Flume: Random Analysis (March-November)

(adapted from Jarad Niemi - Soilpad Analysis)

Jessica Nelson

2022-05-02

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  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     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()
```

```
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
```

```
library("stringr")
library("ggpmisc")
```

```
## Loading required package: ggpp

##
## Attaching package: 'ggpp'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
options(width = 120, scipen = 999)

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

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
##
## 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_1.14.2  ggResidpanel_0.3.0 emmeans_1.7.2      forcats_0.5.1      stringr_1.4.0      
##  [7] purrr_0.3.4        readr_2.1.2        tidyr_1.2.0        tibble_3.1.6       ggplot2_3.3.5      
## [13] lmerTest_3.1-3     lme4_1.1-28        Matrix_1.4-0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2          viridisLite_0.4.0   jsonlite_1.8.0      splines_4.1.3       modelr_0.1.8
##  [6] assertthat_0.2.1    cellranger_1.1.0    robustbase_0.93-9   yaml_2.3.5          numDeriv_2016.8
## [11] pillar_1.7.0        backports_1.4.1     lattice_0.20-45     glue_1.6.2          digest_0.6.29
## [16] rvest_1.0.2         minqa_1.2.4         colorspace_2.0-3    cowplot_1.1.1       htmltools_0.5.2
## [21] pkgconfig_2.0.3     broom_0.7.12        haven_2.4.3         xtable_1.8-4        mvtnorm_1.1-3
## [26] scales_1.1.1        tzdb_0.2.0          generics_0.1.2      ellipsis_0.3.2      withr_2.5.0
## [31] lazyeval_0.2.2      cli_3.2.0           magrittr_2.0.2      crayon_1.5.0        readxl_1.3.1
## [36] estimability_1.3    evaluate_0.15       fs_1.5.2            fansi_1.0.2         nlme_3.1-155
## [41] MASS_7.3-55         xml2_1.3.3          tools_4.1.3         hms_1.1.1           lifecycle_1.0.1
## [46] plotly_4.10.0       munsell_0.5.0       reprex_2.0.1        qqplotr_0.0.5       compiler_4.1.3
## [51] rlang_1.0.2         grid_4.1.3          nloptr_2.0.0        rstudioapi_0.13     htmlwidgets_1.5
## [56] rmarkdown_2.13      boot_1.3-28         gtable_0.3.0        DBI_1.1.2           R6_2.5.1
## [61] lubridate_1.8.0     knitr_1.38          fastmap_1.1.0       utf8_1.2.2          stringi_1.7.6
## [66] Rcpp_1.0.8          vctrs_0.3.8         DEoptimR_1.0-10     dbplyr_2.1.1        tidyselect_1.1.1
## [71] xfun_0.30
```

Read in data

```
library("tidyverse")

options(scipen = 999)

flume <- read_csv("../data/tidy/flume_event_data612_UPDATE.csv") %>%
  mutate(Year = factor(Year)) %>%
  subset(SiteID != 'MAR') %>%
  subset(subset=!(SiteID=="MCN" & Year == 2016)) %>%
  subset(subset=!(SiteID=="MCN" & Year == 2017)) %>%
  subset(subset=!(SiteID=="MCN" & Year == 2018)) %>%
  subset(subset=!(SiteID=="MCN" & Year == 2019)) %>%
  subset(subset=!(SiteID=="MCN" & Year == 2020)) %>%
```

```
  subset(subset=!(SiteID=="RHO" & Year == 2016)) %>%
  subset(subset=!(SiteID=="RHO" & Year == 2017)) %>%
  subset(subset=!(SiteID == "WOR" & Year == 2018 & precipitation == "NA"))
```

```
## Rows: 439 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): SiteID, subtreatment, Treatment, sampleID, rf_event, random, crop, f_loc
## dbl (15): precipitation, rain_time, slope75, Lfactor, Sfactor, LSfactor, sample_event, ro_event, Yea
##
## 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, crop, slope75, LSfactor, Lfactor,Sfactor) %>%
  summarize(tss_load = tss_sum) %>%
  distinct()
```

```
## `summarise()` has grouped output by 'Treatment', 'Year', 'SiteID', 'sample_event', 'tss_sum', 'crop'
## 'LSfactor', 'Lfactor', 'Sfactor'. You can override using the `.groups` argument.
```

```
ppt_sum <- flume %>%
  group_by(Treatment, Year, SiteID, sample_event, crop, slope75, LSfactor, Lfactor,Sfactor) %>%
  #group_by(Treatment, Year, SiteID, sample_event, crop) %>%
  summarize(ppt_sum = sum(precipitation)) %>%
  ungroup() %>%
  filter(!duplicated(cbind(Year, SiteID, sample_event)))
```

```
## `summarise()` has grouped output by 'Treatment', 'Year', 'SiteID', 'sample_event', 'crop', 'slope75'
## 'Lfactor'. You can override using the `.groups` argument.
```

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

```
## Adding missing grouping variables: `slope75`, `LSfactor`, `Lfactor`, `Sfactor`
## `summarise()` has grouped output by 'SiteID', 'Year', 'Treatment', 'sample_event'. You can override u
## argument.
```

```
pivot_sample <- sampl_anova %>%
  inner_join(ppt_sum,by=c("SiteID", "Year", "sample_event", "crop")) %>%
  filter(!is.na(strips)) %>%
  mutate(ln_ppt = log(ppt_sum)) %>%
  subset(select = -c(Treatment))
```

```
long_load <- pivot_sample %>%
  gather(Treatment, tss_load, control:strips) %>%
  arrange(Treatment, tss_load) %>%
  filter(!is.na(diff)) %>%
  select(SiteID, Treatment, Year, sample_event, tss_load, ppt_sum, crop)
```

```
## Warning in is.na(diff): is.na() applied to non-(list or vector) of type 'closure'
```

```
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 == 'RHO', 'R',
  ifelse(SiteID == 'WHI', 'NR',
  ifelse(SiteID == 'WOR', 'R', 0)))))))))

long_load <- long_load %>%
  mutate(random = (ifelse(SiteID == 'ARM', 'NR',
  ifelse(SiteID == 'EIA', 'R',
  ifelse(SiteID == 'MCN', 'R',
  ifelse(SiteID == 'HOE', '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","crop")) %>%
  drop_na(tss_load) %>%
  mutate(ppt_sum = ppt_sum.x,
         ln_ppt = log(ppt_sum.x),
         Treatment = factor(Treatment.x, levels=c('strips','control'))) %>%
  subset(select = -c(Treatment.y, Treatment.x, ppt_sum.x, ppt_sum.y)) %>%
  arrange(Year, SiteID, Treatment, sample_event)

save(full_df, file = "full_df.RData")

#write.csv(full_df,"D:/ISU/ResearchProject/flume_analysis/data/tidy/full_df.csv", row.names = FALSE)
```

```
load("full_df.RData")
```

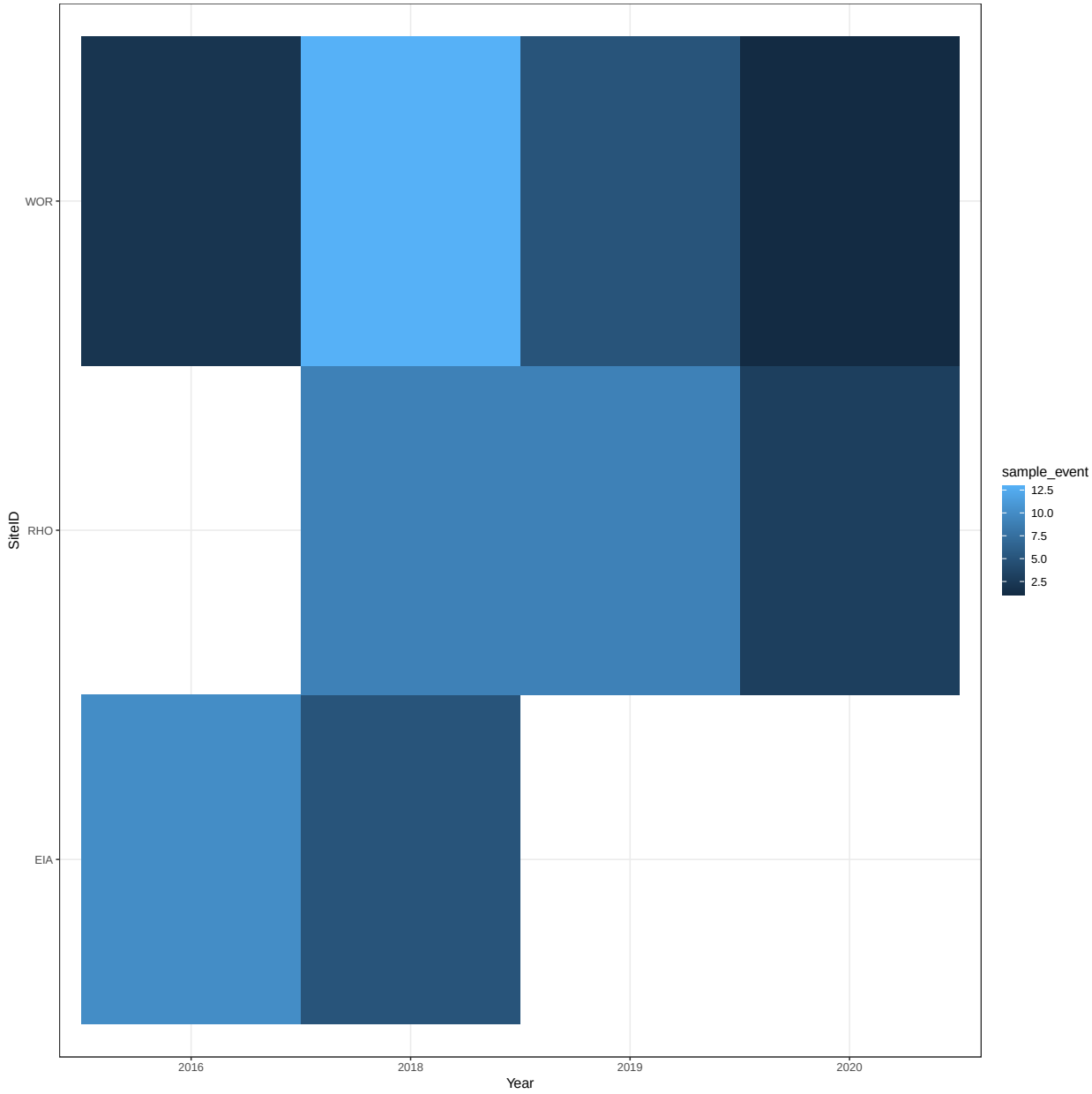
```
flumeR <- full_df %>%
  #filter(!is.na(ro_event)) %>%
    subset(random == 'R')
```

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.

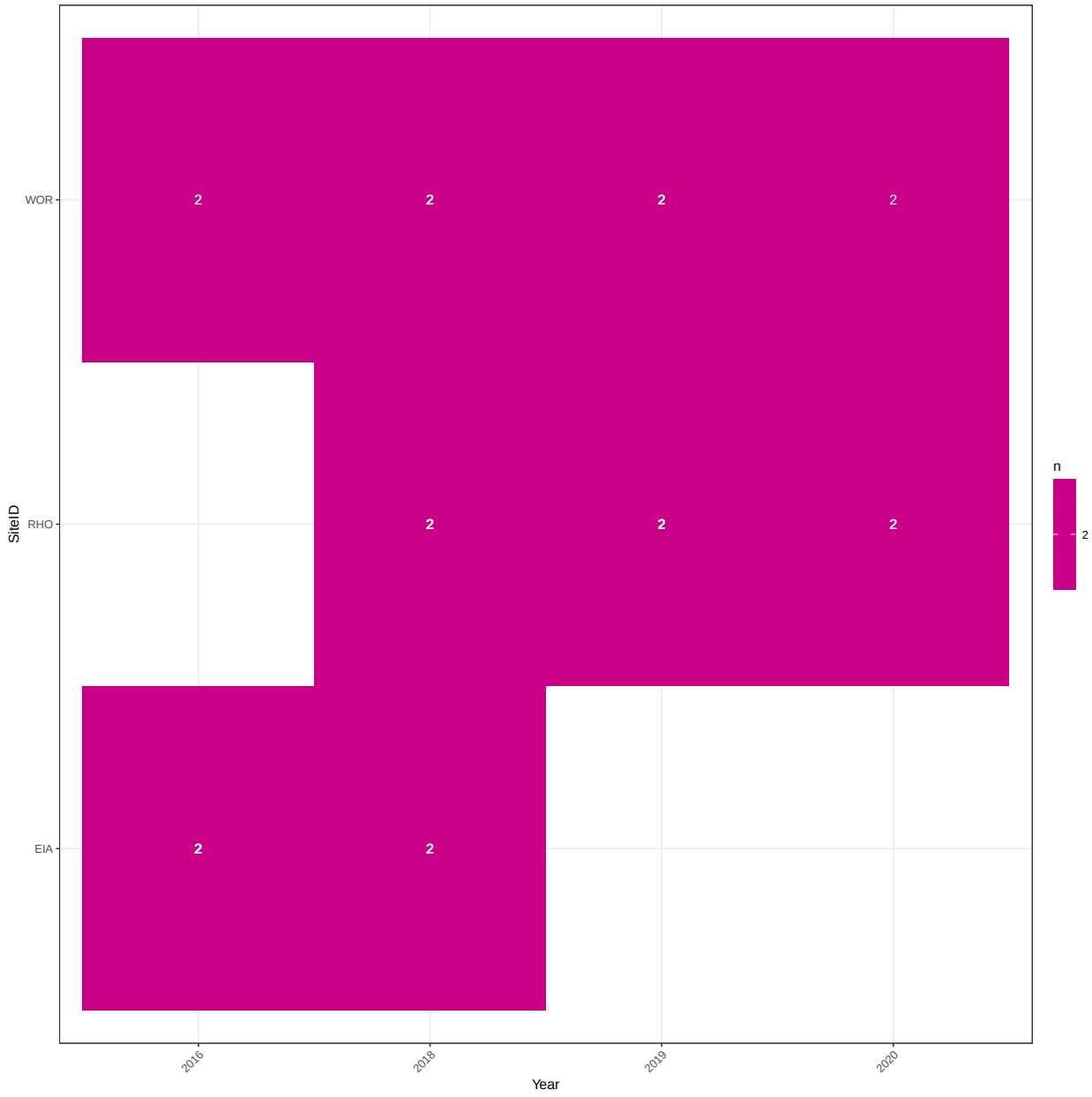
```
TSS_countsR <- flumeR %>%
  group_by(Year, SiteID, sample_event) %>%
  distinct() %>%
  summarize(n = n(), .groups = "drop")
```

```
TSS_sumR <- flumeR %>%
  group_by(Year, SiteID, Treatment) %>%
  distinct() %>%
  summarize(tss_sum = sum(tss_load), .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
```

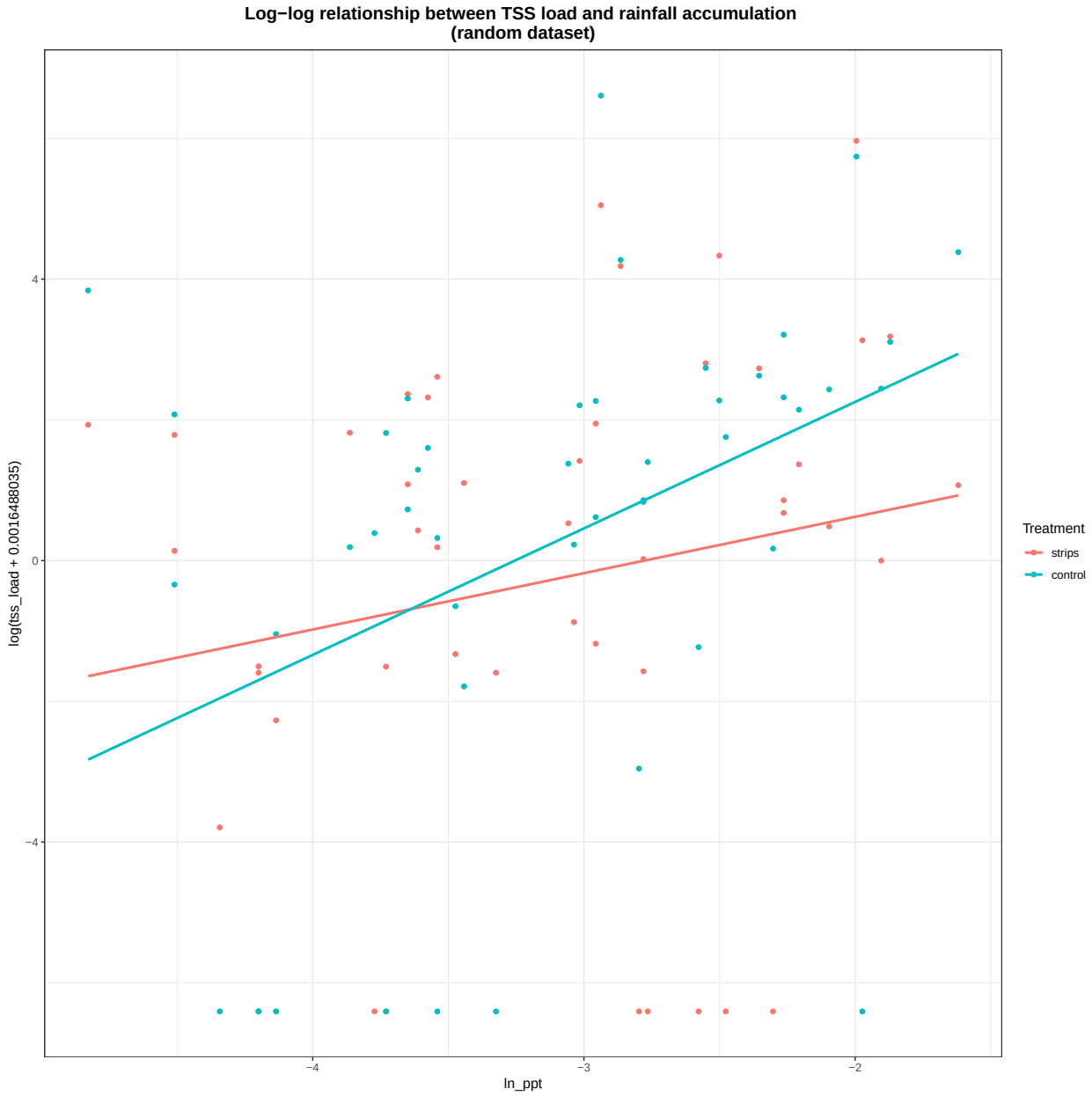



```
##ggsave("fig/soilpad_counts_no_diversion.png", g, width = 12, height = 12)
```

Data visualization

```
hR <- ggplot(flumeR, aes(x=ln_ppt, y=log(tss_load+0.0016488035), 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))
hR
```

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




```
#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

```r
mR_flume <- lmerTest::lmer(log(tss_load+0.0016) ~
                             Treatment*ln_ppt +
                            Treatment*crop +
                            Year*Treatment +

                            (1 | SiteID) +
                             (1 | SiteID:Treatment) + #removed due to singular fit
                            #(1|Year:sample_event) + #removed due to singular fit
                            (1|SiteID:Year:sample_event),

                          data = flumeR)


summary(mR_flume)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: log(tss_load + 0.0016) ~ Treatment * ln_ppt + Treatment * crop +
##     Year * Treatment + (1 | SiteID) + (1 | SiteID:Treatment) +      (1 | SiteID:Year:sample_event)
##    Data: flumeR
##
## REML criterion at convergence: 433.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3246 -0.4455  0.1256  0.5249  1.7918
##
## Random effects:
##  Groups                    Name        Variance    Std.Dev.
##  SiteID:Year:sample_event (Intercept) 1.878911613 1.370734
##  SiteID:Treatment         (Intercept) 0.000001584 0.001259
##  SiteID                   (Intercept) 0.111049000 0.333240
##  Residual                             6.419341517 2.533642
## Number of obs: 94, groups:  SiteID:Year:sample_event, 47; SiteID:Treatment, 6; SiteID, 3
##
## Fixed effects:
##                              Estimate Std. Error      df t value  Pr(>|t|)
## (Intercept)                   -0.0750     2.1019 73.7524  -0.036    0.9716
## Treatmentcontrol               4.9519     2.6007 40.9979   1.904    0.0639 .
## ln_ppt                         1.3377     0.5414 77.5111   2.471    0.0157 *
## cropsoybean                    2.8966     1.2322  8.4052   2.351    0.0451 *
## Year2018                       2.3493     1.2776 77.7838   1.839    0.0697 .
## Year2019                       6.6500     1.5234 77.6841   4.365 0.0000387 ***
## Year2020                       1.6980     1.8197 75.7538   0.933    0.3537
## Treatmentcontrol:ln_ppt        0.6474     0.6727 40.9980   0.962    0.3415
## Treatmentcontrol:cropsoybean  -3.5816     1.4665 40.9748  -2.442    0.0190 *
## Treatmentcontrol:Year2018     -0.2812     1.5864 40.9980  -0.177    0.8602
## Treatmentcontrol:Year2019     -3.3258     1.8923 40.9980  -1.758    0.0863 .
## Treatmentcontrol:Year2020     -2.0497     2.2531 40.9979  -0.910    0.3683
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Trtmnt ln_ppt crpsyb Yr2018 Yr2019 Yr2020 Trtm:_ Trtmn: T:Y2018 T:Y2019
```

```
## Trtmntcntrl -0.619
## ln_ppt       0.781 -0.487
## cropsoybean -0.396  0.241  0.069
## Year2018    -0.558  0.351 -0.064  0.462
## Year2019    -0.419  0.263  0.134  0.603  0.696
## Year2020    -0.393  0.248 -0.039  0.325  0.515  0.503
## Trtmntcnt:_ -0.485  0.785 -0.621 -0.041  0.041 -0.083  0.025
## Trtmntcntr:  0.251 -0.406 -0.042 -0.595 -0.306 -0.398 -0.226  0.068
## Trtmn:Y2018  0.349 -0.565  0.041 -0.293 -0.621 -0.432 -0.318 -0.067  0.493
## Trtmn:Y2019  0.262 -0.424 -0.083 -0.381 -0.432 -0.621 -0.311  0.133  0.640  0.695
## Trtmn:Y2020  0.248 -0.401  0.025 -0.217 -0.319 -0.312 -0.619 -0.041  0.365  0.513   0.502
```

```
mR_flume_step <- step(mR_flume, reduce.random = FALSE, alpha.fixed = 0.1)
```

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

```
mR_flume_model <- get_model(mR_flume_step)
summary(mR_flume_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: log(tss_load + 0.0016) ~ Treatment + ln_ppt + crop + Year + (1 |
##     SiteID) + (1 | SiteID:Treatment) + (1 | SiteID:Year:sample_event) +      Treatment:crop + Treatm
##    Data: flumeR
##
## REML criterion at convergence: 435.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2221 -0.4348  0.1160  0.5014  1.6929
##
## Random effects:
##  Groups                   Name        Variance Std.Dev.
##  SiteID:Year:sample_event (Intercept) 1.89382  1.3762
##  SiteID:Treatment         (Intercept) 0.04855  0.2203
##  SiteID                   (Intercept) 0.08685  0.2947
##  Residual                             6.38990  2.5278
## Number of obs: 94, groups:  SiteID:Year:sample_event, 47; SiteID:Treatment, 6; SiteID, 3
##
## Fixed effects:
##                               Estimate Std. Error      df t value  Pr(>|t|)
## (Intercept)                     0.9036     1.8397 54.0824   0.491  0.625293
## Treatmentcontrol                2.9947     1.6212 25.6561   1.847  0.076281 .
## ln_ppt                          1.6614     0.4243 40.6845   3.916  0.000336 ***
## cropsoybean                     2.9481     1.2416  6.9638   2.374  0.049463 *
## Year2018                        2.3098     1.2759 78.1842   1.810  0.074095 .
## Year2019                        6.7804     1.5173 77.7495   4.469 0.0000264 ***
## Year2020                        1.6698     1.8196 75.5445   0.918  0.361715
## Treatmentcontrol:cropsoybean   -3.6846     1.4979  4.2531  -2.460  0.065965 .
## Treatmentcontrol:Year2018      -0.2023     1.5811 41.9809  -0.128  0.898794
## Treatmentcontrol:Year2019      -3.5865     1.8726 41.8062  -1.915  0.062310 .
## Treatmentcontrol:Year2020      -1.9933     2.2525 40.6086  -0.885  0.381399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation of Fixed Effects:
##             (Intr) Trtmnt ln_ppt crpsyb Yr2018 Yr2019 Yr2020 Trtmn: T:Y2018 T:Y2019
## Trtmntcntrl -0.441
## ln_ppt       0.698  0.000
## cropsoybean -0.474  0.440  0.055
## Year2018    -0.614  0.508 -0.049  0.458
## Year2019    -0.525  0.525  0.106  0.594  0.703
## Year2020    -0.433  0.363 -0.030  0.318  0.515  0.508
## Trtmntcntr:  0.322 -0.730  0.000 -0.603 -0.298 -0.380 -0.214
## Trtmn:Y2018  0.361 -0.820  0.000 -0.290 -0.620 -0.440 -0.318  0.481
## Trtmn:Y2019  0.375 -0.851  0.000 -0.372 -0.441 -0.617 -0.318  0.616  0.712
## Trtmn:Y2020  0.258 -0.587  0.000 -0.208 -0.318 -0.317 -0.619  0.345  0.513   0.513
```

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

```
trt_yrR = emmeans(mR_flume, pairwise ~ Treatment|Year,
                  type = "response",
                  lmer.df = "asymptotic")
confint(trt_yrR)$contrasts
```

```
## Year = 2016:
##  contrast         ratio     SE  df asymp.LCL asymp.UCL
##  strips / control 0.320  0.376 Inf    0.0319      3.21
##
## Year = 2018:
##  contrast         ratio     SE  df asymp.LCL asymp.UCL
##  strips / control 0.423  0.386 Inf    0.0708      2.53
##
## Year = 2019:
##  contrast         ratio     SE  df asymp.LCL asymp.UCL
##  strips / control 8.889 10.928 Inf    0.7987     98.92
##
## Year = 2020:
##  contrast         ratio     SE  df asymp.LCL asymp.UCL
##  strips / control 2.481  4.544 Inf    0.0685     89.83
##
## Results are averaged over the levels of: crop
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
```

```
trtR    = emmeans(mR_flume, pairwise ~ Treatment,
                  type = "response",
                  lmer.df = "asymptotic")
```

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

```
confint(trtR)
```

```
## $emmeans
```

```
##  Treatment response    SE  df asymp.LCL asymp.UCL
##  strips       0.879 0.498 Inf     0.289      2.66
##  control      0.669 0.379 Inf     0.220      2.03
##
## Results are averaged over the levels of: crop, Year
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log(mu + 0.002) scale
##
## $contrasts
##  contrast         ratio    SE  df asymp.LCL asymp.UCL
##  strips / control  1.31 0.864 Inf     0.362      4.77
##
## Results are averaged over the levels of: crop, Year
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
```

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

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

```
confint(yearR)
```

```
##  Year response      SE  df asymp.LCL asymp.UCL
##  2016    0.106  0.0828 Inf    0.0222     0.485
##  2018    0.977  0.5963 Inf    0.2946     3.229
##  2019   15.743 12.5976 Inf    3.2801    75.541
##  2020    0.209  0.2472 Inf    0.0195     2.099
##
## Results are averaged over the levels of: Treatment, crop
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log(mu + 0.002) scale
```

```
trt_pptR  = emmeans(mR_flume, pairwise ~ Treatment|ln_ppt,
                  at=list(ln_ppt=c(-4,-3,-2,-1,-0.25)),
                  type = "response",
                  lmer.df = "asymptotic")

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

```
## ln_ppt = -4:
##  contrast         ratio    SE  df asymp.LCL asymp.UCL
##  strips / control 2.322 2.047 Inf   0.41274     13.07
##
## ln_ppt = -3:
##  contrast         ratio    SE  df asymp.LCL asymp.UCL
##  strips / control 1.216 0.806 Inf   0.33119      4.46
##
```

```
## ln_ppt = -2:
##  contrast         ratio    SE  df asymp.LCL asymp.UCL
##  strips / control 0.636 0.639 Inf   0.08889      4.55
##
## ln_ppt = -1:
##  contrast         ratio    SE  df asymp.LCL asymp.UCL
##  strips / control 0.333 0.525 Inf   0.01519      7.30
##
## ln_ppt = -0.25:
##  contrast         ratio    SE  df asymp.LCL asymp.UCL
##  strips / control 0.205 0.419 Inf   0.00373     11.27
##
## Results are averaged over the levels of: crop, Year
## Degrees-of-freedom method: asymptotic
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
```

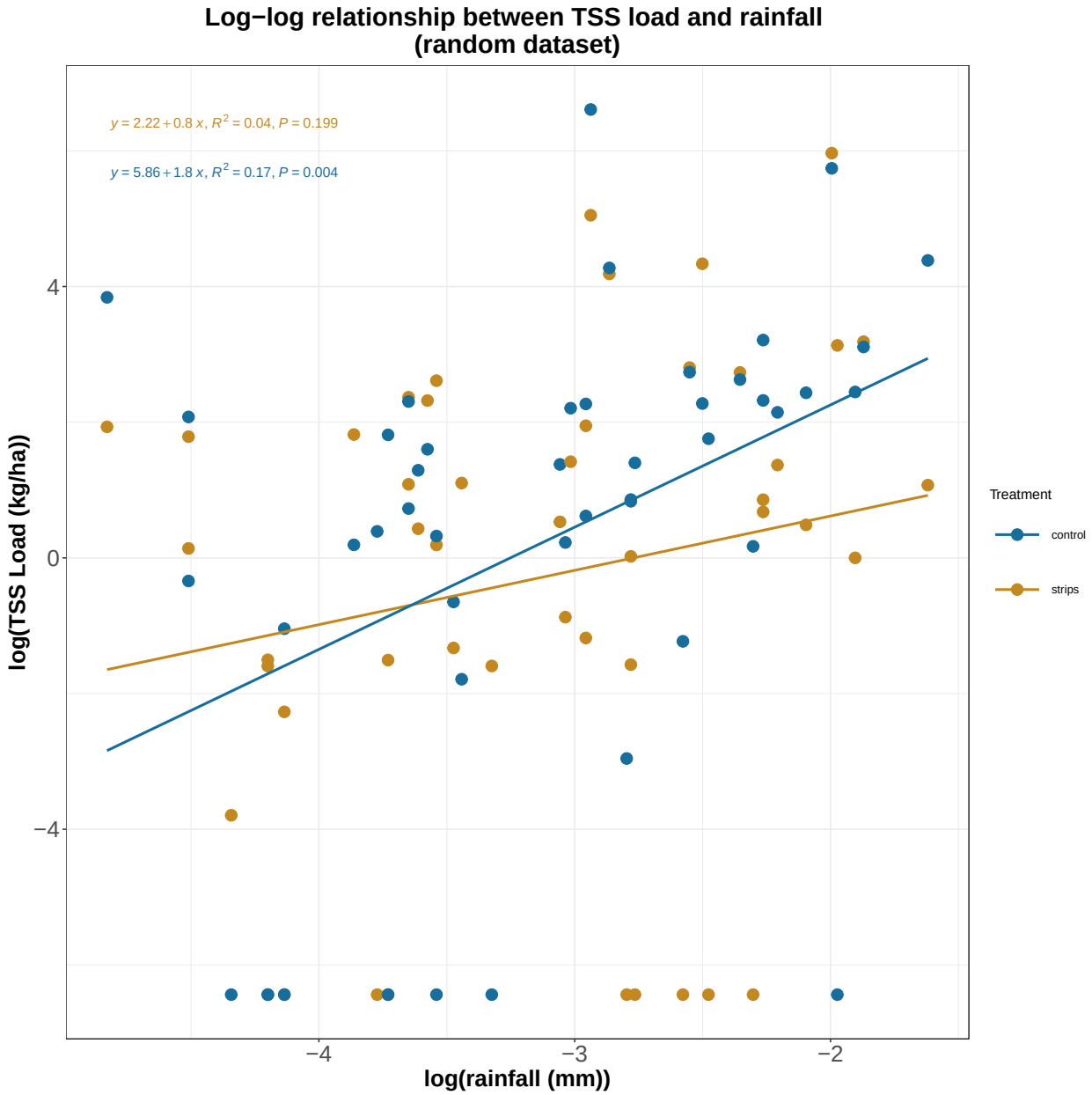
```
#cropR      = emmeans(mR_flume, pairwise ~ Treatment|crop,
#                     type = "response",
#                     lmer.df = "asymptotic")

#confint(cropR)
```

```
h <- ggplot(flumeR, aes(x=log(ppt_sum), y=log(tss_load+0.0016), color=Treatment), inherit.aes = FALSE) +
 scale_color_manual(values = c("control" = "#176D9C",
                               "strips" = "#C38820")) +
 geom_point(size=4) +
 xlab("log(rainfall (mm))") +
 ylab("log(TSS Load (kg/ha))") +
 geom_smooth(method=lm, se=FALSE, fullrange=TRUE) +
 ggtitle("Log-log relationship between TSS load and rainfall \n(random dataset)") +
 theme(plot.title = element_text(size=14, face="bold",hjust = 0.5),
       legend.key.size = unit(3,"line")) +
 theme(plot.title = element_text(size=20, face="bold", hjust=0.5),
       axis.title.x = element_text(size=18, face="bold"),
       axis.title.y = element_text(size=18, face="bold"),
       axis.text.x = element_text(size=18),
       axis.text.y = element_text(size=18)) +
 #xlim(-5, -1.5) +
 stat_poly_eq(formula = y ~ x,
              aes(label = paste(..eq.label.., ..rr.label.., ..p.value.label.., sep = "*`,`~")),
              parse = TRUE,
              label.x.npc = "left",
              label.y.npc = "top",
              vstep = 0.05) #+
 #scale_y_log10()

h
```

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




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

Check assumptions

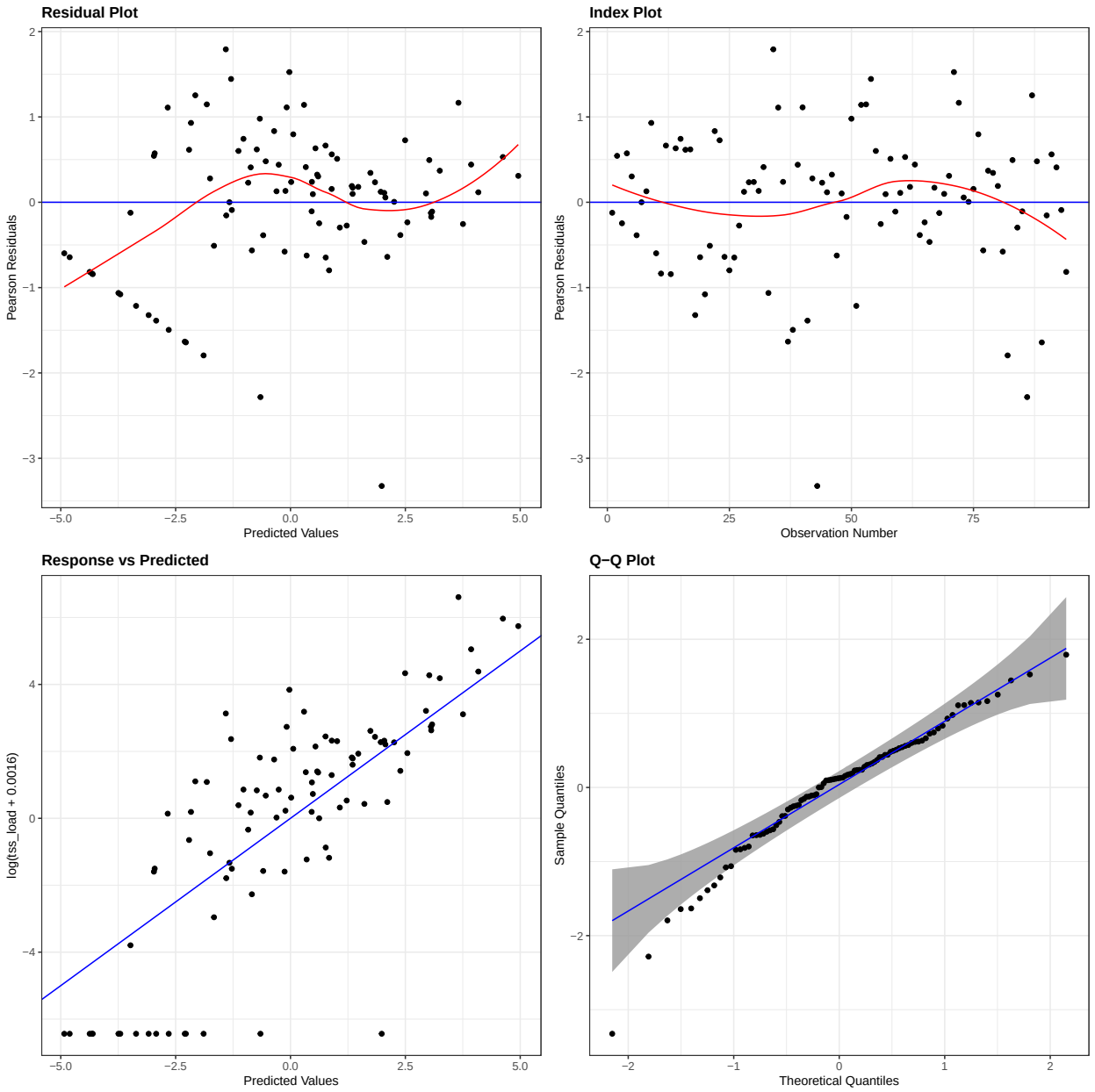
There are two possible models:

- mR_flume: full model design, design-based analysis
- mR_flume_model: model design selected based on backward step selection

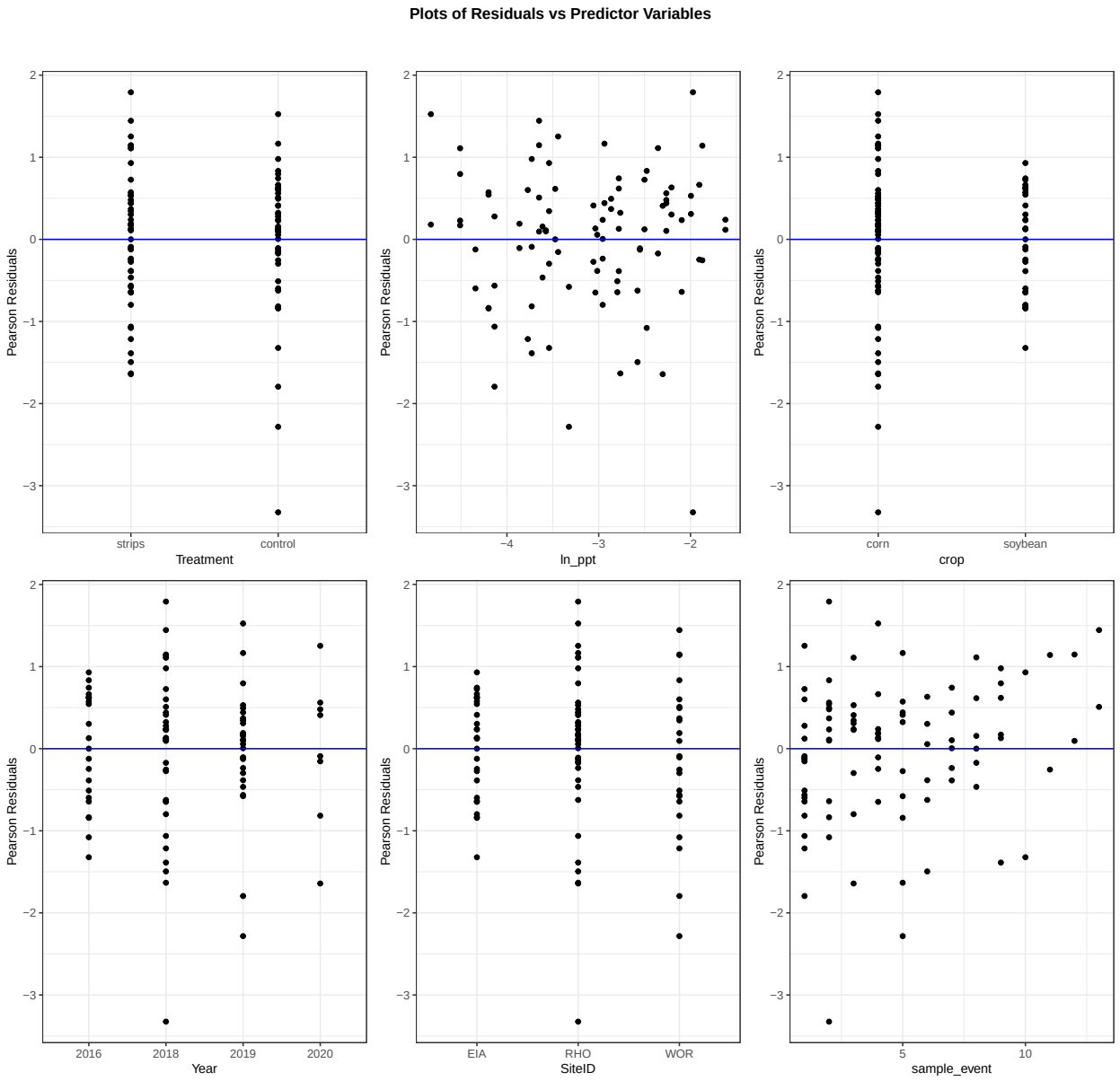
Full model design

```
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'
```

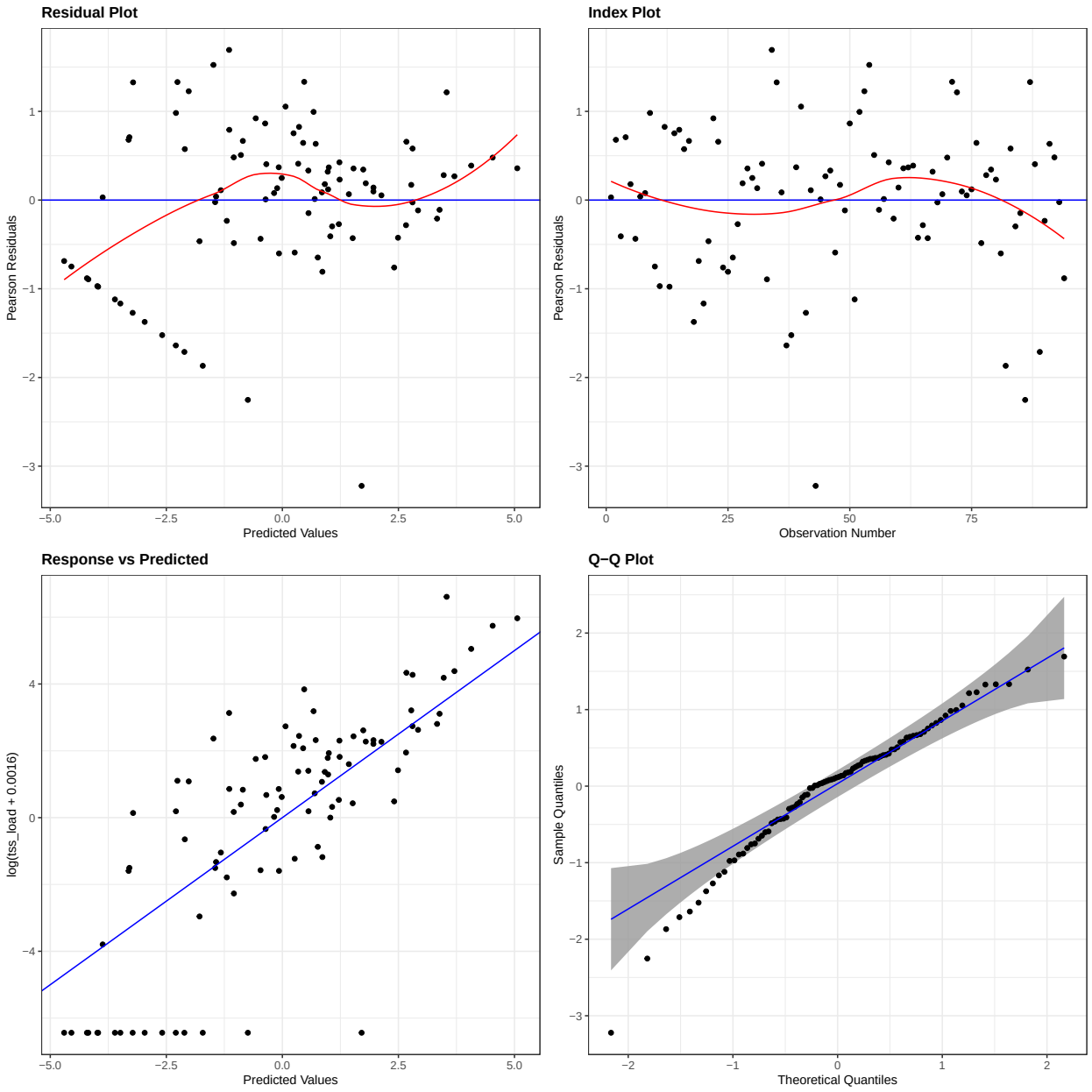



```
resid_xpanel(mR_flume)
```

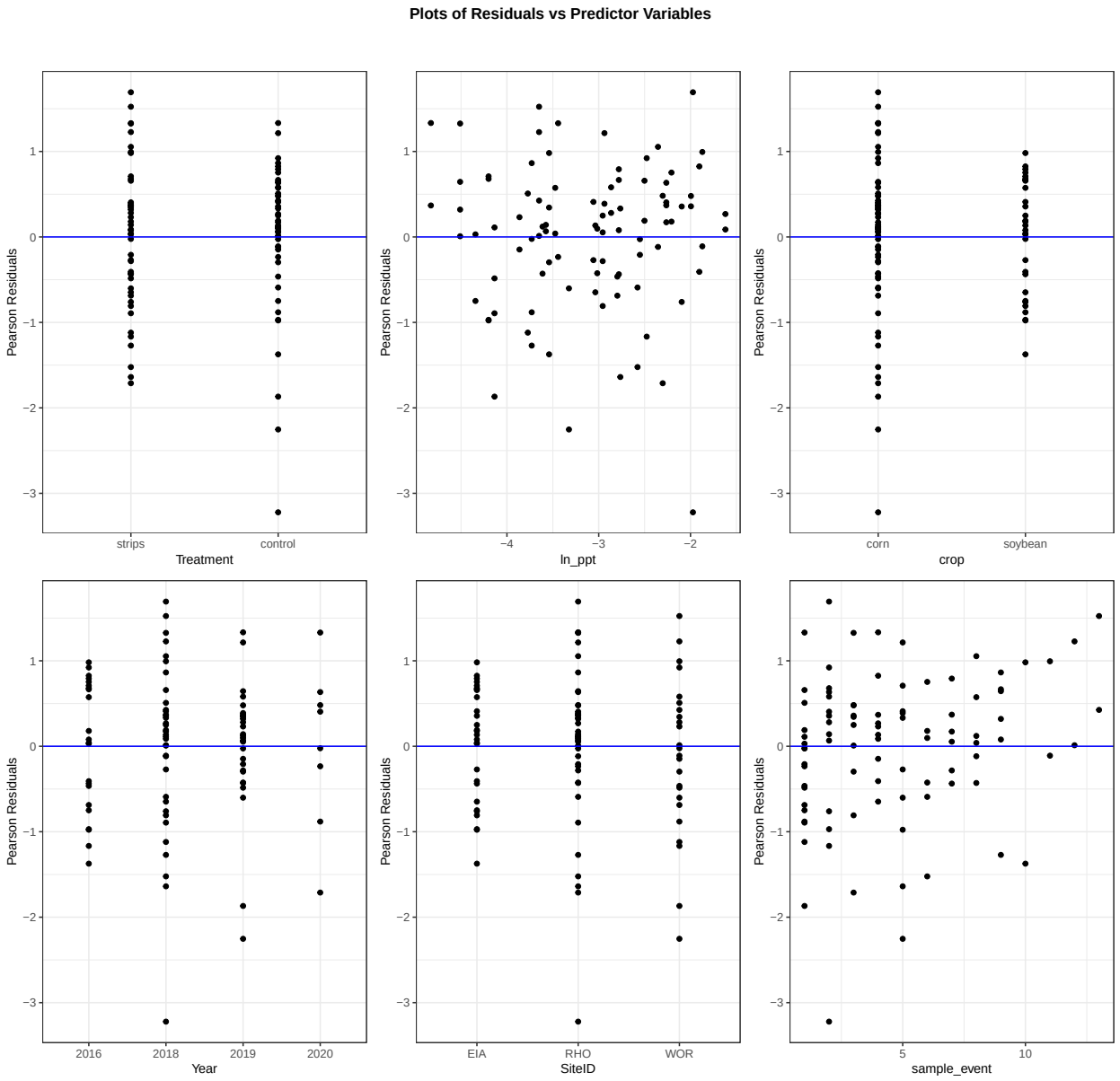



```
resid_panel(mR_flume_model,
            plots = c("resid","index","yvp","qq"),
            smoother = TRUE, qqbands = TRUE)
```

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




```
resid_xpanel(mR_flume_model)
```



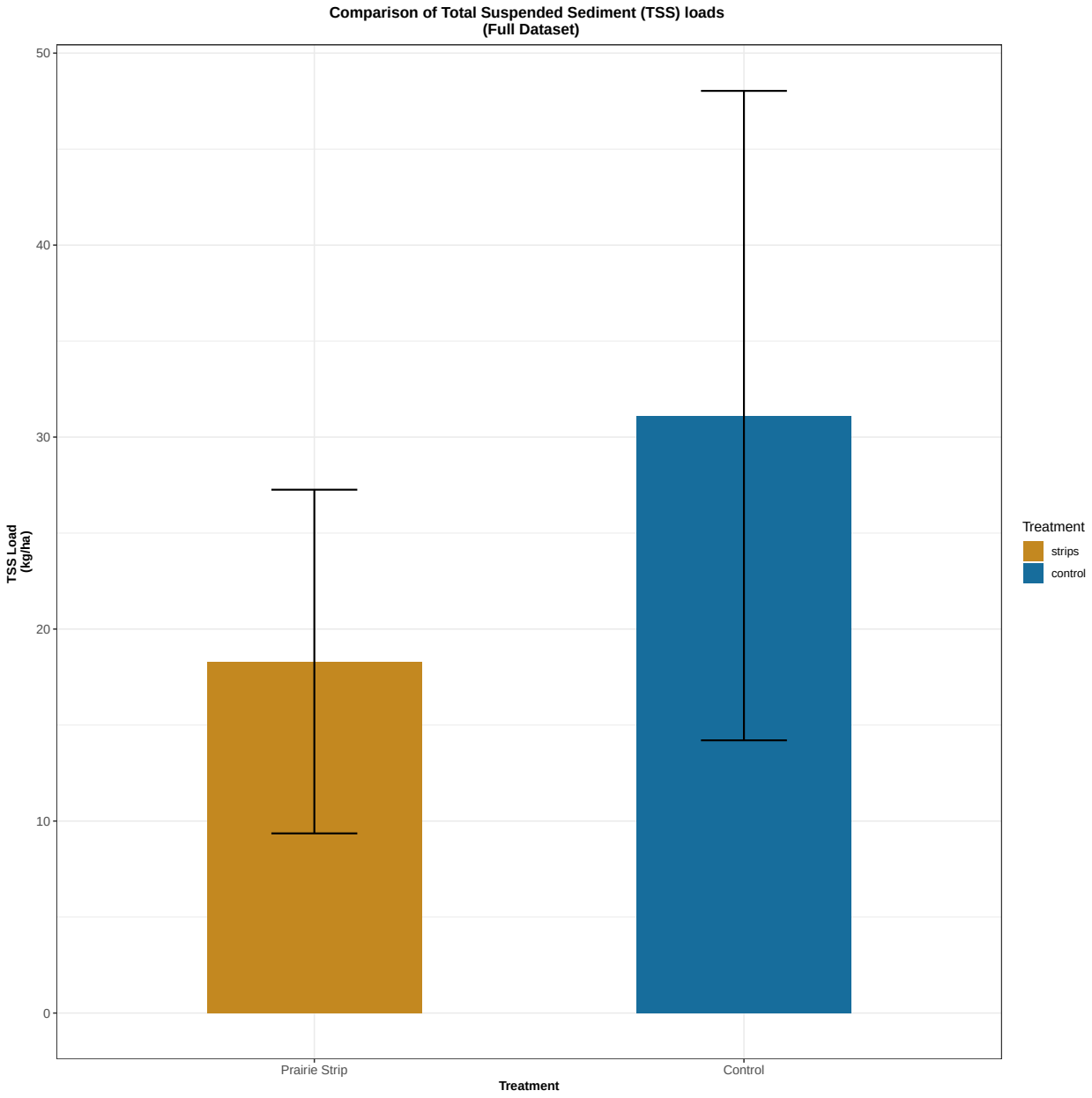

```
groupR <- flumeR %>%
  group_by(Treatment) %>%
  summarize(n = n(),
            mean = mean(tss_load),
            sd = sd(tss_load),
            .groups = "drop") %>%
  mutate(se = sd / sqrt(n),
         lb = mean + qt(0.025, df = n-1)*se,
         ub = mean - qt(0.025, df = n-1)*se)

trt_plotR <- groupR %>%
  ggplot(aes(x=Treatment, y=mean, fill=Treatment))+
  geom_bar(width = 0.5, position = position_dodge(), stat="summary") +
  geom_errorbar(aes(ymin = (mean-se), ymax = (mean+se)),
```

```
                width = 0.2,
                linetype = "solid",
                position = position_dodge(width = 0.5),
                color="black", size=0.7) +
  scale_fill_manual(values = c("strips" = "#C38820", "control" = "#176D9C")) +
  ggtitle("Comparison of Total Suspended Sediment (TSS) loads \n (Full Dataset)") +
  xlab("Treatment") +
  ylab("TSS Load \n(kg/ha)") +
  theme(plot.title = element_text(size=12, face="bold", hjust=0.5),
        axis.title.x = element_text(size=10, face="bold"),
        axis.title.y = element_text(size=10, face="bold"),
        axis.text.x = element_text(size=10),
        axis.text.y = element_text(size=10)) +
  scale_x_discrete(labels= c("Prairie Strip", "Control"))

trt_plotR
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
#ggsave("E:/ISU/Project/SoilMove/data/statistics/flume_analysis/code/fig/flume_plot.png", trt_plot)
```