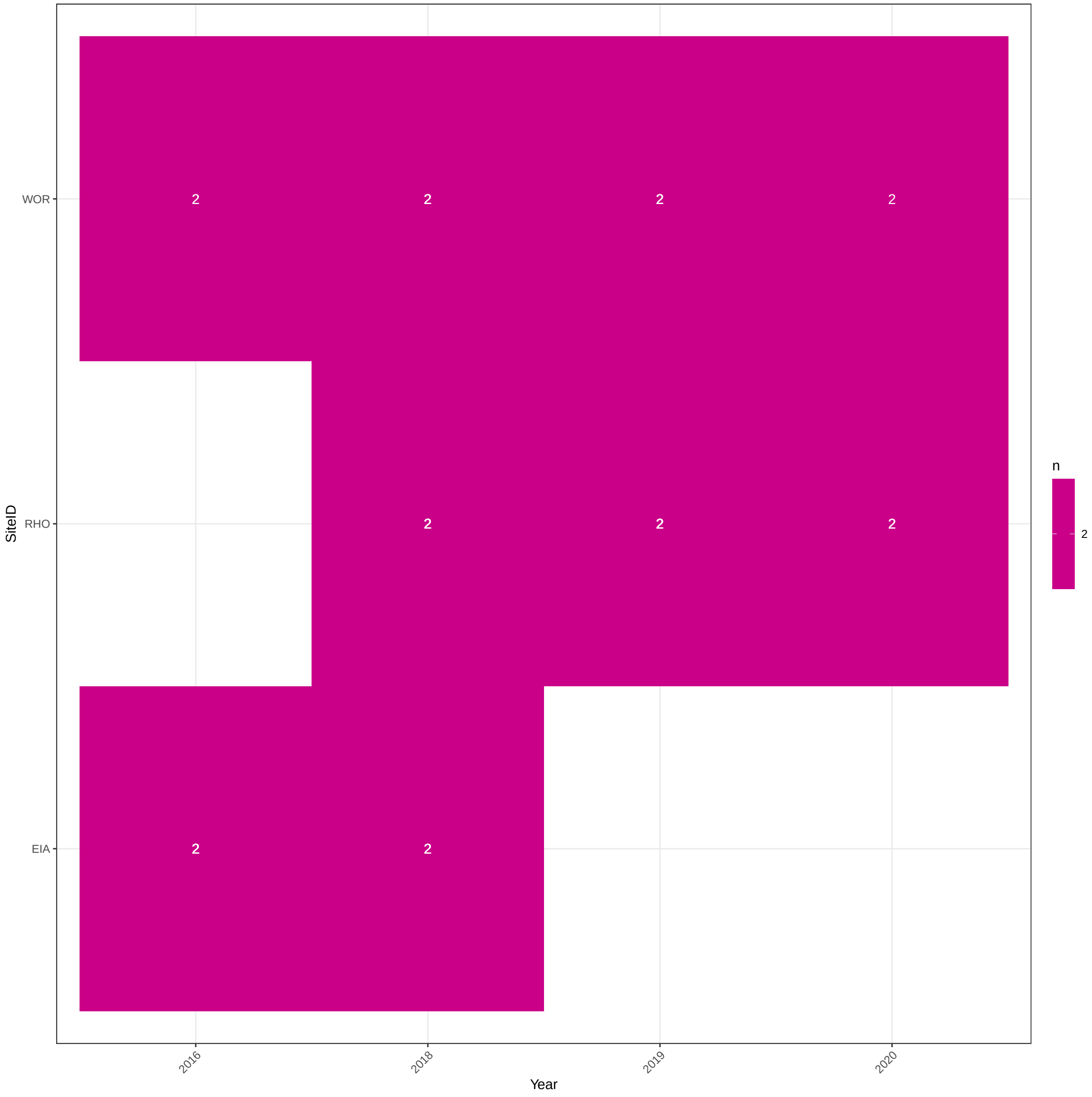
Calculate the number of observations for each treatment-position-year-site-time combination.

```
TSS_countsR <- flumeR %>%
  group_by(Year, SiteID, sample_event) %>%
  distinct() %>%
  summarize(n = n(), .groups = "drop")
```

Plot the number of observations for each combination.

```
g <- ggplot(TSS_countsR, 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

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


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

Main Analyses

There are three main analyses of interest:

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

```
mR_flume <- lmerTest::lmer(log(tss_load+0.005322915) ~
                             #(1 | SiteID) +
                             (1 | SiteID:Treatment) +
                             Treatment*ln_ppt +
                            Year,
                          data = flumeR)
summary(mR_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: flumeR
##
## REML criterion at convergence: 441.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0533 -0.3752  0.2366  0.6493  1.6999
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  SiteID:Treatment (Intercept) 0.1527   0.3908
##  Residual                     6.4009   2.5300
## Number of obs: 96, groups:  SiteID:Treatment, 6
##
## Fixed effects:
##                         Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)               4.3659     1.4501 76.2117   3.011  0.00353 **
## Treatmentstrips          -2.2907     1.8630 77.3261  -1.230  0.22259
## ln_ppt                    1.8466     0.4163 85.2418   4.436 2.72e-05 ***
## Year2018                  1.5913     0.7110 50.8269   2.238  0.02964 *
## Year2019                  3.9138     0.7728 21.3928   5.064 4.88e-05 ***
## Year2020                 0.1517     1.0829 50.9683   0.140   0.88911
## Treatmentstrips:ln_ppt  -0.5993     0.5750 85.0914  -1.042   0.30030
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Trtmnt ln_ppt Yr2018 Yr2019 Yr2020
## Trtmntstrps -0.642
## ln_ppt       0.891 -0.653
## Year2018    -0.410  0.000 -0.111
## Year2019    -0.226  0.000  0.077  0.602
## Year2020    -0.248  0.000 -0.040  0.453  0.430
## Trtmntstr:_ -0.607  0.945 -0.691  0.000  0.000  0.000
```

```
treatmentR = emmeans(mR_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(treatmentR)$contrasts
```

```
##  contrast         ratio    SE  df asymp.LCL asymp.UCL
##  control / strips  1.58 0.963 Inf     0.478      5.22
##
## 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
```

```
treatment = emmeans(mR_flume, pairwise ~ Treatment|ln_ppt,
                    at=list(ln_ppt=c(-3,-2, -1)),
                    type = "response",
                    lmer.df = "asymptotic")

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

```
## ln_ppt = -3:
##  contrast         ratio   SE  df asymp.LCL asymp.UCL
##  control / strips  1.64 1.00 Inf     0.494      5.42
##
## ln_ppt = -2:
##  contrast         ratio   SE  df asymp.LCL asymp.UCL
##  control / strips  2.98 2.57 Inf     0.549     16.17
##
## ln_ppt = -1:
##  contrast         ratio   SE  df asymp.LCL asymp.UCL
##  control / strips  5.43 7.23 Inf     0.398     74.01
##
## 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
```

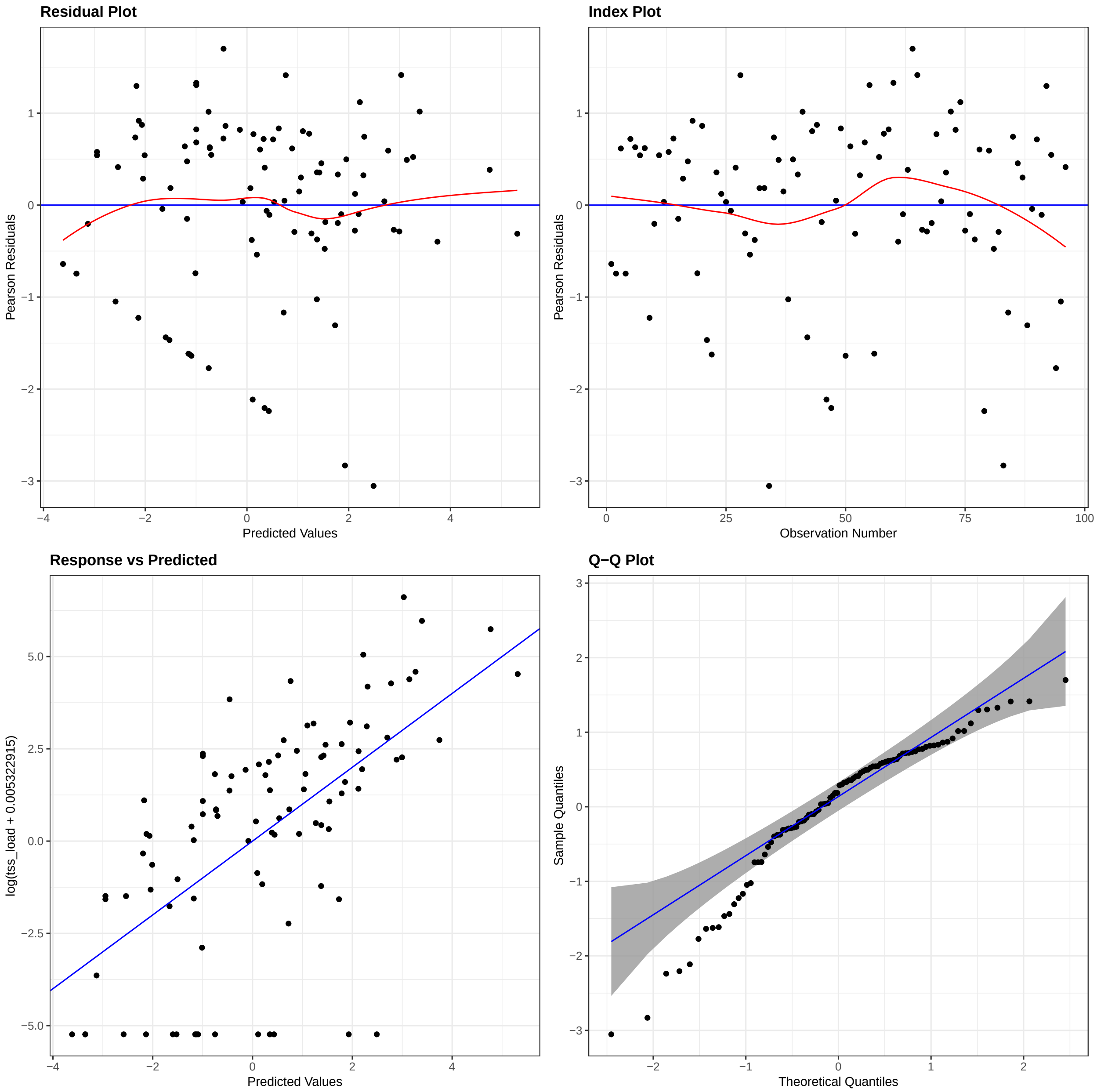
```
year = emmeans(mR_flume, ~ Year,
                    type = "response",
                    lmer.df = "asymptotic")

confint(year)## exp. the values
```

```
##  Year response    SE  df asymp.LCL asymp.UCL
##  2016    0.215 0.128 Inf    0.0648     0.686
##  2018    1.075 0.487 Inf    0.4417     2.607
##  2019   11.019 5.819 Inf    3.9126    31.017
##  2020    0.251 0.235 Inf    0.0370     1.546
##
## 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
```

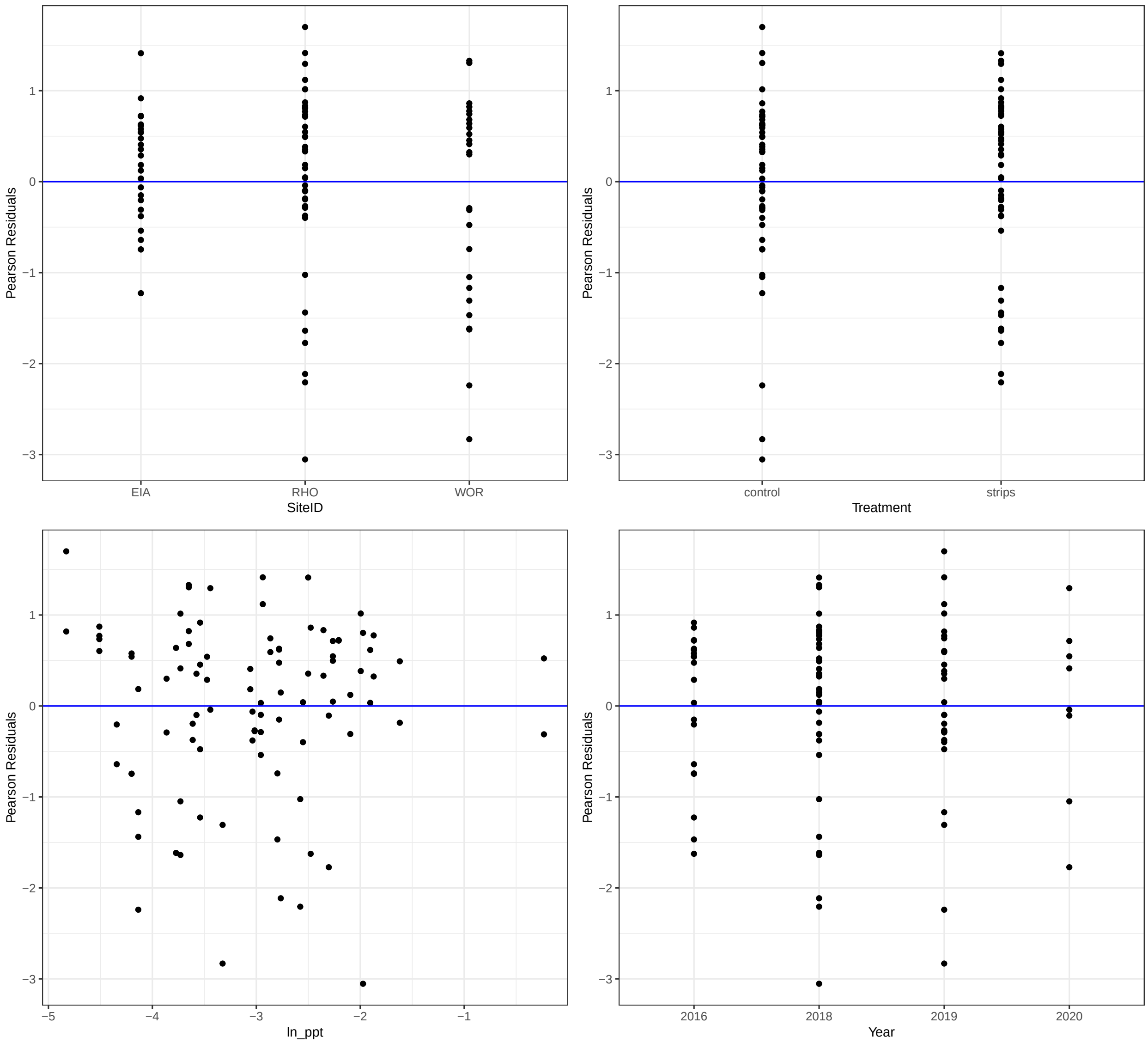
```
resid_panel(mR_flume,
            plots = c("resid","index","yvp","qq"),
            smoother = TRUE, qqbands = TRUE)
```

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



```
resid_xpanel(mR_flume)
```

Plots of Residuals vs Predictor Variables


Pearson Residuals
EIA
RHO
WOR
SiteID
Pearson Residuals
control
strips
Treatment
Pearson Residuals
ln_ppt
Pearson Residuals
Year