Canopy Reflectance Measurements from Aerial and Proximal Sensors Improve the Precision of Mid-Season N Fertilizer Management in Direct-Seeded Rice

Telha H. Rehman

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

START

```
Sys.time()
```

```
## [1] "2023-05-23 15:34:49 PDT"
```

```
library(tinytex)
library(tidyverse)
library(dplyr)
library(cowplot)
library(Cairo)
library(modelr)
library(gridExtra)
library(nlme)
library(car)
library(emmeans)
library(MuMIn)
library(ggpmisc)
library(gtable)
library(grid)
library(segmented)
library(scales)
library(sm)
library(rcompanion)
library(nlstools)
library(magick)
library(multcomp)
library(wesanderson)
```

DATA

GreenSeeker NDVI Data

```
gs_ndvi_data <- read_csv(file = "DATA/PI_greenseeker_data.csv")
```

```
## Rows: 327 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): site_year, NDVI_1, NDVI_2, NDVI_3, NDVI_4
## dbl (12): year, exp_plot_number, block, plot, N_level, N_level_kgha, biomass...
##
## 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.
```

```
str(gs_ndvi_data , give.attr = FALSE)
```

```
## spc_tbl_ [327 x 17] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ site_year         : chr [1:327] "Arbuckle-15" "Arbuckle-15" "Arbuckle-15" "Arbuckle-15" ...
##  $ year              : num [1:327] 2015 2015 2015 2015 2015 ...
##  $ exp_plot_number   : num [1:327] 101 102 103 104 105 201 202 203 204 205 ...
##  $ block             : num [1:327] 1 1 1 1 1 2 2 2 2 2 ...
##  $ plot              : num [1:327] 1 2 3 4 5 6 7 8 9 10 ...
##  $ N_level           : num [1:327] 125 225 0 75 175 0 175 125 225 75 ...
##  $ N_level_kgha      : num [1:327] 125 225 0 75 175 0 175 125 225 75 ...
##  $ biomass_plus_bag_g: num [1:327] 414 472 281 386 455 304 402 322 418 336 ...
##  $ paper_bag_g       : num [1:327] 45 45 45 45 45 45 45 45 45 45 ...
##  $ ring_size         : num [1:327] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ num_of_paper_bags : num [1:327] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sample_weight_mg  : num [1:327] 4.84 5.12 4.78 5.15 4.93 ...
##  $ sample_N_ug       : num [1:327] 117.1 153.4 64.9 92.9 116 ...
##  $ NDVI_1            : chr [1:327] "0.77" "0.82" "0.56" "0.72" ...
##  $ NDVI_2            : chr [1:327] "n/a" "n/a" "n/a" "n/a" ...
##  $ NDVI_3            : chr [1:327] "n/a" "n/a" "n/a" "n/a" ...
##  $ NDVI_4            : chr [1:327] "n/a" "n/a" "n/a" "n/a" ...
```

```
gs_ndvi_data <- gs_ndvi_data %>%
  filter(!year %in% c("2015" , "2016" , "2018") & N_level_kgha != 275) #remove the years we don't need

str(gs_ndvi_data , give.attr = FALSE)
```

```
## spc_tbl_ [144 x 17] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ site_year         : chr [1:144] "Nicolaus-17" "Nicolaus-17" "Nicolaus-17" "Nicolaus-17" ...
##  $ year              : num [1:144] 2017 2017 2017 2017 2017 ...
##  $ exp_plot_number   : num [1:144] 101 102 103 104 106 107 202 203 204 205 ...
##  $ block             : num [1:144] 21 21 21 21 21 21 22 22 22 22 ...
##  $ plot              : num [1:144] 109 110 111 112 114 115 117 118 119 120 ...
##  $ N_level           : num [1:144] 225 0 125 175 45 75 75 125 0 175 ...
##  $ N_level_kgha      : num [1:144] 225 0 125 175 45 75 75 125 0 175 ...
##  $ biomass_plus_bag_g: num [1:144] 361 264 318 360 285 ...
##  $ paper_bag_g       : num [1:144] 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 ...
##  $ ring_size         : num [1:144] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ num_of_paper_bags : num [1:144] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sample_weight_mg  : num [1:144] 3.17 3.42 3.09 3.07 3.2 ...
##  $ sample_N_ug       : num [1:144] 79.4 61.1 80.7 95.9 63.6 ...
##  $ NDVI_1            : chr [1:144] "0.77" "0.50" "0.67" "0.79" ...
##  $ NDVI_2            : chr [1:144] "0.77" "0.52" "0.67" "0.78" ...
##  $ NDVI_3            : chr [1:144] "0.79" "0.46" "0.71" "0.79" ...
```

```
##  $ NDVI_4            : chr [1:144] "n/a" "n/a" "n/a" "n/a" ...
```

```
gs_ndvi_data <- gs_ndvi_data[c(1:144), c(1:17)] #removes the empty rows and columns from the data frame

gs_ndvi_data$block <- factor(gs_ndvi_data$block)
gs_ndvi_data$year <- factor(gs_ndvi_data$year)
gs_ndvi_data$plot <- factor(gs_ndvi_data$plot)
gs_ndvi_data$N_level_kgha_f <- factor(gs_ndvi_data$N_level_kgha)
gs_ndvi_data$exp_plot_number <- factor(gs_ndvi_data$exp_plot_number)
gs_ndvi_data$site_year <- factor(gs_ndvi_data$site_year , levels = c( "Nicolaus-17" , "Williams-17" , "
gs_ndvi_data$NDVI_1 <- as.numeric(as.character(gs_ndvi_data$NDVI_1))
gs_ndvi_data$NDVI_2 <- as.numeric(as.character(gs_ndvi_data$NDVI_2))
```

```
## Warning: NAs introduced by coercion
```

```
gs_ndvi_data$NDVI_3 <- as.numeric(as.character(gs_ndvi_data$NDVI_3))
gs_ndvi_data$NDVI_4 <- as.numeric(as.character(gs_ndvi_data$NDVI_4)) #gets the data right
```

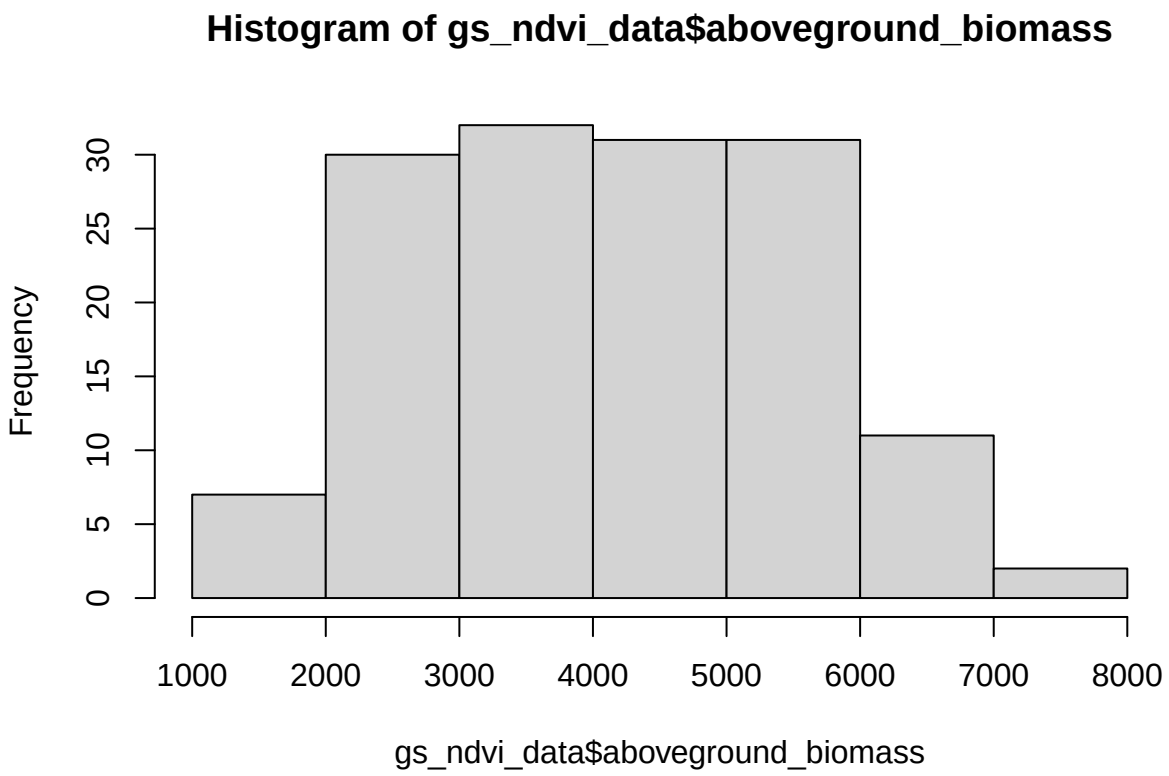
```
## Warning: NAs introduced by coercion
```

```
str(gs_ndvi_data , give.attr = FALSE)
```

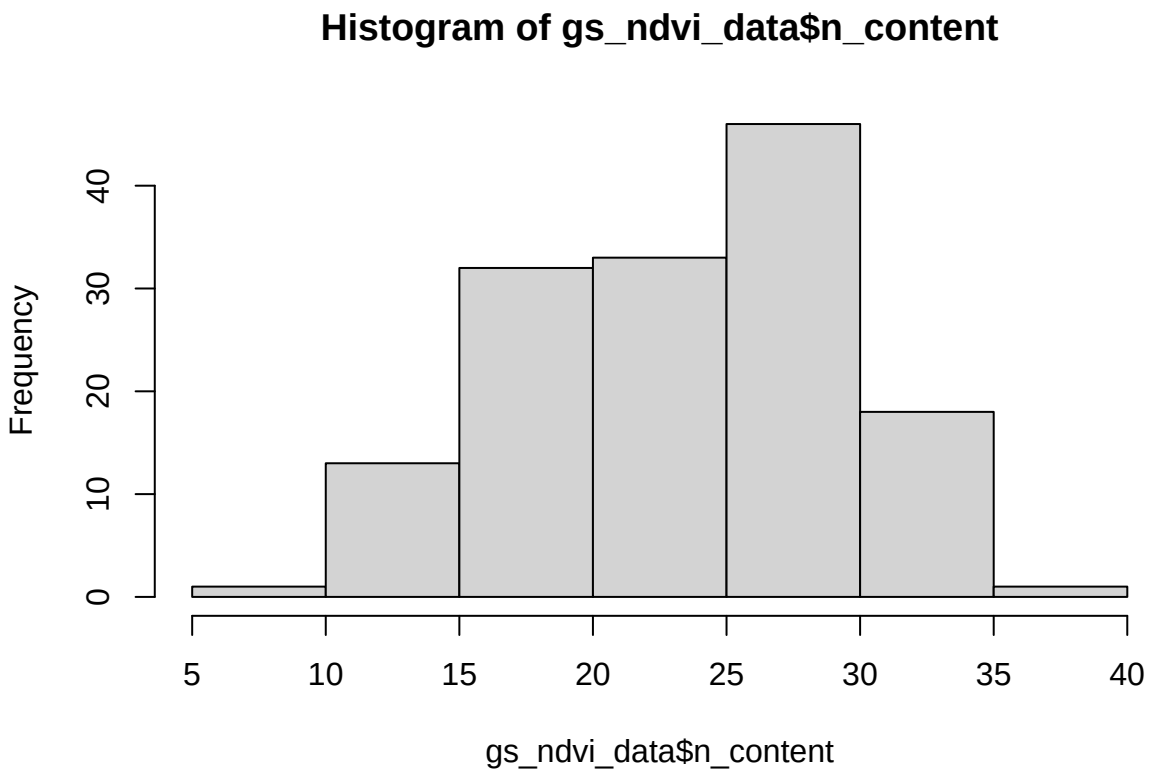
```
## tibble [144 x 18] (S3: tbl_df/tbl/data.frame)
##  $ site_year         : Factor w/ 6 levels "Nicolaus-17",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year              : Factor w/ 2 levels "2017","2019": 1 1 1 1 1 1 1 1 1 1 ...
##  $ exp_plot_number   : Factor w/ 27 levels "101","102","103",..: 1 2 3 4 6 7 9 10 11 12 ...
##  $ block             : Factor w/ 24 levels "17","18","19",..: 5 5 5 5 5 5 6 6 6 6 ...
##  $ plot              : Factor w/ 144 levels "81","82","83",..: 25 26 27 28 29 30 31 32 33 34 ...
##  $ N_level           : num [1:144] 225 0 125 175 45 75 75 125 0 175 ...
##  $ N_level_kgha      : num [1:144] 225 0 125 175 45 75 75 125 0 175 ...
##  $ biomass_plus_bag_g: num [1:144] 361 264 318 360 285 ...
##  $ paper_bag_g       : num [1:144] 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 ...
##  $ ring_size         : num [1:144] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ num_of_paper_bags : num [1:144] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sample_weight_mg  : num [1:144] 3.17 3.42 3.09 3.07 3.2 ...
##  $ sample_N_ug       : num [1:144] 79.4 61.1 80.7 95.9 63.6 ...
##  $ NDVI_1            : num [1:144] 0.77 0.5 0.67 0.79 0.68 0.63 0.66 0.68 0.54 0.77 ...
##  $ NDVI_2            : num [1:144] 0.77 0.52 0.67 0.78 0.69 0.56 0.65 0.68 0.5 0.75 ...
##  $ NDVI_3            : num [1:144] 0.79 0.46 0.71 0.79 0.68 0.53 0.63 0.64 0.5 0.77 ...
##  $ NDVI_4            : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ N_level_kgha_f    : Factor w/ 11 levels "0","45","75",..: 10 1 5 8 2 3 3 5 1 8 ...
```

```
gs_ndvi_data <-  gs_ndvi_data %>%
  mutate( biomass_dry_wt = biomass_plus_bag_g - (paper_bag_g * num_of_paper_bags) ,
          aboveground_biomass = (biomass_dry_wt / ring_size) * 10 ,
          n_content = sample_N_ug / sample_weight_mg ,
          PI_N_Uptake = (aboveground_biomass * n_content) / 1000 #n uptake in kg per ha
          )#processes the data


gs_ndvi_data <- gs_ndvi_data %>%
  rowwise() %>%
  mutate(gs_NDVI = mean(c( NDVI_1 , NDVI_2 , NDVI_3 , NDVI_4) , na.rm = T)) #takes average of four NDVI

gs_ndvi_data <- dplyr::select(gs_ndvi_data ,
                      site_year,
```

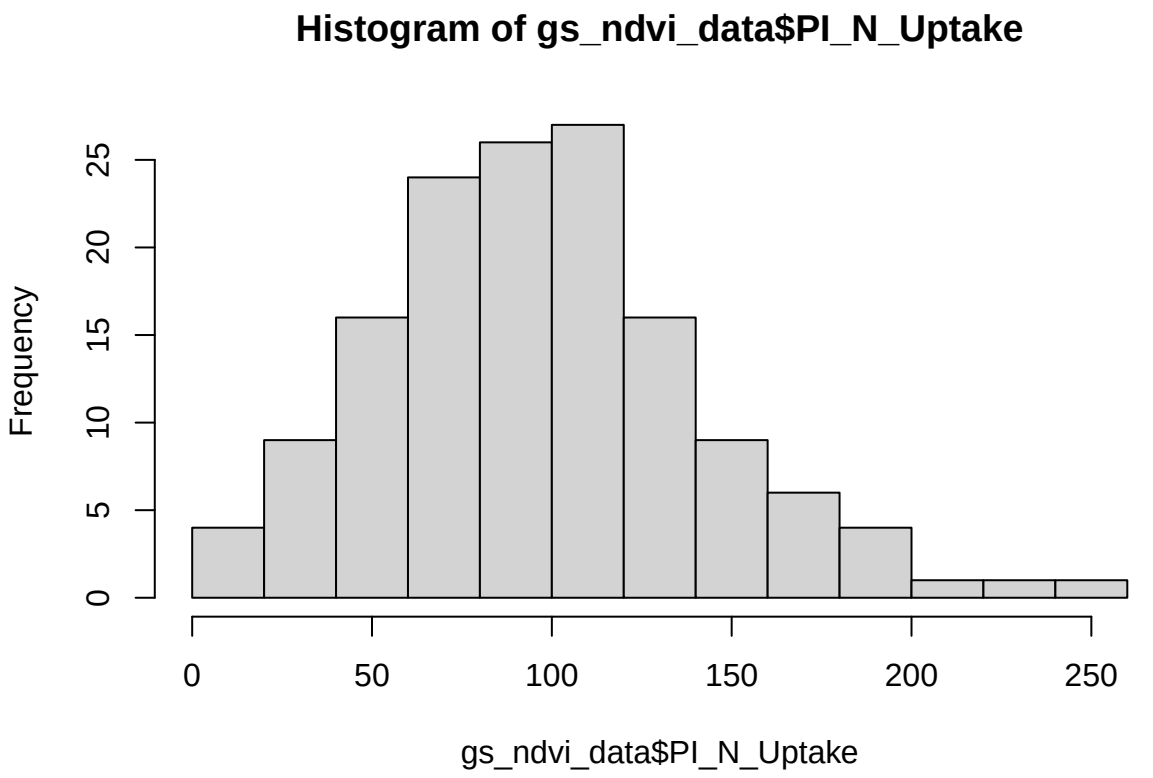
```
                      year,
                      exp_plot_number,
                      N_level_kgha,
                      aboveground_biomass,
                      n_content,
                      PI_N_Uptake,
                      gs_NDVI) #selects the relevant columns

gs_ndvi_data$site_year <- factor(gs_ndvi_data$site_year , levels = c( "Nicolaus-17" , "Williams-17" , "

hist(gs_ndvi_data$aboveground_biomass)
```



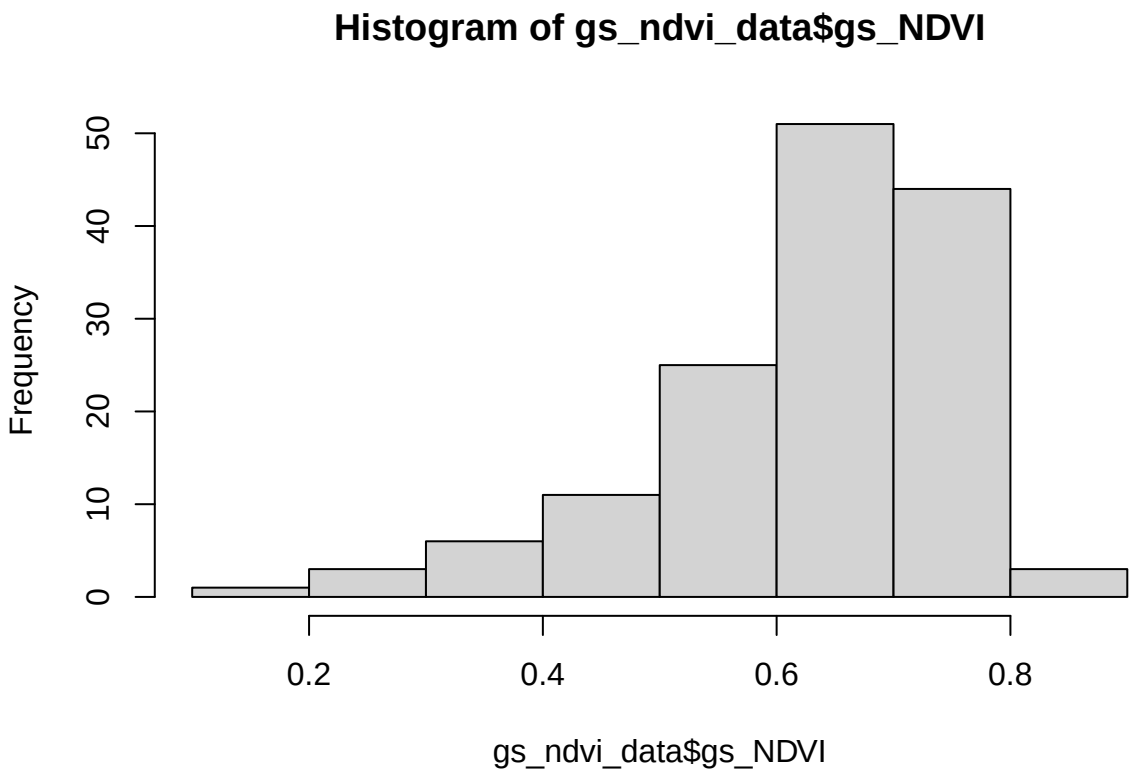

```
hist(gs_ndvi_data$n_content)
```




```
hist(gs_ndvi_data$PI_N_Uptake)
```




```
hist(gs_ndvi_data$gs_NDVI)
```



Drone Data

```
drone_data <- read_csv(file = "DATA/PI_drone_data.csv")
```

```
## Rows: 248 Columns: 38
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): site_year
## dbl (37): year, exp_plot_number, Block, MainPlot, N_level, N_level_kgha, bio...
##
## 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.
```

```
str(drone_data , give.attr = FALSE)
```

```
## spc_tbl_ [248 x 38] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ site_year         : chr [1:248] "Nicolaus-17" "Nicolaus-17" "Nicolaus-17" "Nicolaus-17" ...
##  $ year              : num [1:248] 2017 2017 2017 2017 2017 ...
##  $ exp_plot_number   : num [1:248] 101 102 103 104 105 106 107 201 202 203 ...
##  $ Block             : num [1:248] 1 1 1 1 1 1 1 2 2 2 ...
##  $ MainPlot          : num [1:248] 1 2 3 4 5 6 7 1 2 3 ...
##  $ N_level           : num [1:248] 225 0 125 175 275 45 75 275 75 125 ...
##  $ N_level_kgha      : num [1:248] 225 0 125 175 275 45 75 275 75 125 ...
##  $ biomass_plus_bag_g: num [1:248] 361 264 318 360 394 ...
##  $ ring_size         : num [1:248] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ paper_bag_g       : num [1:248] 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 ...
##  $ num_of_paper_bags : num [1:248] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sample_weight_mg  : num [1:248] 3.17 3.42 3.09 3.07 3.35 ...
##  $ sample_N_ug       : num [1:248] 79.4 61.1 80.7 95.9 111.3 ...
##  $ greenmean         : num [1:248] 0.0467 0.0581 0.0498 0.0488 0.0505 0.0608 0.0589 0.0461 0.0537 0.0
```

```
##  $ greenmedia        : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ greenstdev        : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ greenmin          : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ greenmax          : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemean          : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemedian        : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluestdev         : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemin           : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemax           : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ redmean           : num [1:248] 0.023 0.0286 0.0243 0.0237 0.0245 0.0292 0.0281 0.0234 0.0263 0.0
##  $ redmedian         : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ redstdev          : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ redmin            : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ redmax            : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgemean          : num [1:248] 0.0738 0.0867 0.0768 0.0764 0.0791 0.0903 0.0877 0.0744 0.0812 0.0
##  $ edgemedian        : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgestdev         : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgemin           : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgemax           : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirmean           : num [1:248] 0.55 0.357 0.482 0.537 0.599 ...
##  $ nirmedian         : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirstdev          : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirmin            : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirmax            : num [1:248] NA NA NA NA NA NA NA NA NA NA ...
```

```
drone_data <- drone_data %>%
  filter(N_level_kgha != 275 & year != 2018) %>%
  mutate(year = factor(year) ,
         exp_plot_number = factor(exp_plot_number) ,
         Block = factor(Block) ,
         MainPlot = factor(MainPlot) ,
         N_level = factor(N_level) ,
         N_level_kgha_f = factor(N_level_kgha)
  )

drone_data$site_year <- factor(drone_data$site_year , levels = c("Nicolaus-17" , "Williams-17" , "Arbuc

str(drone_data , give.attr = FALSE)
```

```
## tibble [144 x 39] (S3: tbl_df/tbl/data.frame)
##  $ site_year         : Factor w/ 6 levels "Nicolaus-17",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year              : Factor w/ 2 levels "2017","2019": 1 1 1 1 1 1 1 1 1 1 ...
##  $ exp_plot_number   : Factor w/ 27 levels "101","102","103",..: 1 2 3 4 6 7 9 10 11 12 ...
##  $ Block             : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...
##  $ MainPlot          : Factor w/ 7 levels "1","2","3","4",..: 1 2 3 4 6 7 2 3 4 5 ...
##  $ N_level           : Factor w/ 11 levels "0","45","75",..: 11 1 6 8 2 3 3 6 1 8 ...
##  $ N_level_kgha      : num [1:144] 225 0 125 175 45 75 75 125 0 175 ...
##  $ biomass_plus_bag_g: num [1:144] 361 264 318 360 285 ...
##  $ ring_size         : num [1:144] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ paper_bag_g       : num [1:144] 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 ...
##  $ num_of_paper_bags : num [1:144] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sample_weight_mg  : num [1:144] 3.17 3.42 3.09 3.07 3.2 ...
##  $ sample_N_ug       : num [1:144] 79.4 61.1 80.7 95.9 63.6 ...
##  $ greenmean         : num [1:144] 0.0467 0.0581 0.0498 0.0488 0.0608 0.0589 0.0537 0.0488 0.0598 0.0
##  $ greenmedia        : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
```

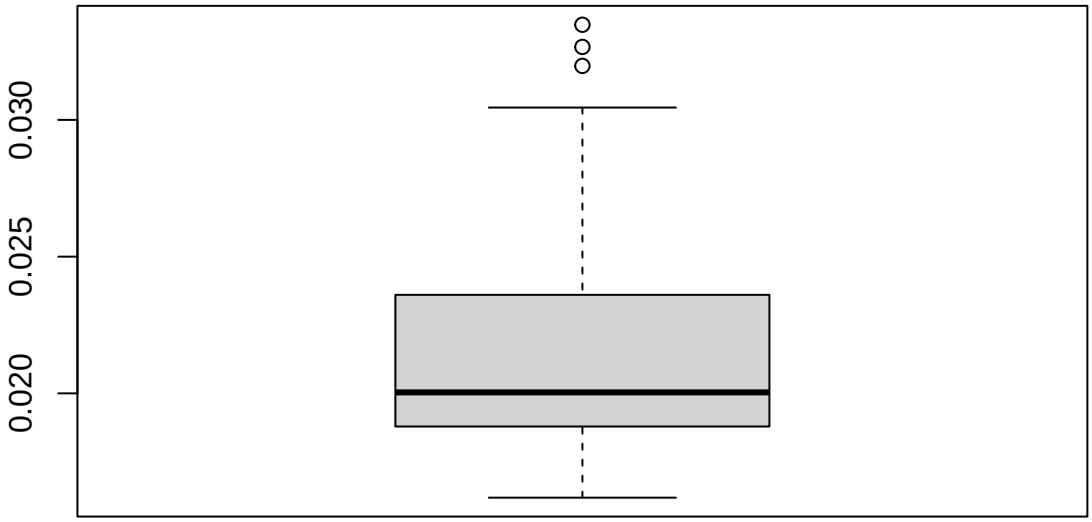
```
##  $ greenstdev        : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ greenmin          : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ greenmax          : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemean          : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemedian        : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluestdev         : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemin           : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ bluemax           : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ redmean           : num [1:144] 0.023 0.0286 0.0243 0.0237 0.0292 0.0281 0.0263 0.0237 0.0293 0.0
##  $ redmedian         : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ redstdev          : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ redmin            : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ redmax            : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgemean          : num [1:144] 0.0738 0.0867 0.0768 0.0764 0.0903 0.0877 0.0812 0.0754 0.0873 0.0
##  $ edgemedian        : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgestdev         : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgemin           : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ edgemax           : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirmean           : num [1:144] 0.55 0.357 0.482 0.537 0.431 ...
##  $ nirmedian         : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirstdev          : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirmin            : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ nirmax            : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ N_level_kgha_f    : Factor w/ 11 levels "0","45","75",..: 10 1 5 8 2 3 3 5 1 8 ...
```

```
drone_data <- dplyr::select(drone_data ,
                      site_year,
                      year,
                      exp_plot_number,
                      Block,
                      MainPlot,
                      N_level,
                      N_level_kgha,
                      N_level_kgha_f,
                      biomass_plus_bag_g,
                      ring_size,
                      paper_bag_g,
                      num_of_paper_bags,
                      sample_weight_mg,
                      sample_N_ug,
                      bluemean,
                      greenmean,
                      redmean,
                      edgemean,
                      nirmean
                      )#selects the relevant columns

str(drone_data , give.attr = FALSE)
```

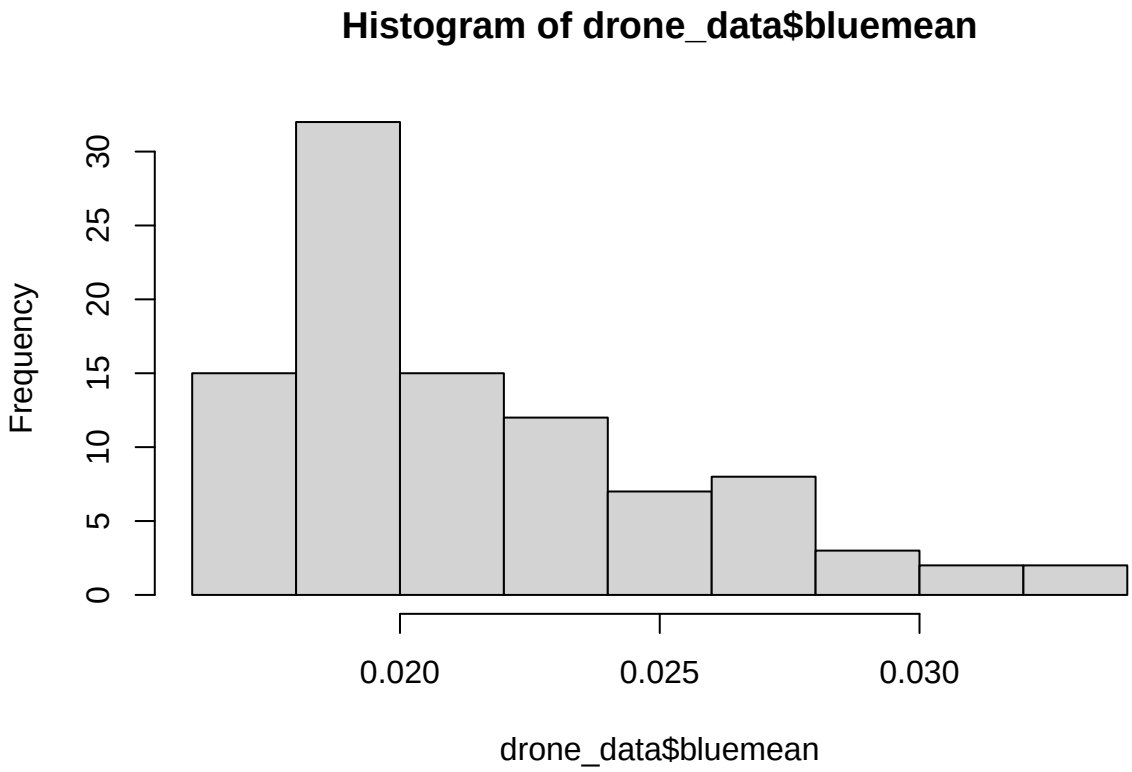
```
## tibble [144 x 19] (S3: tbl_df/tbl/data.frame)
##  $ site_year         : Factor w/ 6 levels "Nicolaus-17",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year              : Factor w/ 2 levels "2017","2019": 1 1 1 1 1 1 1 1 1 1 ...
##  $ exp_plot_number   : Factor w/ 27 levels "101","102","103",..: 1 2 3 4 6 7 9 10 11 12 ...
##  $ Block             : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...
##  $ MainPlot          : Factor w/ 7 levels "1","2","3","4",..: 1 2 3 4 6 7 2 3 4 5 ...
```

```
##  $ N_level           : Factor w/ 11 levels "0","45","75",..: 11 1 6 8 2 3 3 6 1 8 ...
##  $ N_level_kgha      : num [1:144] 225 0 125 175 45 75 75 125 0 175 ...
##  $ N_level_kgha_f    : Factor w/ 11 levels "0","45","75",..: 10 1 5 8 2 3 3 5 1 8 ...
##  $ biomass_plus_bag_g: num [1:144] 361 264 318 360 285 ...
##  $ ring_size         : num [1:144] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ paper_bag_g       : num [1:144] 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 45.5 ...
##  $ num_of_paper_bags : num [1:144] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sample_weight_mg  : num [1:144] 3.17 3.42 3.09 3.07 3.2 ...
##  $ sample_N_ug       : num [1:144] 79.4 61.1 80.7 95.9 63.6 ...
##  $ bluemean          : num [1:144] NA NA NA NA NA NA NA NA NA NA ...
##  $ greenmean         : num [1:144] 0.0467 0.0581 0.0498 0.0488 0.0608 0.0589 0.0537 0.0488 0.0598 0.0
##  $ redmean           : num [1:144] 0.023 0.0286 0.0243 0.0237 0.0292 0.0281 0.0263 0.0237 0.0293 0.02
##  $ edgemean          : num [1:144] 0.0738 0.0867 0.0768 0.0764 0.0903 0.0877 0.0812 0.0754 0.0873 0.0
##  $ nirmean           : num [1:144] 0.55 0.357 0.482 0.537 0.431 ...
```

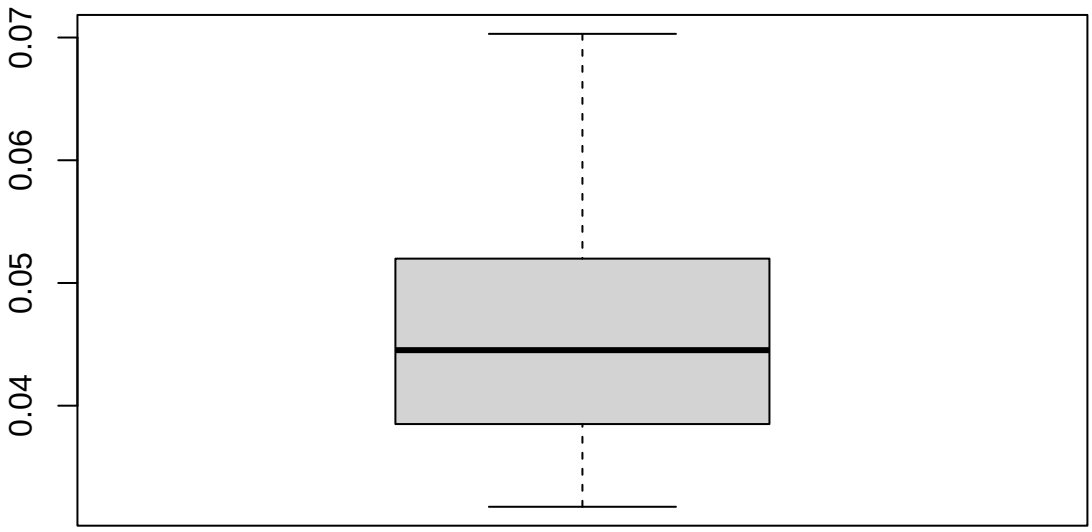
```
#visualize drone_data to look for outliers

boxplot(drone_data$bluemean)
```



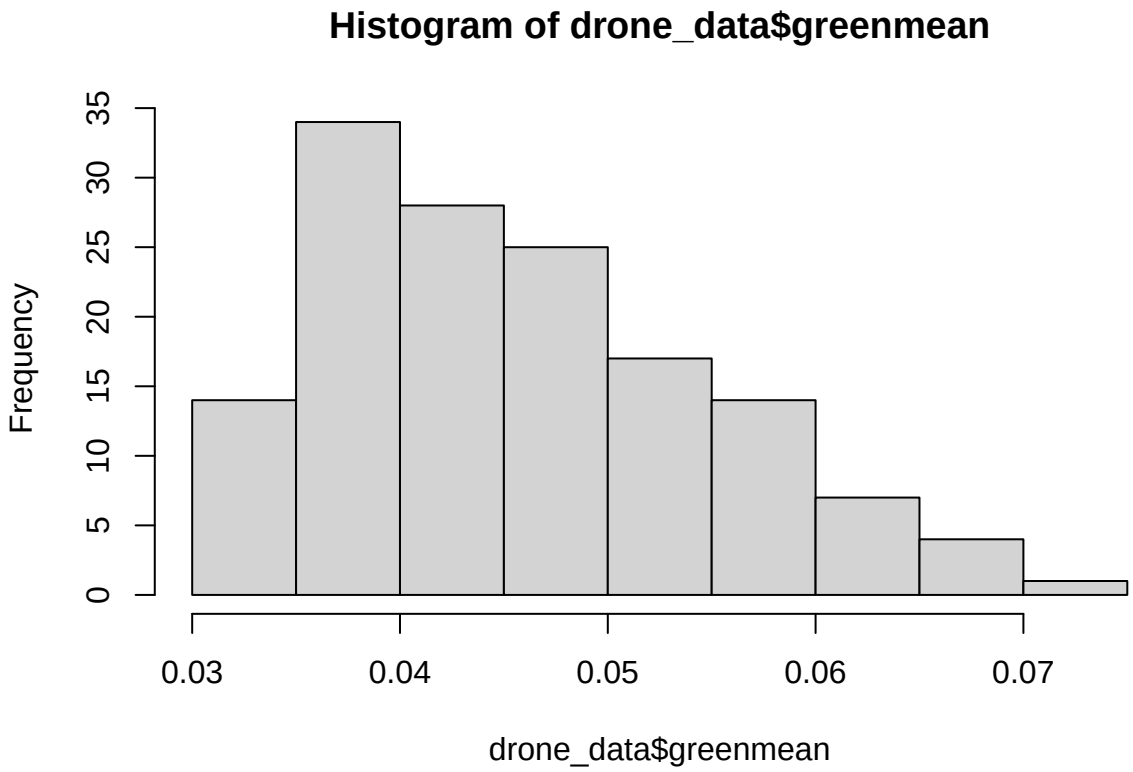

```
hist(drone_data$bluemean)
```



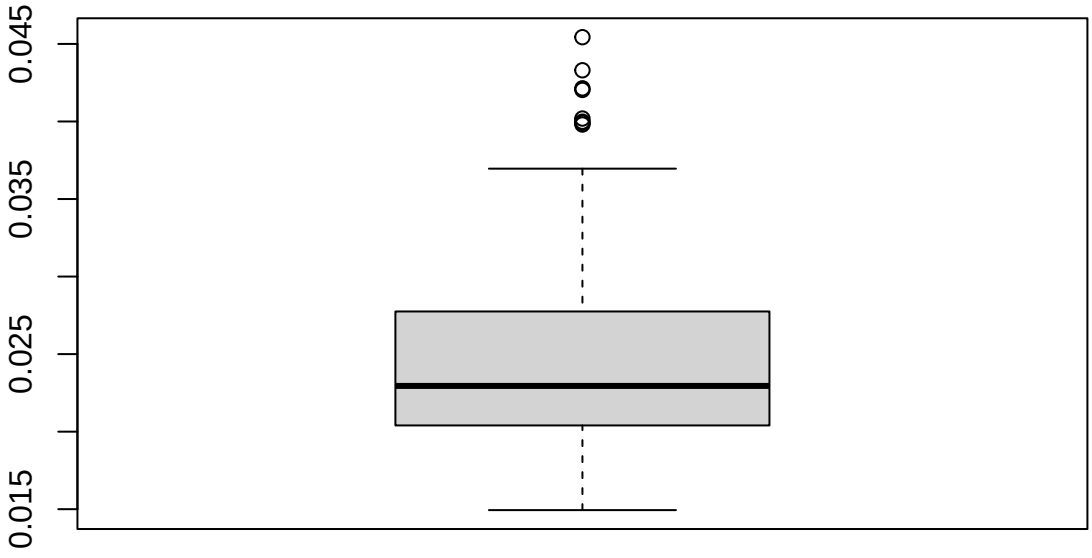

```
boxplot(drone_data$greenmean)
```



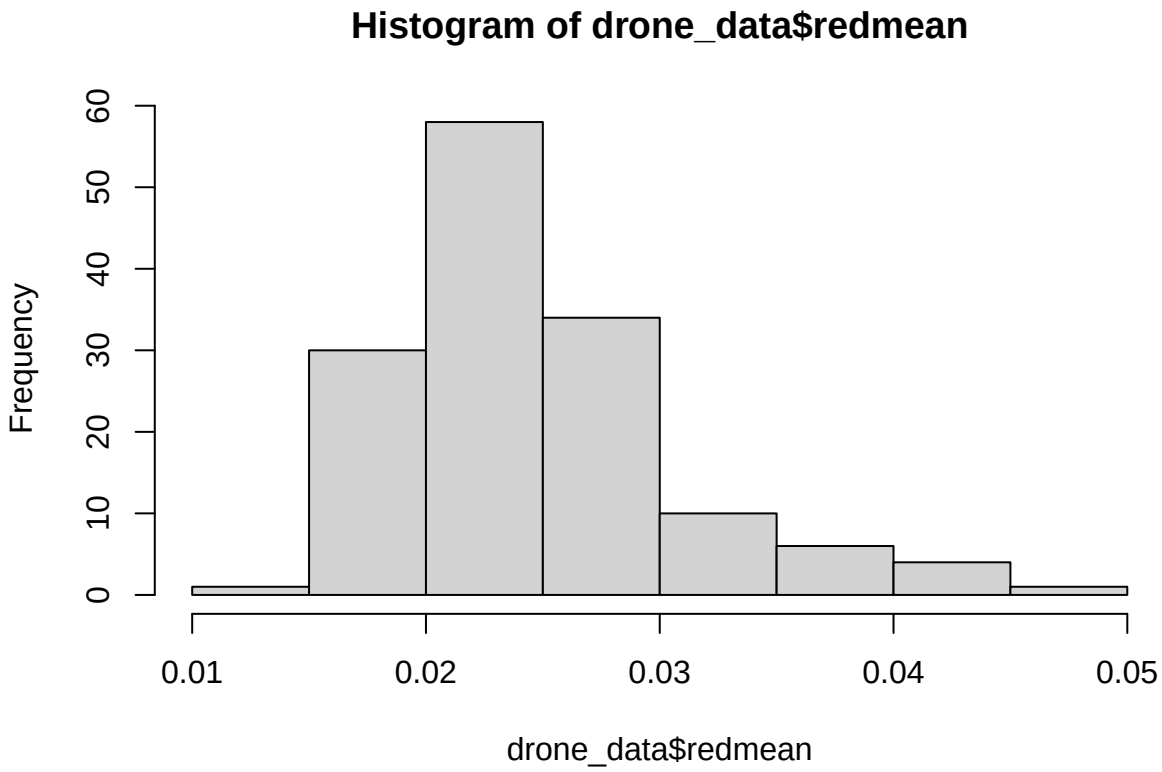

```
hist(drone_data$greenmean)
```



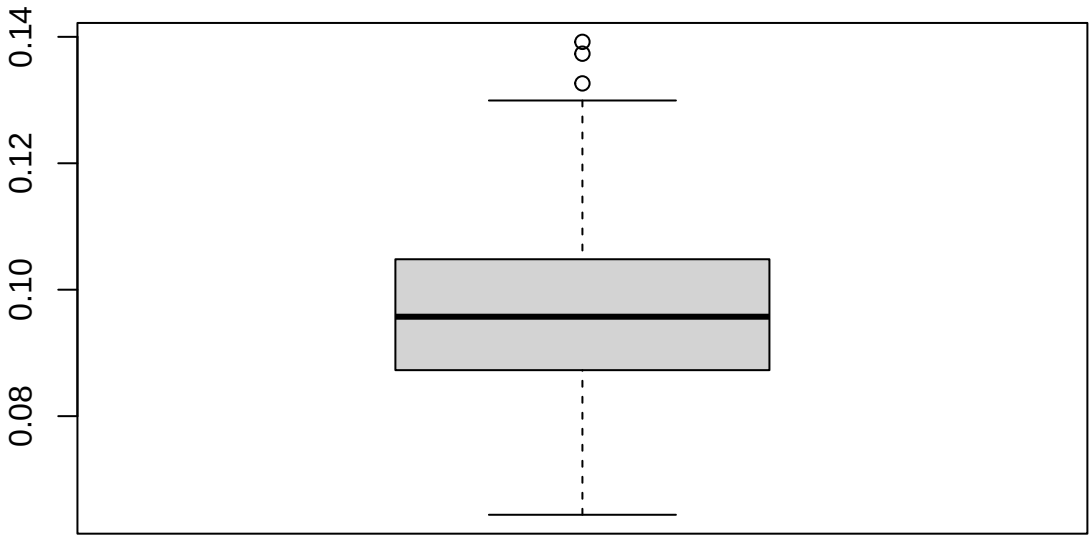

```
boxplot(drone_data$redmean)
```



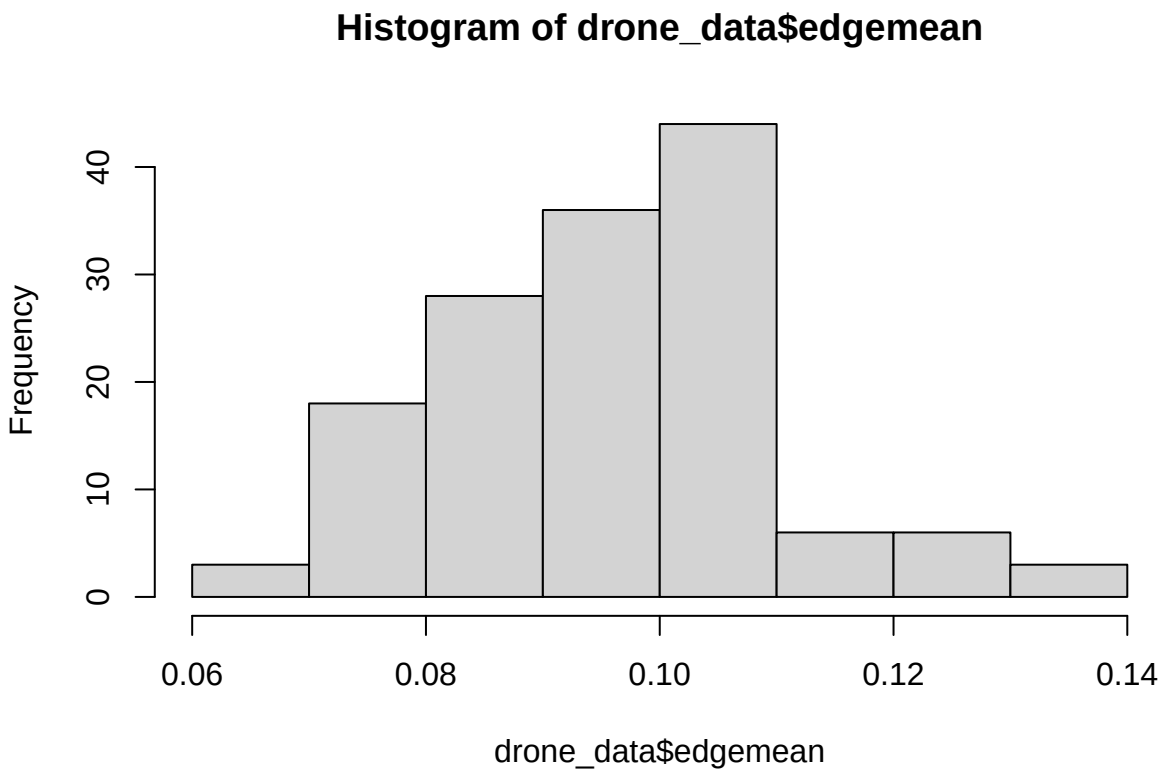

```
hist(drone_data$redmean)
```



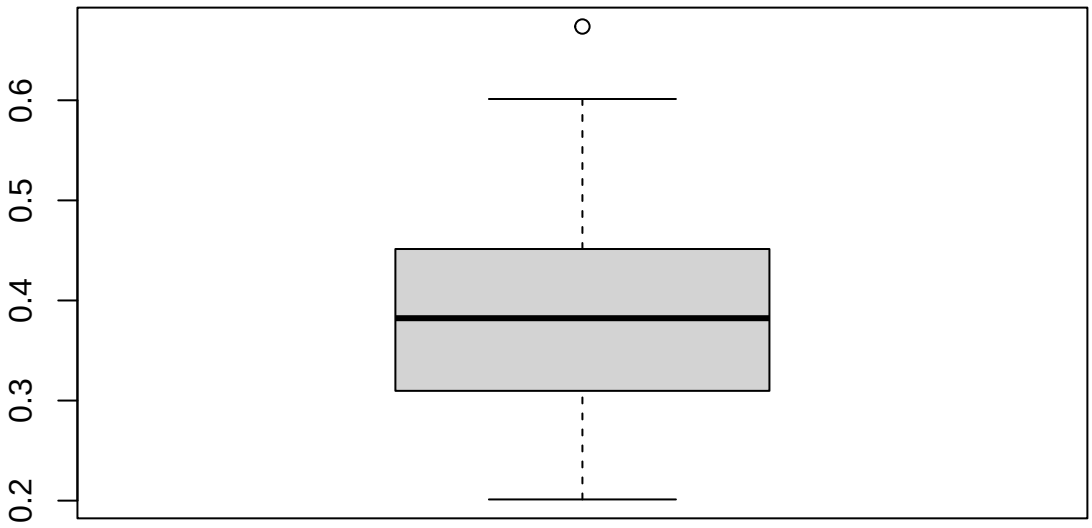

```
boxplot(drone_data$edgemean)
```



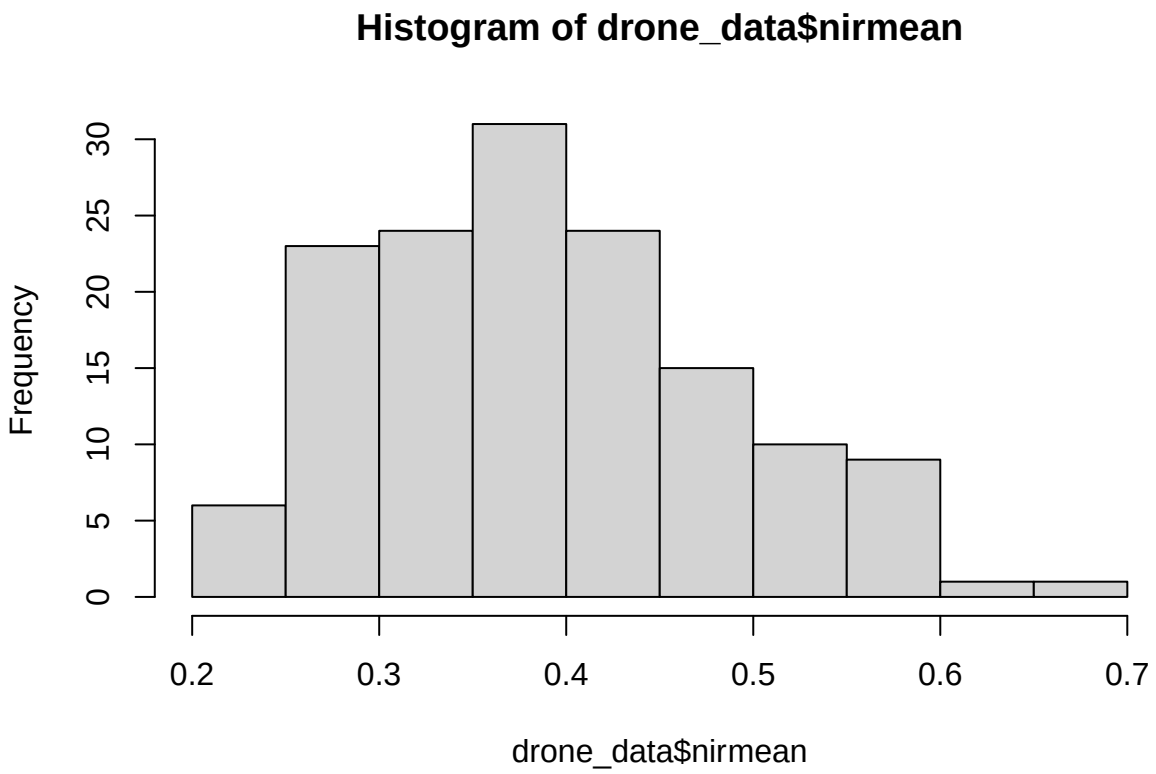

```
hist(drone_data$edgemean)
```



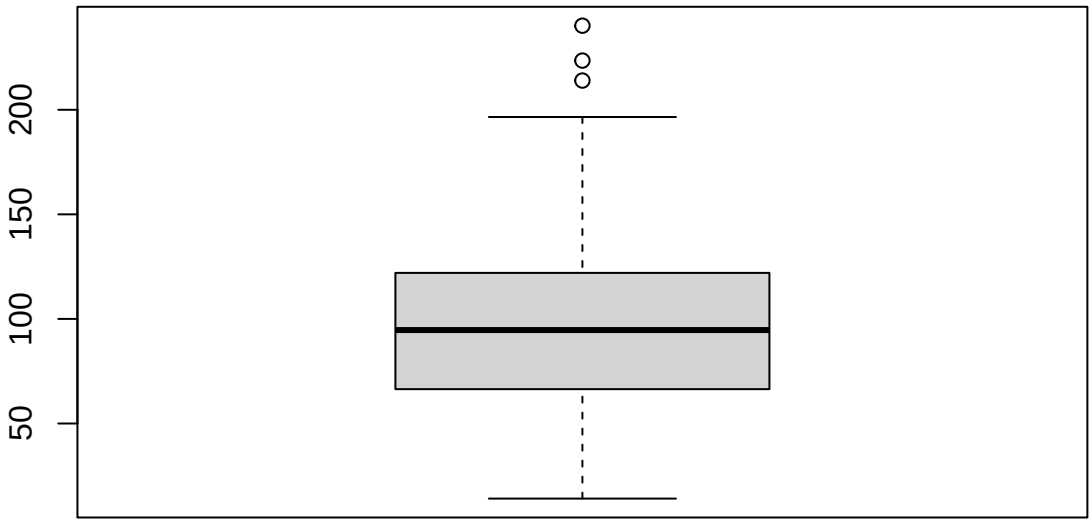

```
boxplot(drone_data$nirmean)
```



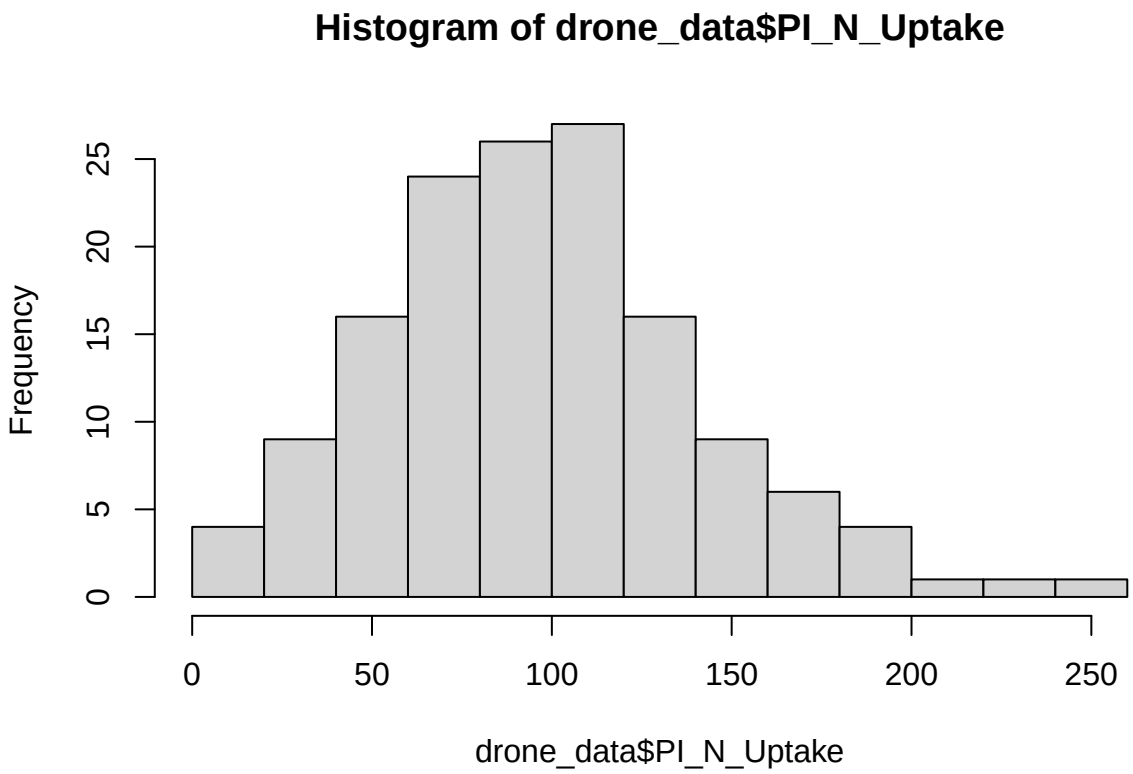

```
hist(drone_data$nirmean)
```



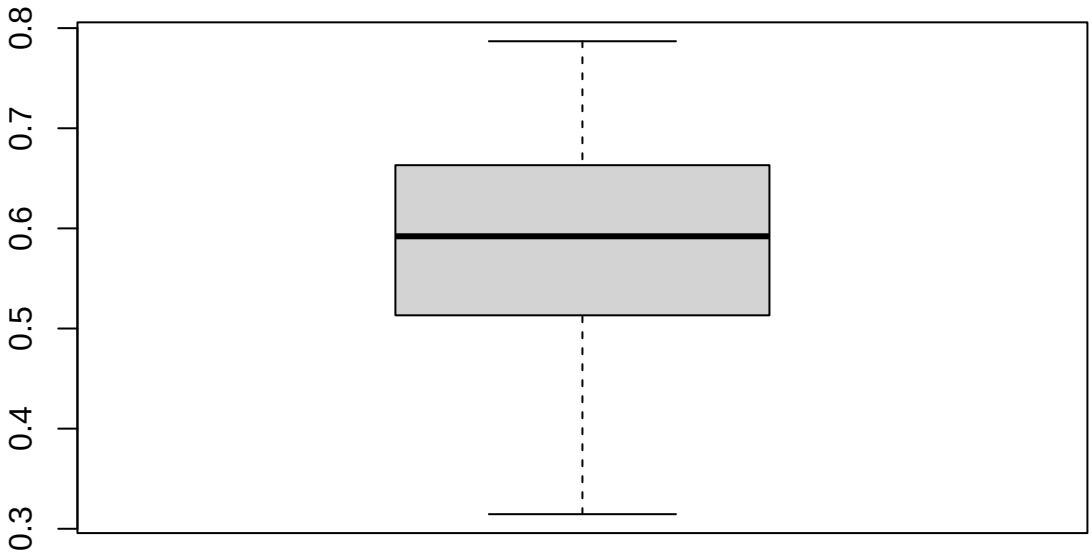

```
drone_data <-  drone_data %>%
  mutate( biomass_dry_wt = biomass_plus_bag_g - (paper_bag_g * num_of_paper_bags) ,
          aboveground_biomass = (biomass_dry_wt / ring_size) * 10 , #ring size 0.5 m^2 biomass in kg pe
          n_content = sample_N_ug / sample_weight_mg ,
          PI_N_Uptake = (aboveground_biomass * n_content) / 1000 #n uptake in kg per ha
          )#processes the data2

boxplot(drone_data$PI_N_Uptake)
```



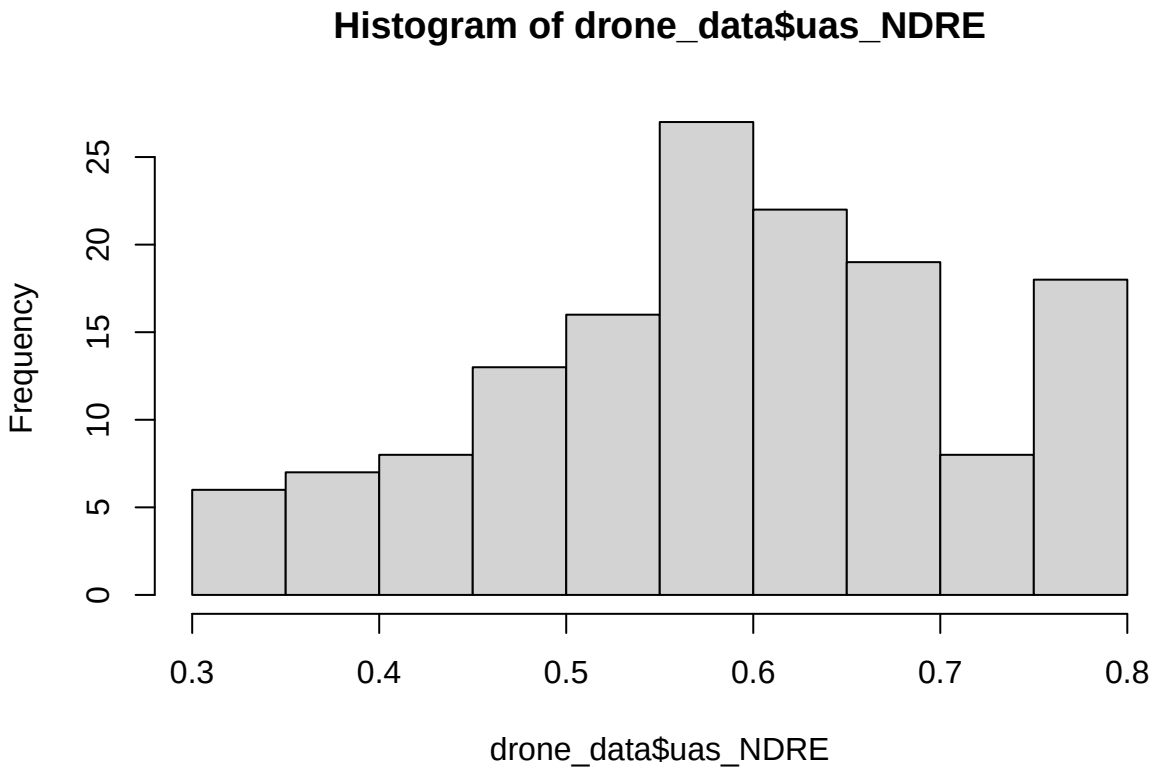

```
hist(drone_data$PI_N_Uptake)
```



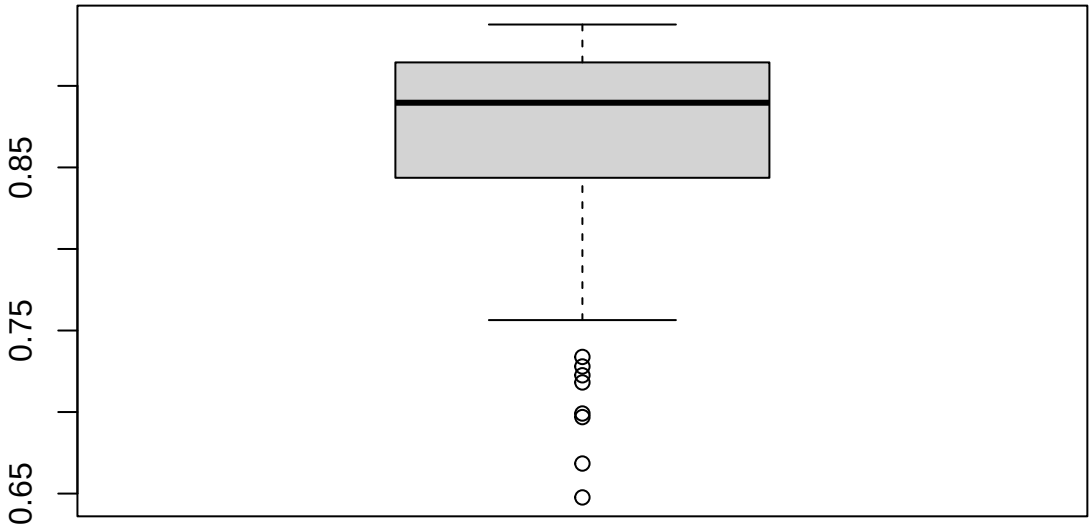

```
drone_data <- drone_data %>%
  mutate(uas_NDRE = ((nirmean - edgemean) / (nirmean + edgemean)) ,
         uas_NDVI = ((nirmean - redmean) / (nirmean + redmean))
         ) #calculates NDRE and NDVI

boxplot(drone_data$uas_NDRE)
```



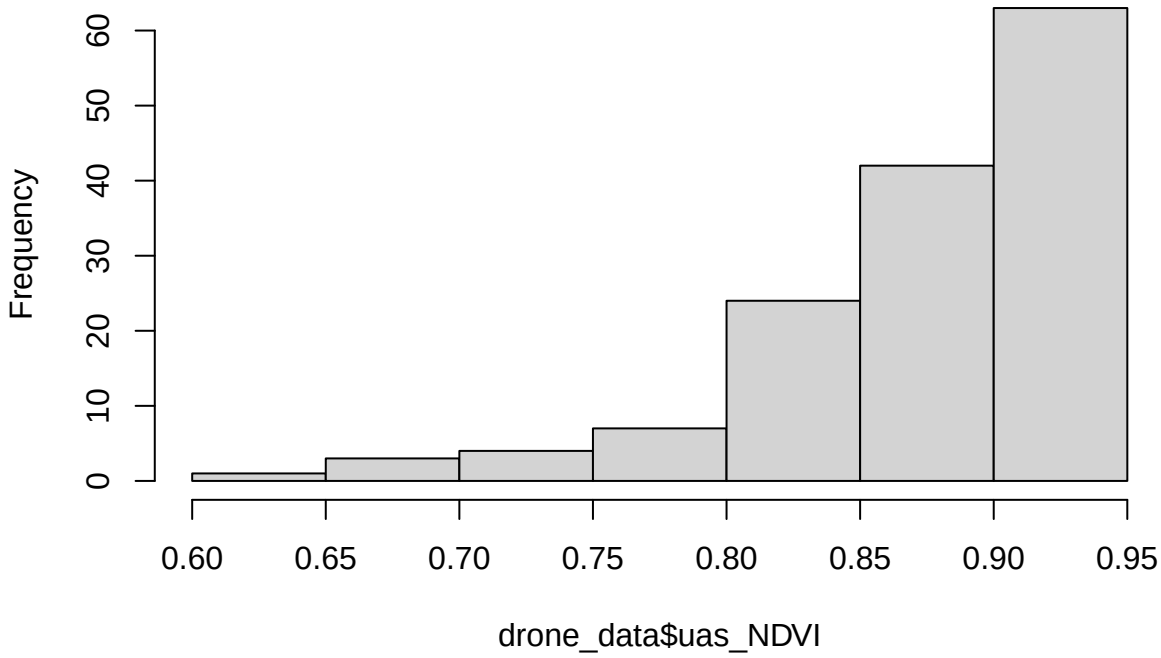

```
hist(drone_data$uas_NDRE)
```



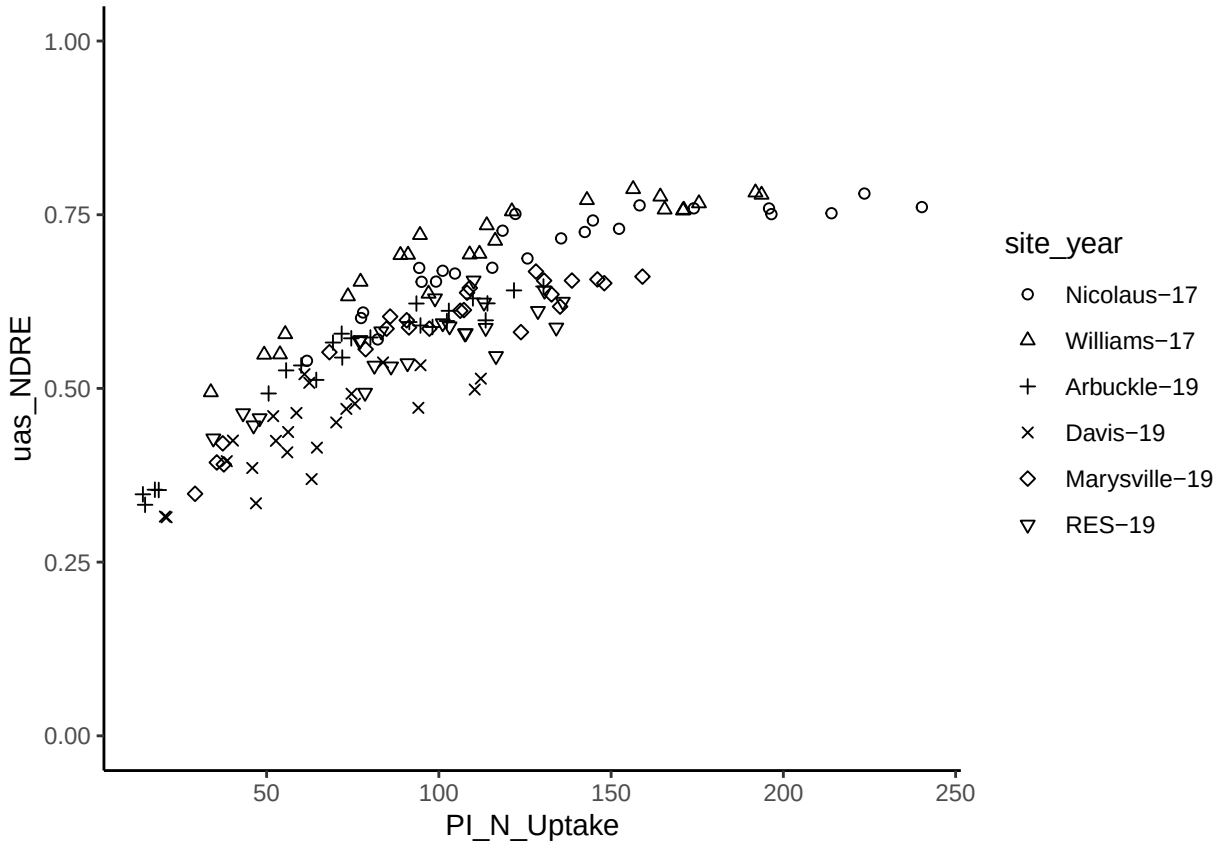

```
boxplot(drone_data$uas_NDVI)
```



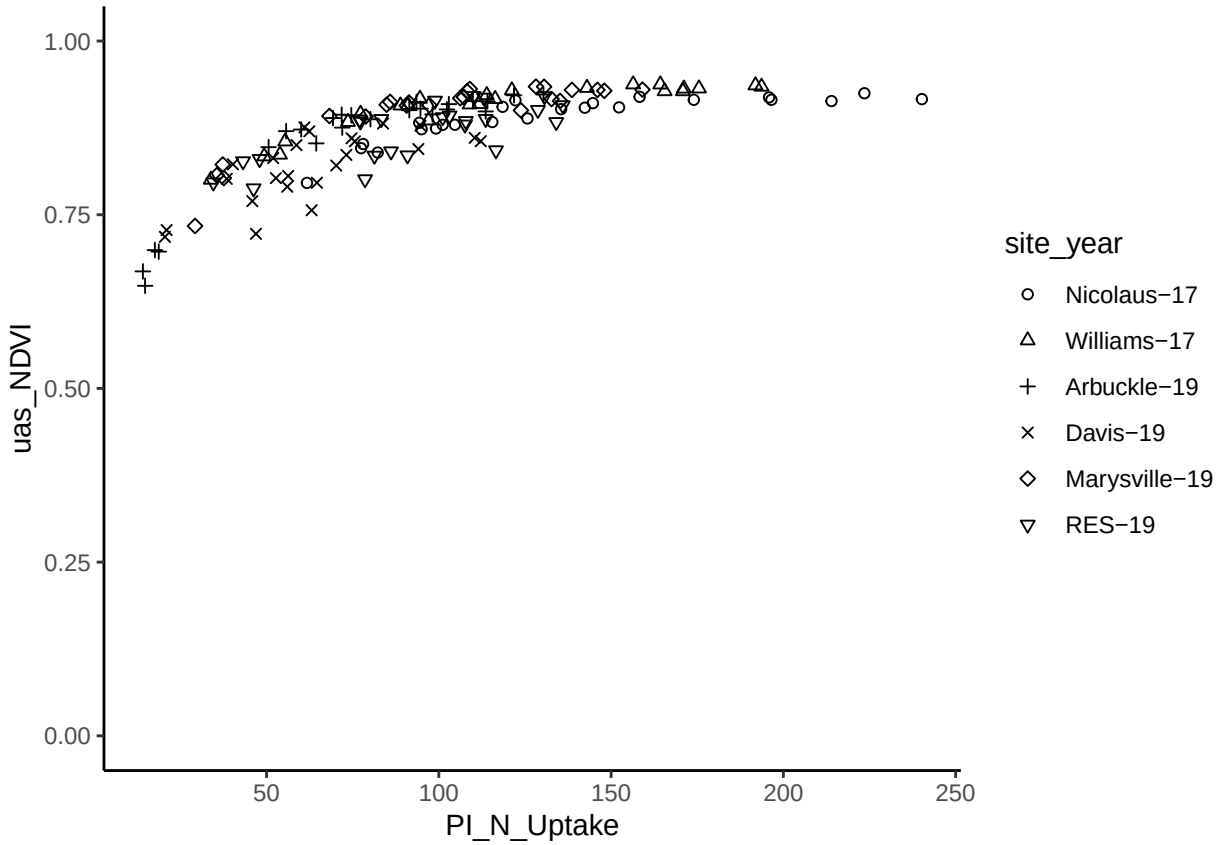

```
hist(drone_data$uas_NDVI)
```




```
ggplot(data = drone_data, aes(x = PI_N_Uptake , y = uas_NDRE )) +
  geom_point(mapping = aes(x = PI_N_Uptake , y = uas_NDRE , shape = site_year) , data = drone_data) +
  theme_classic() +
  scale_shape_manual(values = seq(4:20)) +
  coord_cartesian(ylim = c(0,1))
```




```
ggplot(data = drone_data, aes(x = PI_N_Uptake , y = uas_NDVI )) +
  geom_point(mapping = aes(x = PI_N_Uptake , y = uas_NDVI , shape = site_year) , data = drone_data) +
  theme_classic() +
  scale_shape_manual(values = seq(4:20)) +
  coord_cartesian(ylim = c(0,1))
```




```
drone_data <- dplyr::select(drone_data ,
                      site_year,
                      year,
                      exp_plot_number,
                      Block,
                      MainPlot,
                      uas_NDVI,
                      uas_NDRE) #selects the relevant columns

str(drone_data , give.attr = FALSE)
```

```
## tibble [144 x 7] (S3: tbl_df/tbl/data.frame)
##  $ site_year      : Factor w/ 6 levels "Nicolaus-17",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year           : Factor w/ 2 levels "2017","2019": 1 1 1 1 1 1 1 1 1 1 ...
##  $ exp_plot_number: Factor w/ 27 levels "101","102","103",..: 1 2 3 4 6 7 9 10 11 12 ...
##  $ Block          : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...
##  $ MainPlot       : Factor w/ 7 levels "1","2","3","4",..: 1 2 3 4 6 7 2 3 4 5 ...
##  $ uas_NDVI       : num [1:144] 0.92 0.852 0.904 0.915 0.873 ...
##  $ uas_NDRE       : num [1:144] 0.763 0.609 0.725 0.751 0.653 ...
```

```
PI_data <- full_join(gs_ndvi_data , drone_data ) #joins greenseeker and drone data
```

```
## Joining with `by = join_by(site_year, year, exp_plot_number)`
```

```
PI_data <- PI_data %>%
  mutate(N_level_kgha_f = factor(N_level_kgha))

PI_data <- dplyr::select(PI_data ,
```

```
                      site_year,
                      year,
                      exp_plot_number,
                      Block,
                      MainPlot,
                      N_level_kgha,
                      N_level_kgha_f,
                      aboveground_biomass,
                      n_content,
                      PI_N_Uptake,
                      gs_NDVI,
                      uas_NDVI,
                      uas_NDRE) #selects the relevant columns
```

Calculating SI

```
max_PI_data <- PI_data %>%
  dplyr::select(site_year , gs_NDVI , uas_NDVI , uas_NDRE) %>%
  group_by(site_year) %>%
  summarise_all(.funs = quantile, probs = 0.95) %>%
  ungroup()

nic17 <- subset(max_PI_data, site_year == "Nicolaus-17")
nic17max_gs_NDVI <- as.numeric(nic17$gs_NDVI)
nic17max_uas_NDVI <- as.numeric(nic17$uas_NDVI)
nic17max_uas_NDRE <- as.numeric(nic17$uas_NDRE)


wil17 <- subset(max_PI_data, site_year == "Williams-17")
wil17max_gs_NDVI <- as.numeric(wil17$gs_NDVI)
wil17max_uas_NDVI <- as.numeric(wil17$uas_NDVI)
wil17max_uas_NDRE <- as.numeric(wil17$uas_NDRE)


arb19 <- subset(max_PI_data, site_year == "Arbuckle-19")
arb19max_gs_NDVI <- as.numeric(arb19$gs_NDVI)
arb19max_uas_NDVI <- as.numeric(arb19$uas_NDVI)
arb19max_uas_NDRE <- as.numeric(arb19$uas_NDRE)


mry19 <- subset(max_PI_data, site_year == "Marysville-19")
mry19max_gs_NDVI <- as.numeric(mry19$gs_NDVI)
mry19max_uas_NDVI <- as.numeric(mry19$uas_NDVI)
mry19max_uas_NDRE <- as.numeric(mry19$uas_NDRE)


dav19 <- subset(max_PI_data, site_year == "Davis-19")
dav19max_gs_NDVI <- as.numeric(dav19$gs_NDVI)
dav19max_uas_NDVI <- as.numeric(dav19$uas_NDVI)
dav19max_uas_NDRE <- as.numeric(dav19$uas_NDRE)


res19 <- subset(max_PI_data, site_year == "RES-19")
```

```
res19max_gs_NDVI <- as.numeric(res19$gs_NDVI)
res19max_uas_NDVI <- as.numeric(res19$uas_NDVI)
res19max_uas_NDRE <- as.numeric(res19$uas_NDRE)

PI_data <- PI_data %>%
  mutate(max_gs_NDVI = case_when(
                           site_year == "Nicolaus-17" ~ nic17max_gs_NDVI ,
                           site_year == "Williams-17" ~ wil17max_gs_NDVI ,
                           site_year == "Arbuckle-19" ~ arb19max_gs_NDVI ,
                           site_year == "Davis-19" ~ dav19max_gs_NDVI ,
                           site_year == "Marysville-19" ~ mry19max_gs_NDVI ,
                           site_year == "RES-19" ~ res19max_gs_NDVI))

PI_data <- PI_data %>%
  mutate(max_uas_NDVI = case_when(
                           site_year == "Nicolaus-17" ~ nic17max_uas_NDVI ,
                           site_year == "Williams-17" ~ wil17max_uas_NDVI ,
                           site_year == "Arbuckle-19" ~ arb19max_uas_NDVI ,
                           site_year == "Davis-19" ~ dav19max_uas_NDVI ,
                           site_year == "Marysville-19" ~ mry19max_uas_NDVI ,
                           site_year == "RES-19" ~ res19max_uas_NDVI))

PI_data <- PI_data %>%
  mutate(max_uas_NDRE = case_when(
                           site_year == "Nicolaus-17" ~ nic17max_uas_NDRE ,
                           site_year == "Williams-17" ~ wil17max_uas_NDRE ,
                           site_year == "Arbuckle-19" ~ arb19max_uas_NDRE ,
                           site_year == "Davis-19" ~ dav19max_uas_NDRE ,
                           site_year == "Marysville-19" ~ mry19max_uas_NDRE ,
                           site_year == "RES-19" ~ res19max_uas_NDRE))

PI_data <- PI_data %>%
  mutate(gs_NDVI_Sufficiency_Index =  gs_NDVI / max_gs_NDVI ,
         uas_NDVI_Sufficiency_Index = uas_NDVI / max_uas_NDVI ,
         uas_NDRE_Sufficiency_Index = uas_NDRE / max_uas_NDRE) #calculates the sufficiency indices

PI_data_test <- PI_data %>%
  filter(gs_NDVI_Sufficiency_Index > 1)

length(PI_data_test$gs_NDVI_Sufficiency_Index) / length(PI_data$gs_NDVI_Sufficiency_Index)
```

```
## [1] 0.07638889
```

```
PI_data_test <- PI_data %>%
  filter(uas_NDRE_Sufficiency_Index > 1)

length(PI_data_test$uas_NDRE_Sufficiency_Index) /length(PI_data$uas_NDRE_Sufficiency_Index)
```

```
## [1] 0.08333333
```

Yield Data

```
yield_data <- read_csv(file = "DATA/yield_data.csv")
```

```
## Rows: 672 Columns: 34
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): site_year, SubPlot, Received_TopDress
## dbl (31): year, Block, MainPlot, exp_plot_number, N_level, TopDress, Seasona...
##
## 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.
```

```
str(yield_data , give.attr = FALSE)
```

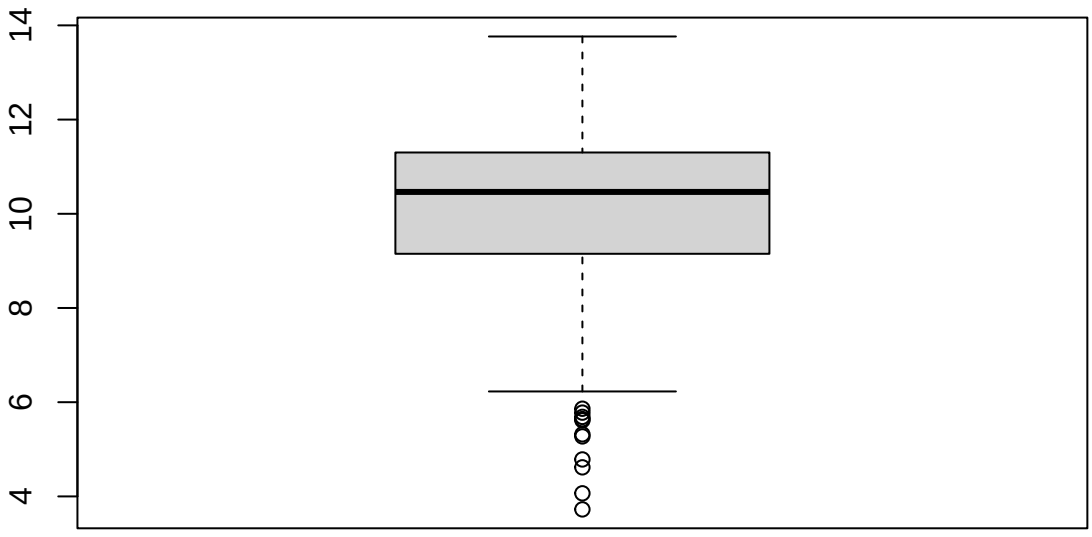
```
## spc_tbl_ [672 x 34] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ site_year            : chr [1:672] "Davis-16" "Davis-16" "Davis-16" "Davis-16" ...
##  $ year                 : num [1:672] 2016 2016 2016 2016 2016 ...
##  $ Block                : num [1:672] 1 1 1 1 1 2 2 2 2 2 ...
##  $ MainPlot             : num [1:672] 1 2 3 4 5 1 2 3 4 5 ...
##  $ SubPlot              : chr [1:672] "3" "1" "3" "2" ...
##  $ exp_plot_number      : num [1:672] 101 102 103 104 105 201 202 203 204 205 ...
##  $ N_level              : num [1:672] 75 225 0 125 175 75 0 225 175 125 ...
##  $ TopDress             : num [1:672] 0 0 0 0 0 0 0 0 0 0 ...
##  $ SeasonalNRate        : num [1:672] 75 225 0 125 175 75 0 225 175 125 ...
##  $ N_level_kgha         : num [1:672] 75 225 0 125 175 75 0 225 175 125 ...
##  $ TopDress_kgha        : num [1:672] 0 0 0 0 0 0 0 0 0 0 ...
##  $ SeasonalNRate_kgha   : num [1:672] 75 225 0 125 175 75 0 225 175 125 ...
##  $ Received_TopDress    : chr [1:672] "NO" "NO" "NO" "NO" ...
##  $ tare                 : num [1:672] 1220 1220 1220 1220 1220 1220 1220 1220 1220 1220 ...
##  $ FW1PlusTare          : num [1:672] 5662 5298 5256 4846 5220 ...
##  $ FW2PlusTare          : num [1:672] 4610 5438 1220 4278 5156 ...
##  $ SSFWPlusTare         : num [1:672] 3070 3262 2266 2482 2650 ...
##  $ SSODW                : num [1:672] 692 705 519 541 538 ...
##  $ HarvestBagPlusTie    : num [1:672] 121 122 122 121 122 ...
##  $ Grain1PlusPaperBag1  : num [1:672] 292 271 188 215 208 ...
##  $ PaperBag1            : num [1:672] 7.92 7.92 7.92 7.92 7.92 7.92 7.92 7.92 7.92 7.92 ...
##  $ Grain2PlusPaperBag2  : num [1:672] 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 ...
##  $ PaperBag2            : num [1:672] 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 ...
##  $ StrawSampleSize      : num [1:672] 8.27 7.74 7.9 8.38 7.52 ...
##  $ StrawN               : num [1:672] 55.1 55.2 49.4 47.1 45.3 ...
##  $ GrainSampleSize      : num [1:672] 4.95 4.97 5.04 5.02 4.57 ...
##  $ GrainN               : num [1:672] 54.5 58.6 47.4 52.5 46.1 ...
##  $ DaysPI2Harvest       : num [1:672] 102 102 102 102 102 102 102 102 102 102 ...
##  $ SeedTray1.1          : num [1:672] 243 243 243 243 243 243 243 243 243 243 ...
##  $ SeedTray1.2          : num [1:672] 0.62 0.62 0.62 0.64 0.64 0.62 0.62 0.62 0.62 0.62 ...
##  $ Grain3PlusSeedTray1  : num [1:672] 435 514 426 455 447 ...
##  $ MoistureContentGrain3: num [1:672] 11.2 11.2 11.2 11.2 11.2 11.2 11.2 11.2 11.2 11.2 ...
##  $ SeedTray2            : num [1:672] 244 244 244 244 244 ...
##  $ Grain4PlusSeedTray2  : num [1:672] 254 254 254 254 254 ...
```

```
yield_data <- yield_data %>%
  filter(!year %in% c( "2016" , "2018") & N_level_kgha != 275) #removing the years and  N rate to match

yield_data <- yield_data %>%
  mutate(
    site_year = factor(site_year),
    year = factor(year),
    Block = factor(Block),
```

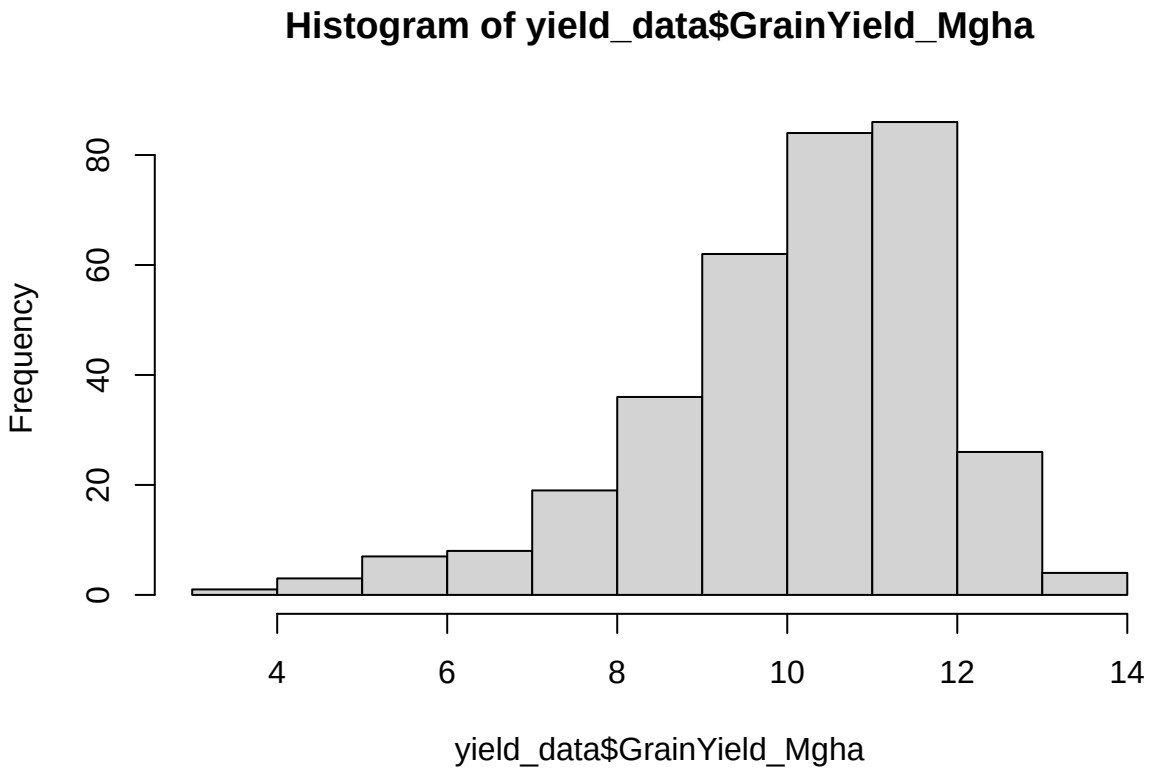
```
    MainPlot = factor(MainPlot),
    exp_plot_number = factor(exp_plot_number),
    N_level = factor(N_level),
    SubPlot = factor(SubPlot),
    TopDress = factor(TopDress),
    SeasonalNRate_f = factor(SeasonalNRate),
    N_level_kgha_f = factor(N_level_kgha),
    TopDress_kgha_f = factor(TopDress_kgha),
    SeasonalNRate_kgha_f = factor(SeasonalNRate_kgha)
    ) #changes these columns to factor

str(yield_data , give.attr = FALSE)
```

```
## tibble [336 x 38] (S3: tbl_df/tbl/data.frame)
##  $ site_year            : Factor w/ 6 levels "Arbuckle-19",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ year                 : Factor w/ 2 levels "2017","2019": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Block                : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...
##  $ MainPlot             : Factor w/ 7 levels "1","2","3","4",..: 1 2 3 4 6 7 2 3 4 5 ...
##  $ SubPlot              : Factor w/ 5 levels "1","2","3","a",..: 2 3 1 1 1 2 1 1 2 3 ...
##  $ exp_plot_number      : Factor w/ 27 levels "101","102","103",..: 1 2 3 4 6 7 9 10 11 12 ...
##  $ N_level              : Factor w/ 11 levels "0","45","75",..: 11 1 6 8 2 3 3 6 1 8 ...
##  $ TopDress             : Factor w/ 4 levels "0","25","30",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SeasonalNRate        : num [1:336] 225 0 125 175 45 75 75 125 0 175 ...
##  $ N_level_kgha         : num [1:336] 225 0 125 175 45 75 75 125 0 175 ...
##  $ TopDress_kgha        : num [1:336] 0 0 0 0 0 0 0 0 0 0 ...
##  $ SeasonalNRate_kgha   : num [1:336] 225 0 125 175 45 75 75 125 0 175 ...
##  $ Received_TopDress    : chr [1:336] "NO" "NO" "NO" "NO" ...
##  $ tare                 : num [1:336] 1220 1220 1220 1220 1220 1220 1220 1220 1220 1220 ...
##  $ FW1PlusTare          : num [1:336] 4360 4818 5376 5598 4852 ...
##  $ FW2PlusTare          : num [1:336] 3254 1220 1220 1220 1220 ...
##  $ SSFWPlusTare         : num [1:336] 2324 1814 1994 2126 2106 ...
##  $ SSODW                : num [1:336] 632 479 570 562 583 ...
##  $ HarvestBagPlusTie    : num [1:336] 162 176 169 121 120 ...
##  $ Grain1PlusPaperBag1  : num [1:336] 230 173 211 233 265 ...
##  $ PaperBag1            : num [1:336] 7.96 7.96 7.96 7.96 7.96 7.96 7.96 7.96 7.96 7.96 ...
##  $ Grain2PlusPaperBag2  : num [1:336] 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 ...
##  $ PaperBag2            : num [1:336] 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 7.01 ...
##  $ StrawSampleSize      : num [1:336] 8.25 7.96 7.76 7.89 8.45 ...
##  $ StrawN               : num [1:336] 63.5 36 46.1 55.9 39.7 ...
##  $ GrainSampleSize      : num [1:336] 4.97 5.52 5.23 5.72 5.82 ...
##  $ GrainN               : num [1:336] 60.5 50 56 66.7 51.1 ...
##  $ DaysPI2Harvest       : num [1:336] 76 76 76 76 76 76 76 76 76 76 ...
##  $ SeedTray1.1          : num [1:336] 243 243 243 243 243 243 243 243 243 243 ...
##  $ SeedTray1.2          : num [1:336] 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 ...
##  $ Grain3PlusSeedTray1  : num [1:336] 470 410 449 471 504 ...
##  $ MoistureContentGrain3: num [1:336] 11.2 11.2 11.2 11.2 11.2 11.2 11.2 11.2 11.2 11.2 ...
##  $ SeedTray2            : num [1:336] 244 244 244 244 244 ...
##  $ Grain4PlusSeedTray2  : num [1:336] 254 254 254 254 254 ...
##  $ SeasonalNRate_f      : Factor w/ 21 levels "0","25","30",..: 18 1 12 14 4 7 7 12 1 14 ...
##  $ N_level_kgha_f       : Factor w/ 11 levels "0","45","75",..: 10 1 5 8 2 3 3 5 1 8 ...
##  $ TopDress_kgha_f      : Factor w/ 4 levels "0","25","34",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SeasonalNRate_kgha_f : Factor w/ 22 levels "0","25","34",..: 18 1 11 15 4 7 7 11 1 15 ...
```

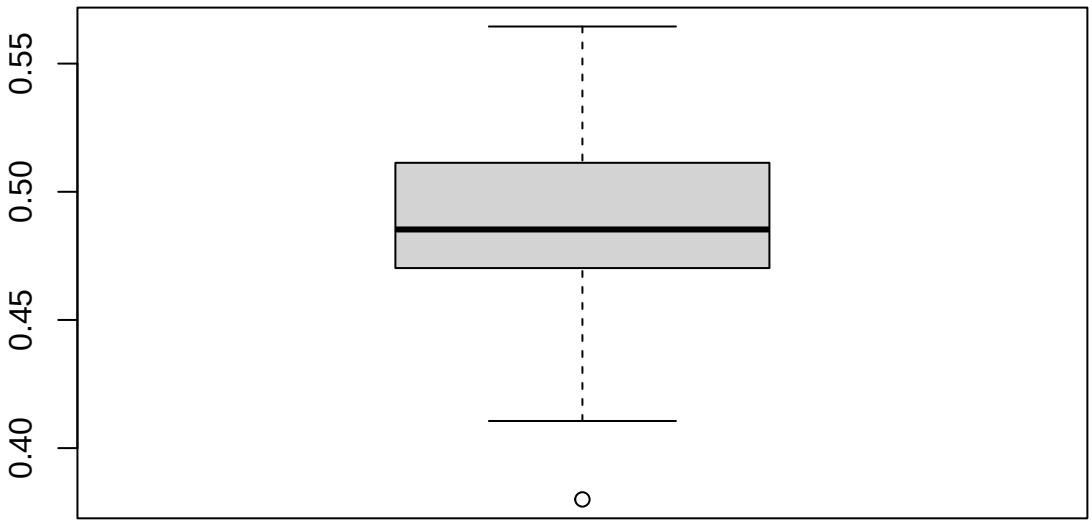
```r
yield_data <- yield_data %>%
  mutate(
    FW1net = FW1PlusTare - tare,
    FW2net = FW2PlusTare - tare,
    TotalFW = FW1net + FW2net,
    SSFWnet = SSFWPlusTare - tare,
    Ratio = SSFWnet / TotalFW,
    SSODWnet = SSODW - HarvestBagPlusTie,
    SeedTray1 = SeedTray1.1 + SeedTray1.2, #adds the decimal to the 243g to get the tare weight for the
    Grain3net = Grain3PlusSeedTray1 - SeedTray1, #subtract tare of seed tray from grain3. Grain3 is the
    Grain4net = Grain4PlusSeedTray2 - SeedTray2, #grain4 is the amount of grain removed for ballmilling
    Grain2net = Grain2PlusPaperBag2 - PaperBag2, #yield component grain sample
    Grain2net = Grain2net * Ratio, #this essentially subsamples the yield component grain sample
    GrainNet = Grain3net + Grain4net + Grain2net, #add the grain removed for ball milling and yield com
    GrainRing = GrainNet / Ratio, #the amount of grain in the entire m^2 ring in grams
    GrainYield = GrainRing * 10, #g/m^2 to kg/ha
    GrainYield_kgha = GrainYield * ((100-MoistureContentGrain3)/86),#corrects for 14% moisture based on
    GrainYield_Mgha = GrainYield_kgha / 1000 , #converts kg/ha to Mg/ha
    Grain5 = GrainRing * ((100-MoistureContentGrain3)/98.1), #grain in the ring if the subsample was at
    Grain6 = GrainNet * ((100-MoistureContentGrain3)/98.1), #grain in the subsample if it was at 1.9% m
    StrawSS = SSODWnet - Grain6 , #just straw in subsample in grams
    StrawRing = StrawSS / Ratio, #straw in ring in grams i.e g/m2
    StrawNcon = StrawN / StrawSampleSize,
    StrawNup = (StrawRing * StrawNcon) / 100, #straw Nup divide by 100 to convert mg/m2 to kg/ha - this
    GrainNcon = (GrainN / GrainSampleSize), #grain in ring in kg/ha
    GrainNup = (Grain5 * GrainNcon) / 100, #grain Nup divide by 100 to convert mg N/m2 to kg N/ha
    TotalSeasonalNup = StrawNup + GrainNup, #in kg/ha
    HarvestIndex = Grain5 / (Grain5 + StrawRing),
    Moisture = SSFWnet / SSODWnet
    )

boxplot(yield_data$GrainYield_Mgha)
```



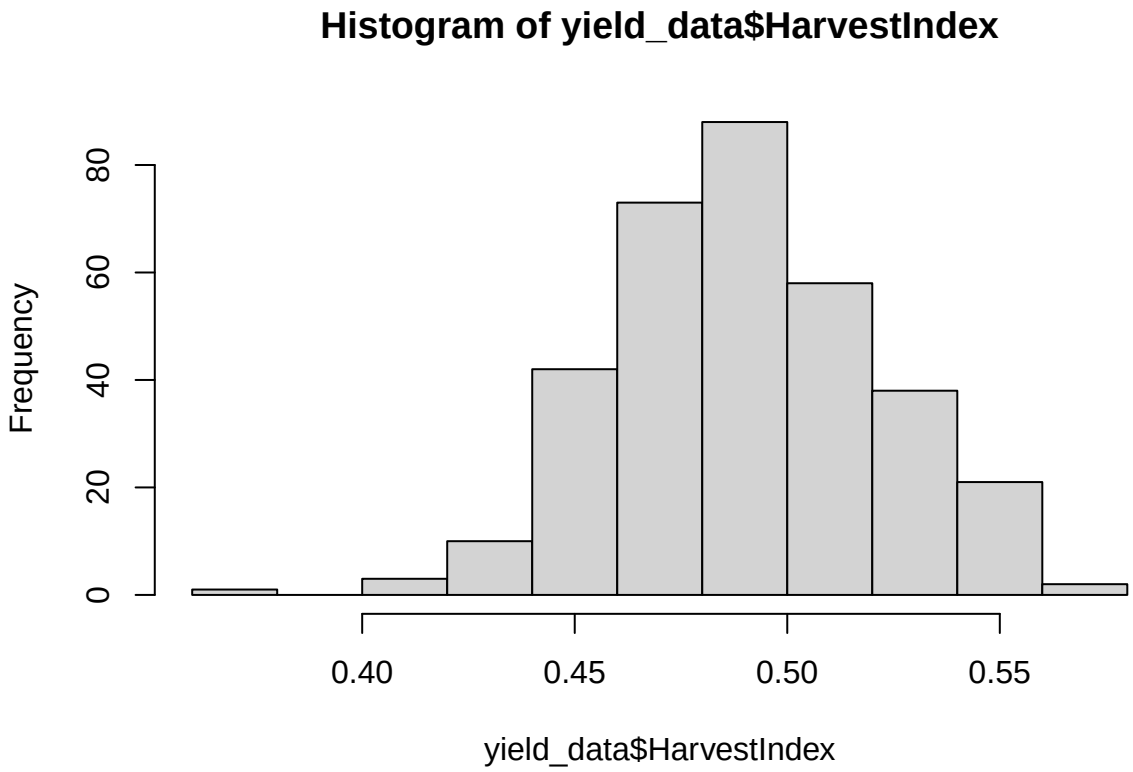

```r
hist(yield_data$GrainYield_Mgha)
```



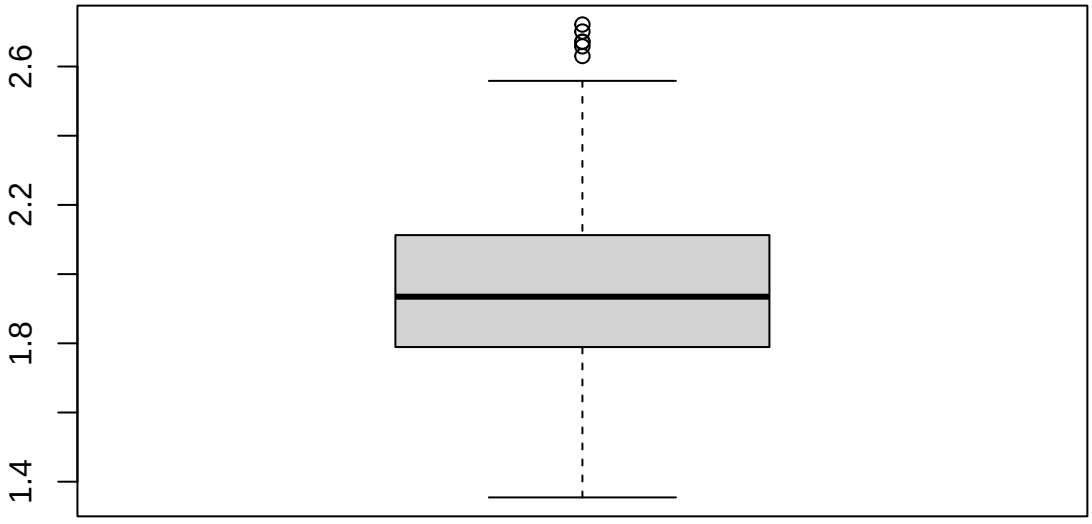

```
boxplot(yield_data$HarvestIndex)
```



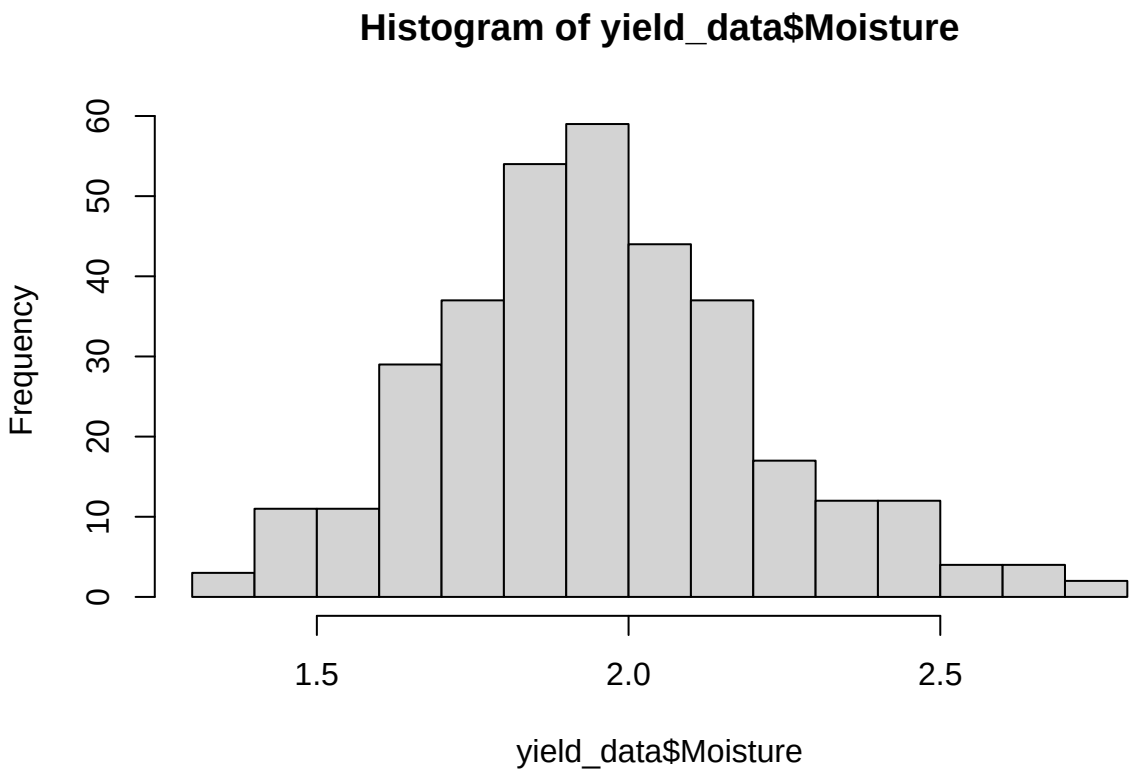

```
hist(yield_data$HarvestIndex)
```




```
boxplot(yield_data$Moisture)
```




```
hist(yield_data$Moisture)
```




```
yield_data_2 <- yield_data

#the data looks good - don't see any unusual values

yield_data <- dplyr::select(yield_data,
                            site_year,
                            year,
                            Block,
                            MainPlot,
                            SubPlot,
                            exp_plot_number,
                            N_level_kgha,
                            N_level_kgha_f,
                            TopDress_kgha,
                            TopDress_kgha_f,
                            GrainYield_Mgha
                            )


paper2_data <- full_join(PI_data , yield_data)
```

```
## Joining with `by = join_by(site_year, year, exp_plot_number, Block, MainPlot,
## N_level_kgha, N_level_kgha_f)`
```

```
paper2_data <- dplyr::select(paper2_data,
                            site_year,
                            year,
                            Block,
                            MainPlot,
                            SubPlot,
```

```
                              exp_plot_number,
                              N_level_kgha,
                              N_level_kgha_f,
                              TopDress_kgha,
                              TopDress_kgha_f,
                              gs_NDVI,
                              uas_NDVI,
                              uas_NDRE,
                              gs_NDVI_Sufficiency_Index,
                              uas_NDVI_Sufficiency_Index,
                              uas_NDRE_Sufficiency_Index,
                              GrainYield_Mgha)
```

Outlier Removal

```
gs_ndvi_SI_mean <- mean(paper2_data$gs_NDVI_Sufficiency_Index)
gs_ndvi_SI_sd <- sd(paper2_data$gs_NDVI_Sufficiency_Index)

lower_limit <- gs_ndvi_SI_mean - (4*gs_ndvi_SI_sd) #lower limit for outlier removal. Observations that

print(lower_limit)
```

```
## [1] 0.2121179
```

```
paper2_data_outliers <- paper2_data %>%
  filter(gs_NDVI_Sufficiency_Index <= lower_limit)

paper2_data_OR <- paper2_data %>%
  filter(gs_NDVI_Sufficiency_Index >= lower_limit)
```

TABLES S2-S4

```
table_data_1 <- paper2_data_OR %>%
  dplyr::select(site_year , year , N_level_kgha , TopDress_kgha , gs_NDVI , uas_NDRE , gs_NDVI_Sufficie
  filter(year == 2017 & TopDress_kgha == 0) #filters data to 2017 and 0 top-dress

table_data_1 <- table_data_1 %>%
  dplyr::select(site_year , N_level_kgha , gs_NDVI , uas_NDRE , gs_NDVI_Sufficiency_Index , uas_NDRE_Su
  group_by(site_year , N_level_kgha) %>%
  summarise_all(list(min = min , max = max , mean = mean)) %>%
  dplyr::select(site_year , N_level_kgha ,
                gs_NDVI_min ,
                gs_NDVI_max ,
                gs_NDVI_mean ,
                gs_NDVI_Sufficiency_Index_min ,
                gs_NDVI_Sufficiency_Index_max ,
                gs_NDVI_Sufficiency_Index_mean ,
                uas_NDRE_min ,
                uas_NDRE_max ,
                uas_NDRE_mean ,
                uas_NDRE_Sufficiency_Index_min ,
                uas_NDRE_Sufficiency_Index_max ,
```

```
                uas_NDRE_Sufficiency_Index_mean
                ) %>%
  ungroup() #takes the min max and mean of all the relevant columns

table_data_1_nic <- table_data_1 %>%
  filter(site_year == "Nicolaus-17") %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) #filters nic-17 site and rounds to 2 places
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(everything(), round, 2)`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
write_excel_csv(table_data_1_nic , "DATA/R_EXPORT/table_data_1_nic.csv" ) #exports a excel csv to copy

table_data_1_wil <- table_data_1 %>%
  filter(site_year == "Williams-17") %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) #same as above, but for wil-17

write_excel_csv(table_data_1_wil , "DATA/R_EXPORT/table_data_1_wil.csv" ) #exports a excel csv to copy

table_data_1_2017_all <- table_data_1 %>%
  dplyr::select(-N_level_kgha) %>%
  group_by(site_year) %>%
  summarise_all(list(min = min , max = max , mean = mean)) %>%
  ungroup() %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) %>%
  mutate(site_year = c("Nicolaus-17" , "Williams-17") , .before = gs_NDVI_min_min,
         site_year = as.factor(site_year)) %>%
  dplyr::select(c(site_year,
                  gs_NDVI_min_min ,
                  gs_NDVI_max_max ,
                  gs_NDVI_mean_mean ,
                  gs_NDVI_Sufficiency_Index_min_min ,
                  gs_NDVI_Sufficiency_Index_max_max ,
                  gs_NDVI_Sufficiency_Index_mean_mean ,
                  uas_NDRE_min_min ,
                  uas_NDRE_max_max ,
                  uas_NDRE_mean_mean ,
                  uas_NDRE_Sufficiency_Index_min_min ,
                  uas_NDRE_Sufficiency_Index_max_max ,
                  uas_NDRE_Sufficiency_Index_mean_mean )) #gets min max and mean for the 2 2017 sites a

write_excel_csv(table_data_1_2017_all , "DATA/R_EXPORT/table_data_1_2017_all.csv" ) #exports a excel cs
```

```r
table_data_2 <- paper2_data_OR %>%
  dplyr::select(site_year , year , N_level_kgha , TopDress_kgha , gs_NDVI , uas_NDRE , gs_NDVI_Sufficie
  filter(year == 2019 & TopDress_kgha == 0) #filters data to 2019 and 0 top-dress

table_data_2 <- table_data_2 %>%
  dplyr::select(site_year , N_level_kgha , gs_NDVI , uas_NDRE , gs_NDVI_Sufficiency_Index , uas_NDRE_Su
  group_by(site_year , N_level_kgha) %>%
  summarise_all(list(min = min , max = max , mean = mean)) %>%
  dplyr::select(site_year , N_level_kgha ,
                gs_NDVI_min ,
                gs_NDVI_max ,
                gs_NDVI_mean ,
                gs_NDVI_Sufficiency_Index_min ,
                gs_NDVI_Sufficiency_Index_max ,
                gs_NDVI_Sufficiency_Index_mean ,
                uas_NDRE_min ,
                uas_NDRE_max ,
                uas_NDRE_mean ,
                uas_NDRE_Sufficiency_Index_min ,
                uas_NDRE_Sufficiency_Index_max ,
                uas_NDRE_Sufficiency_Index_mean
                ) %>%
  ungroup() #takes the min max and mean of all the relevant columns

table_data_2_arb <- table_data_2 %>%
  filter(site_year == "Arbuckle-19") %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) #filters arb-19 site and rounds to 2 places

write_excel_csv(table_data_2_arb , "DATA/R_EXPORT/table_2_data_Arbuckle19.csv" ) #exports a excel csv t

table_data_2_dav <- table_data_2 %>%
  filter(site_year == "Davis-19") %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) #filters dav-19 site and rounds to 2 places

write_excel_csv(table_data_2_dav , "DATA/R_EXPORT/table_2_data_Davis19.csv" ) #exports a excel csv to c

table_data_2_mry <- table_data_2 %>%
  filter(site_year == "Marysville-19") %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) #filters mry-19 site and rounds to 2 places

write_excel_csv(table_data_2_mry , "DATA/R_EXPORT/table_2_data_Marysville19.csv" ) #exports a excel csv

table_data_2_res <- table_data_2 %>%
  filter(site_year == "RES-19") %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) #filters res-19 site and rounds to 2 places

write_excel_csv(table_data_2_res , "DATA/R_EXPORT/table_2_data_RES19.csv" ) #exports a excel csv to cop

table_data_2_2019_all <- table_data_2 %>%
```

```
  dplyr::select(-N_level_kgha) %>%
  group_by(site_year) %>%
  summarise_all(list(min = min , max = max , mean = mean)) %>%
  ungroup() %>%
  dplyr::select(-site_year) %>%
  mutate(across(everything(), round, 2)) %>%
  mutate(site_year = c("Arbuckle-19" , "Davis-19" , "Marysville-19" , "RES-19") , .before = gs_NDVI_min
         site_year = as.factor(site_year)) %>%
  dplyr::select(c(site_year,
                  gs_NDVI_min_min ,
                  gs_NDVI_max_max ,
                  gs_NDVI_mean_mean ,
                  gs_NDVI_Sufficiency_Index_min_min ,
                  gs_NDVI_Sufficiency_Index_max_max ,
                  gs_NDVI_Sufficiency_Index_mean_mean ,
                  uas_NDRE_min_min ,
                  uas_NDRE_max_max ,
                  uas_NDRE_mean_mean ,
                  uas_NDRE_Sufficiency_Index_min_min ,
                  uas_NDRE_Sufficiency_Index_max_max ,
                  uas_NDRE_Sufficiency_Index_mean_mean )) #gets min max and mean for the 2019 sites acr

write_excel_csv(table_data_2_2019_all , "DATA/R_EXPORT/table_2_data_ALL_2019_SITES.csv" ) #exports a ex
```

FIG 3

ALL SITES (Fig S2)

model

```
paper2_data_OR_S1 <- paper2_data_OR %>%
  filter(TopDress_kgha == 0)

lme.1 <- lme(GrainYield_Mgha ~ N_level_kgha + I(N_level_kgha^2),
                  random = ~ N_level_kgha + I(N_level_kgha^2) | site_year,
                  data = paper2_data_OR_S1,
                  na.action = na.omit)

summary(lme.1)
```

```
## Linear mixed-effects model fit by REML
##   Data: paper2_data_OR_S1
##        AIC     BIC    logLik
##   427.7854 457.273 -203.8927
##
## Random effects:
##  Formula: ~N_level_kgha + I(N_level_kgha^2) | site_year
##  Structure: General positive-definite, Log-Cholesky parametrization
##                    StdDev       Corr
## (Intercept)       1.986461e+00 (Intr) N_lvl_
## N_level_kgha      1.754687e-02 -0.783
## I(N_level_kgha^2) 5.316616e-05  0.573 -0.917
## Residual          7.741335e-01
```

```
##
## Fixed effects:  GrainYield_Mgha ~ N_level_kgha + I(N_level_kgha^2)
##                       Value Std.Error  DF   t-value p-value
## (Intercept)        6.970994 0.8250156 136  8.449530   0e+00
## N_level_kgha       0.036650 0.0077018 136  4.758591   0e+00
## I(N_level_kgha^2) -0.000092 0.0000247 136 -3.707755   3e-04
##  Correlation:
##                   (Intr) N_lvl_
## N_level_kgha      -0.765
## I(N_level_kgha^2)  0.543 -0.917
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -4.636833539 -0.427216277 -0.009156218  0.509007581  2.855491090
##
## Number of Observations: 144
## Number of Groups: 6
```

```
car::Anova(lme.1, type = "3")
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: GrainYield_Mgha
##                     Chisq Df Pr(>Chisq)
## (Intercept)       71.395  1  < 2.2e-16 ***
## N_level_kgha      22.644  1  1.949e-06 ***
## I(N_level_kgha^2) 13.747  1  0.0002091 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MuMIn::r.squaredGLMM(lme.1)
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
##            R2m       R2c
## [1,] 0.4305297 0.8512486
```

```
coefficients(lme.1)
```

```
##               (Intercept) N_level_kgha I(N_level_kgha^2)
## Nicolaus-17     10.368082   0.01984158     -5.916342e-05
## Williams-17      6.605821   0.04809997     -1.316718e-04
## Arbuckle-19      4.414444   0.05988307     -1.334913e-04
## Davis-19         7.347929   0.01672837     -1.274998e-05
## Marysville-19    6.978654   0.03840123     -9.981216e-05
## RES-19           6.111033   0.03694357     -1.122348e-04
```

```
lme.1$coefficients$fixed
```

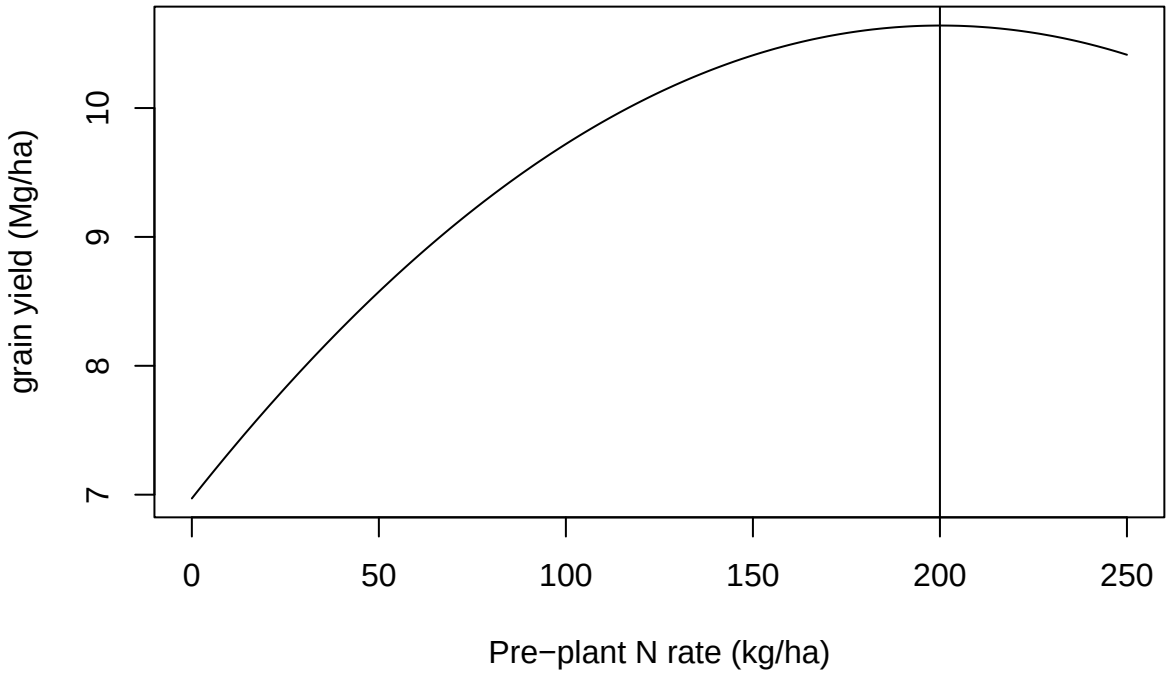
```
##       (Intercept)      N_level_kgha I(N_level_kgha^2)
##      6.970994e+00      3.664963e-02     -9.152059e-05
```

```
xvals.lme.1 <- seq(0,250, by = 1)

yvals.lme.1 <- (lme.1$coefficients$fixed[1]
                + lme.1$coefficients$fixed[2]*seq(0,250, by = 1)
                + lme.1$coefficients$fixed[3]*(seq(0,250, by = 1)^2))
```

```
plot(yvals.lme.1 ~ xvals.lme.1, type = "l",
     ylab = "grain yield (Mg/ha)",
     xlab = "Pre-plant N rate (kg/ha)")

lme.1.vertex <- xvals.lme.1[which(yvals.lme.1 == max(yvals.lme.1))]

print(lme.1.vertex)
```

```
## [1] 200
```

```
abline(v = xvals.lme.1[which(yvals.lme.1 == max(yvals.lme.1))], col = "black")
```



Results for Paper

```
opt_N_rates_data <- paper2_data_OR_S1 %>%
  filter(N_level_kgha >= 165 & N_level_kgha <= 224) %>%
  dplyr::select(site_year, exp_plot_number , N_level_kgha , TopDress_kgha , gs_NDVI_Sufficiency_Index ,

min_gs_ndvi_opt_N <- min(opt_N_rates_data$gs_NDVI_Sufficiency_Index)

round(print(min_gs_ndvi_opt_N) , digits = 2)
```

```
## [1] 0.7412587
```

```
## [1] 0.74
```

```
max_gs_ndvi_opt_N <- max(opt_N_rates_data$gs_NDVI_Sufficiency_Index)

round(print(max_gs_ndvi_opt_N) , digits = 2)
```

```
## [1] 1.047673
```

```
## [1] 1.05
```

```
min_uas_NDRE_opt_N <- min(opt_N_rates_data$uas_NDRE_Sufficiency_Index)

round(print(min_uas_NDRE_opt_N) , digits = 2)
```

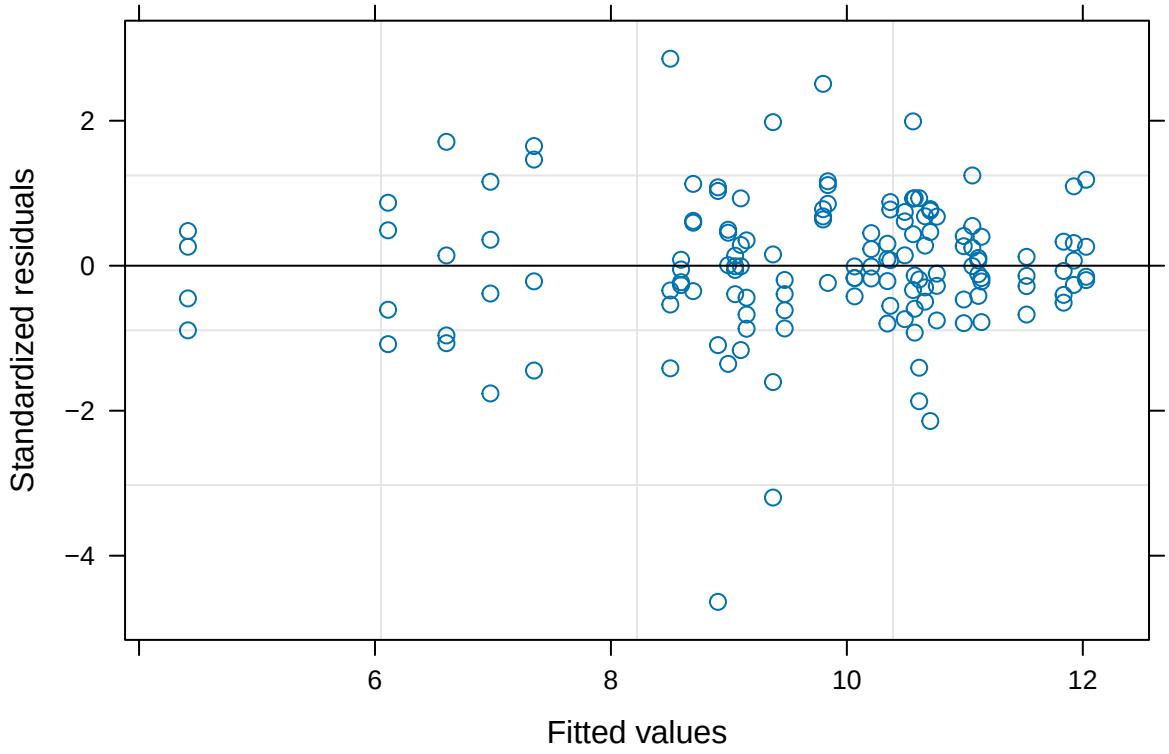
```
## [1] 0.7805444
## [1] 0.78
```

```
max_uas_NDRE_opt_N <- max(opt_N_rates_data$uas_NDRE_Sufficiency_Index)

round(print(max_uas_NDRE_opt_N) , digits = 2)
```

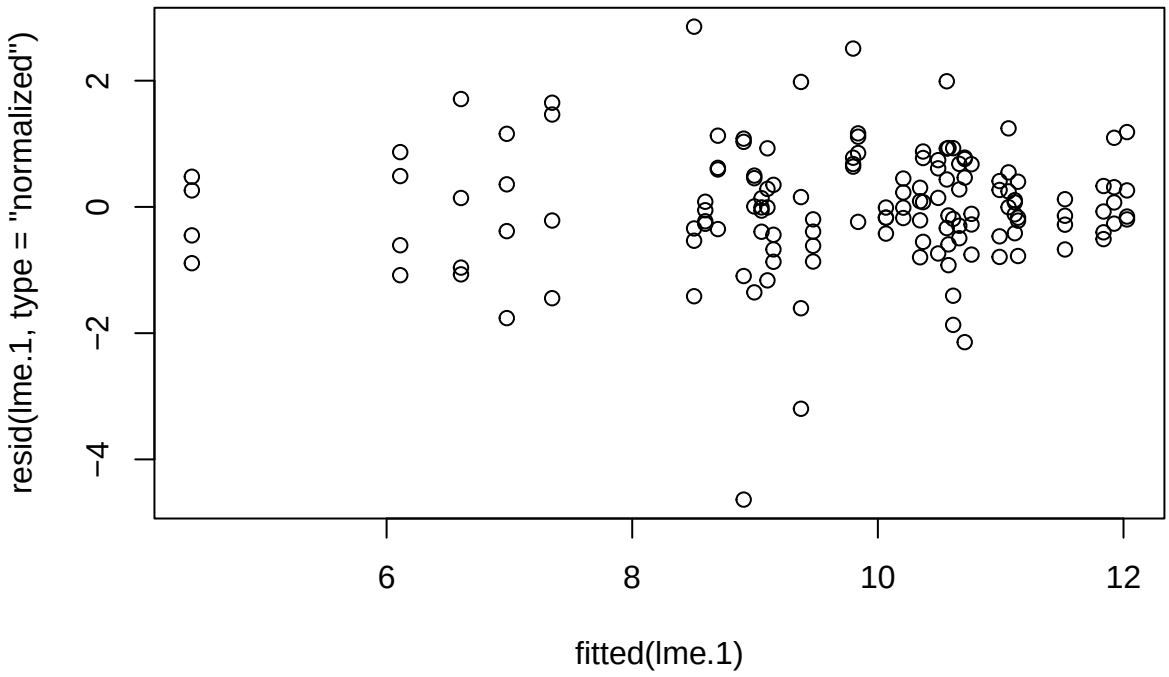
```
## [1] 1.011228
## [1] 1.01
```

diagnostics

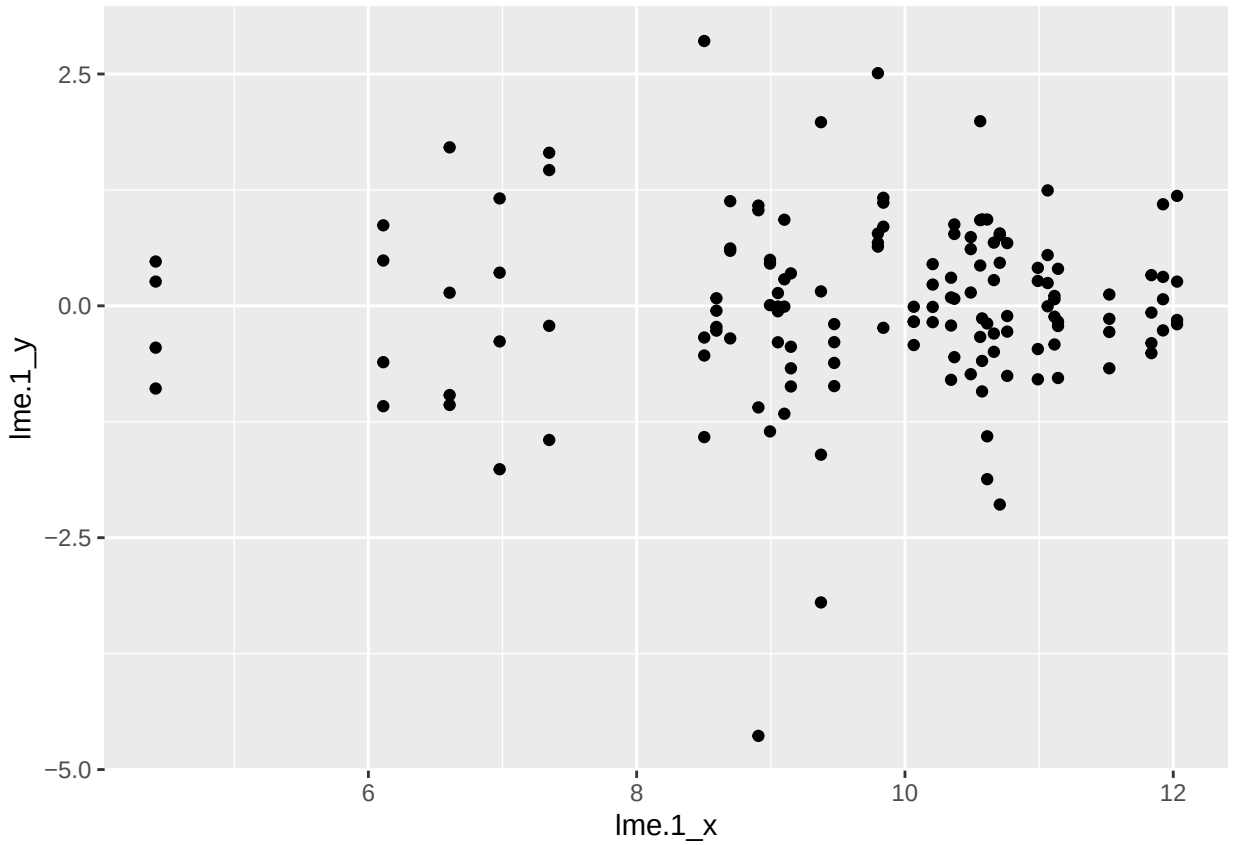
```
plot (lme.1)
```



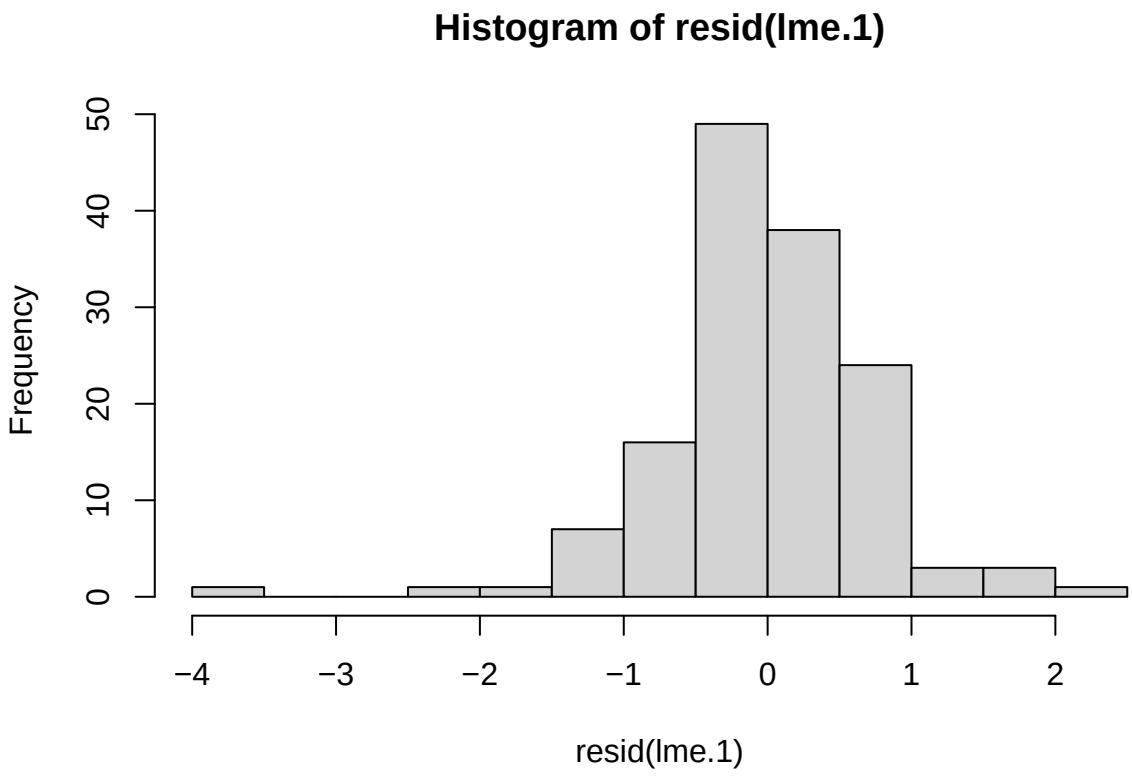

```
plot(resid(lme.1, type = "normalized") ~fitted(lme.1))
```



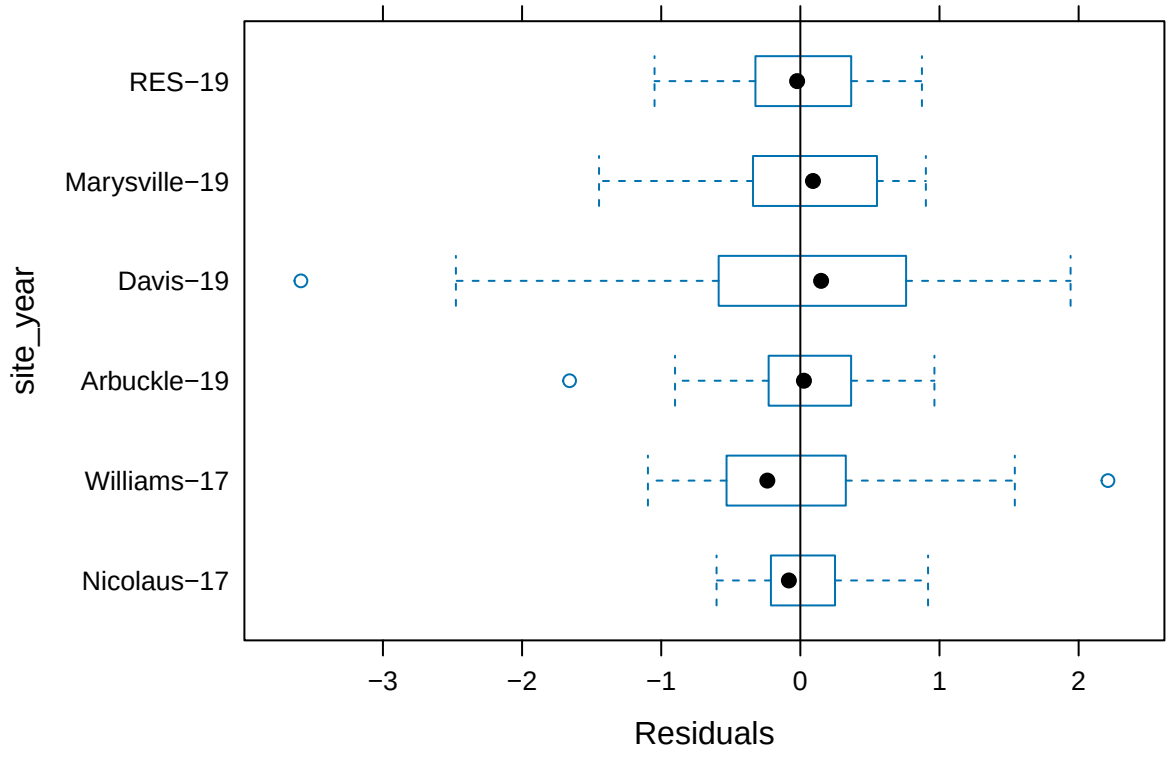

```
lme.1_y <- resid(lme.1, type = "normalized")
lme.1_x <- fitted(lme.1)

lme.1resid_data <- data.frame(lme.1_x , lme.1_y)

ggplot( data = lme.1resid_data , aes( x = lme.1_x , y = lme.1_y)) +
  geom_point(mapping = aes(lme.1_x , lme.1_y) , data = lme.1resid_data)
```



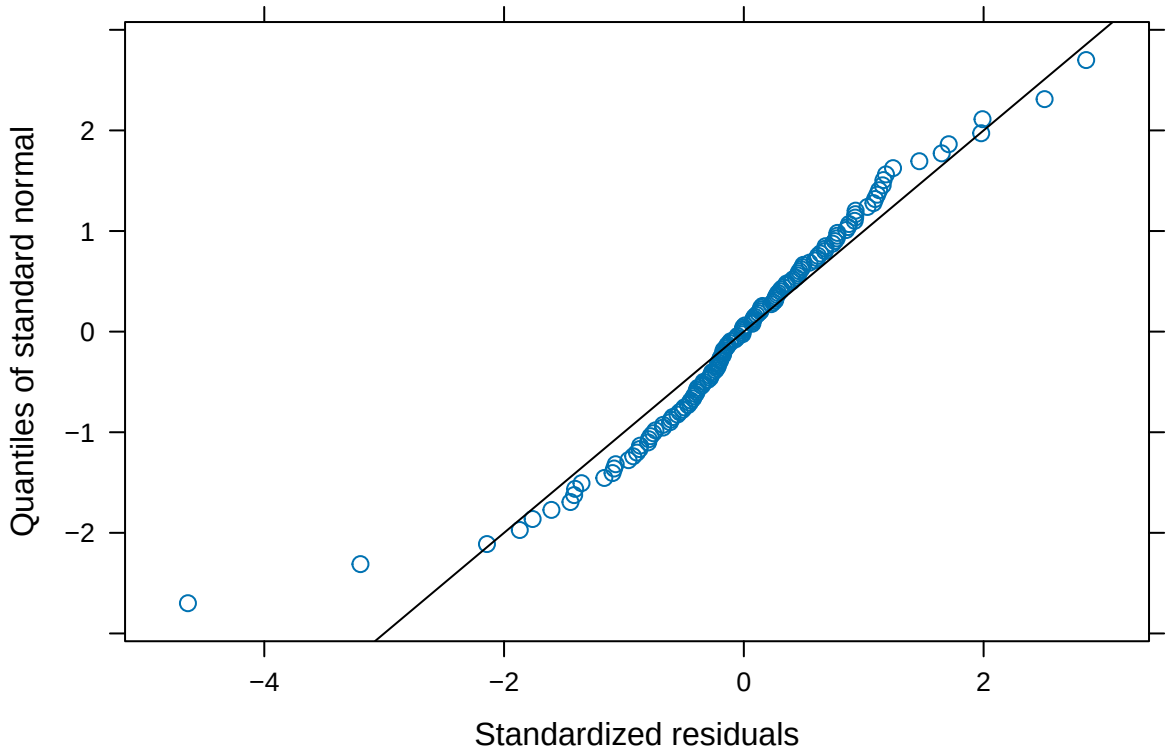

```
hist(resid(lme.1))
```



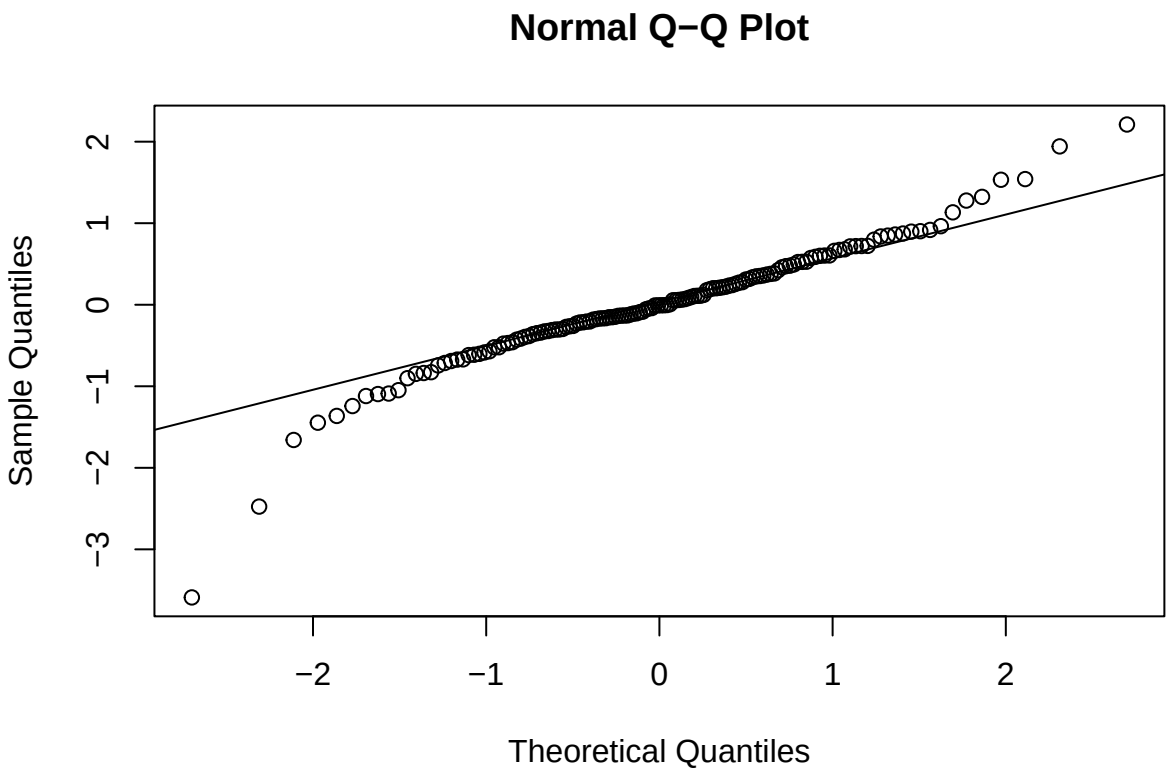
```
plot(lme.1, site_year ~ resid(.), abline = 0)
```



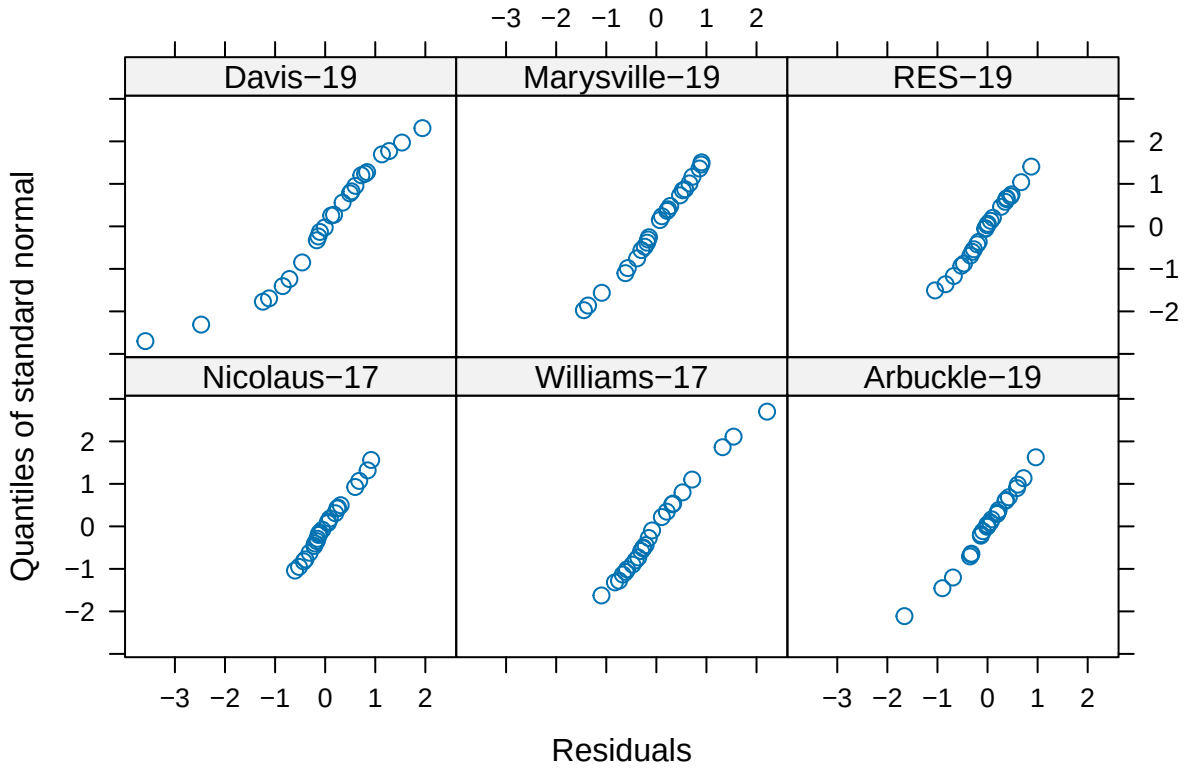

```
qqnorm(lme.1, abline = c(0,1) )
```



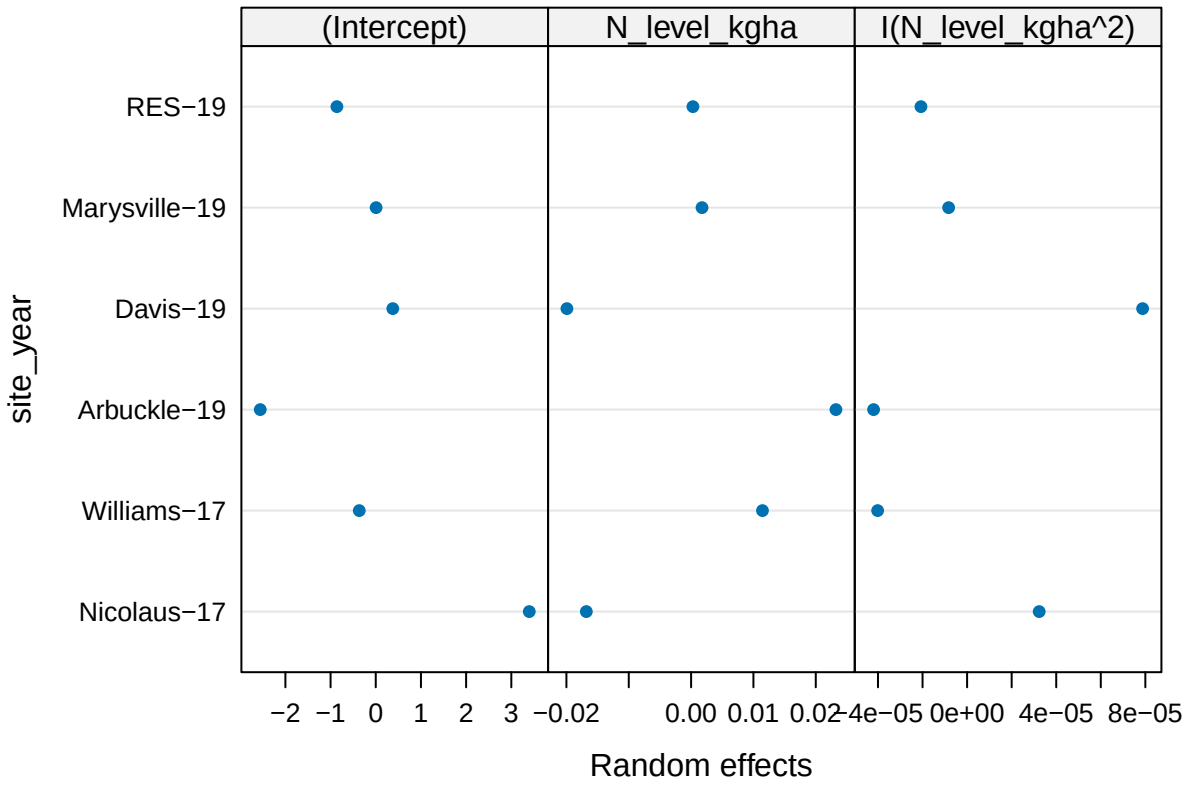

```
qqnorm(resid(lme.1))
qqline(resid(lme.1))
```



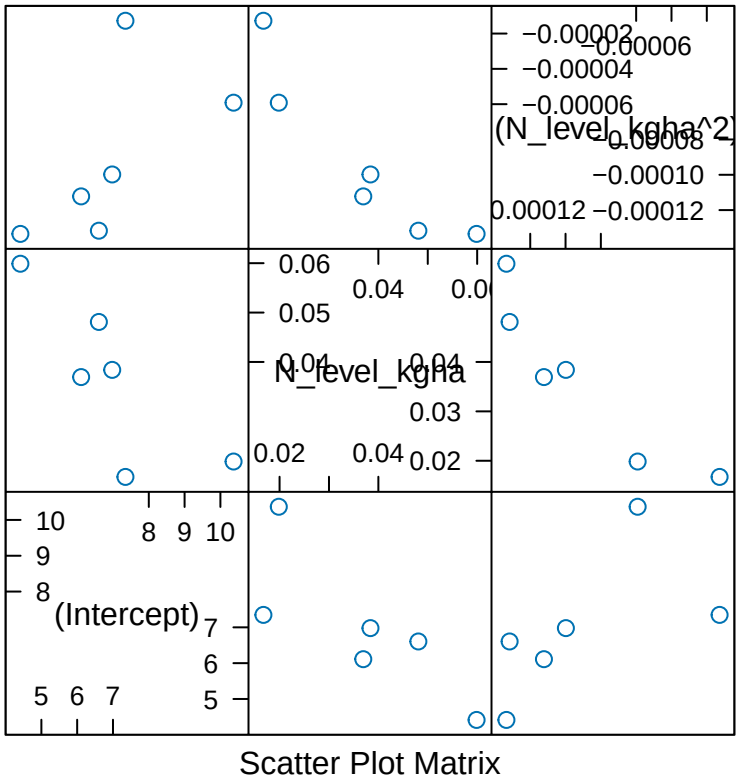

```
qqnorm(lme.1 , ~resid(.) | site_year)
```



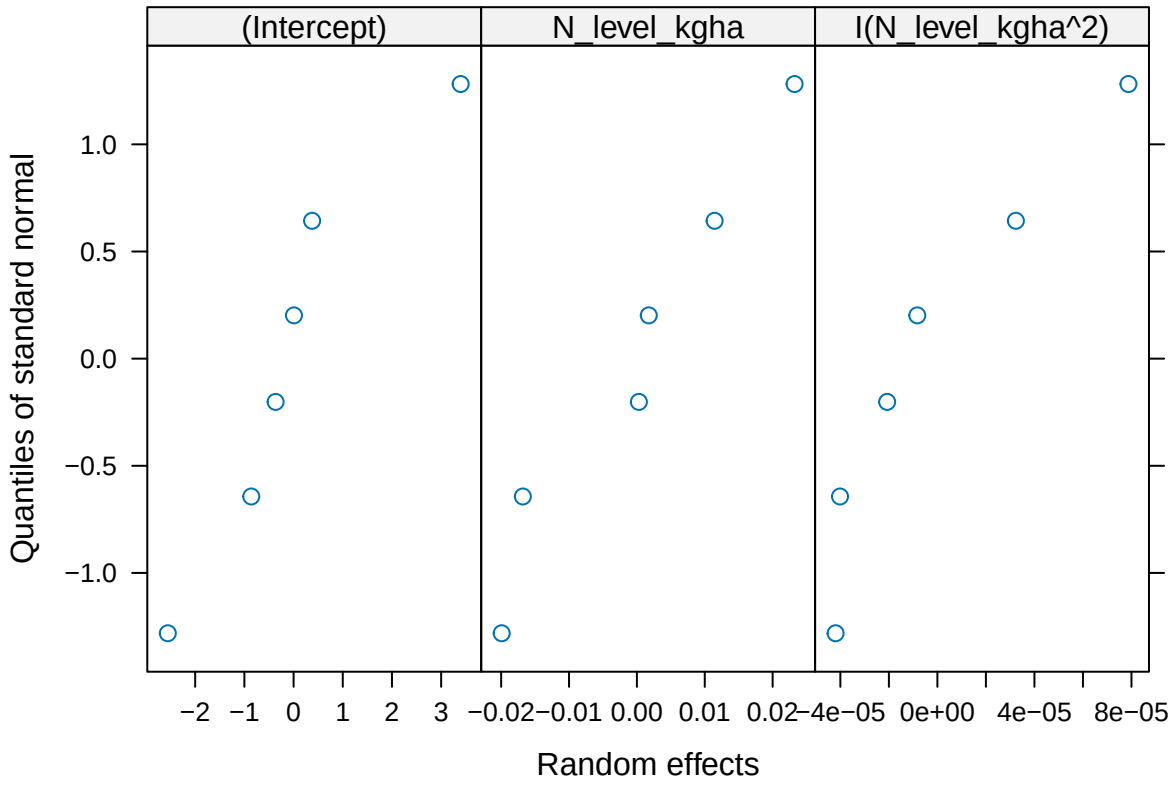

```
plot(ranef(lme.1))
```



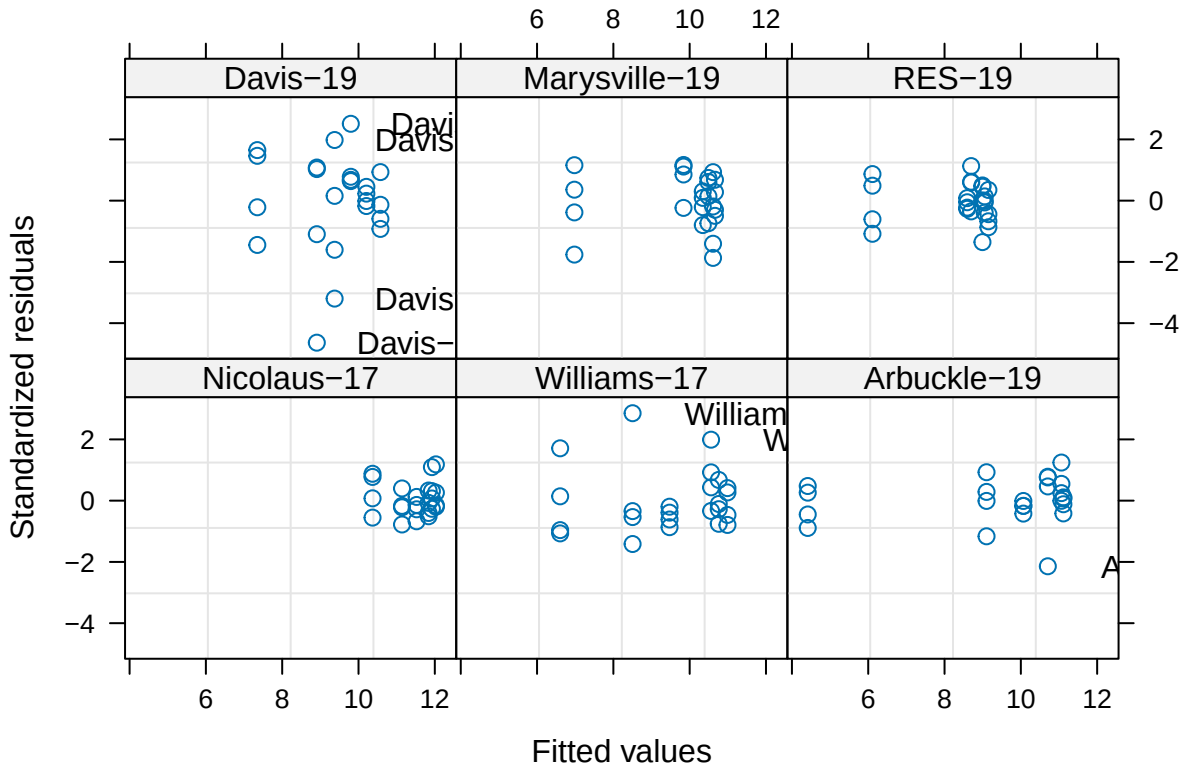

```
pairs(lme.1 , id = 0.1)
```



```
qqnorm(lme.1 , ~ranef(.))
```




```
plot( lme.1, resid(., type = "p") ~ fitted(.) | site_year,
      id = 0.05, adj = -0.3 )
```



emmeans

```
mylist <- list(N_level_kgha = seq(0 , 250 , by = 1) ) #makes the list

yield_emmeans <- as.data.frame(summary(emmeans(lme.1 , ~ N_level_kgha , at = mylist ))) #calculates the

yield_emmeans <- yield_emmeans %>%
  rename(GrainYield_Mgha = emmean ) #renames the grain yield column
```

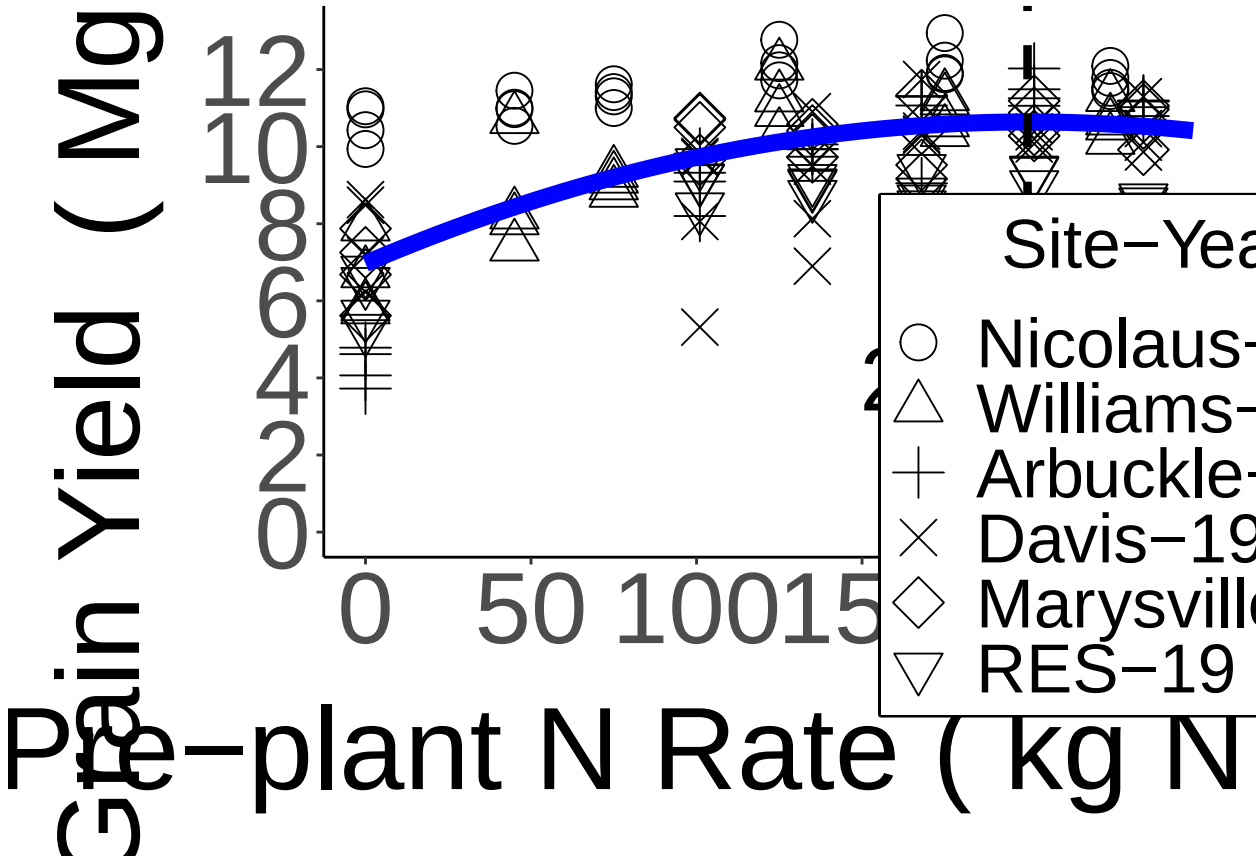
plot

```
FigS2 <- ggplot(data = paper2_data_OR_S1 , aes ( x = N_level_kgha , y = GrainYield_Mgha)) +
  geom_point(data = paper2_data_OR_S1 , aes ( x = N_level_kgha , y = GrainYield_Mgha , shape = site_yea
  geom_line(data = yield_emmeans , aes( x = N_level_kgha , y = GrainYield_Mgha) , size = 3 , color = "bl
  coord_cartesian(ylim = c(0 , 13) , xlim = c(0 , 252)) +
  scale_x_continuous(breaks = seq(0 , 250 , by = 50)) +
  scale_y_continuous(breaks = seq(0 , 13 , by = 2)) +
  theme_classic() +
  labs(x = "Pre-plant N Rate ( kg N ha"^-1~")" , y = "Grain Yield  ( Mg ha"^-1~")" , shape = "Site-Year
  theme(axis.title = element_text(size = 44),
        axis.text = element_text(size = 38),
        legend.text = element_text(size = 26),
        legend.title = element_text(size = 26 , hjust = 0.5),
        legend.box.background = element_rect(size = 1)
        ) +
  annotate("text" , x = (lme.1.vertex - 5) , y = 4 , label = "200" , size = 12 , hjust = 1) +
  geom_vline(xintercept = lme.1.vertex , linetype = "dashed" , size = 1.5) +
  theme(legend.position = c(0.90 , 0.185)) +
  scale_shape_manual(values = c(1:20))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
FigS2
```

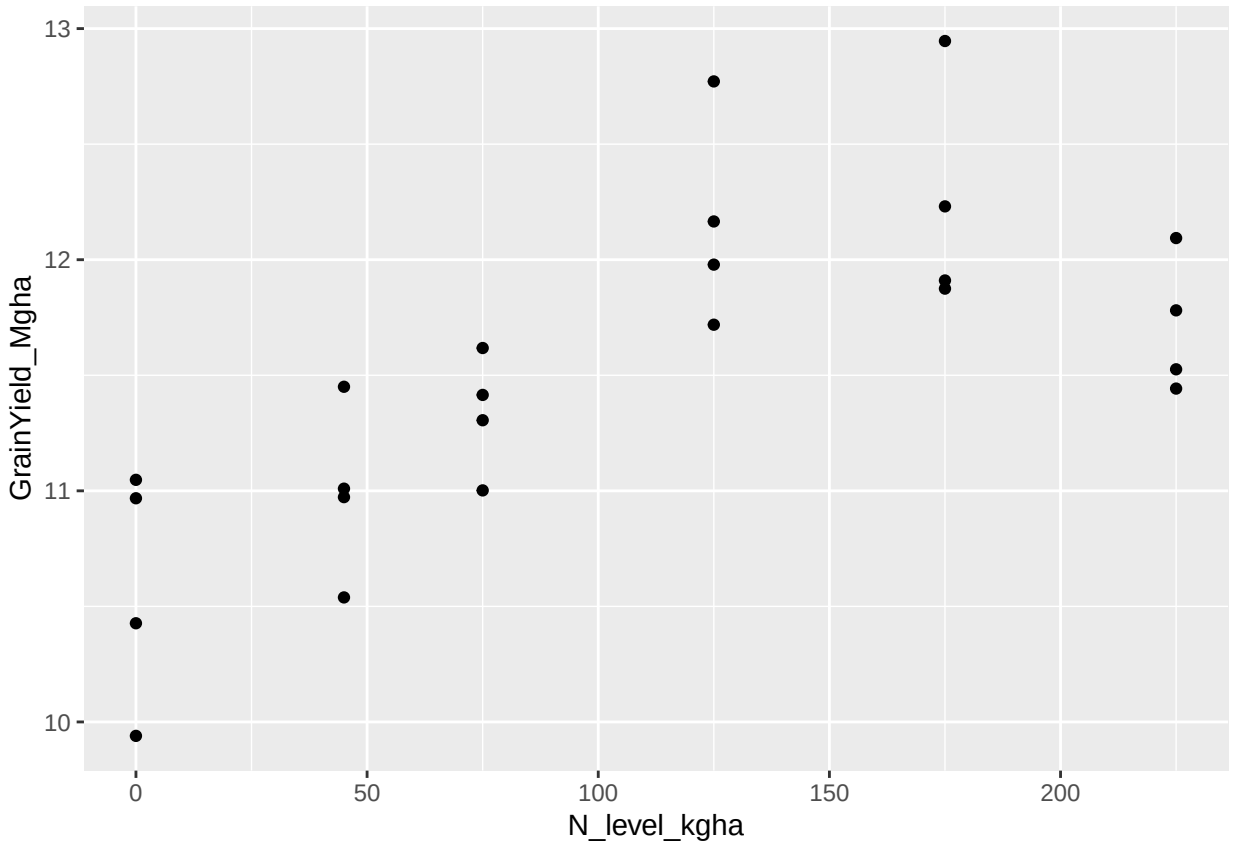


ALL SITES (Fig S2)

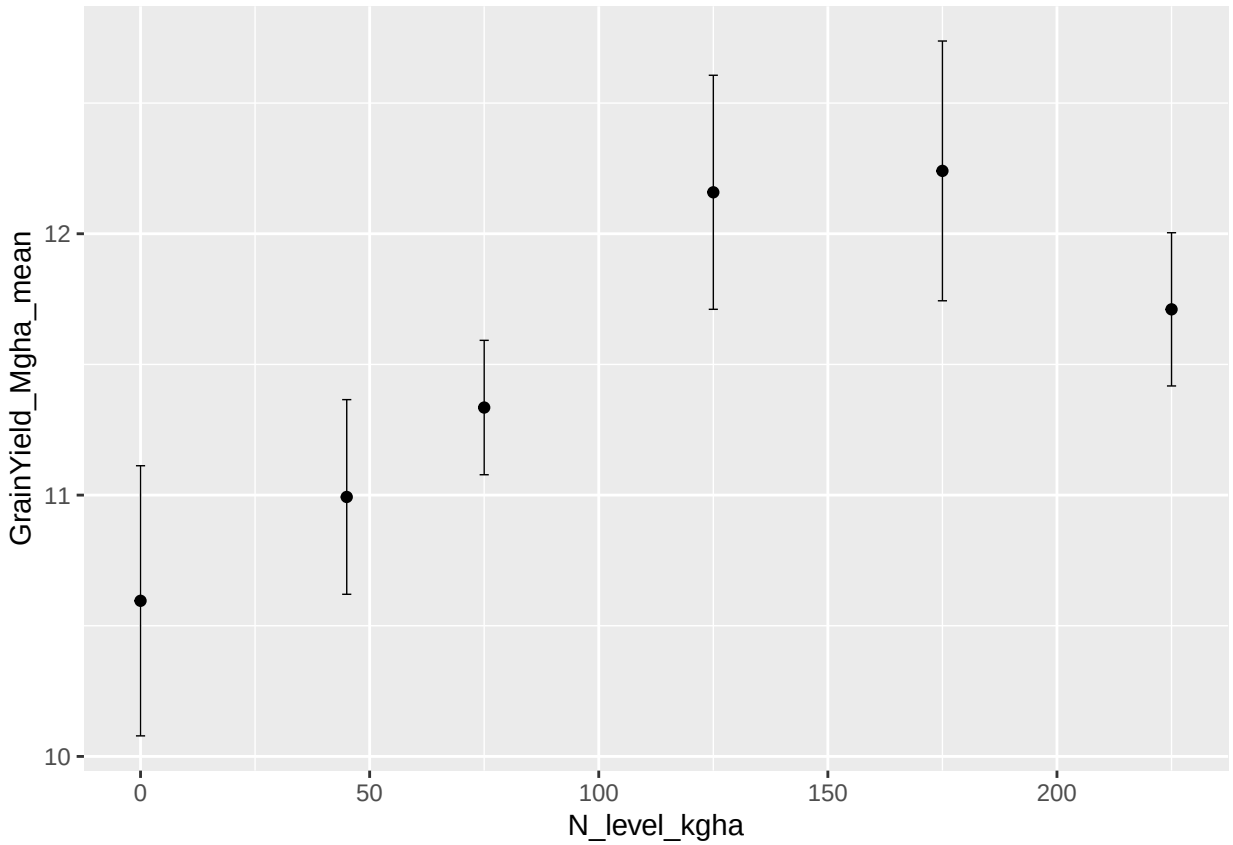
```
ggsave("FIGURES/FigS2.pdf" , FigS2 , width = 17 , height = 12 , dpi = 10000)
```

NICOLAUS-17

```
fig1_data_nic17 <- paper2_data_OR_S1 %>%
  filter(site_year == "Nicolaus-17") #filters to nic 17 site

ggplot(fig1_data_nic17 , aes(x = N_level_kgha , y = GrainYield_Mgha)) +
  geom_point() #creates a plot
```




```
fig1_data_nic17_mean <- fig1_data_nic17 %>%
  group_by(N_level_kgha) %>%
  summarise(GrainYield_Mgha_mean = mean(GrainYield_Mgha) , GrainYield_Mgha_sd = sd(GrainYield_Mgha)) %>%
  mutate(site_year = factor("Nicolaus-17")) %>%
  ungroup()

ggplot(fig1_data_nic17_mean , aes(x = N_level_kgha , y = GrainYield_Mgha_mean)) +
  geom_point() +
  geom_errorbar(data = fig1_data_nic17_mean , aes( y = GrainYield_Mgha_mean , ymax = (GrainYield_Mgha_m
```



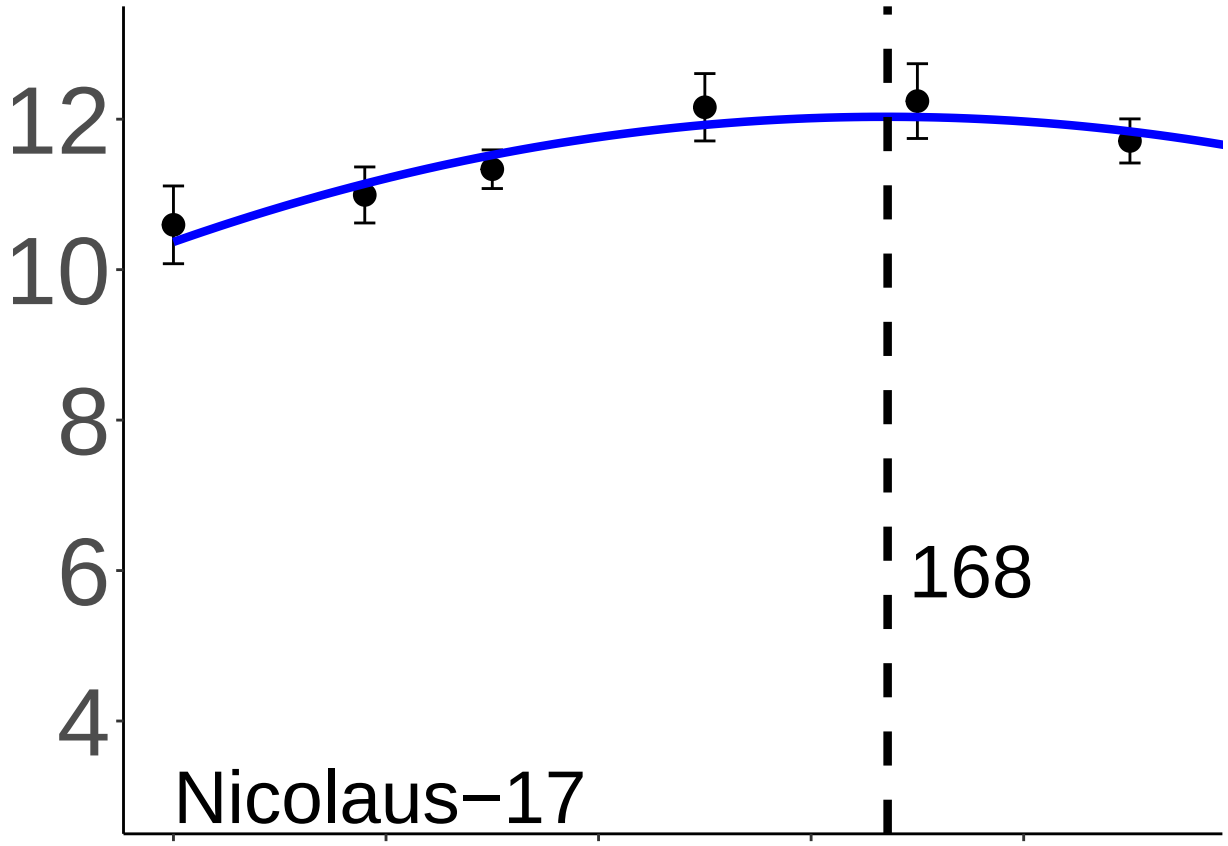
data

```
xvals.site.nic17 <- seq(0 , 250 , by = 1 )

yvals.site.nic17 <- (coefficients(lme.1)[1,1]
                     + coefficients(lme.1)[1,2]*seq(0 , 250, by = 1)
                     + coefficients(lme.1)[1,3]*(seq(0 , 250, by = 1)^2))

lme.1.nic17.vertex <- xvals.site.nic17[which(yvals.site.nic17 == max(yvals.site.nic17))]

lm.nic17.df <- data.frame(xvals.site.nic17 , yvals.site.nic17) #creates dataframe with N rate and fitte
```

plot

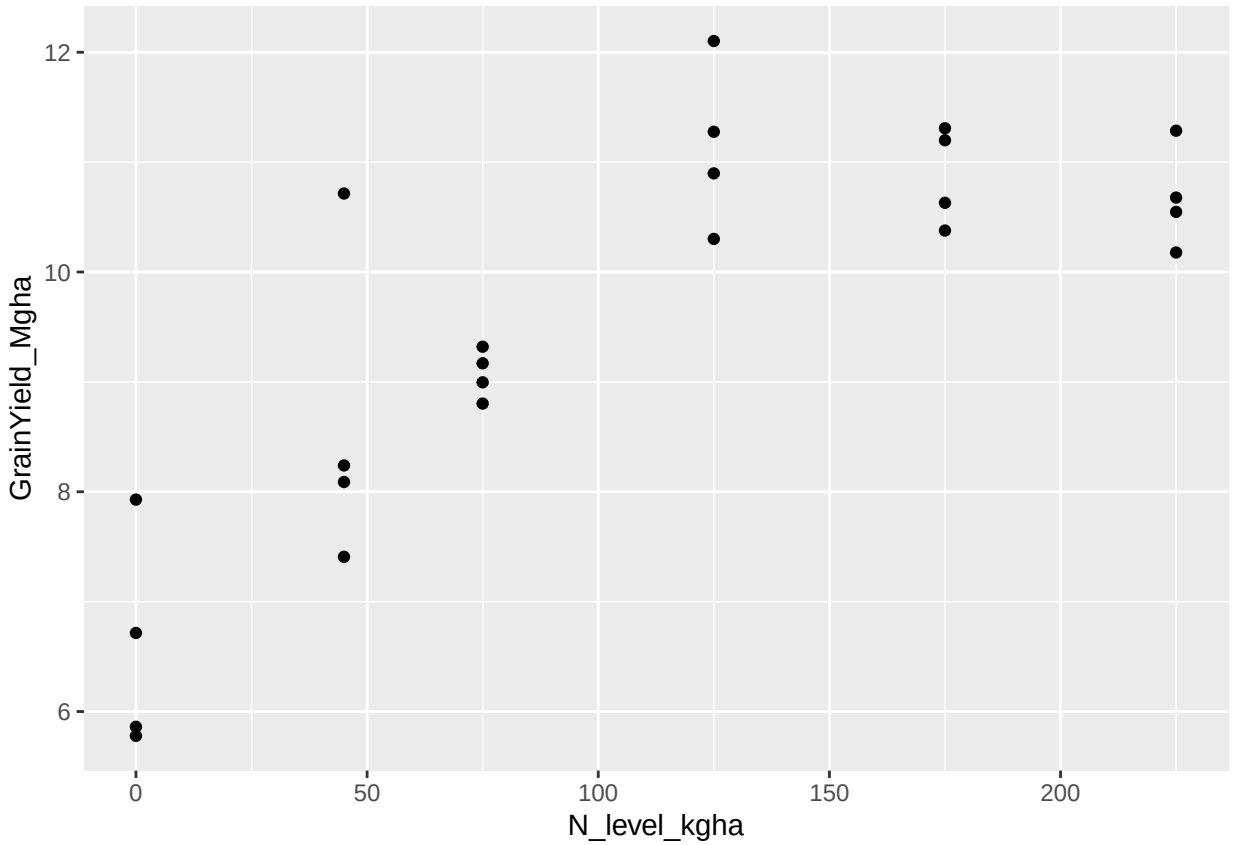
```
nic17_plot <- ggplot(fig1_data_nic17_mean ,
                     aes( x = N_level_kgha ,
                          y = GrainYield_Mgha_mean)) +
  geom_point(aes( x = N_level_kgha ,
                              y = GrainYield_Mgha_mean) , size = 3.5) +
  geom_errorbar(data = fig1_data_nic17_mean , aes( y = GrainYield_Mgha_mean ,
                                                   ymax = (GrainYield_Mgha_mean + GrainYield_Mgha_sd) ,
                                                   ymin = (GrainYield_Mgha_mean - GrainYield_Mgha_sd) ) ,
                alpha = 1 , width = 5 , size = .5 , position = position_dodge(.6) , show.legend = F) +
  geom_line(data = lm.nic17.df , aes(x = xvals.site.nic17 ,
                                     y = yvals.site.nic17) ,
            color = "blue" , size = 1.5) +
```

```
  coord_cartesian(xlim = c(0 , 235) ,
                  ylim = c(3 , 13)) +
  scale_y_continuous(breaks = seq(2 , 13 , by = 2 )) +
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.y = element_text(size = 36) ,
        axis.text.x = element_blank(),
        legend.position = "none" ) +
  annotate("text" , x = 0 , y = 3 , label = "Nicolaus-17" , size = 10 , hjust = 0) +
  annotate("text" , x = (lme.1.nic17.vertex + 5) , y = 6 , label = "168" , size = 10 , hjust = 0) +
  geom_vline(xintercept = lme.1.nic17.vertex , linetype = "dashed" , size = 1.5)

nic17_plot
```

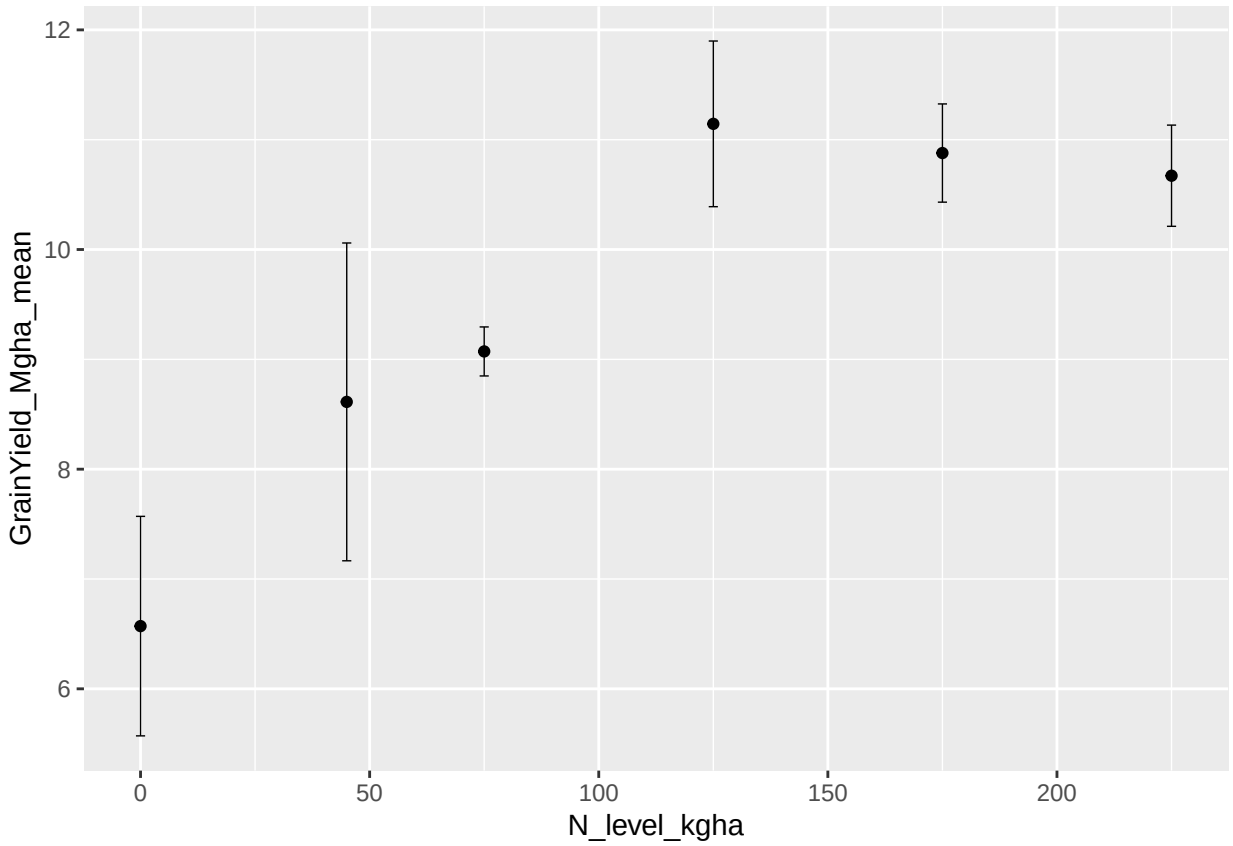


WILLIAMS-17

```
fig1_data_wil17 <- paper2_data_OR_S1 %>%
  filter(site_year == "Williams-17") #filters to wil 17 site

ggplot(fig1_data_wil17 , aes(x = N_level_kgha , y = GrainYield_Mgha)) +
  geom_point() #creates a plot
```




```
fig1_data_wil17_mean <- fig1_data_wil17 %>%
  group_by(N_level_kgha) %>%
  summarise(GrainYield_Mgha_mean = mean(GrainYield_Mgha) , GrainYield_Mgha_sd = sd(GrainYield_Mgha)) %>%
  mutate(site_year = factor("Williams-17")) %>%
  ungroup()

ggplot(fig1_data_wil17_mean , aes(x = N_level_kgha , y = GrainYield_Mgha_mean)) +
  geom_point() +
  geom_errorbar(data = fig1_data_wil17_mean , aes( y = GrainYield_Mgha_mean , ymax = (GrainYield_Mgha_m
```



data

```
xvals.site.wil17 <- seq(0 , 250 , by = 1 )

yvals.site.wil17 <- (coefficients(lme.1)[2,1]
                     + coefficients(lme.1)[2,2]*seq(0 , 250, by = 1)
                     + coefficients(lme.1)[2,3]*(seq(0 , 250, by = 1)^2))

lme.1.wil17.vertex <- xvals.site.wil17[which(yvals.site.wil17 == max(yvals.site.wil17))]

print(lme.1.wil17.vertex)
```

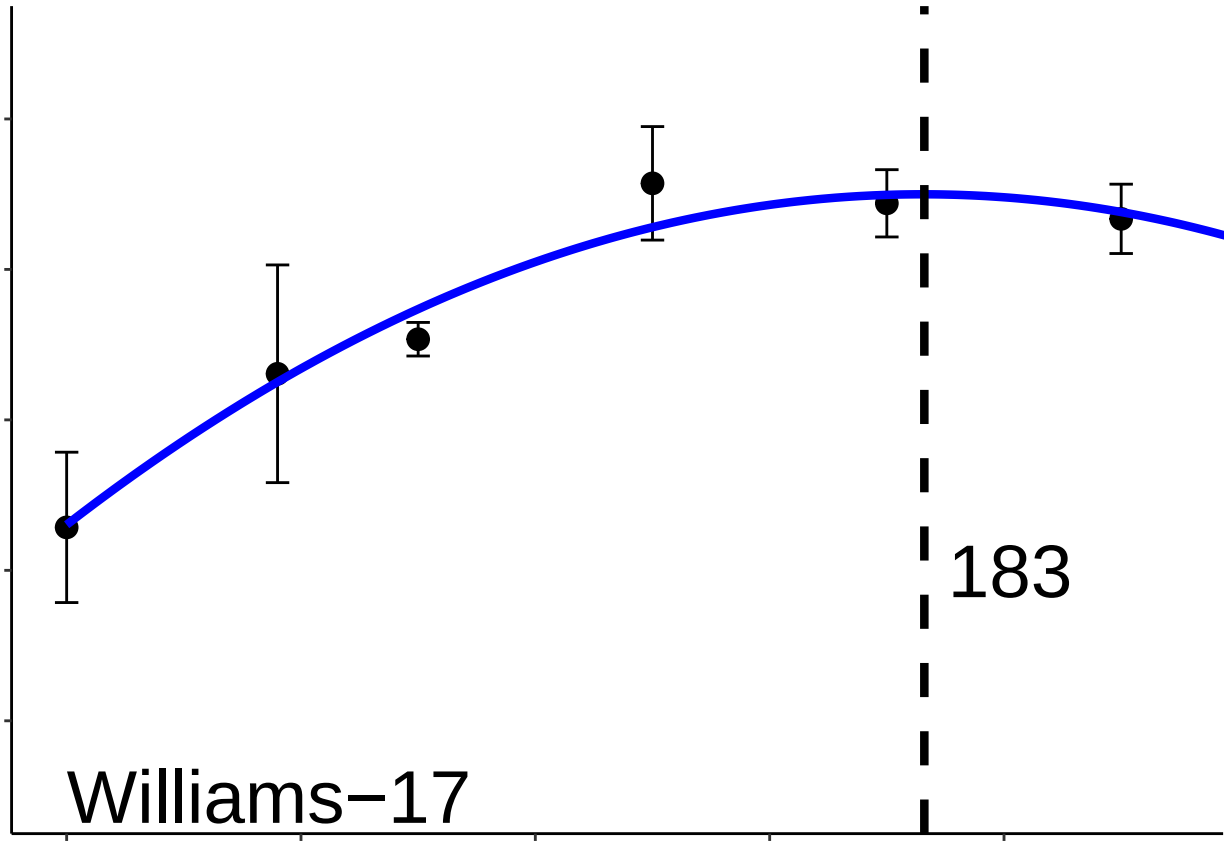
```
## [1] 183
```

```
lm.wil17.df <- data.frame(xvals.site.wil17 , yvals.site.wil17) #creates dataframe with N rate and fitte
```

plot

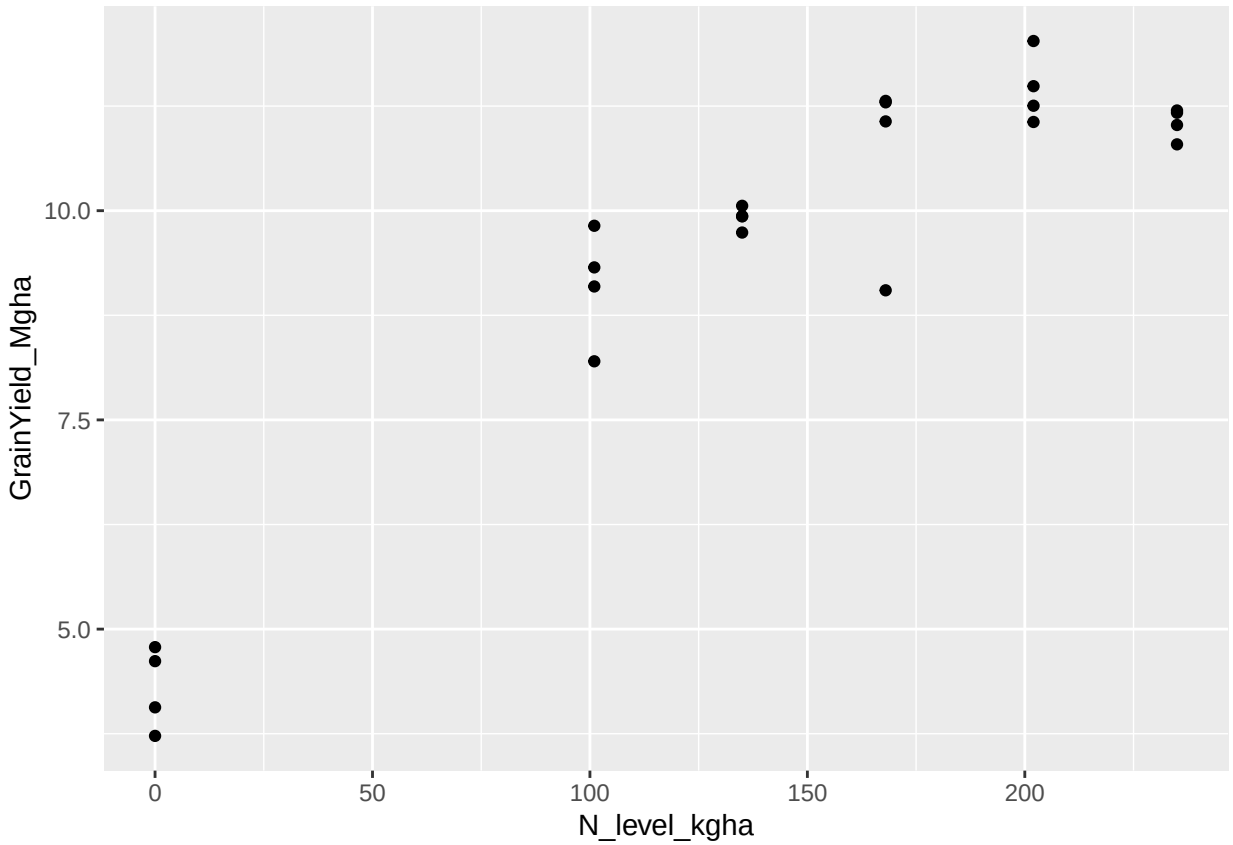
```
wil17_plot <- ggplot(fig1_data_wil17_mean ,
                     aes( x = N_level_kgha ,
                          y = GrainYield_Mgha_mean)) +
  geom_point(aes( x = N_level_kgha ,
                              y = GrainYield_Mgha_mean) , size = 3.5) +
  geom_errorbar(data = fig1_data_wil17_mean , aes( y = GrainYield_Mgha_mean ,
                                                   ymax = (GrainYield_Mgha_mean + GrainYield_Mgha_sd) ,
                                                   ymin = (GrainYield_Mgha_mean - GrainYield_Mgha_sd) ) ,
                alpha = 1 , width = 5 , size = .5 , position = position_dodge(.6) , show.legend = F) +
```

```
  geom_line(data = lm.wil17.df , aes(x = xvals.site.wil17 ,
                                     y = yvals.site.wil17) ,
            color = "blue" , size = 1.5) +
  coord_cartesian(xlim = c(0 , 235) , ylim = c(3 , 13)) +
  scale_y_continuous(breaks = seq(2 , 13 , by = 2 )) +
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.y = element_blank() ,
        axis.text.x = element_blank(),
        legend.position = "none" ) +
  annotate("text" , x = 0 , y = 3 , label = "Williams-17" , size = 10 , hjust = 0) +
  annotate("text" , x = (lme.1.wil17.vertex + 5) , y = 6 , label = "183" , size = 10 , hjust = 0) +
  geom_vline(xintercept = lme.1.wil17.vertex , linetype = "dashed" , size = 1.5)

wil17_plot
```

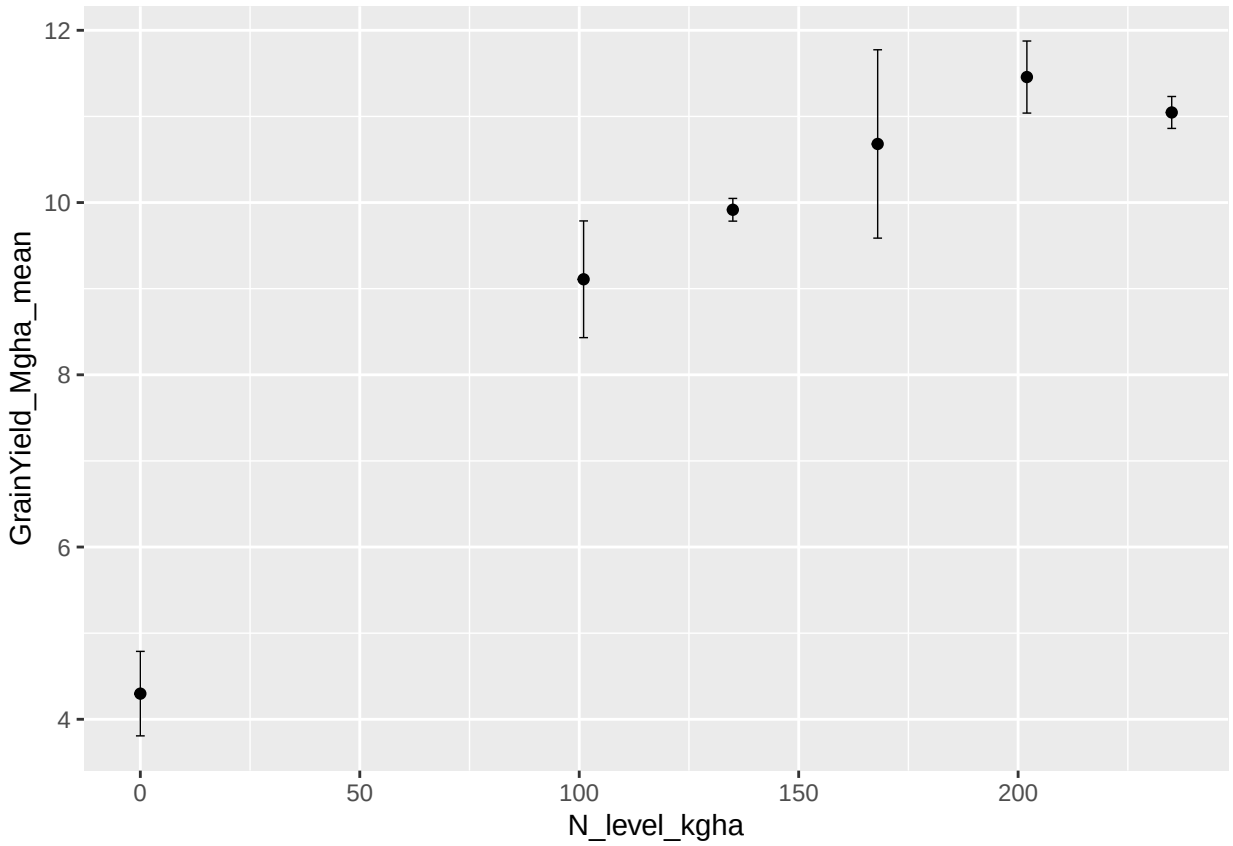


ARBUCKLE-19

```
fig1_data_arb19 <- paper2_data_OR_S1 %>%
  filter(site_year == "Arbuckle-19") #filters to arb 19 site

ggplot(fig1_data_arb19 , aes(x = N_level_kgha , y = GrainYield_Mgha)) +
  geom_point() #creates a plot
```




```
fig1_data_arb19_mean <- fig1_data_arb19 %>%
  group_by(N_level_kgha) %>%
  summarise(GrainYield_Mgha_mean = mean(GrainYield_Mgha) , GrainYield_Mgha_sd = sd(GrainYield_Mgha)) %>%
  mutate(site_year = factor("Arbuckle-19")) %>%
  ungroup()

ggplot(fig1_data_arb19_mean , aes(x = N_level_kgha , y = GrainYield_Mgha_mean)) +
  geom_point() +
  geom_errorbar(data = fig1_data_arb19_mean , aes( y = GrainYield_Mgha_mean , ymax = (GrainYield_Mgha_m
```



data

```
xvals.site.arb19 <- seq(0 , 250 , by = 1 )

yvals.site.arb19 <- (coefficients(lme.1)[3,1]
                     + coefficients(lme.1)[3,2]*seq(0 , 250, by = 1)
                     + coefficients(lme.1)[3,3]*(seq(0 , 250, by = 1)^2))

lme.1.arb19.vertex <- xvals.site.arb19[which(yvals.site.arb19 == max(yvals.site.arb19))]

print(lme.1.arb19.vertex)
```

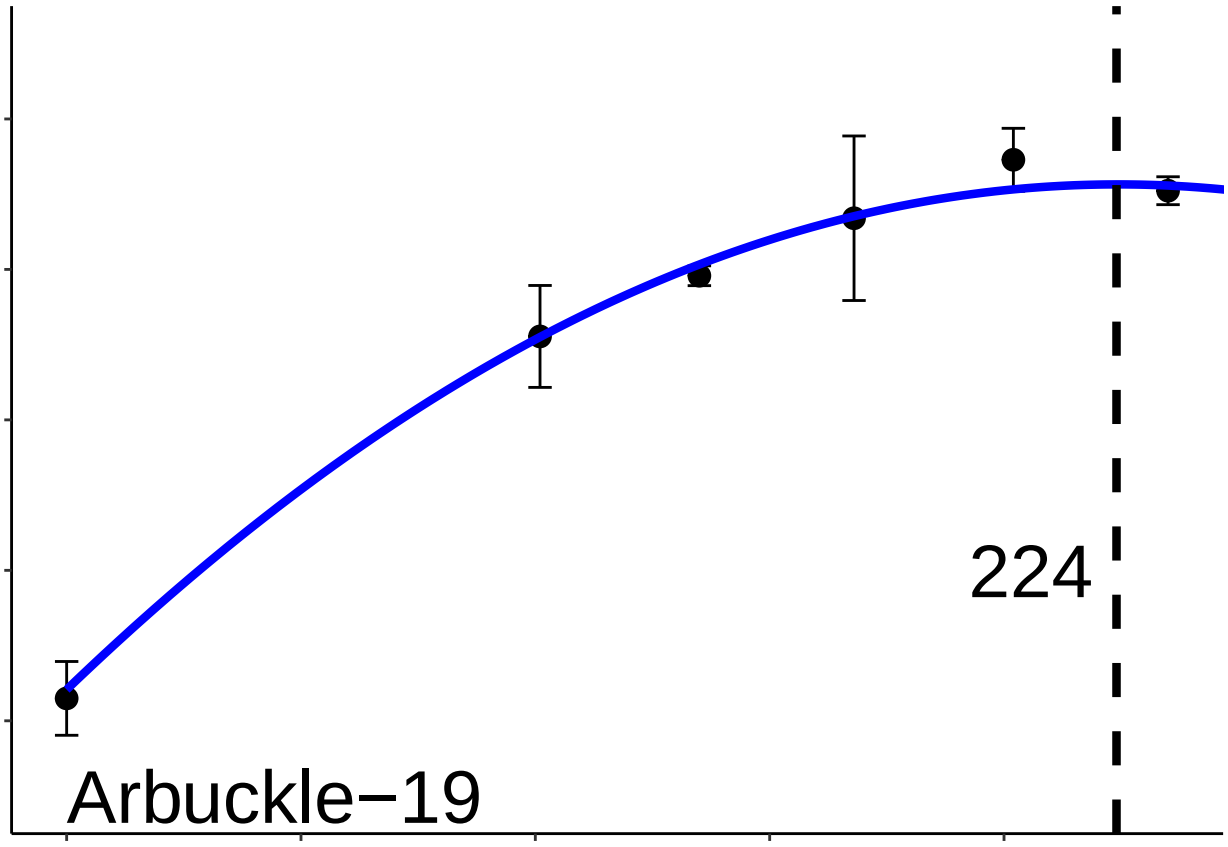
```
## [1] 224
```

```
lm.arb19.df <- data.frame(xvals.site.arb19 , yvals.site.arb19) #creates dataframe with N rate and fitte
```

plot

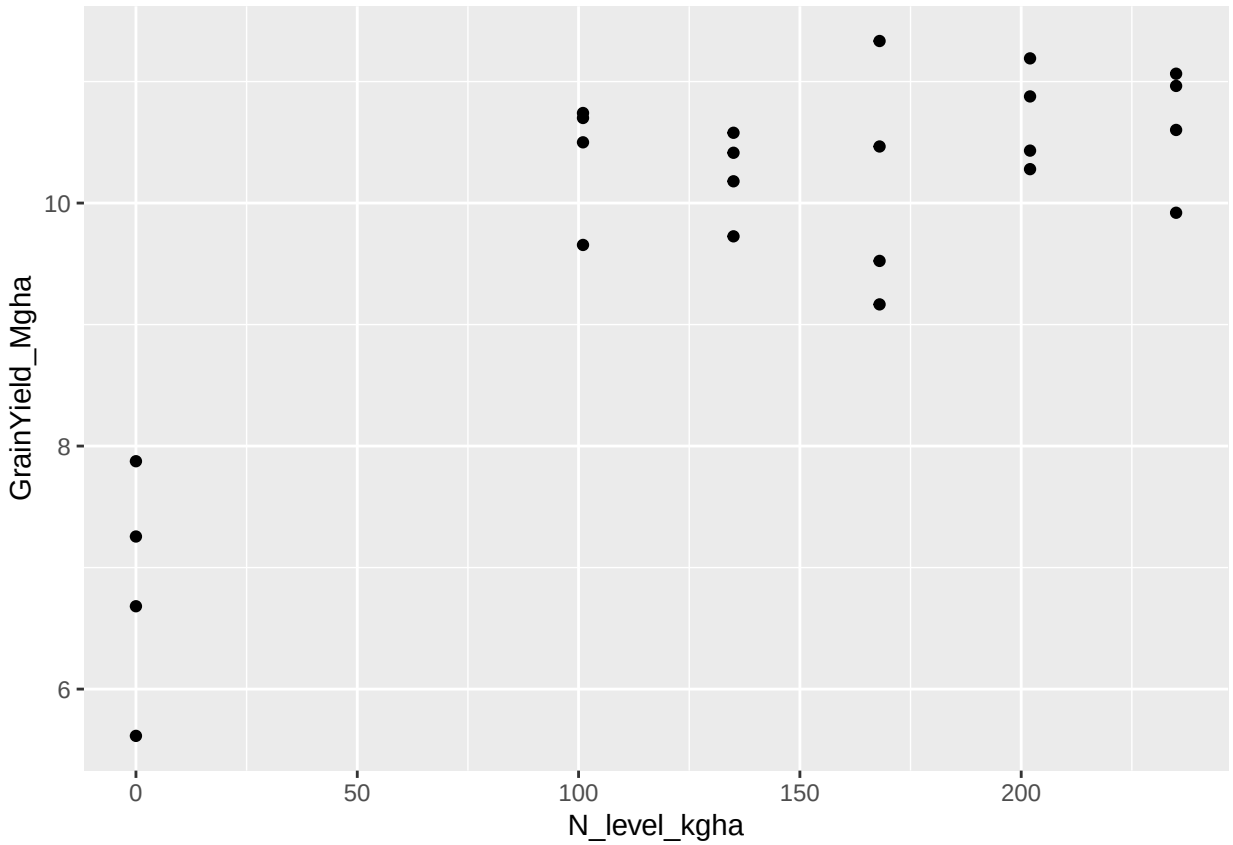
```
arb19_plot <- ggplot(fig1_data_arb19_mean ,
                     aes( x = N_level_kgha ,
                          y = GrainYield_Mgha_mean)) +
  geom_point(aes( x = N_level_kgha ,
                              y = GrainYield_Mgha_mean) , size = 3.5) +
  geom_errorbar(data = fig1_data_arb19_mean , aes( y = GrainYield_Mgha_mean ,
                                                   ymax = (GrainYield_Mgha_mean + GrainYield_Mgha_sd) ,
                                                   ymin = (GrainYield_Mgha_mean - GrainYield_Mgha_sd) ) ,
                alpha = 1 , width = 5 , size = .5 , position = position_dodge(.6) , show.legend = F) +
```

```
  geom_line(data = lm.arb19.df , aes(x = xvals.site.arb19 ,
                                     y = yvals.site.arb19) ,
            color = "blue" , size = 1.5) +
  coord_cartesian(xlim = c(0 , 235) , ylim = c(3 , 13)) +
  scale_y_continuous(breaks = seq(2 , 13 , by = 2 )) +
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.y = element_blank() ,
        axis.text.x = element_blank(),
        legend.position = "none" ) +
  annotate("text" , x = 0 , y = 3 , label = "Arbuckle-19" , size = 10 , hjust = 0) +
  annotate("text" , x = (lme.1.arb19.vertex - 5) , y = 6 , label = "224" , size = 10 , hjust = 1) +
  geom_vline(xintercept = lme.1.arb19.vertex , linetype = "dashed" , size = 1.5)

arb19_plot
```

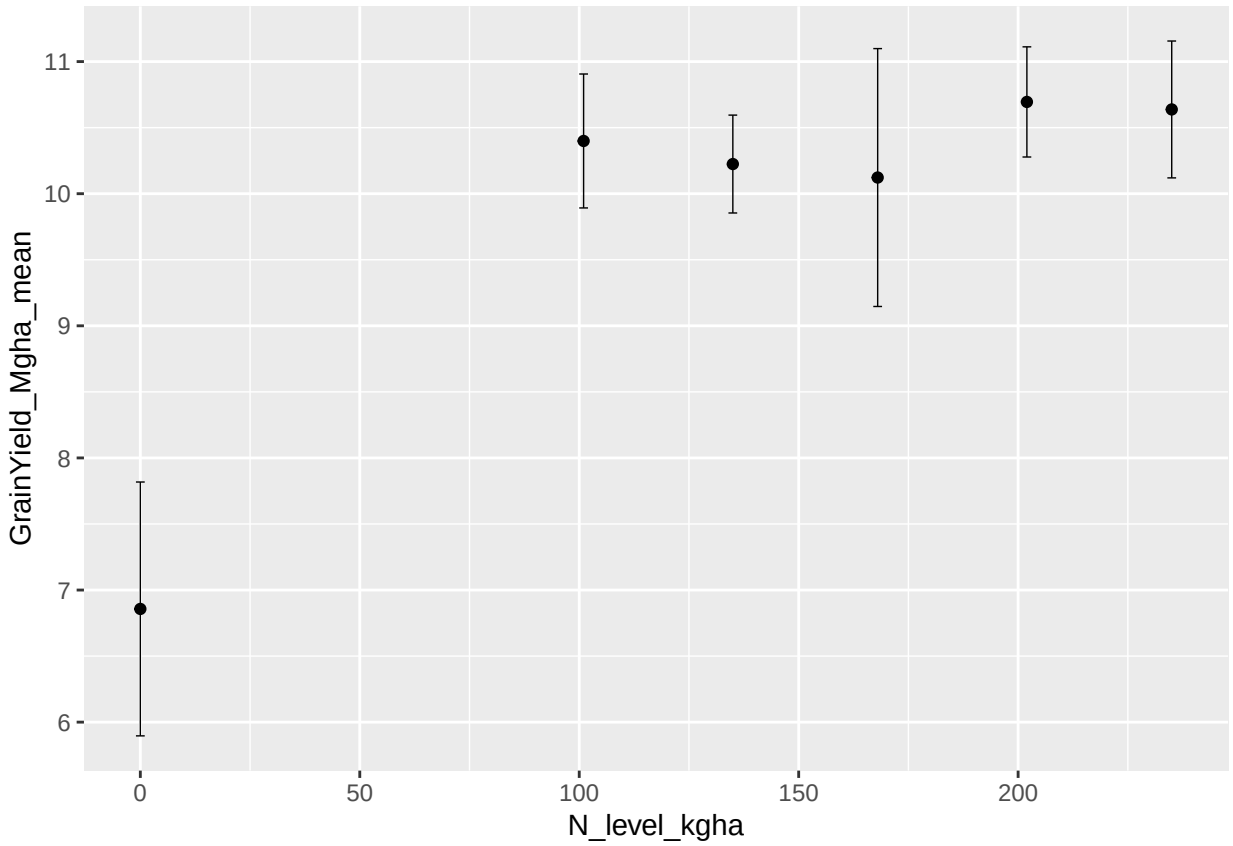


MARYSVILLE-19

```
fig1_data_mry19 <- paper2_data_OR_S1 %>%
  filter(site_year == "Marysville-19") #filters to mry 19 site

ggplot(fig1_data_mry19 , aes(x = N_level_kgha , y = GrainYield_Mgha)) +
  geom_point() #creates a plot
```




```
fig1_data_mry19_mean <- fig1_data_mry19 %>%
  group_by(N_level_kgha) %>%
  summarise(GrainYield_Mgha_mean = mean(GrainYield_Mgha) , GrainYield_Mgha_sd = sd(GrainYield_Mgha)) %>%
  mutate(site_year = factor("Marysville-19")) %>%
  ungroup()

ggplot(fig1_data_mry19_mean , aes(x = N_level_kgha , y = GrainYield_Mgha_mean)) +
  geom_point() +
  geom_errorbar(data = fig1_data_mry19_mean , aes( y = GrainYield_Mgha_mean , ymax = (GrainYield_Mgha_m
```



data

```
xvals.site.mry19 <- seq(0 , 250 , by = 1 )

yvals.site.mry19 <- (coefficients(lme.1)[5,1]
                     + coefficients(lme.1)[5,2]*seq(0 , 250, by = 1)
                     + coefficients(lme.1)[5,3]*(seq(0 , 250, by = 1)^2))

lme.1.mry19.vertex <- xvals.site.mry19[which(yvals.site.mry19 == max(yvals.site.mry19))]

print(lme.1.mry19.vertex)
```

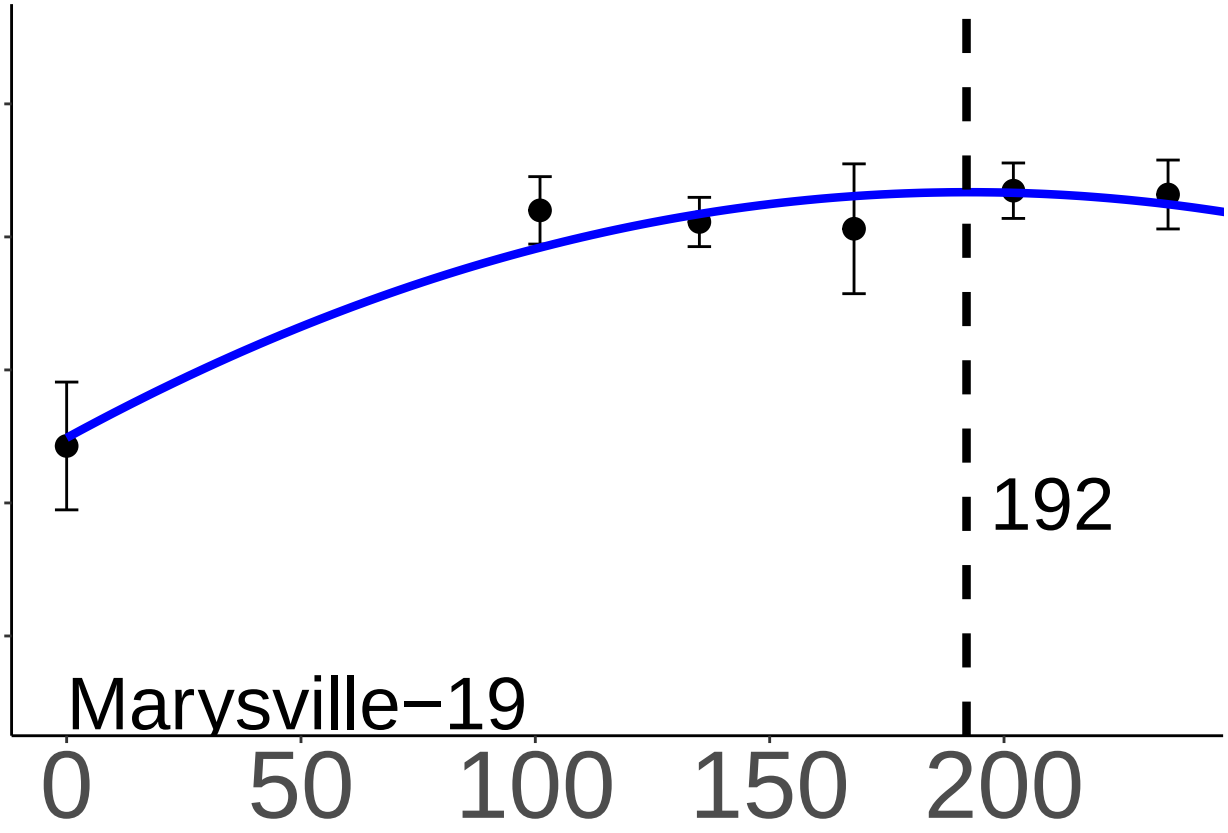
```
## [1] 192
```

```
lm.mry19.df <- data.frame(xvals.site.mry19 , yvals.site.mry19) #creates dataframe with N rate and fitte
```

plot

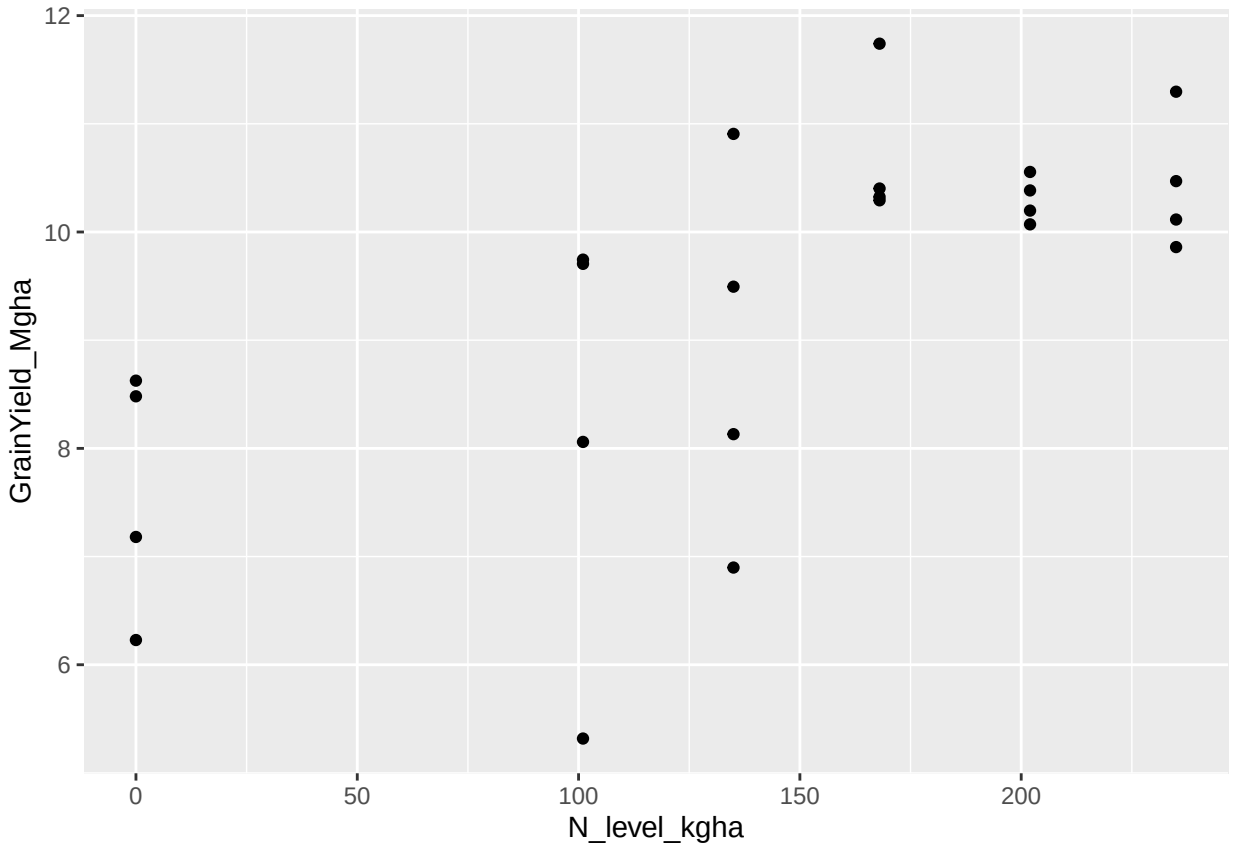
```
mry19_plot <- ggplot(fig1_data_mry19_mean ,
                     aes( x = N_level_kgha ,
                          y = GrainYield_Mgha_mean)) +
  geom_point(aes( x = N_level_kgha ,
                              y = GrainYield_Mgha_mean) , size = 3.5) +
  geom_errorbar(data = fig1_data_mry19_mean , aes( y = GrainYield_Mgha_mean ,
                                                   ymax = (GrainYield_Mgha_mean + GrainYield_Mgha_sd) ,
                                                   ymin = (GrainYield_Mgha_mean - GrainYield_Mgha_sd) ) ,
                alpha = 1 , width = 5 , size = .5 , position = position_dodge(.6) , show.legend = F) +
```

```
  geom_line(data = lm.mry19.df , aes(x = xvals.site.mry19 ,
                                     y = yvals.site.mry19) ,
            color = "blue" , size = 1.5) +
  coord_cartesian(xlim = c(0 , 235) , ylim = c(3 , 13)) +
  scale_y_continuous(breaks = seq(2 , 13 , by = 2 )) +
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.y = element_blank() ,
        axis.text.x = element_text(size = 36),
        legend.position = "none" ) +
  annotate("text" , x = 0 , y = 3 , label = "Marysville-19" , size = 10 , hjust = 0) +
  annotate("text" , x = (lme.1.mry19.vertex + 5) , y = 6 , label = "192" , size = 10 , hjust = 0) +
  geom_vline(xintercept = lme.1.mry19.vertex , linetype = "dashed" , size = 1.5)

mry19_plot
```

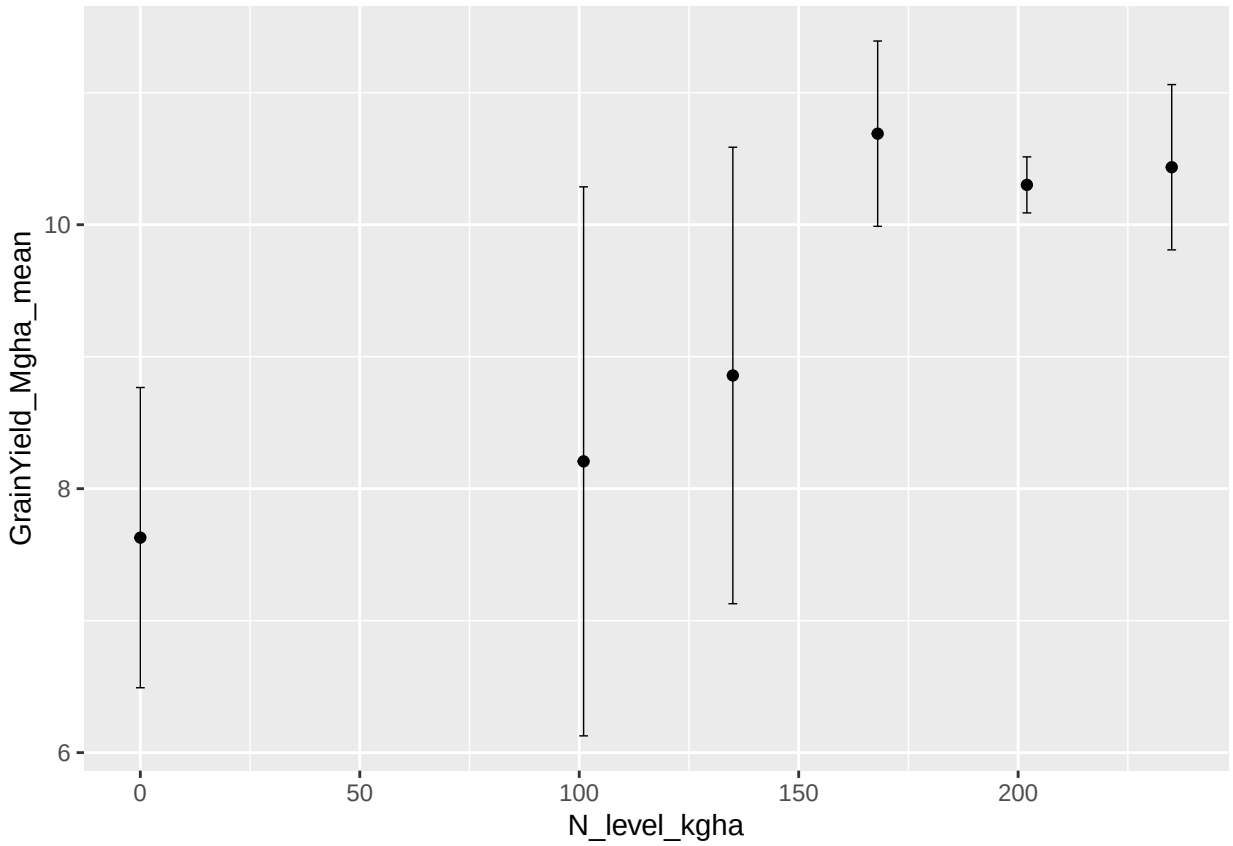


DAVIS-19

```
fig1_data_dav19 <- paper2_data_OR_S1 %>%
  filter(site_year == "Davis-19") #filters to dav 19 site

ggplot(fig1_data_dav19 , aes(x = N_level_kgha , y = GrainYield_Mgha)) +
  geom_point() #creates a plot
```




```
fig1_data_dav19_mean <- fig1_data_dav19 %>%
  group_by(N_level_kgha) %>%
  summarise(GrainYield_Mgha_mean = mean(GrainYield_Mgha) , GrainYield_Mgha_sd = sd(GrainYield_Mgha)) %>%
  mutate(site_year = factor("Davis-19")) %>%
  ungroup()

ggplot(fig1_data_dav19_mean , aes(x = N_level_kgha , y = GrainYield_Mgha_mean)) +
  geom_point() +
  geom_errorbar(data = fig1_data_dav19_mean , aes( y = GrainYield_Mgha_mean , ymax = (GrainYield_Mgha_m
```



data

```
xvals.site.dav19 <- seq(0 , 250 , by = 1 )

yvals.site.dav19 <- (coefficients(lme.1)[4,1]
                     + coefficients(lme.1)[4,2]*seq(0 , 250, by = 1)
                     + coefficients(lme.1)[4,3]*(seq(0 , 250, by = 1)^2))

lme.1.dav19.vertex <- xvals.site.dav19[which(yvals.site.dav19 == max(yvals.site.dav19))]

print(lme.1.dav19.vertex)
```

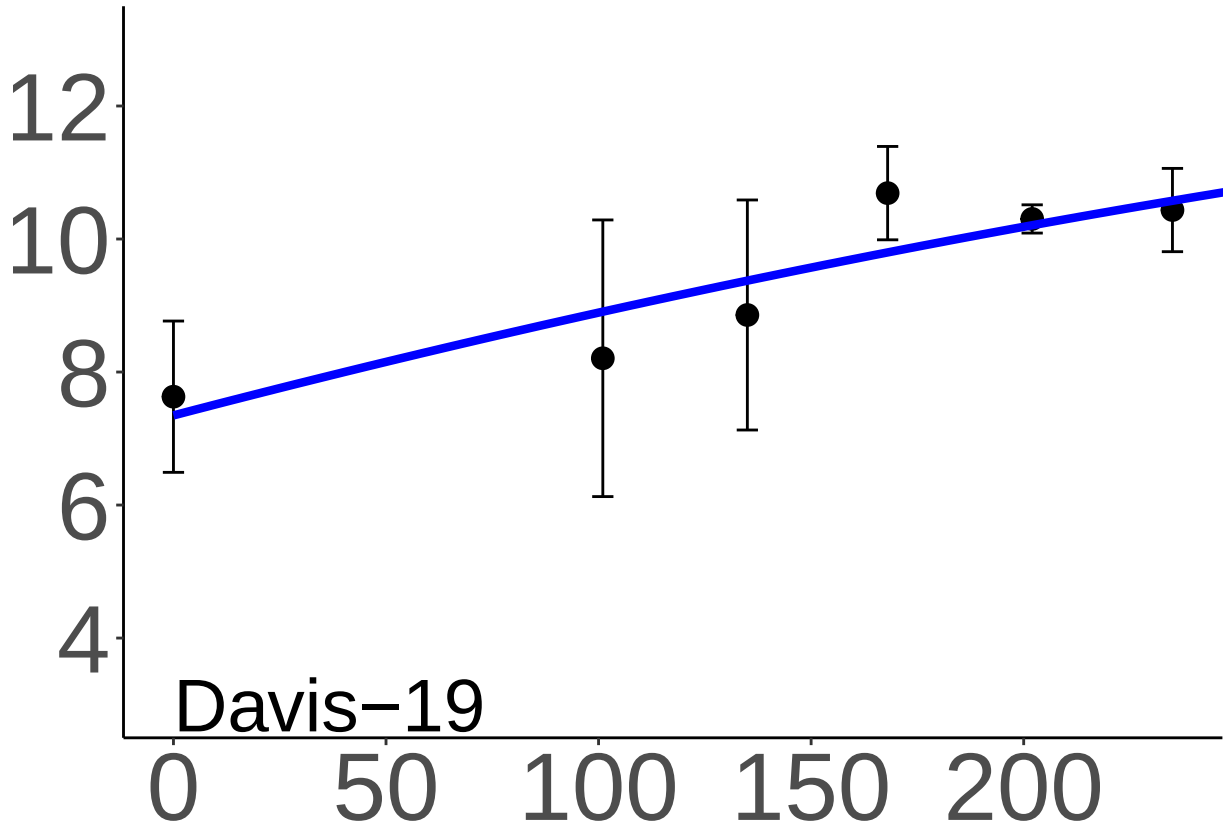
```
## [1] 250
```

```
lm.dav19.df <- data.frame(xvals.site.dav19 , yvals.site.dav19) #creates dataframe with N rate and fitte
```

plot

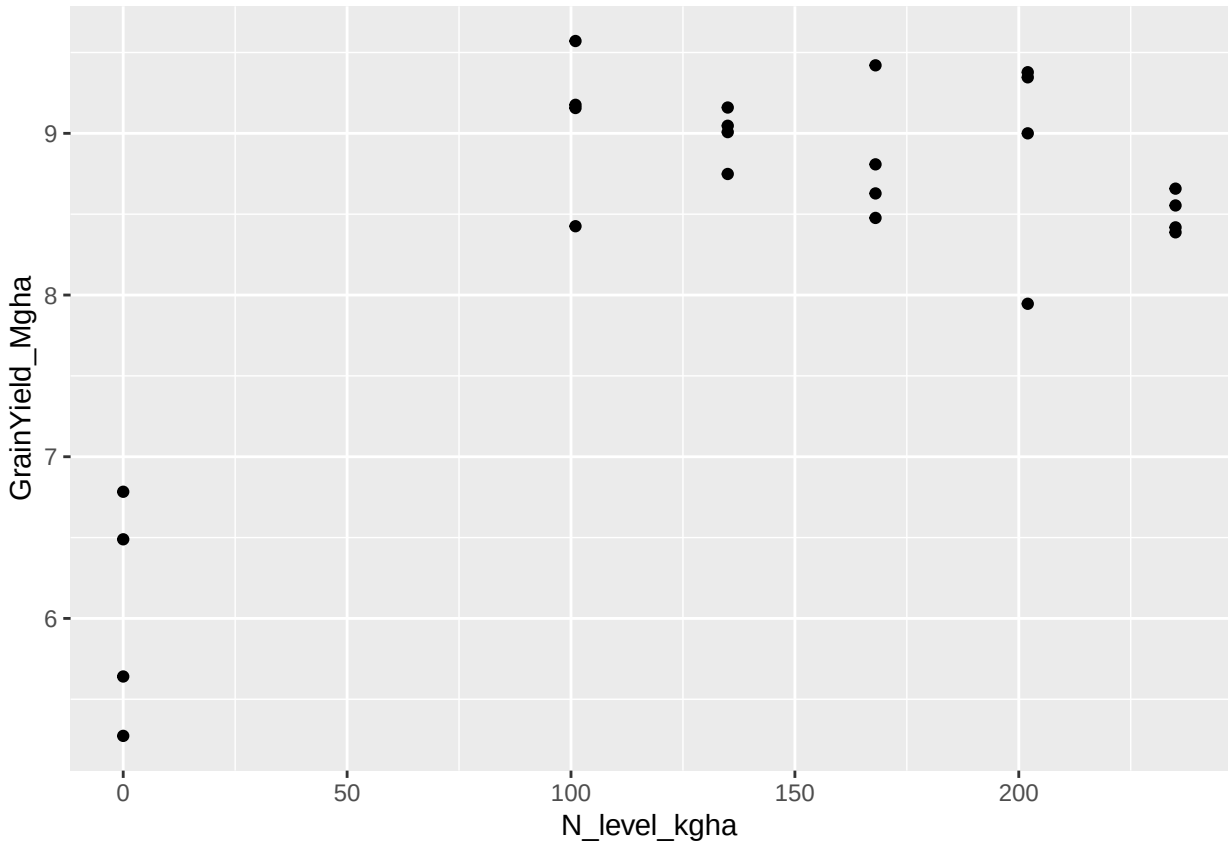
```
dav19_plot <- ggplot(fig1_data_dav19_mean ,
                     aes( x = N_level_kgha ,
                          y = GrainYield_Mgha_mean)) +
  geom_point(aes( x = N_level_kgha ,
                              y = GrainYield_Mgha_mean) , size = 3.5) +
  geom_errorbar(data = fig1_data_dav19_mean , aes( y = GrainYield_Mgha_mean ,
                                                   ymax = (GrainYield_Mgha_mean + GrainYield_Mgha_sd) ,
                                                   ymin = (GrainYield_Mgha_mean - GrainYield_Mgha_sd) ) ,
                alpha = 1 , width = 5 , size = .5 , position = position_dodge(.6) , show.legend = F) +
```

```
  geom_line(data = lm.dav19.df , aes(x = xvals.site.dav19 ,
                                     y = yvals.site.dav19) ,
            color = "blue" , size = 1.5) +
  coord_cartesian(xlim = c(0 , 235) , ylim = c(3 , 13)) +
  scale_y_continuous(breaks = seq(2 , 13 , by = 2 )) +
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.x = element_text(size = 36) ,
        axis.text.y = element_text(size = 36),
        legend.position = "none" ) +
  annotate("text" , x = 0 , y = 3 , label = "Davis-19" , size = 10 , hjust = 0)


dav19_plot
```

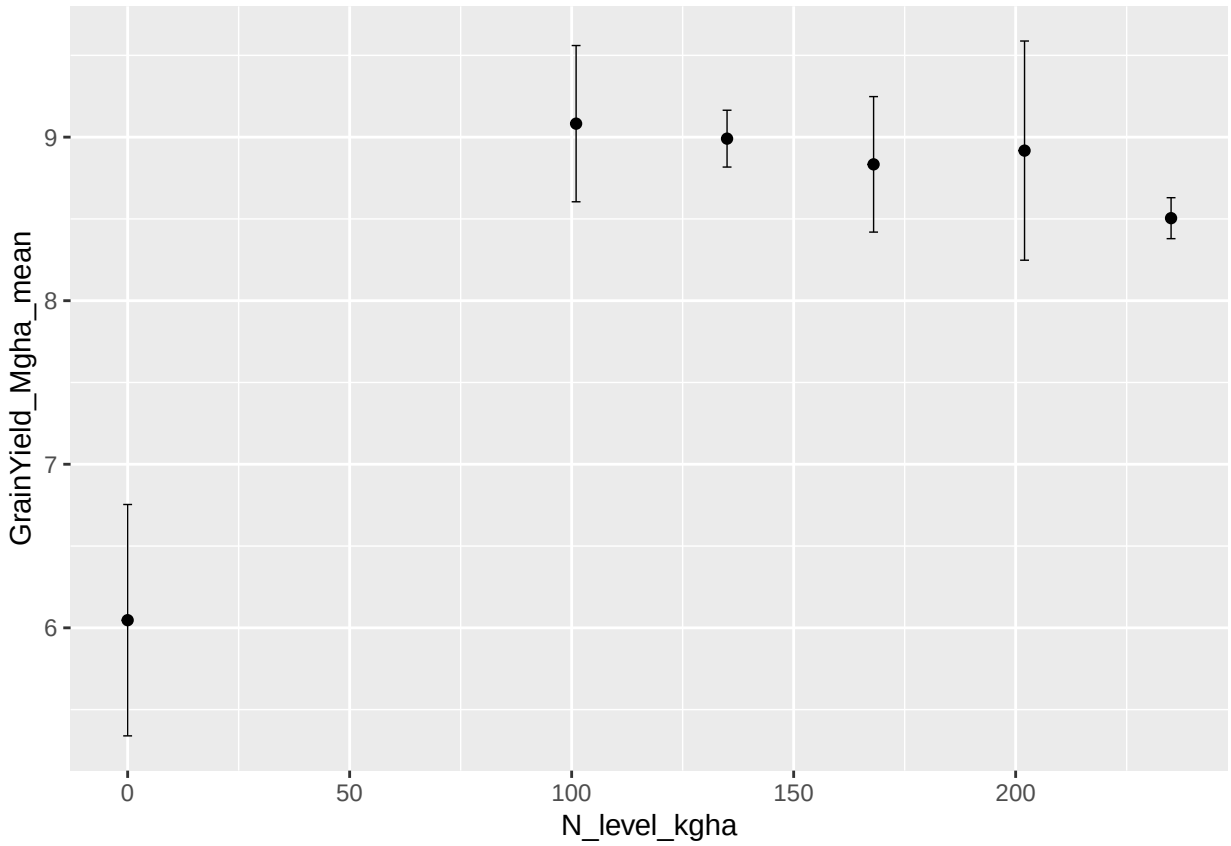


RES-19

```
fig1_data_res19 <- paper2_data_OR_S1 %>%
  filter(site_year == "RES-19") #filters to res 19 site

ggplot(fig1_data_res19 , aes(x = N_level_kgha , y = GrainYield_Mgha)) +
  geom_point() #creates a plot
```




```
fig1_data_res19_mean <- fig1_data_res19 %>%
  group_by(N_level_kgha) %>%
  summarise(GrainYield_Mgha_mean = mean(GrainYield_Mgha) , GrainYield_Mgha_sd = sd(GrainYield_Mgha)) %>%
  mutate(site_year = factor("RES-19")) %>%
  ungroup()

ggplot(fig1_data_res19_mean , aes(x = N_level_kgha , y = GrainYield_Mgha_mean)) +
  geom_point() +
  geom_errorbar(data = fig1_data_res19_mean , aes( y = GrainYield_Mgha_mean , ymax = (GrainYield_Mgha_m
```



data

```
xvals.site.res19 <- seq(0 , 250 , by = 1 )

yvals.site.res19 <- (coefficients(lme.1)[6,1]
                     + coefficients(lme.1)[6,2]*seq(0 , 250, by = 1)
                     + coefficients(lme.1)[6,3]*(seq(0 , 250, by = 1)^2))

lme.1.res19.vertex <- xvals.site.res19[which(yvals.site.res19 == max(yvals.site.res19))]

print(lme.1.res19.vertex)
```

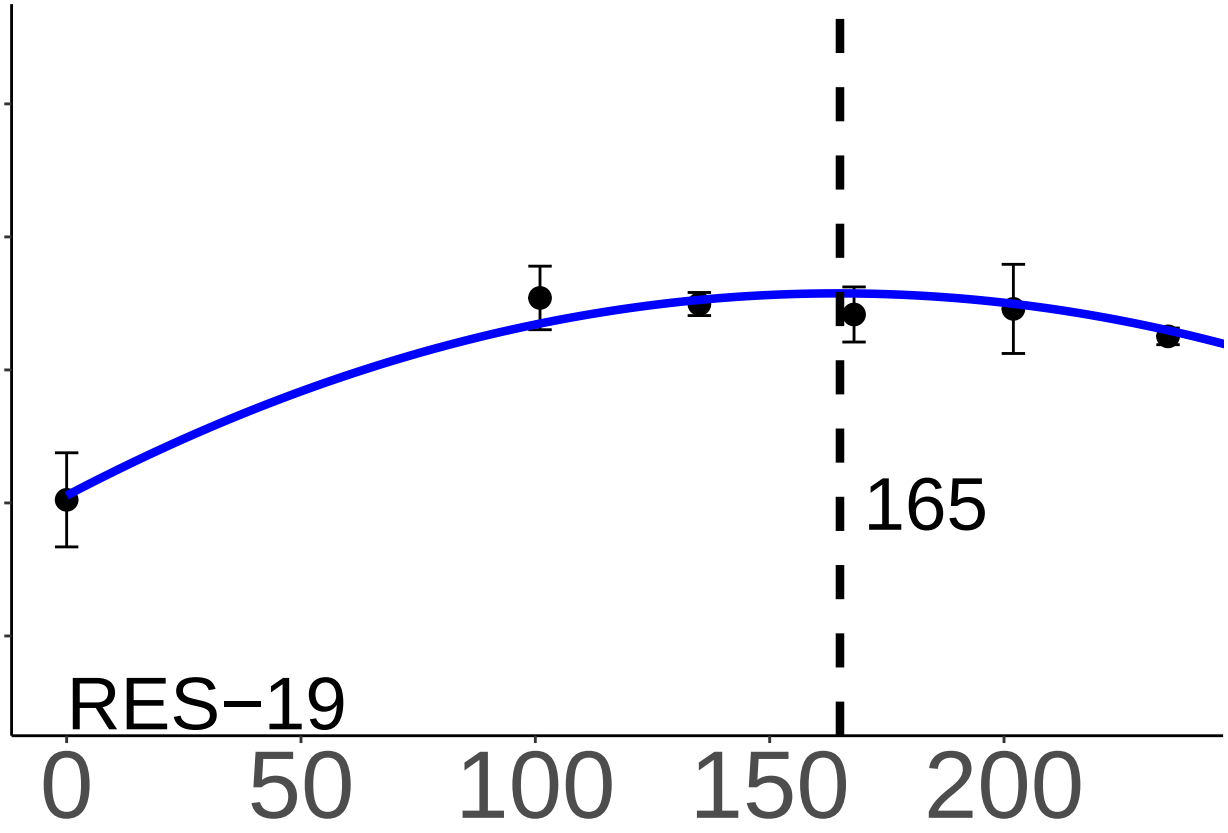
```
## [1] 165
```

```
lm.res19.df <- data.frame(xvals.site.res19 , yvals.site.res19) #creates dataframe with N rate and fitte
```

plot

```
res19_plot <- ggplot(fig1_data_res19_mean ,
                     aes( x = N_level_kgha ,
                          y = GrainYield_Mgha_mean)) +
  geom_point(aes( x = N_level_kgha ,
                              y = GrainYield_Mgha_mean) , size = 3.5) +
  geom_errorbar(data = fig1_data_res19_mean , aes( y = GrainYield_Mgha_mean ,
                                                   ymax = (GrainYield_Mgha_mean + GrainYield_Mgha_sd) ,
                                                   ymin = (GrainYield_Mgha_mean - GrainYield_Mgha_sd) ) ,
                alpha = 1 , width = 5 , size = .5 , position = position_dodge(.6) , show.legend = F) +
```

```
  geom_line(data = lm.res19.df , aes(x = xvals.site.res19 ,
                                     y = yvals.site.res19) ,
            color = "blue" , size = 1.5) +
  coord_cartesian(xlim = c(0 , 235) , ylim = c(3 , 13)) +
  scale_y_continuous(breaks = seq(2 , 13 , by = 2 )) +
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.x = element_text(size = 36) ,
        axis.text.y = element_blank(),
        legend.position = "none" ) +
  annotate("text" , x = 0 , y = 3 , label = "RES-19" , size = 10 , hjust = 0) +
  annotate("text" , x = (lme.1.res19.vertex + 5) , y = 6 , label = "165" , size = 10 , hjust = 0) +
  geom_vline(xintercept = lme.1.res19.vertex , linetype = "dashed" , size = 1.5)

res19_plot
```



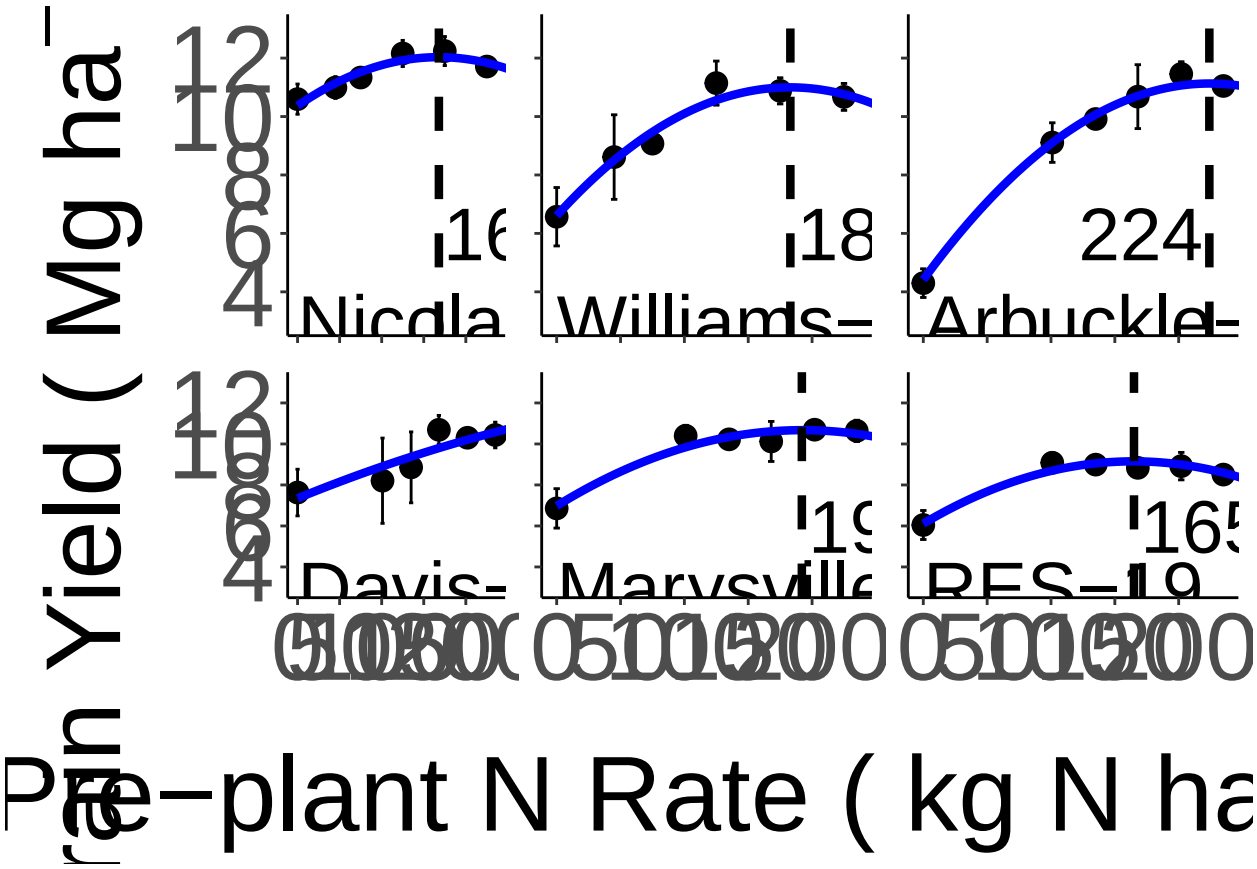
Yield Results for Paper

```
fig1_results <- rbind(fig1_data_nic17_mean,
                      fig1_data_wil17_mean,
                      fig1_data_arb19_mean,
                      fig1_data_mry19_mean,
                      fig1_data_dav19_mean,
                      fig1_data_res19_mean)

fig1_results$GrainYield_Mgha_mean <- round(fig1_results$GrainYield_Mgha_mean , digits = 1)
```

```
fig1_results <- fig1_results %>%
  dplyr::select(-GrainYield_Mgha_sd)

fig1_results_min <- fig1_results %>%
  group_by(site_year) %>%
  summarise(min = min(GrainYield_Mgha_mean)) %>%
  ungroup()

fig1_results_max <- fig1_results %>%
  group_by(site_year) %>%
  summarise(max = max(GrainYield_Mgha_mean)) %>%
  ungroup()
```

FIG 3

```
Fig3 <- grid.arrange(arrangeGrob(
                              nic17_plot,
                              wil17_plot,
                              arb19_plot,
                              dav19_plot,
                              mry19_plot,
                              res19_plot,
                              ncol = 3,
                              nrow = 2,
                              bottom = textGrob("Pre-plant N Rate ( kg N ha"^-1~")" ,
                                                gp = gpar( fontsize = 40) ,
                                                hjust = 0.5),
                              left = textGrob("Grain Yield ( Mg ha"^-1~")" ,
                                              gp = gpar(fontsize = 40) ,
                                              rot = 90,
                                              vjust = 0.5))
                                )
```




```
ggsave("FIGURES/Fig3.pdf" , Fig3,  width = 22 , height = 12 , dpi = 10000)
```

FIG 4

```
fig2.data <- paper2_data_OR %>%
  dplyr::select("site_year" , "exp_plot_number", "N_level_kgha"  , "TopDress_kgha"  , "GrainYield_Mgha"
```

NICOLAUS-17

data

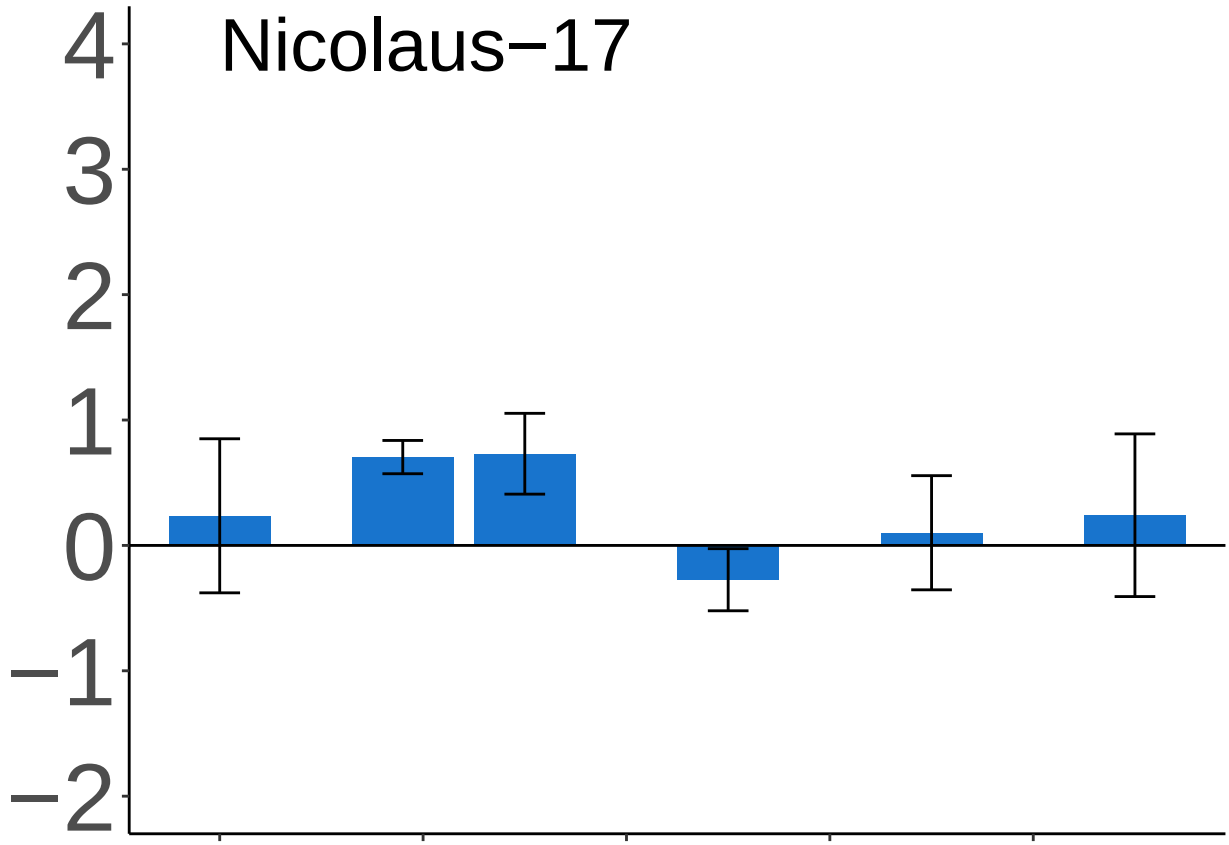
```
fig2.data.nic17 <- fig2.data %>%
  filter(site_year == "Nicolaus-17") #filter nic 17 site

fig2.data.nic17 <- spread(data = fig2.data.nic17 , key = TopDress_kgha , value = GrainYield_Mgha) #spre

fig2.data.nic17 <- fig2.data.nic17 %>%
  rename(yield_0 = "0",
         yield_25 = "25" ,
         yield_50 = "50") %>%
  mutate(yield_resp_25 = yield_25 - yield_0,
         yield_resp_50 = yield_50 - yield_0) %>%
  dplyr::select("site_year" , "exp_plot_number", "N_level_kgha"  , "yield_resp_25" , "yield_resp_50") %
  mutate(yield_response_mean = (yield_resp_25 + yield_resp_50) / 2 ) %>%
  mutate(site_year = factor("Nicolaus-17")) %>%
```

```
  dplyr::select(-c("yield_resp_25" , "yield_resp_50"))#processes data.

fig2.data.nic17 <- fig2.data.nic17 %>%
  group_by(site_year , N_level_kgha) %>%
  summarise(yield_response = mean(yield_response_mean) , yield_response_sd = sd((yield_response_mean)))
  ungroup()
```

```
## `summarise()` has grouped output by 'site_year'. You can override using the
## `.groups` argument.
```

plot

```
nic17.fig2.plot <- ggplot(data = fig2.data.nic17 , aes ( x = N_level_kgha  , y = yield_response , fill =
  geom_bar(position = "dodge" , stat = "identity" , fill = "dodgerblue3" , width = 25) +
  geom_errorbar( data = fig2.data.nic17 , aes( y = yield_response , ymax = (yield_response + yield_respo
  theme_classic() +
theme(axis.title = element_blank(),
        axis.text.y = element_text(size = 36) ,
        axis.text.x = element_blank(),
        legend.title = element_text(size = 22),
        legend.text = element_text(size = 22),
        legend.position = "none" ) +
  geom_hline(yintercept = 0) +
  coord_cartesian( xlim = c (-10 , 235) , ylim = c(-2 , 4)) +
  scale_y_continuous(breaks = seq(-2 , 4 , by = 1 )) +
  labs(fill = "Pre-plant N \nRate (kg N ha"^-1~")") +
  annotate("text" , x = 0 , y = 4 , label = "Nicolaus-17" , size = 10 , hjust = 0)

nic17.fig2.plot
```



WILLIAMS-17

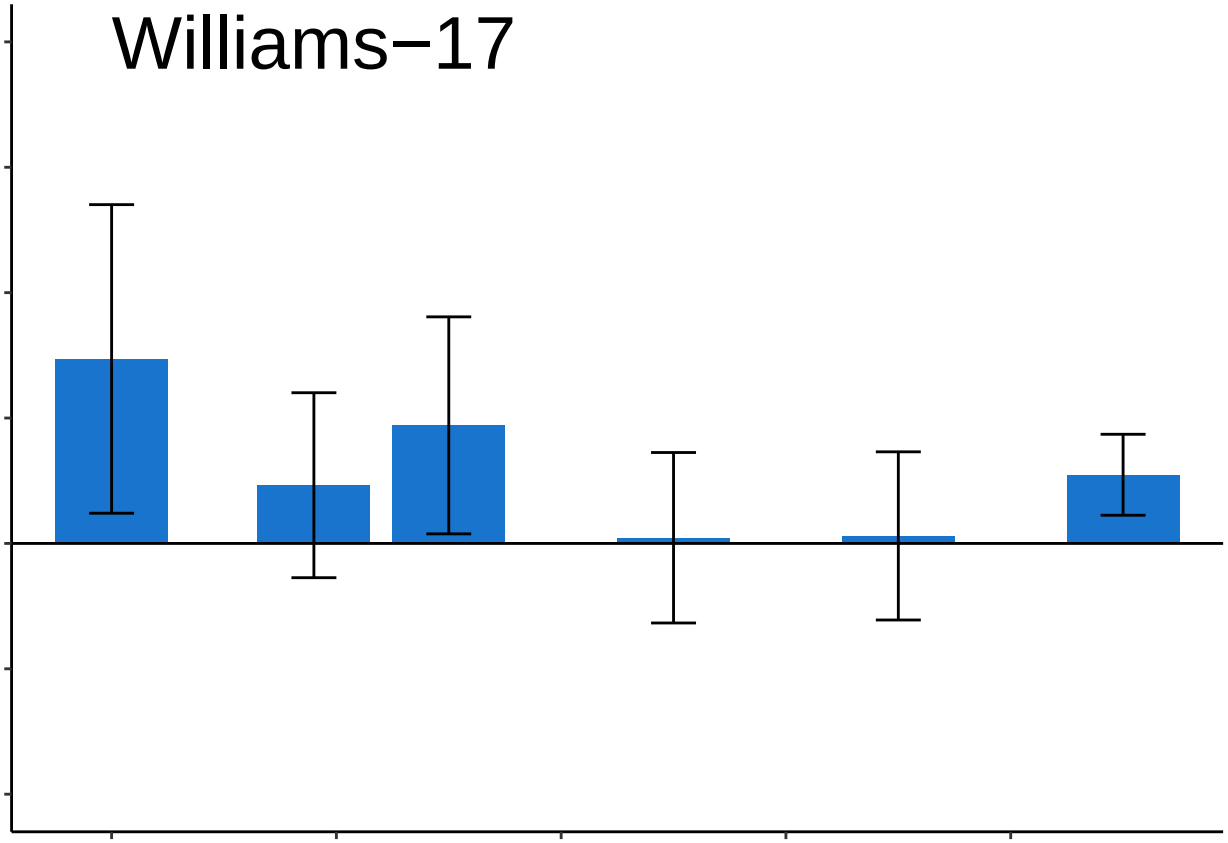
data

```
fig2.data.wil17 <- fig2.data %>%
  filter(site_year == "Williams-17") #filter nic 17 site

fig2.data.wil17 <- spread(data = fig2.data.wil17 , key = TopDress_kgha , value = GrainYield_Mgha) #spre

fig2.data.wil17 <- fig2.data.wil17 %>%
  rename(yield_0 = "0",
         yield_25 = "25" ,
         yield_50 = "50") %>%
  mutate(yield_resp_25 = yield_25 - yield_0,
         yield_resp_50 = yield_50 - yield_0) %>%
  dplyr::select("site_year" , "exp_plot_number", "N_level_kgha"  , "yield_resp_25" , "yield_resp_50") %
  mutate(yield_response_mean = (yield_resp_25 + yield_resp_50) / 2 ) %>%
  mutate(site_year = factor("Williams-17")) %>%
  dplyr::select(-c("yield_resp_25" , "yield_resp_50"))#processes data.

fig2.data.wil17 <- fig2.data.wil17 %>%
  group_by(site_year , N_level_kgha) %>%
  summarise(yield_response = mean(yield_response_mean) , yield_response_sd = sd((yield_response_mean))) 
  ungroup()
```

```
## `summarise()` has grouped output by 'site_year'. You can override using the
## `.groups` argument.
```

plot

```
wil17.fig2.plot <- ggplot(data = fig2.data.wil17 , aes ( x = N_level_kgha  , y = yield_response , fill =
  geom_bar(position = "dodge" , stat = "identity" , fill = "dodgerblue3" , width = 25) +
  geom_errorbar( data = fig2.data.wil17 , aes( y = yield_response , ymax = (yield_response + yield_respo
  theme_classic() +
theme(axis.title = element_blank(),
        axis.text = element_blank(),
        legend.position = "none") +
  geom_hline(yintercept = 0) +
  coord_cartesian( xlim = c (-10 , 235) , ylim = c(-2 , 4)) +
  scale_y_continuous(breaks = seq(-2 , 4 , by = 1 )) +
  labs(fill = "Pre-plant N \nRate (kg N ha"^-1~")") +
  annotate("text" , x = 0 , y = 4 , label = "Williams-17" , size = 10 , hjust = 0)

wil17.fig2.plot
```



ARBUCKLE-19

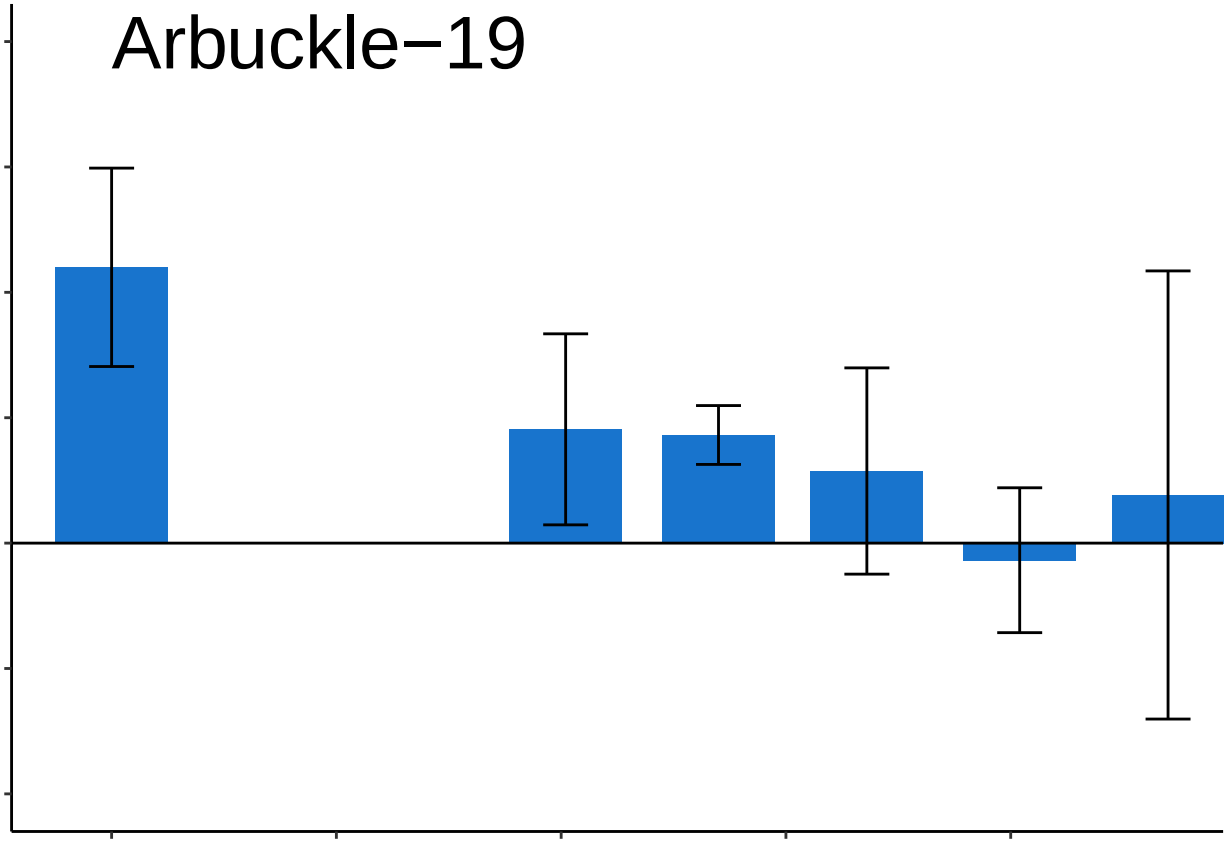
data

```
fig2.data.arb19 <- fig2.data %>%
  filter(site_year == "Arbuckle-19") #filter arb19 site

fig2.data.arb19 <- spread(data = fig2.data.arb19 , key = TopDress_kgha , value = GrainYield_Mgha) #spre

fig2.data.arb19 <- fig2.data.arb19 %>%
```

```
  rename(yield_0 = "0",
         yield_34 = "34" ) %>%
  mutate(yield_resp = yield_34 - yield_0) %>%
  dplyr::select("site_year" , "exp_plot_number", "N_level_kgha"  , "yield_resp") %>%
  group_by(N_level_kgha) %>%
  summarise(yield_response = mean(yield_resp) , yield_response_sd = sd(yield_resp)) %>%
  mutate(site_year = factor("Arbuckle-19"))  #processes data.
```

plot

```
arb19.fig2.plot <- ggplot(data = fig2.data.arb19 , aes ( x = N_level_kgha  , y = yield_response , fill =
  geom_bar(position = "dodge" , stat = "identity" , fill = "dodgerblue3" , width = 25) +
  geom_errorbar( data = fig2.data.arb19 , aes( y = yield_response , ymax = (yield_response + yield_respo
  theme_classic() +
theme(axis.title = element_blank(),
        axis.text = element_blank(),
        legend.position = "none") +
  geom_hline(yintercept = 0) +
  coord_cartesian( xlim = c (-10 , 235) , ylim = c(-2 , 4)) +
  scale_y_continuous(breaks = seq(-2 , 4 , by = 1 )) +
  labs(fill = "Pre-plant N \nRate (kg N ha"^-1~")") +
  annotate("text" , x = 0 , y = 4 , label = "Arbuckle-19" , size = 10 , hjust = 0)

arb19.fig2.plot
```



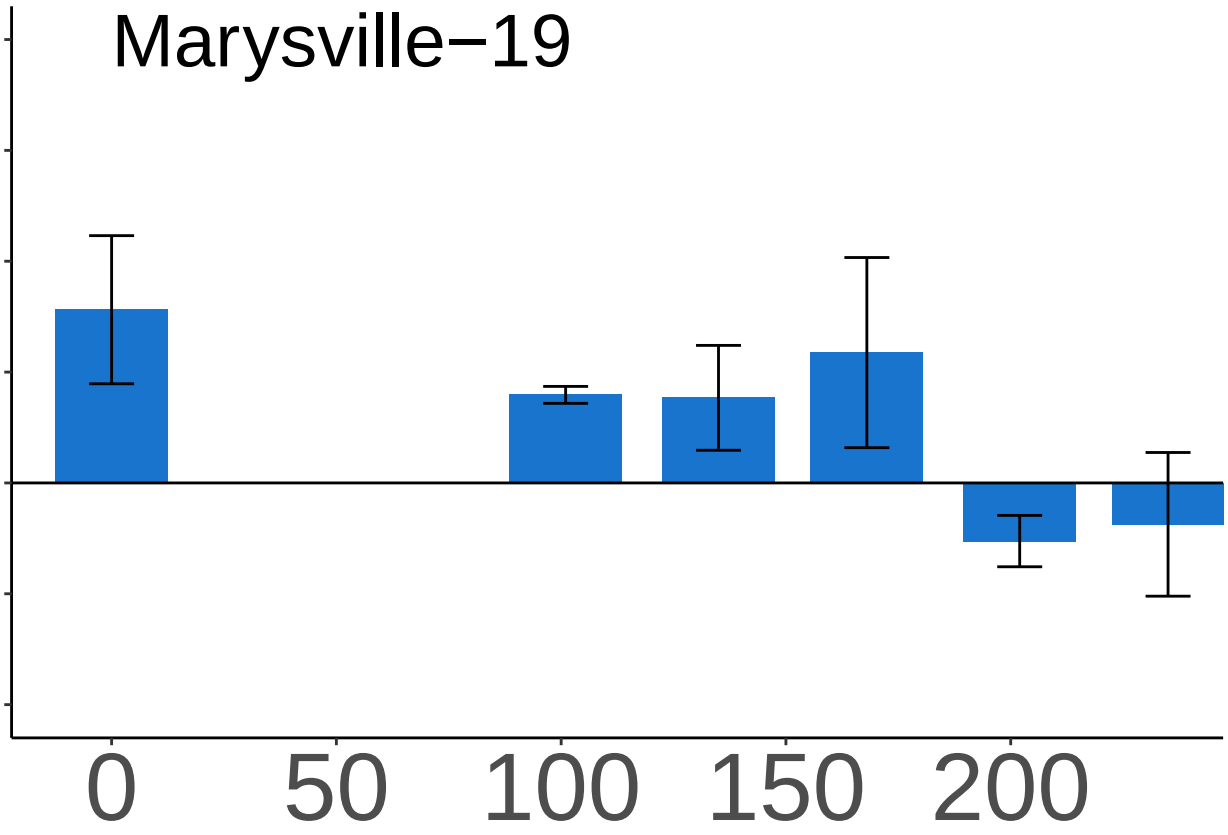
MARYSVILLE-19

data

```
fig2.data.mry19 <- fig2.data %>%
  filter(site_year == "Marysville-19") #filter mry19 site

fig2.data.mry19 <- spread(data = fig2.data.mry19 , key = TopDress_kgha , value = GrainYield_Mgha) #spre

fig2.data.mry19 <- fig2.data.mry19 %>%
  rename(yield_0 = "0",
         yield_34 = "34" ) %>%
  mutate(yield_resp = yield_34 - yield_0) %>%
  dplyr::select("site_year" , "exp_plot_number", "N_level_kgha"  , "yield_resp") %>%
  group_by(N_level_kgha) %>%
  summarise(yield_response = mean(yield_resp) , yield_response_sd = sd(yield_resp)) %>%
  mutate(site_year = factor("Marysville-19"))  #processes data.
```

plot

```
mry19.fig2.plot <- ggplot(data = fig2.data.mry19 , aes ( x = N_level_kgha  , y = yield_response , fill =
  geom_bar(position = "dodge" , stat = "identity" , fill = "dodgerblue3" , width = 25) +
  geom_errorbar( data = fig2.data.mry19 , aes( y = yield_response , ymax = (yield_response + yield_resp
  theme_classic() +
theme(axis.title = element_blank(),
        axis.text.y = element_blank() ,
        axis.text.x = element_text(size = 36),
        legend.position = "none" ) +
  geom_hline(yintercept = 0) +
  coord_cartesian( xlim = c (-10 , 235) , ylim = c(-2 , 4)) +
  scale_y_continuous(breaks = seq(-2 , 4 , by = 1 )) +
  labs(fill = "Pre-plant N \nRate (kg N ha"^-1~")") +
  annotate("text" , x = 0 , y = 4 , label = "Marysville-19" , size = 10 , hjust = 0)

mry19.fig2.plot
```



DAVIS-19

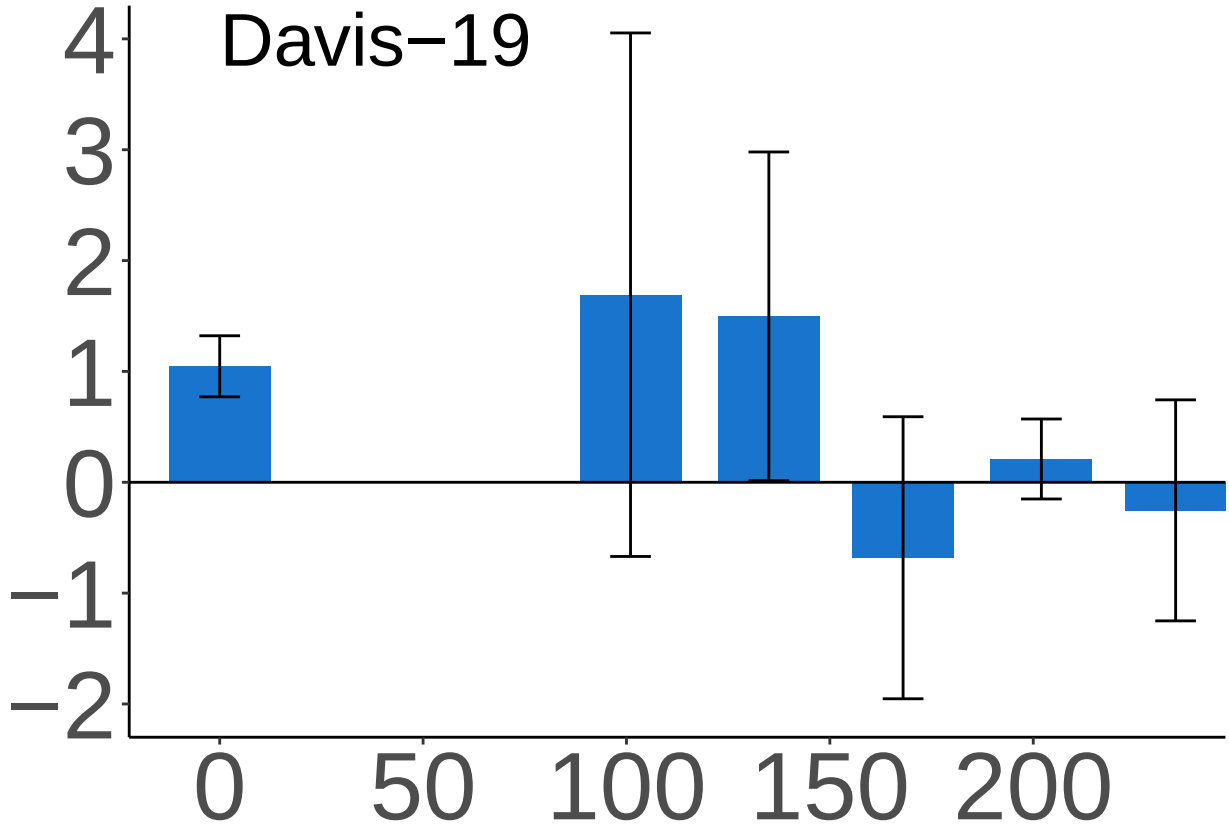
data

```
fig2.data.dav19 <- fig2.data %>%
  filter(site_year == "Davis-19") #filter dav19 site

fig2.data.dav19 <- spread(data = fig2.data.dav19 , key = TopDress_kgha , value = GrainYield_Mgha) #spre

fig2.data.dav19 <- fig2.data.dav19 %>%
  rename(yield_0 = "0",
         yield_34 = "34" ) %>%
  mutate(yield_resp = yield_34 - yield_0) %>%
  dplyr::select("site_year" , "exp_plot_number", "N_level_kgha"  , "yield_resp") %>%
  group_by(N_level_kgha) %>%
  summarise(yield_response = mean(yield_resp) , yield_response_sd = sd(yield_resp)) %>%
  mutate(site_year = factor("Davis-19"))  #processes data.
```

plot

```
dav19.fig2.plot <- ggplot(data = fig2.data.dav19 , aes ( x = N_level_kgha  , y = yield_response , fill =
  geom_bar(position = "dodge" , stat = "identity" , fill = "dodgerblue3" , width = 25) +
  geom_errorbar( data = fig2.data.dav19 , aes( y = yield_response , ymax = (yield_response + yield_resp
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.x = element_text(size = 36) ,
        axis.text.y = element_text(size = 36),
```

```
        legend.position = "none" ) +
  geom_hline(yintercept = 0) +
  coord_cartesian( xlim = c (-10 , 235) , ylim = c(-2 , 4)) +
  scale_y_continuous(breaks = seq(-2 , 4 , by = 1 )) +
  labs(fill = "Pre-plant N \nRate (kg N ha"^-1~")") +
  annotate("text" , x = 0 , y = 4 , label = "Davis-19" , size = 10 , hjust = 0)

dav19.fig2.plot
```



RES-19

data

```
fig2.data.res19 <- fig2.data %>%
  filter(site_year == "RES-19") #filter res19 site

fig2.data.res19 <- spread(data = fig2.data.res19 , key = TopDress_kgha , value = GrainYield_Mgha) #spre

fig2.data.res19 <- fig2.data.res19 %>%
  rename(yield_0 = "0",
         yield_34 = "34" ) %>%
  mutate(yield_resp = yield_34 - yield_0) %>%
  dplyr::select("site_year" , "exp_plot_number", "N_level_kgha"  , "yield_resp") %>%
  group_by(N_level_kgha) %>%
  summarise(yield_response = mean(yield_resp) , yield_response_sd = sd(yield_resp)) %>%
  mutate(site_year = factor("RES-19"))  #processes data.
```

plot

```
res19.fig2.plot <- ggplot(data = fig2.data.res19 , aes ( x = N_level_kgha  , y = yield_response , fill =
  geom_bar(position = "dodge" , stat = "identity" , fill = "dodgerblue3" , width = 25) +
  geom_errorbar( data = fig2.data.res19 , aes( y = yield_response , ymax = (yield_response + yield_respo
  theme_classic() +
  theme(axis.title = element_blank(),
        axis.text.x = element_text(size = 36) ,
        axis.text.y = element_blank(),
        legend.position = "none" ) +
  geom_hline(yintercept = 0) +
  coord_cartesian( xlim = c (-10 , 235) , ylim = c(-2 , 4)) +
  scale_y_continuous(breaks = seq(-2 , 4 , by = 1 )) +
  labs(fill = "Pre-plant N \nRate (kg N ha"^-1~")") +
  annotate("text" , x = 0 , y = 4 , label = "RES-19" , size = 10 , hjust = 0)

res19.fig2.plot
```

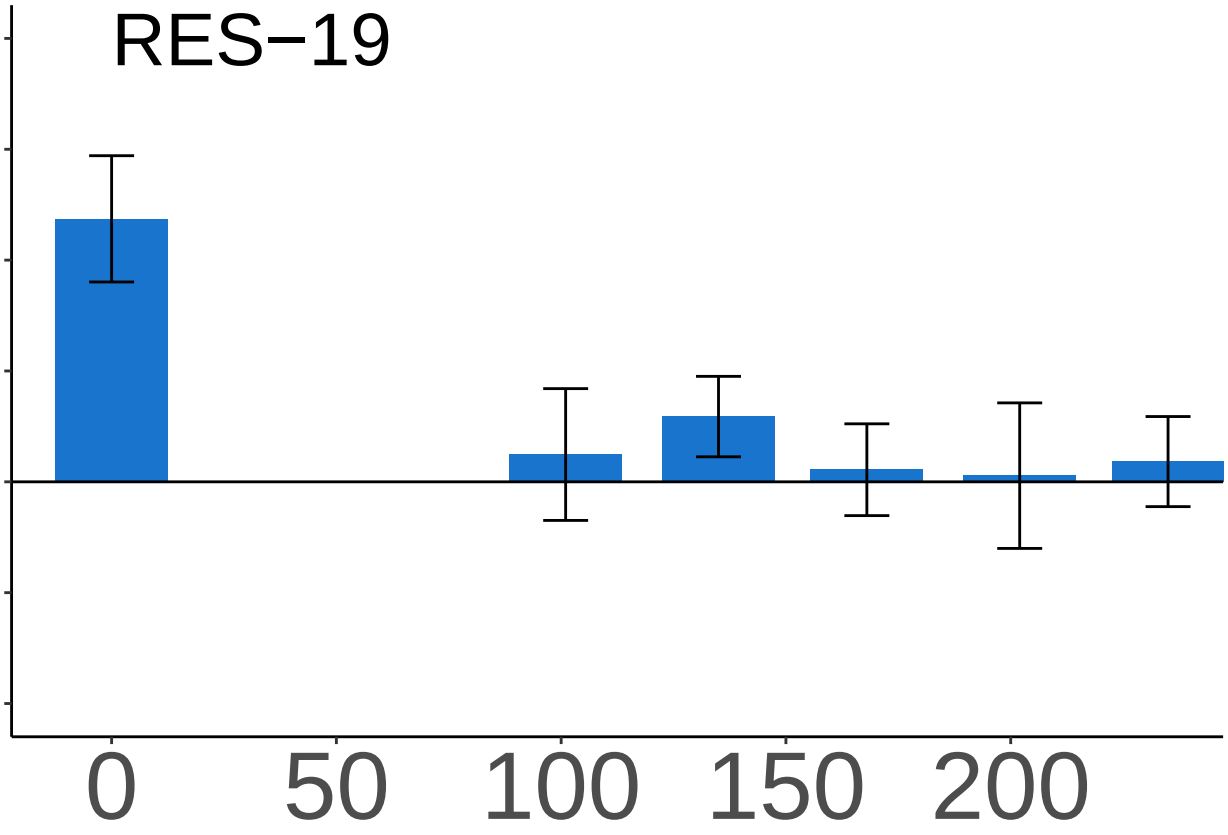
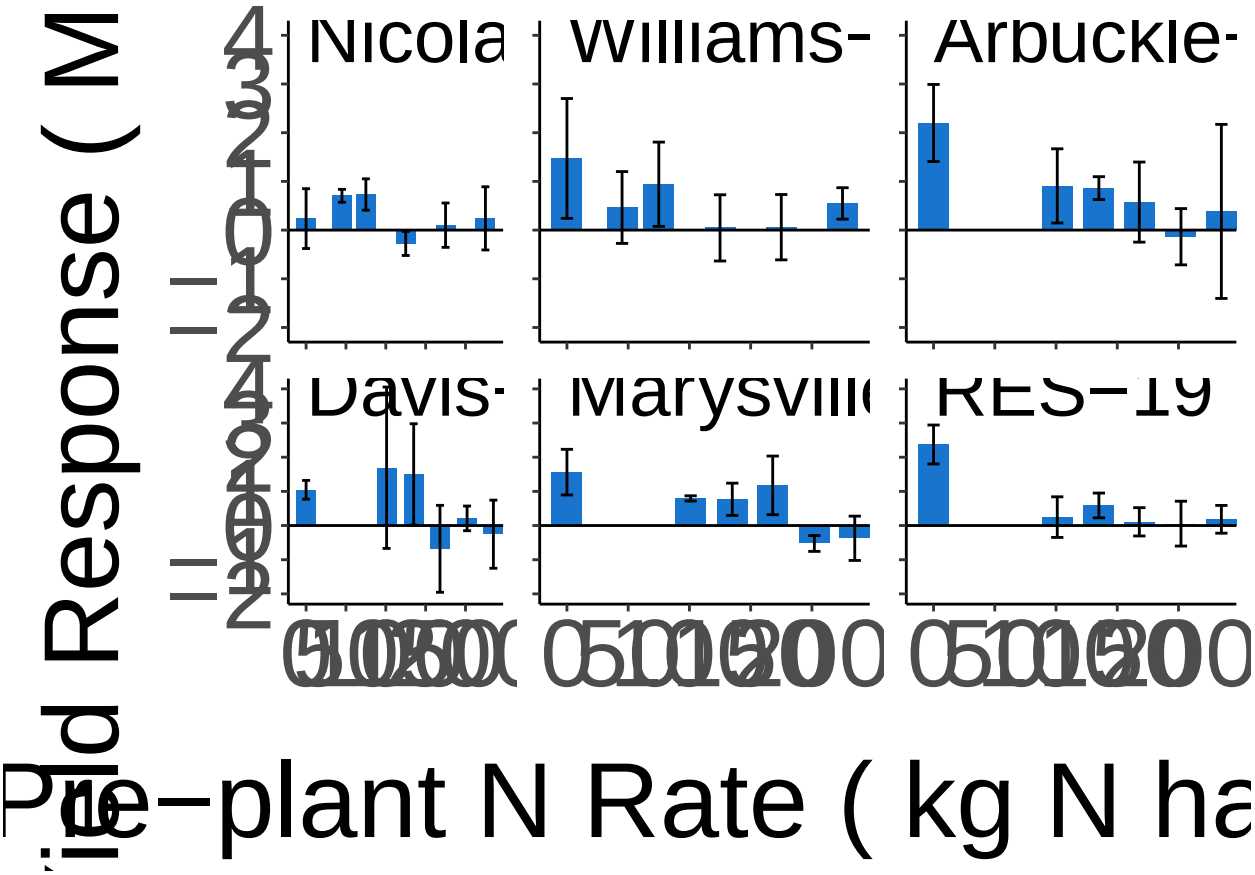


FIG 4

```
Fig4 <- grid.arrange(arrangeGrob(
                              nic17.fig2.plot,
                              wil17.fig2.plot,
                              arb19.fig2.plot,
                              dav19.fig2.plot,
                              mry19.fig2.plot,
```

```
                        res19.fig2.plot,
                        ncol = 3,
                        nrow = 2,
                        bottom = textGrob("Pre-plant N Rate ( kg N ha"^-1~")" ,
                                          gp = gpar( fontsize = 40) ,
                                          hjust = 0.5),
                        left = textGrob("Grain Yield Response ( Mg ha"^-1~")" ,
                                        gp = gpar(fontsize = 40) ,
                                        rot = 90,
                                        vjust = 0.5))
                          )
```




```
ggsave("FIGURES/Fig4.pdf" , Fig4,  width = 22 , height = 12 , dpi = 10000)
```

FIG 5

models

gs NDVI

```
fig3.lm.gsndvi <- lme(gs_NDVI_Sufficiency_Index ~ N_level_kgha + I(N_level_kgha^2),
                      random = ~N_level_kgha + I(N_level_kgha^2)|site_year,
                      data= paper2_data_OR_S1,
                      na.action = na.omit) #creates mixed model

summary(fig3.lm.gsndvi) # summary of model
```

```
## Linear mixed-effects model fit by REML
##   Data: paper2_data_OR_S1
##         AIC       BIC   logLik
##   -294.8162 -265.3286 157.4081
##
## Random effects:
##  Formula: ~N_level_kgha + I(N_level_kgha^2) | site_year
##  Structure: General positive-definite, Log-Cholesky parametrization
##                   StdDev       Corr
## (Intercept)       1.308802e-01 (Intr) N_lvl_
## N_level_kgha      1.831749e-03 -0.860
## I(N_level_kgha^2) 5.886440e-06  0.745 -0.980
## Residual          6.221036e-02
##
## Fixed effects:  gs_NDVI_Sufficiency_Index ~ N_level_kgha + I(N_level_kgha^2)
##                        Value  Std.Error  DF   t-value p-value
## (Intercept)        0.5685407 0.05479903 136 10.375013  0.0000
## N_level_kgha       0.0036516 0.00078147 136  4.672660  0.0000
## I(N_level_kgha^2) -0.0000085 0.00000258 136 -3.285746  0.0013
##  Correlation:
##                   (Intr) N_lvl_
## N_level_kgha      -0.850
## I(N_level_kgha^2)  0.720 -0.975
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.43779779 -0.42196176  0.08211079  0.53963270  2.24798317
##
## Number of Observations: 144
## Number of Groups: 6
```

```
car::Anova(fig3.lm.gsndvi, type = "3")
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: gs_NDVI_Sufficiency_Index
##                      Chisq Df Pr(>Chisq)
## (Intercept)       107.641  1  < 2.2e-16 ***
## N_level_kgha       21.834  1  2.973e-06 ***
## I(N_level_kgha^2)  10.796  1   0.001017 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MuMIn::r.squaredGLMM(fig3.lm.gsndvi)
```

```
##            R2m       R2c
## [1,] 0.6966089 0.8637795
```

```
coefficients(fig3.lm.gsndvi)
```

```
##               (Intercept) N_level_kgha I(N_level_kgha^2)
## Nicolaus-17     0.6645431  0.002530473     -5.355691e-06
## Williams-17     0.5631309  0.004885946     -1.360099e-05
## Arbuckle-19     0.3365180  0.005951827     -1.413451e-05
## Davis-19        0.6827222  0.000933393      9.508980e-07
```

```
## Marysville-19   0.6306106  0.003522628     -8.922008e-06
## RES-19          0.5337193  0.004085133     -9.813552e-06
```

```
fig3.lm.gsndvi$coefficients$fixed
```

```
##       (Intercept)      N_level_kgha I(N_level_kgha^2)
##      5.685407e-01      3.651567e-03     -8.479309e-06
```

```
xvals.fig3.lm.gsndvi <- seq(0,250, by = 1)

yvals.fig3.lm.gsndvi <- (fig3.lm.gsndvi$coefficients$fixed[1]
                         + fig3.lm.gsndvi$coefficients$fixed[2]*seq(0,250, by = 1)
                         + fig3.lm.gsndvi$coefficients$fixed[3]*(seq(0,250, by = 1)^2))

h.fig3.lm.gsndvi <- xvals.fig3.lm.gsndvi[which(yvals.fig3.lm.gsndvi == max(yvals.fig3.lm.gsndvi))]

mylist <- list(N_level_kgha = seq(0 , 250 , by = 1) ) #makes the list

fig3.lm.gsndvi.df <- as.data.frame(summary(emmeans(fig3.lm.gsndvi , ~ N_level_kgha , at = mylist ))) #c

colnames(fig3.lm.gsndvi.df)
```

```
## [1] "N_level_kgha" "emmean"       "SE"           "df"           "lower.CL"
## [6] "upper.CL"
```

```
fig3.lm.gsndvi.df <- fig3.lm.gsndvi.df %>%
  rename(N_rate = N_level_kgha,
         Fit = emmean) %>%
  mutate(Index = "gs_NDVI")

max_opt_N <- list(N_level_kgha = 215) #makes the list
max_opt_SI <- emmeans(fig3.lm.gsndvi , ~ N_level_kgha , at = max_opt_N ) #calculates the emmeans SI for
max_opt_SI
```

```
##  N_level_kgha emmean     SE df lower.CL upper.CL
##           215  0.962 0.0124  5     0.93    0.994
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
```

uas NDRE

```
fig3.lm.uasndre <- lme(uas_NDRE_Sufficiency_Index ~ N_level_kgha + I(N_level_kgha^2),
                  random = ~N_level_kgha + I(N_level_kgha^2)|site_year,
                  data= paper2_data_OR_S1,
                  na.action = na.omit) #creates mixed model

summary(fig3.lm.uasndre) # summary of model
```

```
## Linear mixed-effects model fit by REML
##   Data: paper2_data_OR_S1
##         AIC       BIC   logLik
##   -406.1827 -376.6951 213.0913
##
## Random effects:
##  Formula: ~N_level_kgha + I(N_level_kgha^2) | site_year
```

```
##  Structure: General positive-definite, Log-Cholesky parametrization
##                    StdDev       Corr
## (Intercept)       8.208336e-02 (Intr) N_lvl_
## N_level_kgha      5.041233e-04 -0.528
## I(N_level_kgha^2) 1.435543e-06 -0.256 -0.681
## Residual          4.185303e-02
##
## Fixed effects:  uas_NDRE_Sufficiency_Index ~ N_level_kgha + I(N_level_kgha^2)
##                        Value  Std.Error  DF   t-value p-value
## (Intercept)        0.6567409 0.03449606 136 19.038140       0
## N_level_kgha       0.0026113 0.00025604 136 10.198972       0
## I(N_level_kgha^2) -0.0000054 0.00000086 136 -6.309234       0
##  Correlation:
##                   (Intr) N_lvl_
## N_level_kgha      -0.516
## I(N_level_kgha^2) -0.075 -0.788
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.62031914 -0.45351573  0.07453487  0.50164297  2.85634957
##
## Number of Observations: 144
## Number of Groups: 6
```

```
car::Anova(fig3.lm.uasndre, type = "3")
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: uas_NDRE_Sufficiency_Index
##                      Chisq Df Pr(>Chisq)
## (Intercept)       362.451  1  < 2.2e-16 ***
## N_level_kgha      104.019  1  < 2.2e-16 ***
## I(N_level_kgha^2)  39.806  1  2.804e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MuMIn::r.squaredGLMM(fig3.lm.uasndre)
```

```
##           R2m       R2c
## [1,] 0.718635 0.8966055
```

```
coefficients(fig3.lm.uasndre)
```

```
##               (Intercept) N_level_kgha I(N_level_kgha^2)
## Nicolaus-17     0.7565335  0.002468930     -6.436800e-06
## Williams-17     0.7101138  0.002767194     -6.733537e-06
## Arbuckle-19     0.5521163  0.003130763     -5.543102e-06
## Davis-19        0.6193688  0.002135867     -3.334667e-06
## Marysville-19   0.5989988  0.003105201     -6.155315e-06
## RES-19          0.7033140  0.002059978     -4.403405e-06
```

```
fig3.lm.uasndre$coefficients$fixed
```

```
##       (Intercept)      N_level_kgha I(N_level_kgha^2)
##      6.567409e-01      2.611322e-03     -5.434471e-06
```

```
xvals.fig3.lm.uasndre <- seq(0,250, by = 1)

yvals.fig3.lm.uasndre <- (fig3.lm.uasndre$coefficients$fixed[1]
                          + fig3.lm.uasndre$coefficients$fixed[2]*seq(0,250, by = 1)
                          + fig3.lm.uasndre$coefficients$fixed[3]*(seq(0,250, by = 1)^2))

h.fig3.lm.uasndre <- xvals.fig3.lm.uasndre[which(yvals.fig3.lm.uasndre == max(yvals.fig3.lm.uasndre))]

mylist <- list(N_level_kgha = seq(0 , 250 , by = 1) ) #makes the list

fig3.lm.uasndre.df <- as.data.frame(summary(emmeans(fig3.lm.uasndre , ~ N_level_kgha , at = mylist ))) 

colnames(fig3.lm.uasndre.df)
```

```
## [1] "N_level_kgha" "emmean"       "SE"           "df"           "lower.CL"
## [6] "upper.CL"
```

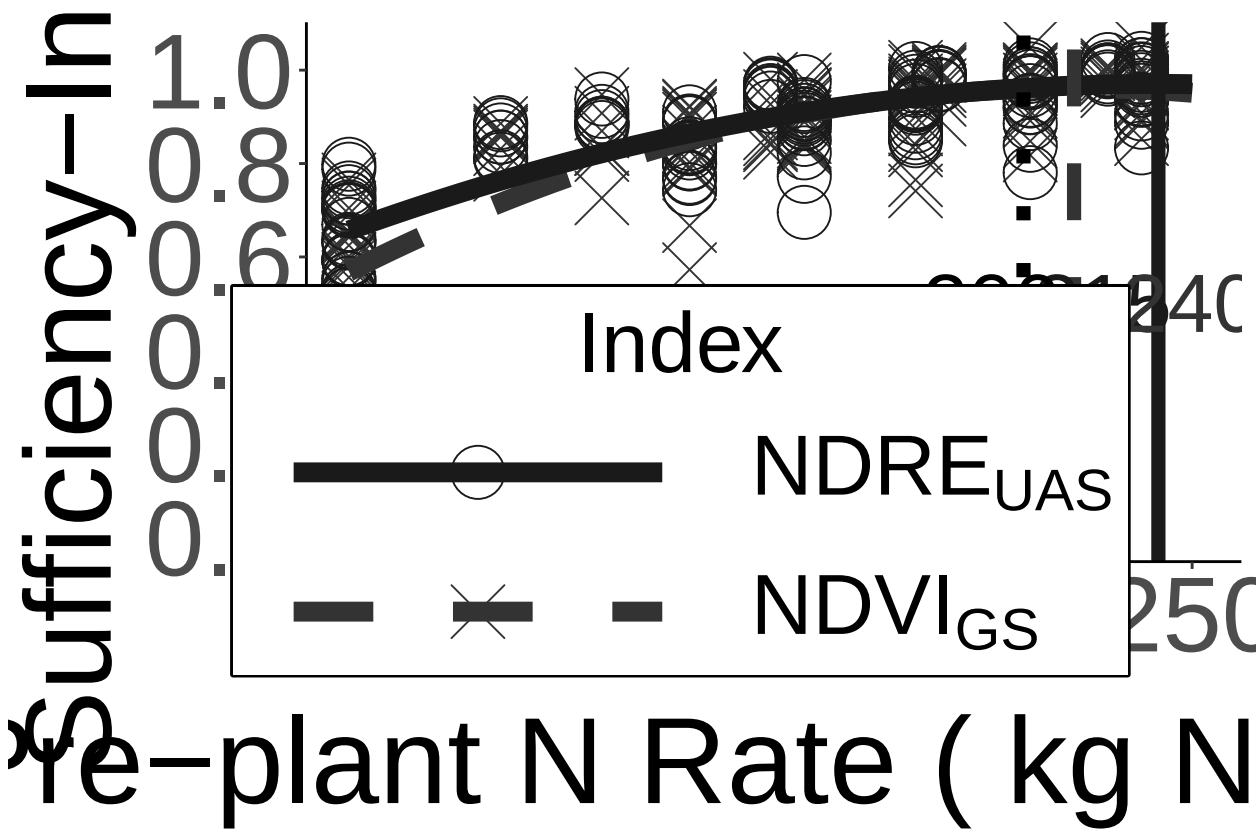
```
fig3.lm.uasndre.df <- fig3.lm.uasndre.df %>%
  rename(N_rate = N_level_kgha,
         Fit = emmean) %>%
  mutate(Index = "uas_NDRE")


max_opt_N <- list(N_level_kgha = 240) #makes the list
max_opt_SI <- emmeans(fig3.lm.uasndre , ~ N_level_kgha , at = max_opt_N ) #calculates the emmeans for m
max_opt_SI
```

```
##  N_level_kgha emmean     SE df lower.CL upper.CL
##           240   0.97 0.0133  5    0.936        1
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
```

```
new_fig3_data <- rbind(fig3.lm.gsndvi.df,
                       fig3.lm.uasndre.df)

new_fig3_data_ndvi <- paper2_data_OR_S1 %>%
  dplyr::select(N_level_kgha , gs_NDVI_Sufficiency_Index) %>%
  rename(N_rate = N_level_kgha ,
         SI = gs_NDVI_Sufficiency_Index) %>%
  mutate(Index = "gs_NDVI" )

new_fig3_data_ndre <- paper2_data_OR_S1 %>%
  dplyr::select(N_level_kgha , uas_NDRE_Sufficiency_Index) %>%
  rename(N_rate = N_level_kgha ,
         SI = uas_NDRE_Sufficiency_Index) %>%
  mutate(Index = "uas_NDRE" )

new_fig3_data_2 <- rbind(new_fig3_data_ndvi,
                         new_fig3_data_ndre)
```

FIG 5

```
Fig5 <- ggplot(data = new_fig3_data_2 , aes ( x = N_rate , y = SI)) +
  geom_point(data = new_fig3_data_2 , aes ( x = N_rate , y = SI , shape = Index , fill = Index , color =
  geom_line(data = new_fig3_data , aes( x = N_rate , y = Fit , color = Index , linetype = Index) , size
  coord_cartesian(ylim = c(0 , 1.05) , xlim = c(0 , 252)) +
  scale_x_continuous(breaks = seq(0 , 250 , by = 50)) +
  scale_y_continuous(breaks = seq(0 , 1 , by = .2)) +
  theme_classic() +
  labs(x = "Pre-plant N Rate ( kg N ha"^-1~")" , y = "Sufficiency-Index" , shape = "Index" , color = "In
  guides(color = guide_legend(keywidth = 12 , keyheight = 1.5 , unit = "cm" , title.hjust = 0.5 , byrow
         shape = guide_legend(keywidth = 12 , keyheight = 1.5 , unit = "cm" , title.hjust = 0.5 , byrow
         fill = guide_legend(keywidth = 12 , keyheight = 1.5 , unit = "cm" , title.hjust = 0.5 , byrow =
         alpha = guide_legend(keywidth = 12 , keyheight = 1.5 , unit = "cm" , title.hjust = 0.5 , byrow
         linetype = guide_legend(keywidth = 12 , keyheight = 1.5 , unit = "cm" , title.hjust = 0.5 , by
  theme(axis.title = element_text(size = 46),
        axis.text = element_text(size = 40),
        legend.text = element_text(size = 32),
        legend.title = element_text(size = 32 , hjust = 0.5),
        legend.text.align = 0,
        legend.box.background = element_rect(size = 1),
        legend.position = c(0.40 , 0.15 )) +
  geom_vline(xintercept = h.fig3.lm.gsndvi , linetype = "dashed" , size = 2.5 , color = "grey20") +
  geom_vline(xintercept = h.fig3.lm.uasndre , linetype = "solid" , size = 2.5 , color = "gray10") +
  geom_vline(xintercept = 200 , linetype = "dotted" , size = 2.5 , color = "black") +
  scale_linetype_manual("Index" , breaks = c(  "uas_NDRE" , "gs_NDVI"   ) , values = c( "solid", "dashed
  scale_shape_manual("Index" , breaks = c( "uas_NDRE" , "gs_NDVI"   ) , values = c( 1 , 4  ) , labels =
  scale_color_manual("Index" , breaks = c(     "uas_NDRE" , "gs_NDVI"  ) , values = c( "grey10" ,  "grey
  scale_fill_manual("Index" , breaks = c(    "uas_NDRE" ,  "gs_NDVI"    ) , values = c( "white"  ,  "whi
scale_alpha_manual("Index" , breaks = c(    "uas_NDRE" , "gs_NDVI"   ) , values = c( 1 ,  1 ) , labels =
  scale_size_manual("Index" , breaks = c(  "uas_NDRE"  , "gs_NDVI"  ) , values = c( 9 ,  9 ) , labels =
  annotate("text" , x = (h.fig3.lm.gsndvi + 8.5) , y = 0.5 , label = "215" , size = 11 , color = "grey1
  annotate("text" , x = (h.fig3.lm.uasndre + 9.5) , y = 0.5 , label = "240" , size = 11 , color = "gray1
  annotate("text" , x = (200 - 9.5) , y = 0.5 , label = "200" , size = 11 , color = "black" , parse = T

Fig5
```




```
ggsave("FIGURES/Fig5.pdf" , Fig5 , width = 17 , height = 12 ,  dpi = 10000)
```

FIG 6 - MODELS

GS NDVI

model

```
ctrl <- lmeControl(opt = "optim") #changes control to "optimal" settings

gs_ndvi_model <- lme(GrainYield_Mgha ~ gs_NDVI_Sufficiency_Index * TopDress_kgha   ,
             control = ctrl ,
             random = ~ I(TopDress_kgha) | site_year  ,
             data = paper2_data_OR)

summary(gs_ndvi_model)
```

```
## Linear mixed-effects model fit by REML
##   Data: paper2_data_OR
##       AIC      BIC    logLik
##   912.355 942.7961 -448.1775
##
## Random effects:
##  Formula: ~I(TopDress_kgha) | site_year
##  Structure: General positive-definite, Log-Cholesky parametrization
##                  StdDev      Corr
## (Intercept)      1.032504500 (Intr)
```

```
## I(TopDress_kgha) 0.004850162 -0.95
## Residual         0.867076906
##
## Fixed effects:  GrainYield_Mgha ~ gs_NDVI_Sufficiency_Index * TopDress_kgha
##                                        Value Std.Error  DF   t-value
## (Intercept)                         2.546983 0.5665809 327  4.495356
## gs_NDVI_Sufficiency_Index           8.400526 0.4380235 327 19.178255
## TopDress_kgha                       0.087972 0.0139089 327  6.324840
## gs_NDVI_Sufficiency_Index:TopDress_kgha -0.085524 0.0159228 327 -5.371130
##                                        p-value
## (Intercept)                                  0
## gs_NDVI_Sufficiency_Index                    0
## TopDress_kgha                                0
## gs_NDVI_Sufficiency_Index:TopDress_kgha      0
##  Correlation:
##                                        (Intr) gs_NDVI_S_I TpDrs_
## gs_NDVI_Sufficiency_Index              -0.657
## TopDress_kgha                          -0.581  0.706
## gs_NDVI_Sufficiency_Index:TopDress_kgha  0.477 -0.726      -0.973
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.04071596 -0.54764734  0.03466277  0.67613807  2.99803648
##
## Number of Observations: 336
## Number of Groups: 6
```

```
summary(gs_ndvi_model)$tTable
```

```
##                                         Value  Std.Error  DF   t-value
## (Intercept)                          2.54698292 0.56658095 327  4.495356
## gs_NDVI_Sufficiency_Index            8.40052580 0.43802347 327 19.178255
## TopDress_kgha                        0.08797181 0.01390894 327  6.324840
## gs_NDVI_Sufficiency_Index:TopDress_kgha -0.08552360 0.01592283 327 -5.371130
##                                             p-value
## (Intercept)                           9.654553e-06
## gs_NDVI_Sufficiency_Index             1.841668e-55
## TopDress_kgha                         8.325000e-10
## gs_NDVI_Sufficiency_Index:TopDress_kgha 1.488137e-07
```

```
Anova(gs_ndvi_model , type = 3)
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: GrainYield_Mgha
##                                          Chisq Df Pr(>Chisq)
## (Intercept)                             20.208  1  6.945e-06 ***
## gs_NDVI_Sufficiency_Index              367.805  1  < 2.2e-16 ***
## TopDress_kgha                           40.004  1  2.535e-10 ***
## gs_NDVI_Sufficiency_Index:TopDress_kgha  28.849  1  7.824e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
r.squaredGLMM(gs_ndvi_model)
```

```
##            R2m       R2c
```

```
## [1,] 0.4401273 0.7436078
```

```
r_sq <- r.squaredGLMM(gs_ndvi_model)

r_sq_fixed <- round(r_sq[1] , digits = 2)
r_sq_fixed
```

```
## [1] 0.44
```

```
r_sq_total <- round(r_sq[2] , digits = 2)
r_sq_total
```

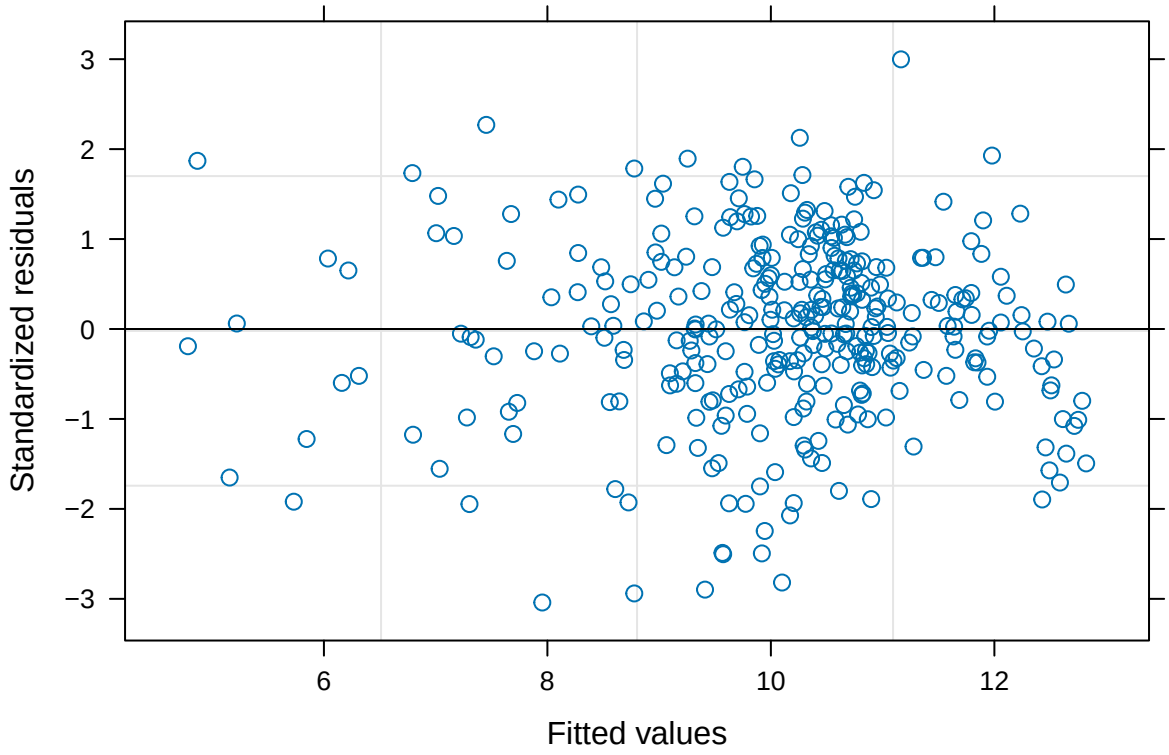
```
## [1] 0.74
```

```
r_sq_random <- r_sq_total - r_sq_fixed
r_sq_random
```

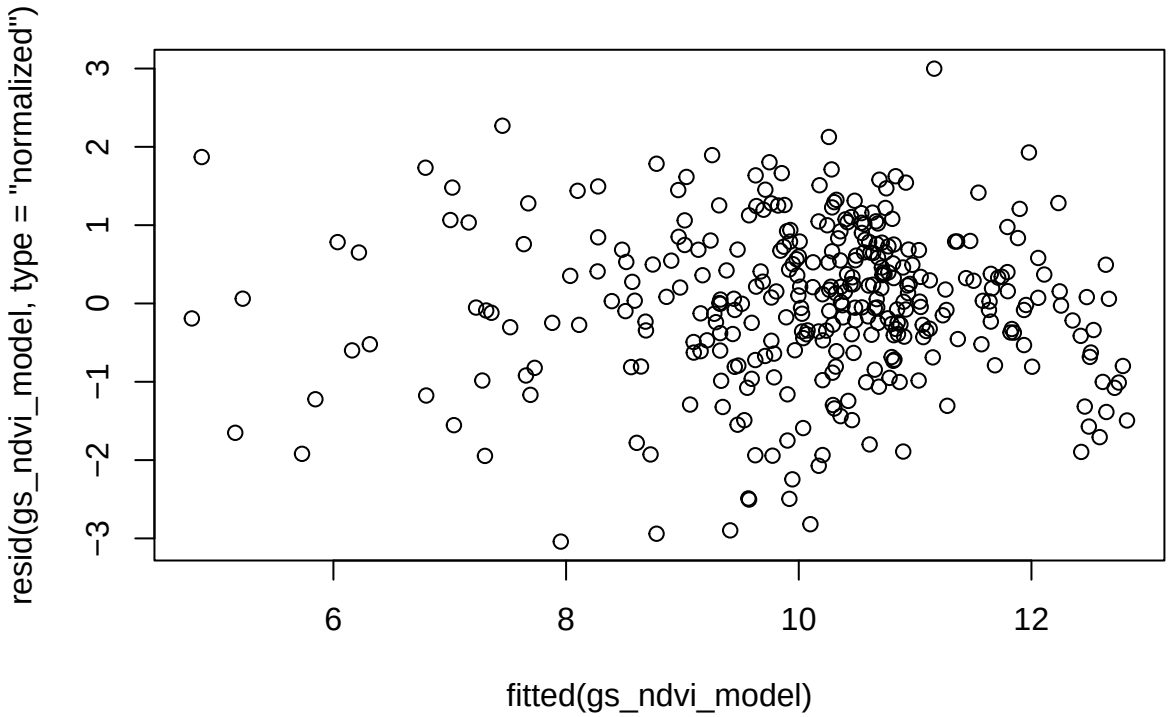
```
## [1] 0.3
```

diagnostics

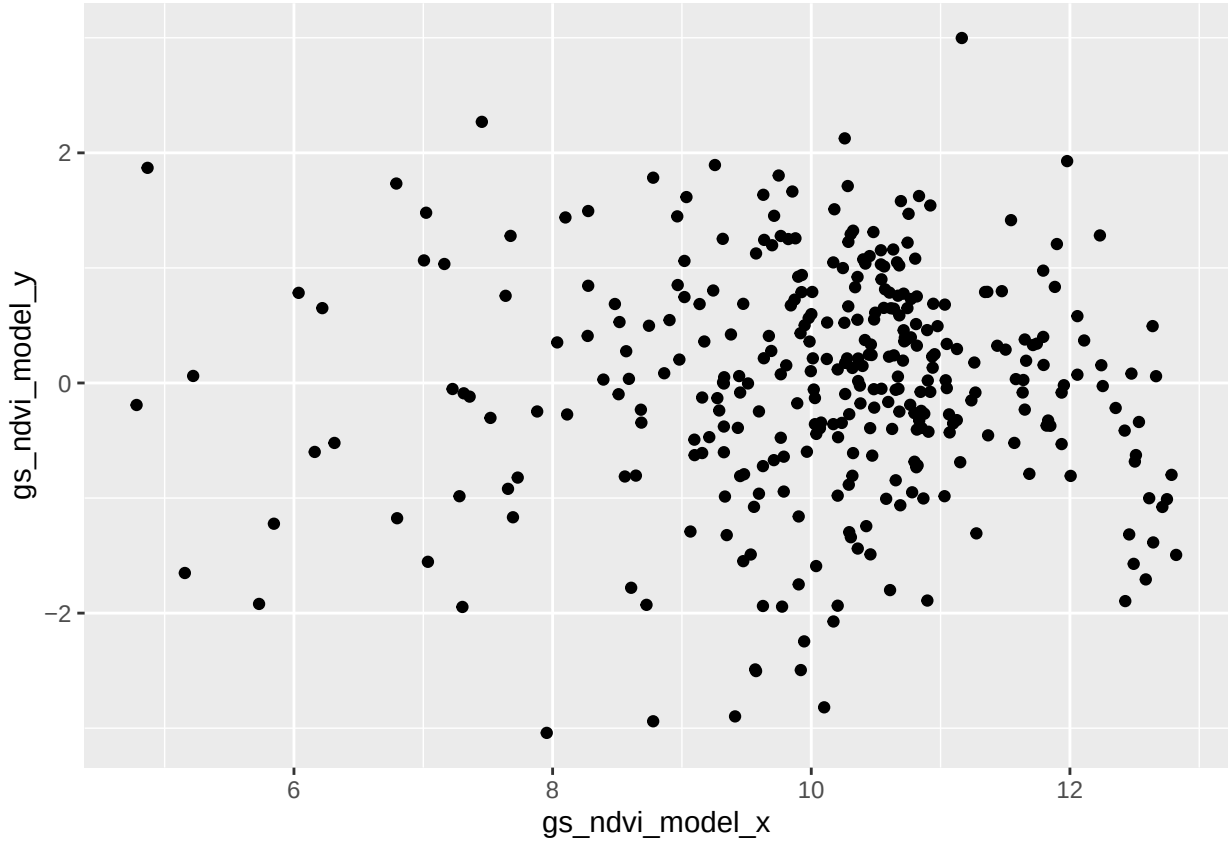
```
plot (gs_ndvi_model)
```



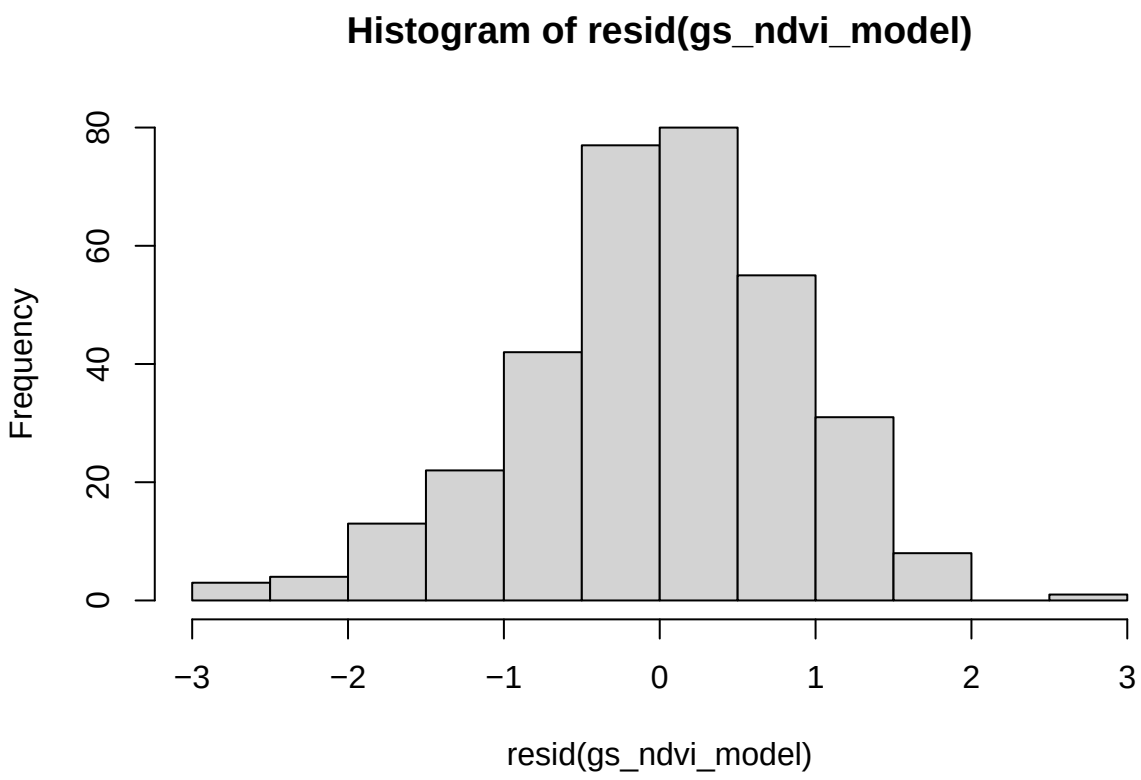

```
plot(resid(gs_ndvi_model, type = "normalized") ~fitted(gs_ndvi_model))
```



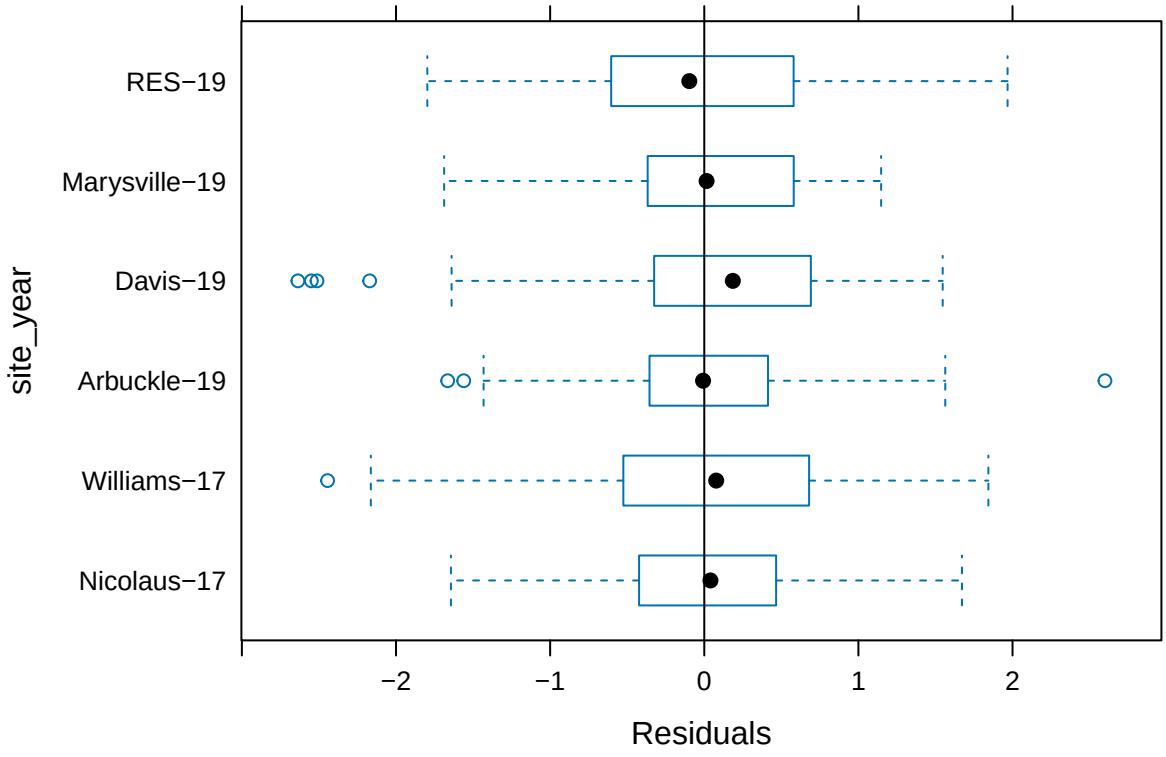

```
gs_ndvi_model_y <- resid(gs_ndvi_model, type = "normalized")
gs_ndvi_model_x <- fitted(gs_ndvi_model)

gs_ndvi_modelresid_data <- data.frame(gs_ndvi_model_x , gs_ndvi_model_y)

ggplot( data = gs_ndvi_modelresid_data , aes( x = gs_ndvi_model_x , y = gs_ndvi_model_y)) +
  geom_point(mapping = aes(gs_ndvi_model_x , gs_ndvi_model_y) , data = gs_ndvi_modelresid_data)
```



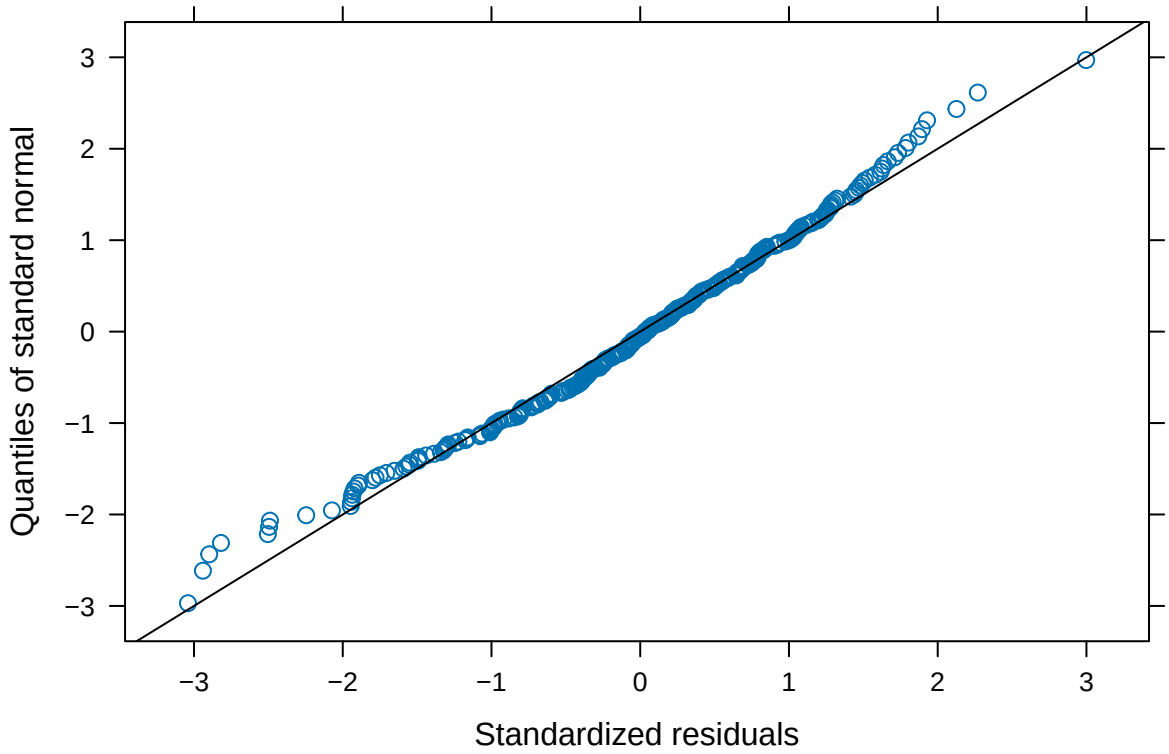

```
hist(resid(gs_ndvi_model))
```



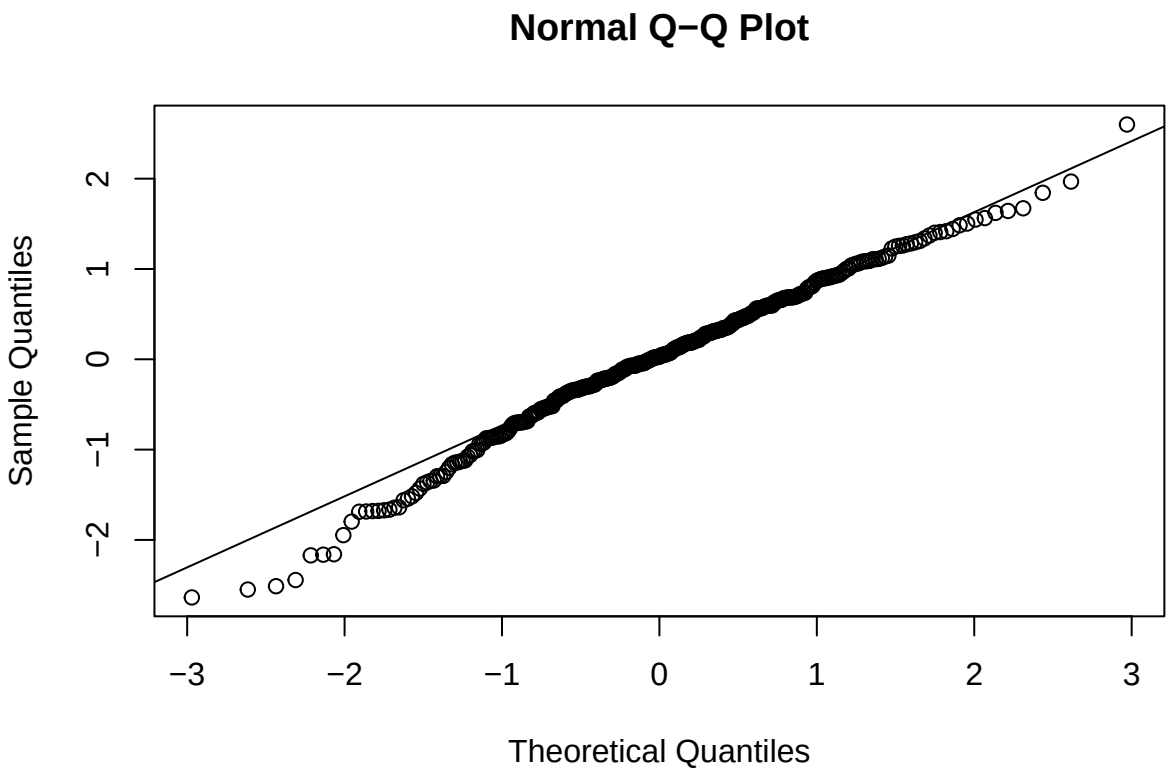
```
plot(gs_ndvi_model, site_year ~ resid(.), abline = 0)
```



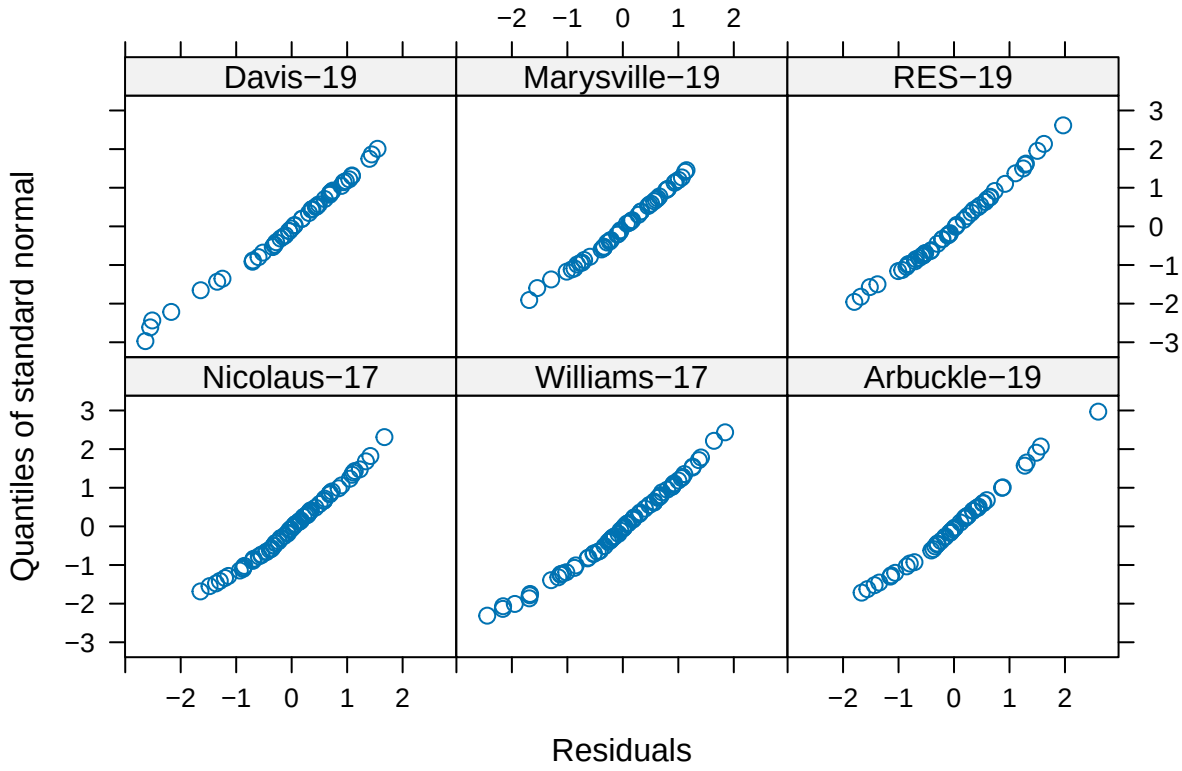

```
qqnorm(gs_ndvi_model, abline = c(0,1) )
```



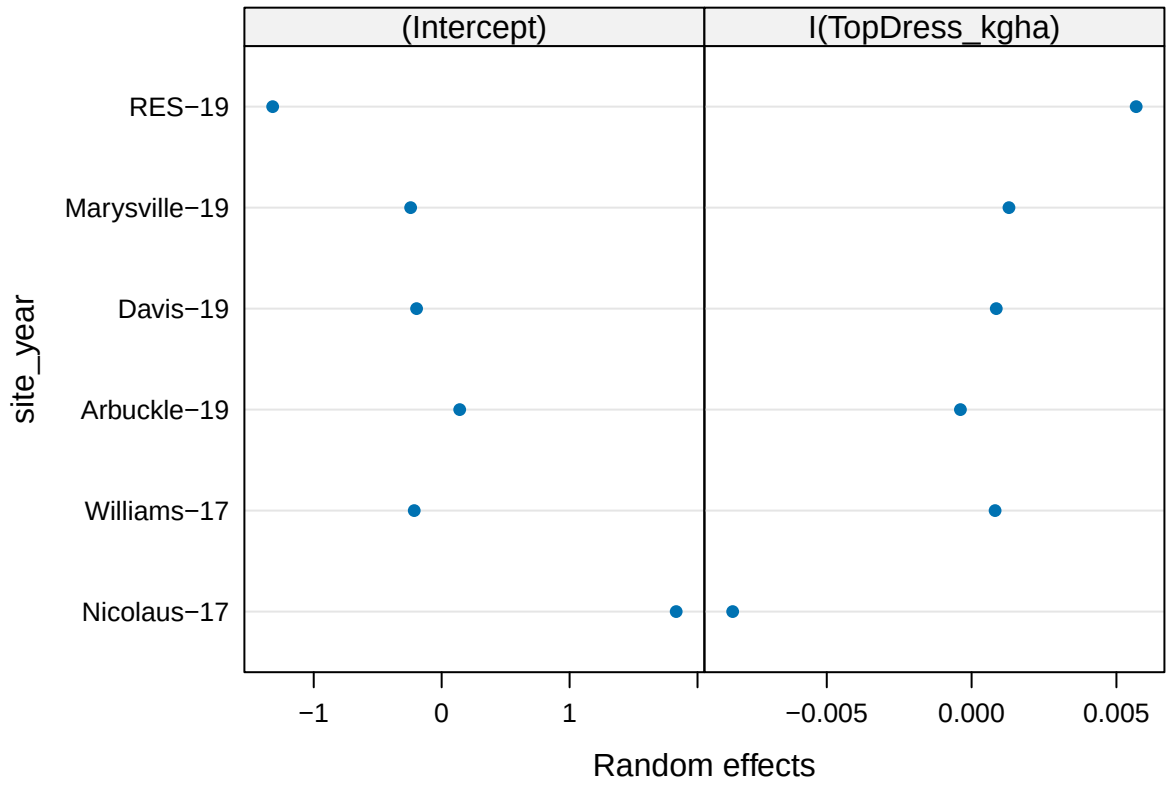

```
qqnorm(resid(gs_ndvi_model))
qqline(resid(gs_ndvi_model))
```



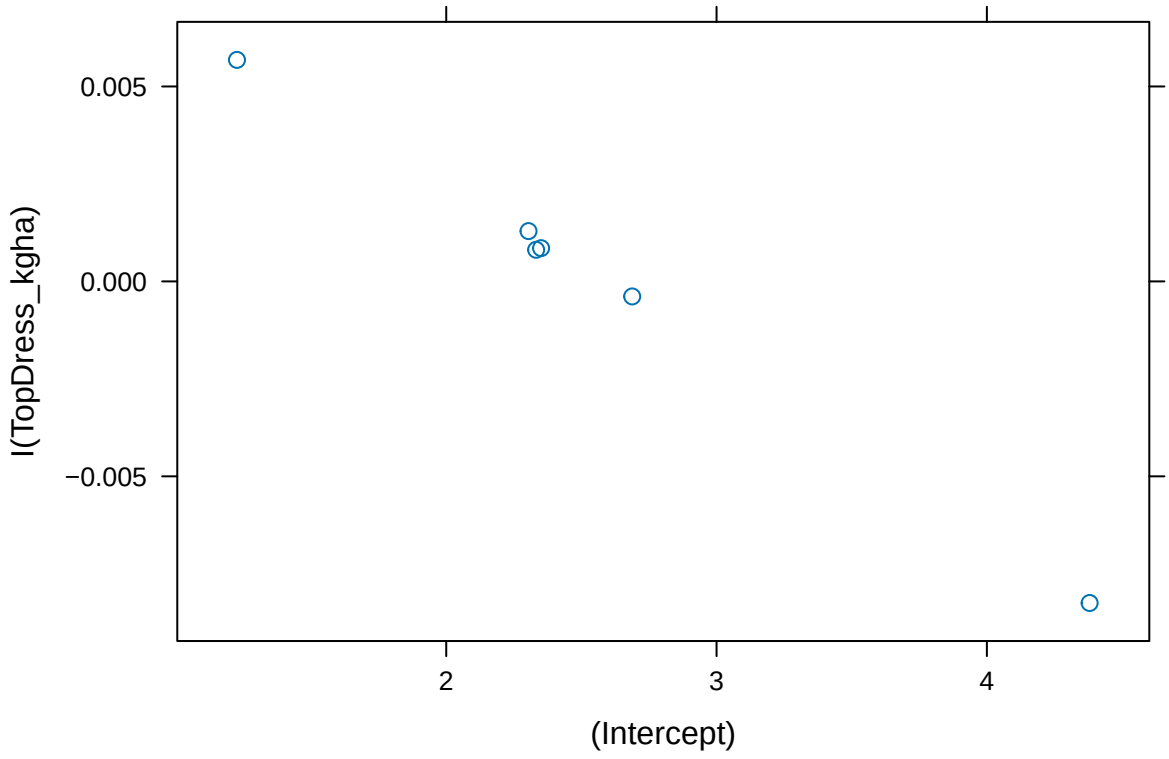

```
qqnorm(gs_ndvi_model , ~resid(.) | site_year)
```



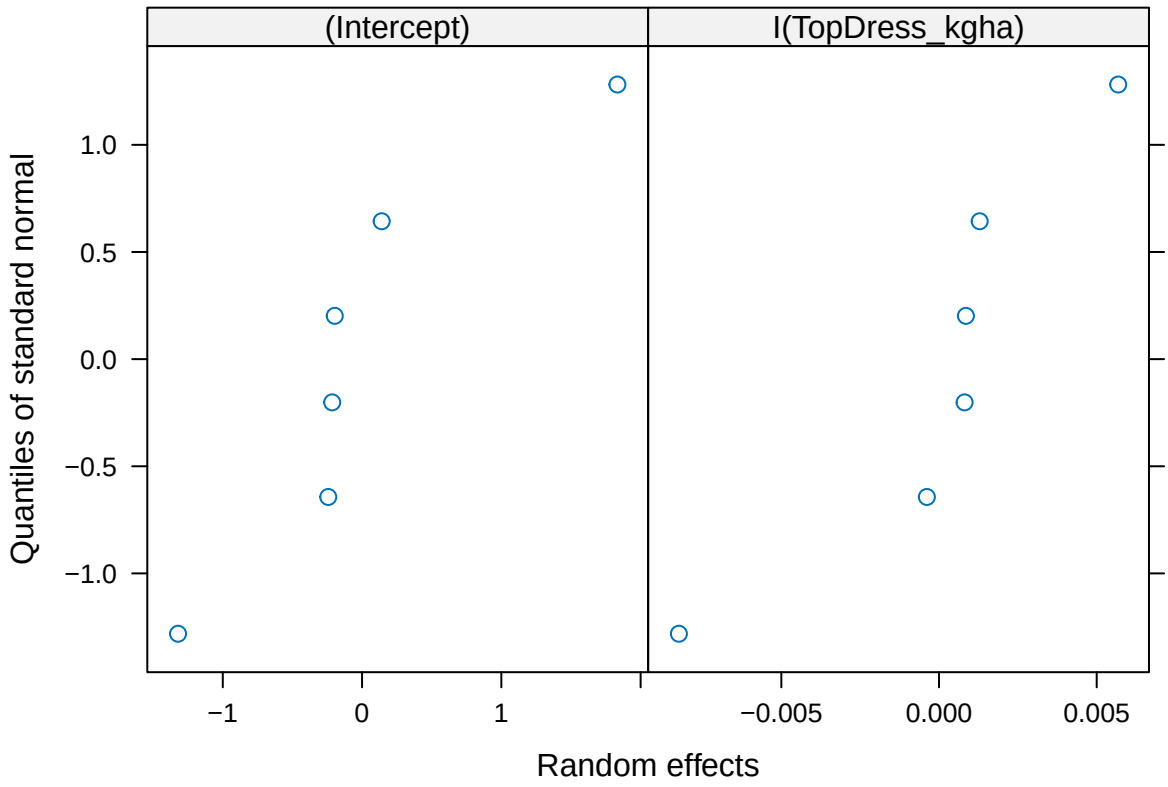

```
plot(ranef(gs_ndvi_model))
```



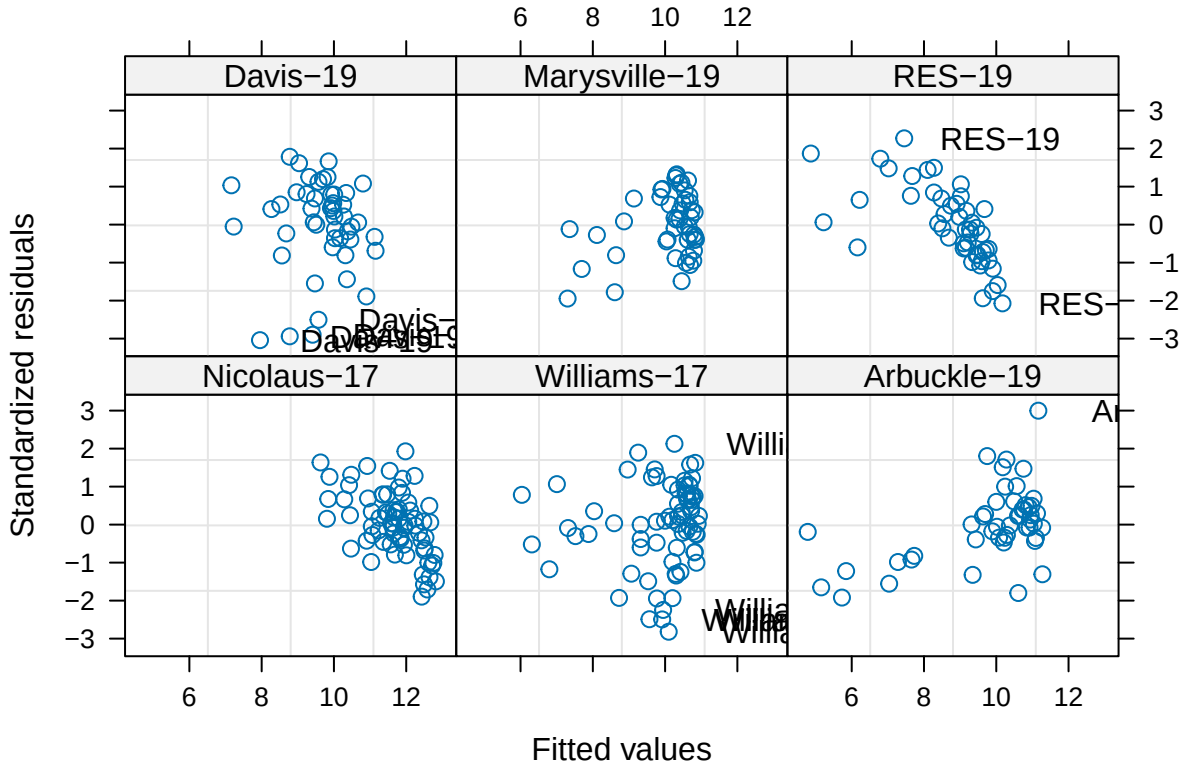

```
pairs(gs_ndvi_model , id = 0.1)
```




```
qqnorm(gs_ndvi_model , ~ranef(.))
```




```
plot( gs_ndvi_model, resid(., type = "p") ~ fitted(.) | site_year,
      id = 0.05, adj = -0.3 )
```



emmeans

```
mylist <- list(gs_NDVI_Sufficiency_Index=seq(round(min(paper2_data_OR$gs_NDVI_Sufficiency_Index) , digi

gs_ndvi_emmeans <- emmeans(gs_ndvi_model, ~TopDress_kgha * gs_NDVI_Sufficiency_Index , at = mylist )

gs_ndvi_emmeans_contrast <- as.data.frame(summary(contrast(gs_ndvi_emmeans , "pairwise" , side = ">" , l

gs_ndvi_emmeans_contrast <- gs_ndvi_emmeans_contrast %>%
  mutate(prob_postive_resp = (1 - p.value) *100) #calculates probability of positive yield response

gs_ndvi_emmeans_contrast$prob_postive_resp <- round(gs_ndvi_emmeans_contrast$prob_postive_resp , digits

gs_ndvi_emmeans_contrast <- gs_ndvi_emmeans_contrast %>%
  mutate(t_score = abs((estimate - 0.30)) / SE) #calculates t-score of response being greater than 0.30

gs_ndvi_emmeans_contrast <- gs_ndvi_emmeans_contrast %>%
  mutate(prob_economical_resp = if_else(estimate < 0.30 , pt(q = t_score , df = df , lower.tail = F) , p

gs_ndvi_emmeans_contrast$prob_economical_resp <- gs_ndvi_emmeans_contrast$prob_economical_resp * 100 # t

gs_ndvi_emmeans_contrast$prob_economical_resp <- round(gs_ndvi_emmeans_contrast$prob_economical_resp, di

gs_ndvi_emmeans_contrast <- gs_ndvi_emmeans_contrast %>%
  mutate(Yield_Response = round(estimate , digits = 2),
         Standard_Error = round(SE , digits = 2)) #round yield response to 2 decimal places and renames

gs_ndvi_emmeans_contrast <- gs_ndvi_emmeans_contrast %>%
  mutate(gs_NDVI_Sufficiency_Index_r = round(gs_NDVI_Sufficiency_Index , digits = 2)) %>%
  dplyr::select(-t_score) #new column with NDVI SI rounded to 2 digits and removes t score column

gs_ndvi_rise <-  max(gs_ndvi_emmeans_contrast$estimate) - min(gs_ndvi_emmeans_contrast$estimate)

gs_ndvi_run <- round(min(paper2_data_OR$gs_NDVI_Sufficiency_Index) , digits = 2) - round(max(paper2_data

gs_ndvi_slope <- gs_ndvi_rise / gs_ndvi_run

gs_ndvi_slope #slope per 1 unit SI. Need to divide by 10 to get slope for every 0.1 unit of SI.
```

```
## [1] -2.907802
```

```
gs_ndvi_slope <- round((gs_ndvi_slope / 10) , digits = 2)

gs_ndvi_slope
```

```
## [1] -0.29
```

```
gs_ndvi_mean_se <- gs_ndvi_emmeans_contrast %>%
  dplyr::select(SE) %>%
  summarise(mean_se = mean(SE))

gs_ndvi_mean_se$mean_se <- round(gs_ndvi_mean_se$mean_se , digits = 2)

gs_ndvi_mean_se$mean_se
```

```
## [1] 0.19
```

UAS NDRE

model

```
UAS_ndre_model <- lme(GrainYield_Mgha ~ uas_NDRE_Sufficiency_Index * TopDress_kgha,
             control = ctrl ,
             random = ~ I(TopDress_kgha) | site_year  ,
             data = paper2_data_OR)

summary(UAS_ndre_model)
```

```
## Linear mixed-effects model fit by REML
##   Data: paper2_data_OR
##        AIC      BIC    logLik
##   890.2701 920.7112 -437.1351
##
## Random effects:
##  Formula: ~I(TopDress_kgha) | site_year
##  Structure: General positive-definite, Log-Cholesky parametrization
##                  StdDev      Corr
## (Intercept)      0.882484995 (Intr)
## I(TopDress_kgha) 0.003084929 -0.887
## Residual         0.841557771
##
## Fixed effects:  GrainYield_Mgha ~ uas_NDRE_Sufficiency_Index * TopDress_kgha
##                                         Value Std.Error  DF   t-value
## (Intercept)                          -0.293921 0.6187216 327 -0.475045
## uas_NDRE_Sufficiency_Index           11.418633 0.5693582 327 20.055271
## TopDress_kgha                         0.110655 0.0187365 327  5.905872
## uas_NDRE_Sufficiency_Index:TopDress_kgha -0.108834 0.0210261 327 -5.176154
##                                         p-value
## (Intercept)                              0.6351
## uas_NDRE_Sufficiency_Index               0.0000
## TopDress_kgha                            0.0000
## uas_NDRE_Sufficiency_Index:TopDress_kgha  0.0000
##  Correlation:
##                                         (Intr) us_NDRE_S_I TpDrs_
## uas_NDRE_Sufficiency_Index              -0.806
## TopDress_kgha                           -0.617  0.709
## uas_NDRE_Sufficiency_Index:TopDress_kgha  0.579 -0.717      -0.989
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -4.29346952 -0.60486574  0.05835305  0.69716981  2.92307024
##
## Number of Observations: 336
## Number of Groups: 6
```

```
summary(UAS_ndre_model)$tTable
```

```
##                                          Value  Std.Error  DF    t-value
## (Intercept)                          -0.2939207 0.61872164 327 -0.4750451
## uas_NDRE_Sufficiency_Index           11.4186328 0.56935819 327 20.0552709
## TopDress_kgha                         0.1106554 0.01873651 327  5.9058719
## uas_NDRE_Sufficiency_Index:TopDress_kgha -0.1088345 0.02102613 327 -5.1761545
```

```
##                                          p-value
## (Intercept)                            6.350720e-01
## uas_NDRE_Sufficiency_Index             6.671794e-59
## TopDress_kgha                          8.792474e-09
## uas_NDRE_Sufficiency_Index:TopDress_kgha 3.962794e-07
```

```
Anova(UAS_ndre_model , type = 3)
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: GrainYield_Mgha
##                                            Chisq Df Pr(>Chisq)
## (Intercept)                                0.2257  1     0.6348
## uas_NDRE_Sufficiency_Index               402.2139  1  < 2.2e-16 ***
## TopDress_kgha                             34.8793  1  3.508e-09 ***
## uas_NDRE_Sufficiency_Index:TopDress_kgha  26.7926  1  2.265e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
r.squaredGLMM(UAS_ndre_model)
```

```
##            R2m       R2c
## [1,] 0.4971712 0.7448585
```

```
r_sq <- r.squaredGLMM(UAS_ndre_model)

r_sq_fixed <- round(r_sq[1] , digits = 2)
r_sq_fixed
```

```
## [1] 0.5
```

```
r_sq_total <- round(r_sq[2] , digits = 2)
r_sq_total
```

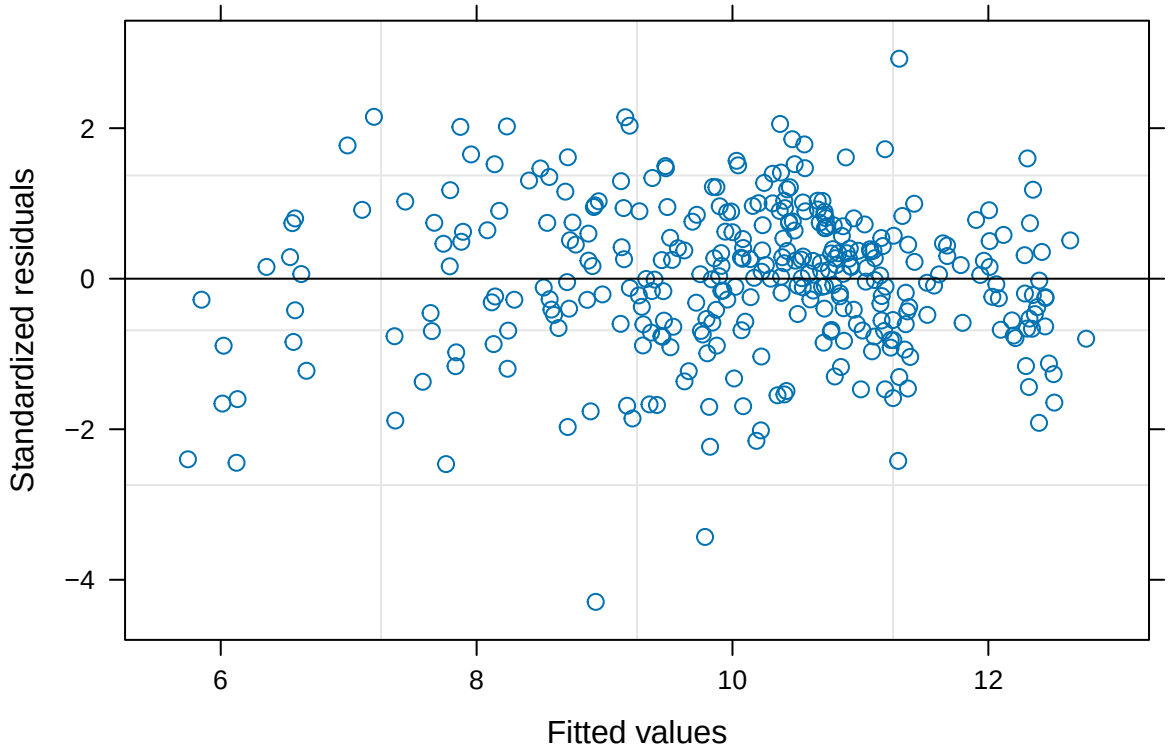
```
## [1] 0.74
```

```
r_sq_random <- r_sq_total - r_sq_fixed
r_sq_random
```

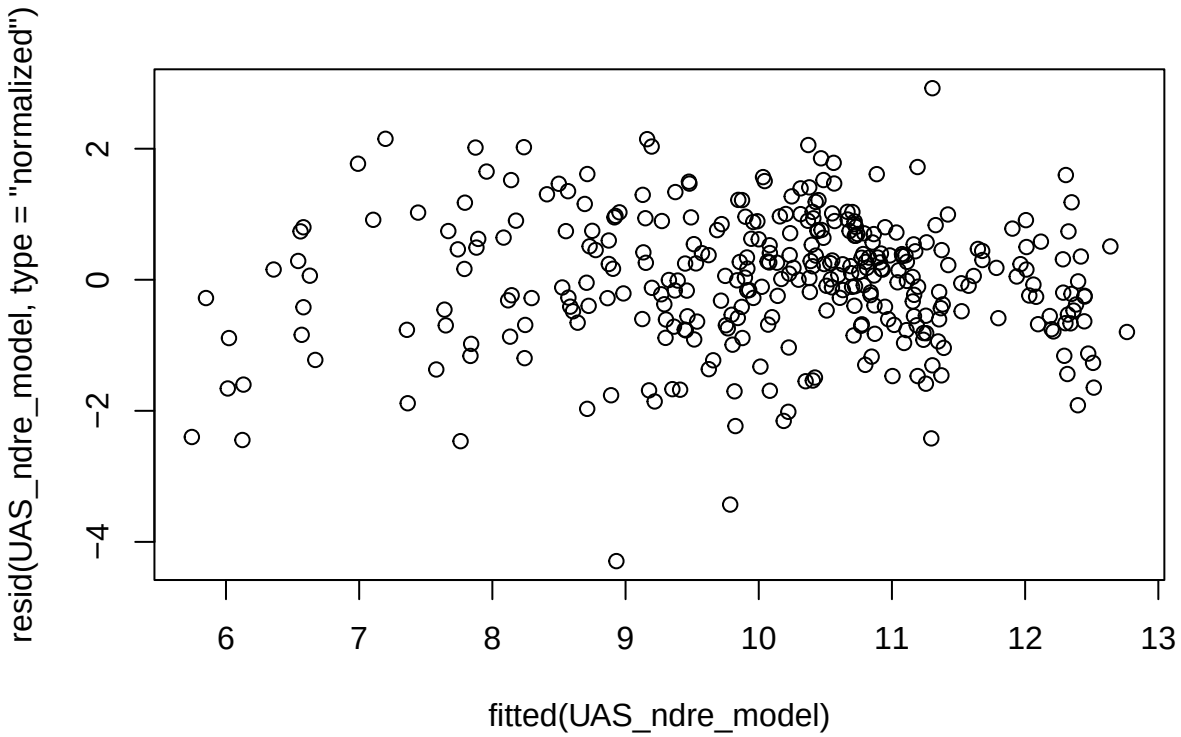
```
## [1] 0.24
```

diagnostics

```
plot (UAS_ndre_model)
```

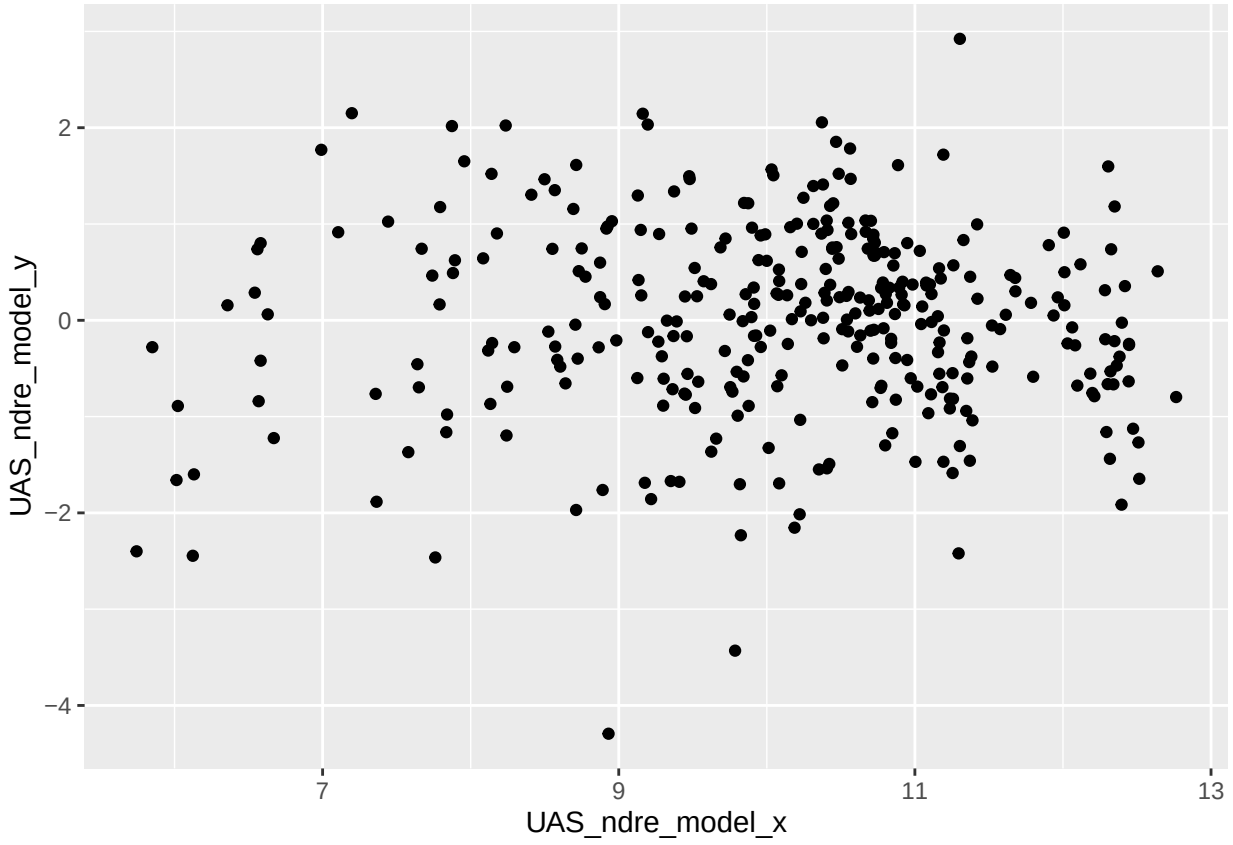



```
plot(resid(UAS_ndre_model, type = "normalized") ~fitted(UAS_ndre_model))
```

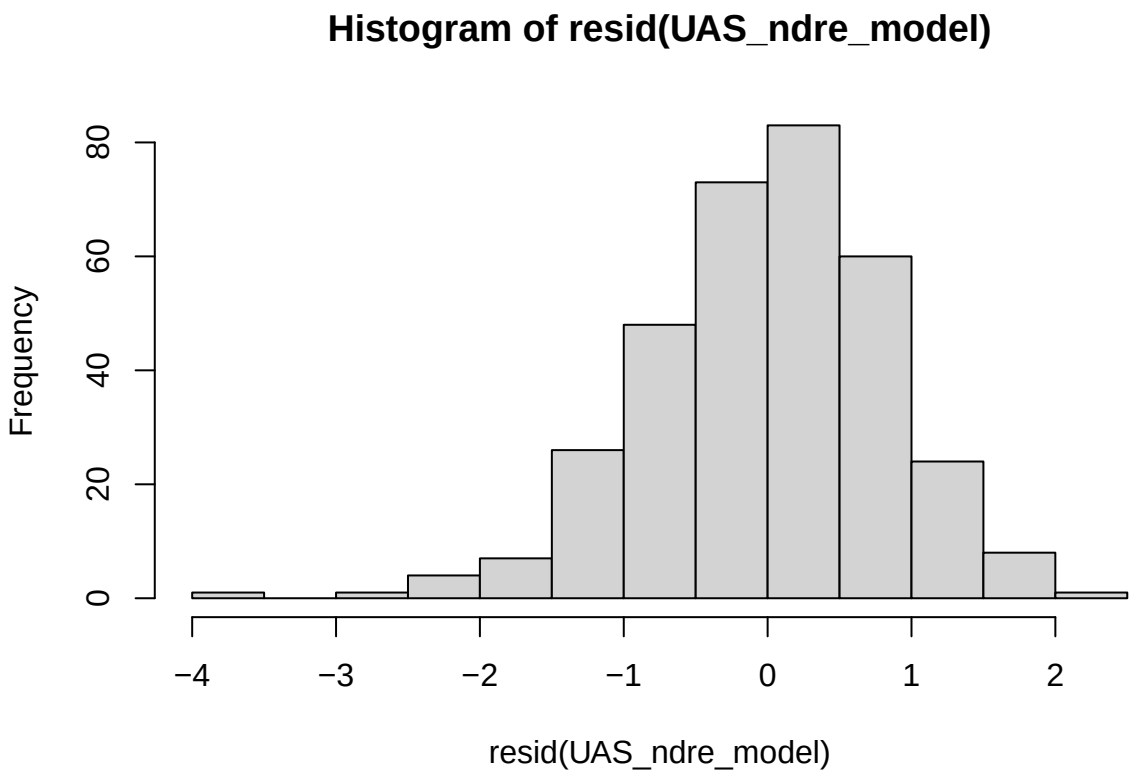



```
UAS_ndre_model_y <- resid(UAS_ndre_model, type = "normalized")
UAS_ndre_model_x <- fitted(UAS_ndre_model)

UAS_ndre_modelresid_data <- data.frame(UAS_ndre_model_x , UAS_ndre_model_y)
```

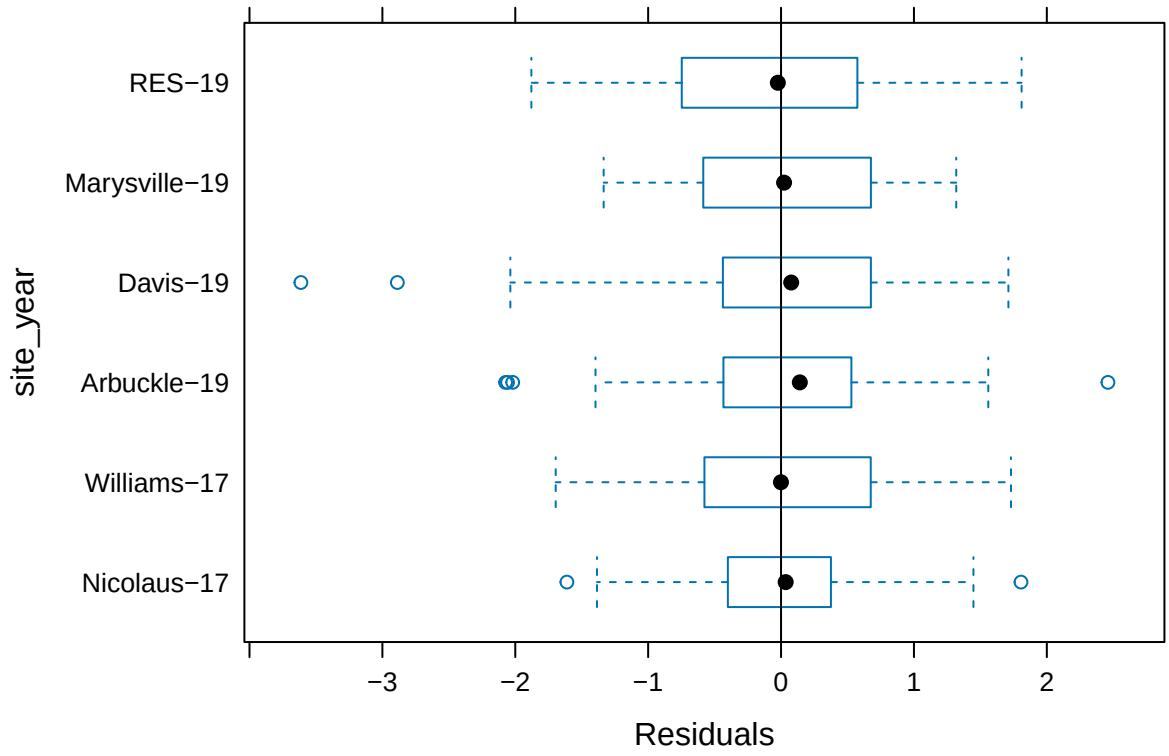
```
ggplot( data = UAS_ndre_modelresid_data , aes( x = UAS_ndre_model_x , y = UAS_ndre_model_y)) +
  geom_point(mapping = aes(UAS_ndre_model_x , UAS_ndre_model_y) , data = UAS_ndre_modelresid_data)
```



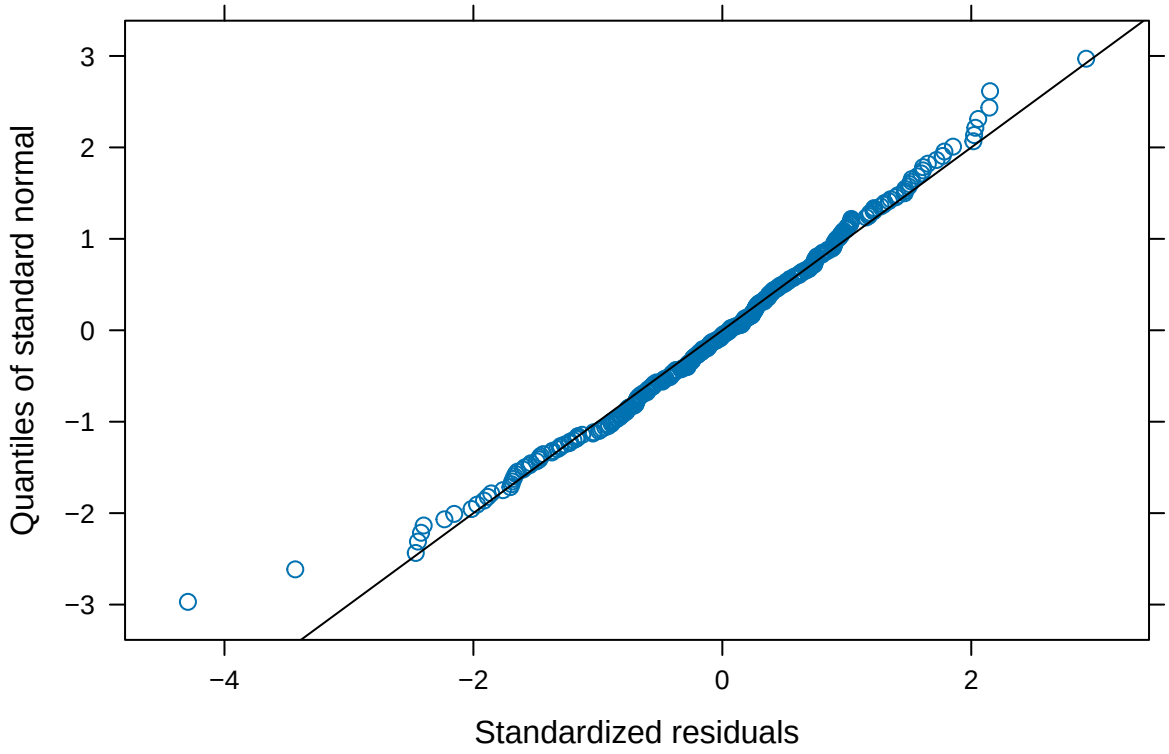

```
hist(resid(UAS_ndre_model))
```



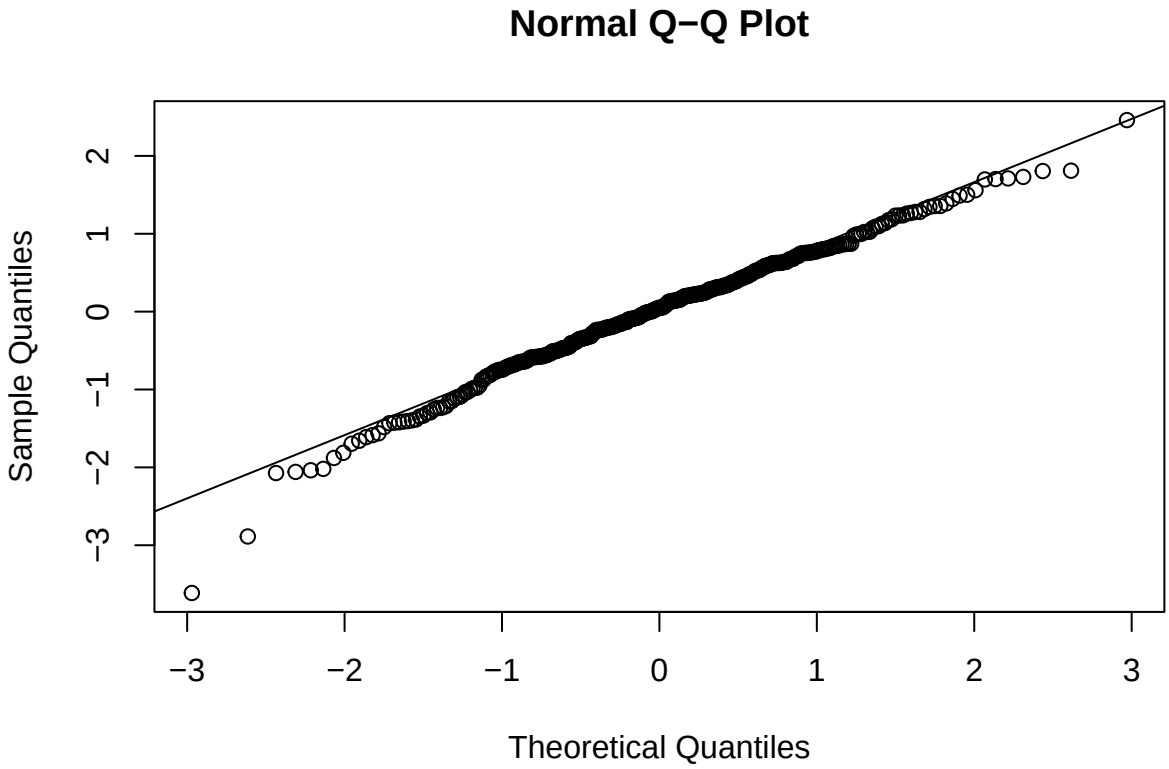

```
plot(UAS_ndre_model, site_year ~ resid(.), abline = 0)
```



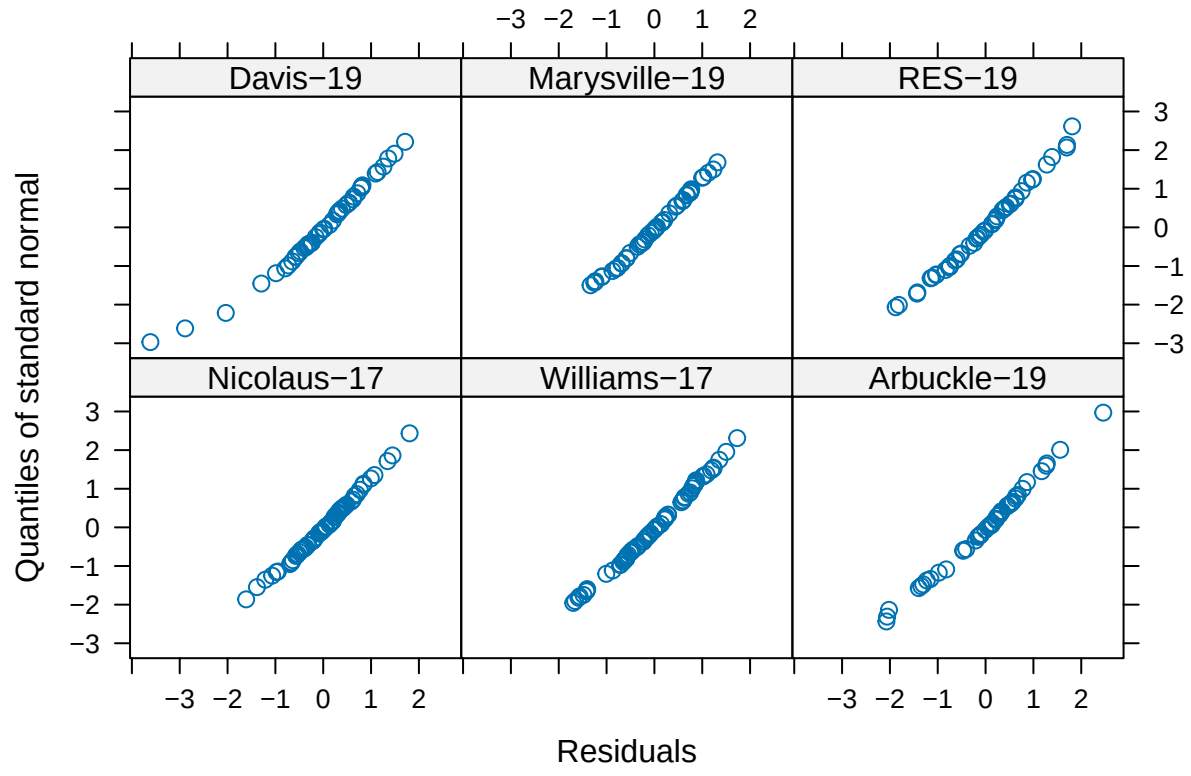

```
qqnorm(UAS_ndre_model, abline = c(0,1) )
```



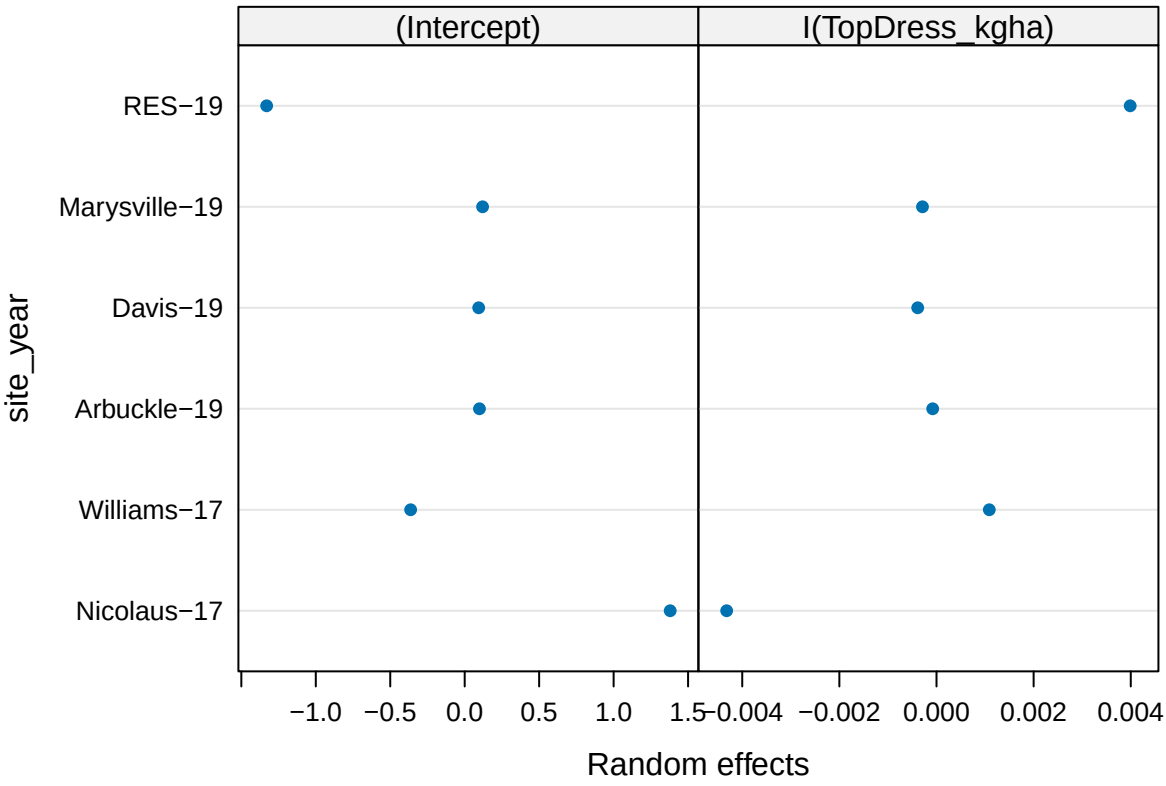

```
qqnorm(resid(UAS_ndre_model))
qqline(resid(UAS_ndre_model))
```



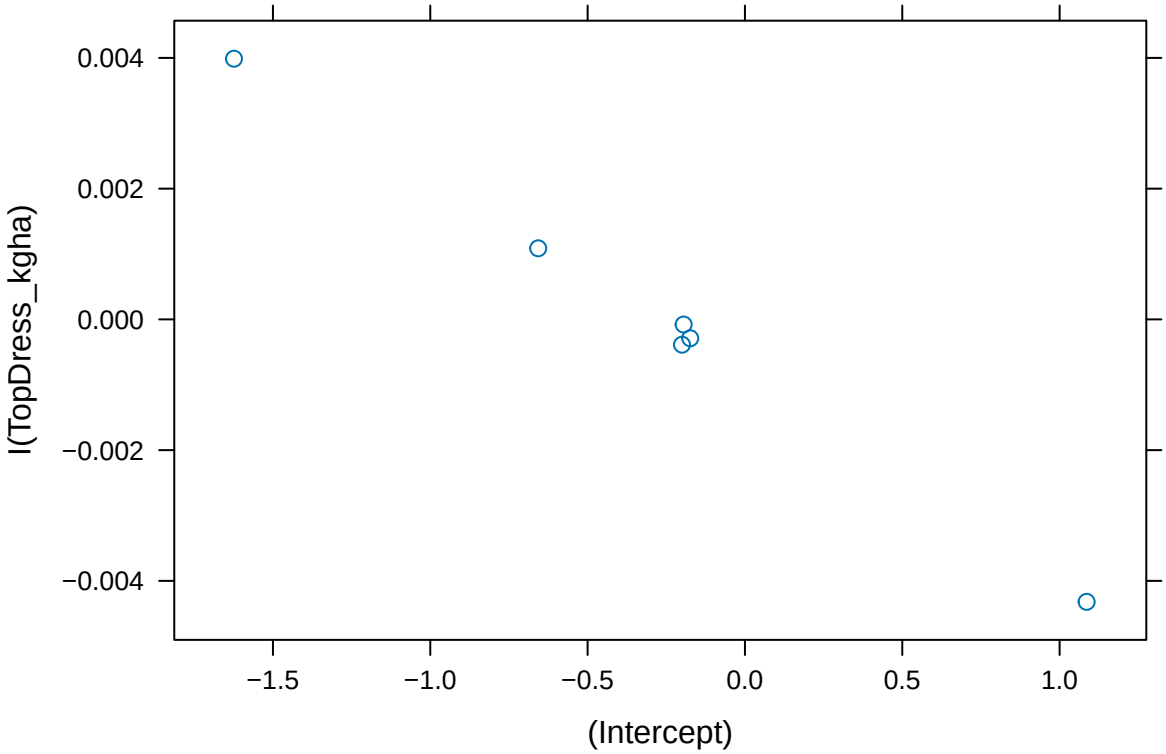

```
qqnorm(UAS_ndre_model , ~resid(.) | site_year)
```



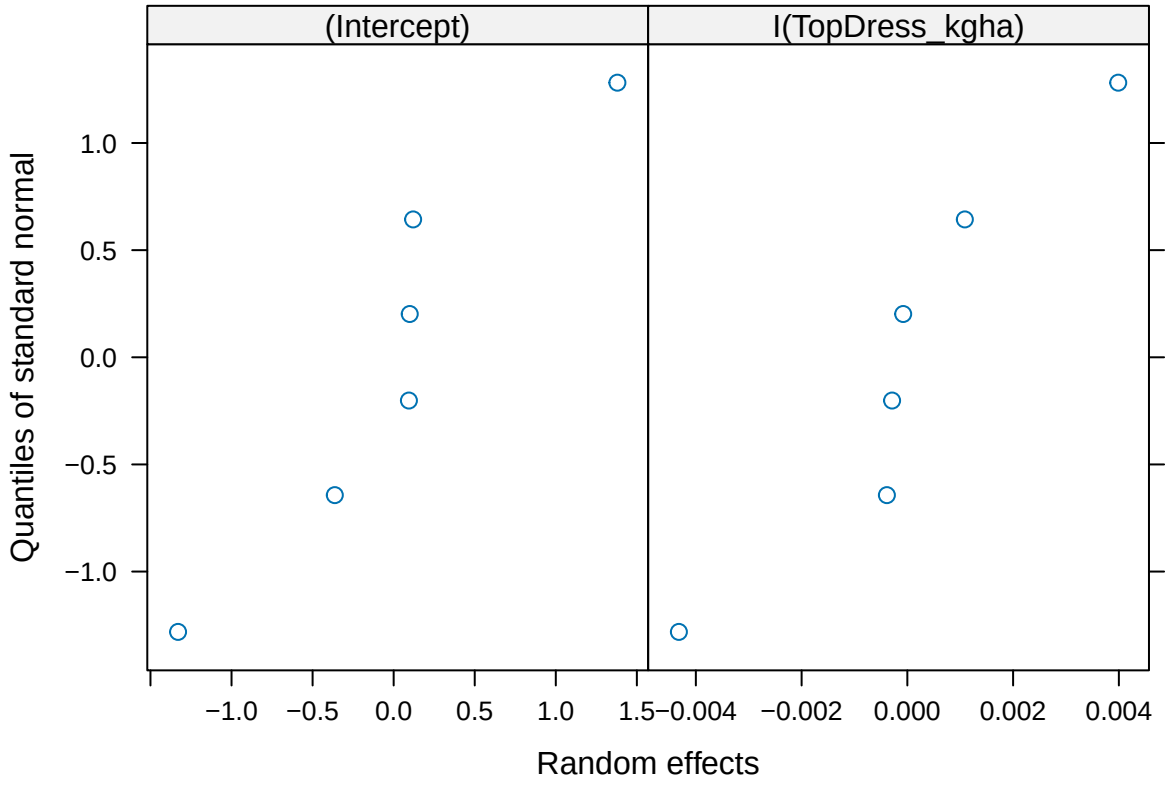

```
plot(ranef(UAS_ndre_model))
```



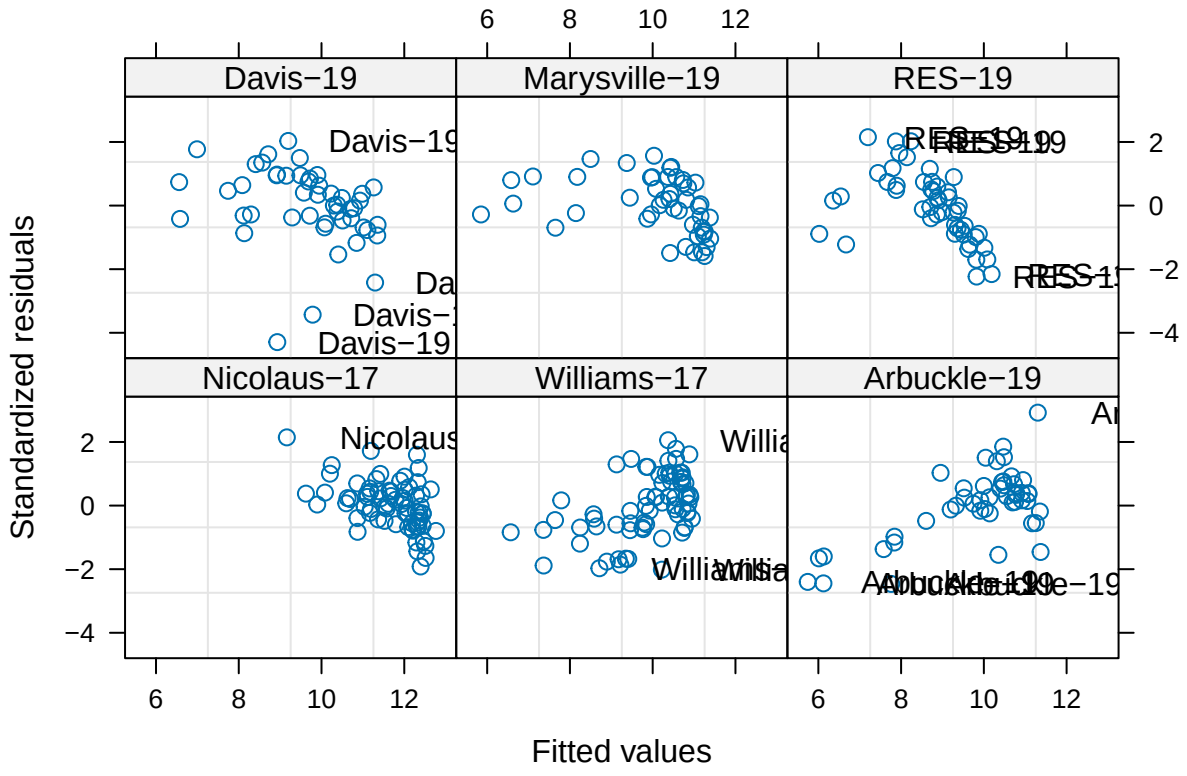

```
pairs(UAS_ndre_model , id = 0.1)
```




```
qqnorm(UAS_ndre_model , ~ranef(.))
```




```
plot( UAS_ndre_model, resid(., type = "p") ~ fitted(.) | site_year,
      id = 0.05, adj = -0.3 )
```



emmeans

```
mylist <- list(uas_NDRE_Sufficiency_Index=seq(round(min(paper2_data_OR$uas_NDRE_Sufficiency_Index) , dig

UAS_ndre_emmeans <- emmeans(UAS_ndre_model , ~TopDress_kgha * uas_NDRE_Sufficiency_Index , at = mylist )

UAS_ndre_emmeans_contrast <- as.data.frame(summary(contrast(UAS_ndre_emmeans , "pairwise" , side = ">"

UAS_ndre_emmeans_contrast <- UAS_ndre_emmeans_contrast %>%
  mutate(prob_postive_resp = (1 - p.value) *100) #calculates probability of positive yield response

UAS_ndre_emmeans_contrast$prob_postive_resp <- round(UAS_ndre_emmeans_contrast$prob_postive_resp , digi

UAS_ndre_emmeans_contrast <- UAS_ndre_emmeans_contrast %>%
  mutate(t_score = abs((estimate - 0.30)) / SE) #calculates t-score of response being greater than 0.30

UAS_ndre_emmeans_contrast <- UAS_ndre_emmeans_contrast %>%
  mutate(prob_economical_resp = if_else(estimate < 0.30 , pt(q = t_score , df = df , lower.tail = F) , 

UAS_ndre_emmeans_contrast$prob_economical_resp <- UAS_ndre_emmeans_contrast$prob_economical_resp * 100 

UAS_ndre_emmeans_contrast$prob_economical_resp <- round(UAS_ndre_emmeans_contrast$prob_economical_resp,

UAS_ndre_emmeans_contrast <- UAS_ndre_emmeans_contrast %>%
  mutate(Yield_Response = round(estimate , digits = 2),
         Standard_Error = round(SE , digits = 2)) #round yield response to 2 decimal places and renames

UAS_ndre_emmeans_contrast <- UAS_ndre_emmeans_contrast %>%
```

```
  mutate(uas_NDRE_Sufficiency_Index_r = round(uas_NDRE_Sufficiency_Index , digits = 2)) %>%
  dplyr::select(-t_score) #new column with NDRE SI rounded to 2 digits and removes t score column

UAS_ndre_rise <-  max(UAS_ndre_emmeans_contrast$estimate) - min(UAS_ndre_emmeans_contrast$estimate)

UAS_ndre_run <- round(min(paper2_data_OR$uas_NDRE_Sufficiency_Index) , digits = 2) - round(max(paper2_da

UAS_ndre_slope <- UAS_ndre_rise / UAS_ndre_run

UAS_ndre_slope #slope per 1 unit SI. Need to divide by 10 to get slope for every 0.1 unit of SI.
```

```
## [1] -3.700372
```

```
UAS_ndre_slope <- round((UAS_ndre_slope / 10) , digits = 2)

UAS_ndre_slope
```

```
## [1] -0.37
```

```
UAS_ndre_mean_se <- UAS_ndre_emmeans_contrast %>%
  dplyr::select(SE) %>%
  summarise(mean_se = mean(SE))

UAS_ndre_mean_se$mean_se <- round(UAS_ndre_mean_se$mean_se , digits = 2)

UAS_ndre_mean_se$mean_se
```

```
## [1] 0.15
```

FIG 6 - PLOTS

Results for Paper

```
test <- paper2_data_OR %>%
  filter(N_level_kgha >= 150 & N_level_kgha <= 200 & TopDress_kgha == 0) %>%
  dplyr::select(site_year , exp_plot_number , N_level_kgha , TopDress_kgha , gs_NDVI_Sufficiency_Index

print(min(test$gs_NDVI_Sufficiency_Index))
```

```
## [1] 0.7412587
```

```
print(max(test$gs_NDVI_Sufficiency_Index))
```

```
## [1] 1.006218
```

```
print(min(test$uas_NDRE_Sufficiency_Index))
```

```
## [1] 0.8487227
```

```
print(max(test$uas_NDRE_Sufficiency_Index))
```

```
## [1] 1.003624
```

```
round(print(sum(test$gs_NDVI_Sufficiency_Index > 0.70 & test$gs_NDVI_Sufficiency_Index < 1) / length(tes
```

```
## [1] 91.66667
## [1] 92
```

```
round(print(sum(test$uas_NDRE_Sufficiency_Index > 0.70 & test$uas_NDRE_Sufficiency_Index < 1) / length(t
```

```
## [1] 95.83333
```

```
## [1] 96
```

GS NDVI

```
mylist <- list(gs_NDVI_Sufficiency_Index = seq( 0.7 , 1 , by = 0.1 ) , TopDress_kgha = c(34 , 0))

benefit_gs_ndvi <- emmeans(gs_ndvi_model , ~TopDress_kgha*gs_NDVI_Sufficiency_Index , at=mylist)

benefit_gs_ndvi <- as.data.frame(summary(contrast(benefit_gs_ndvi , "pairwise" , side = ">" , by = "gs_

benefit_gs_ndvi <- benefit_gs_ndvi %>%
  mutate(Platform = factor("NDVI[GS]"),
         Sufficiency_Index = factor(round(gs_NDVI_Sufficiency_Index , digits = 3)) ,
         Yield_Mg = round(estimate , digits = 3),
         SE_Mg = round(SE , digits = 3),
         Probability = round(((1 - p.value)*100), digits = 2))

benefit_gs_ndvi <- benefit_gs_ndvi %>%
  dplyr::select(Platform , Yield_Mg , SE_Mg , Sufficiency_Index , Probability)
```

UAS NDRE

```
mylist <- list(uas_NDRE_Sufficiency_Index = seq( 0.7 , 1 , by = 0.1 ) , TopDress_kgha = c(34 , 0))

benefit_uas_NDRE <- emmeans(UAS_ndre_model,  ~TopDress_kgha | uas_NDRE_Sufficiency_Index , at = mylist)

benefit_uas_NDRE <- as.data.frame(summary(contrast(benefit_uas_NDRE , "pairwise" , side = ">" , by = "ua


benefit_uas_NDRE <- benefit_uas_NDRE %>%
  mutate(Platform = factor("NDRE[UAS]"),
         Sufficiency_Index = factor(round(uas_NDRE_Sufficiency_Index , digits = 3)) ,
         Yield_Mg = round(estimate , digits = 3),
         SE_Mg = round(SE , digits = 3) ,
         Probability = round(((1 - p.value)*100), digits = 2))

benefit_uas_NDRE <- benefit_uas_NDRE %>%
  dplyr::select(Platform , Yield_Mg , SE_Mg , Sufficiency_Index , Probability )
```

W/O SI

```
mylist <- list(gs_NDVI_Sufficiency_Index = seq( 0.7 , 1.0 , by = 0.1 ) , TopDress_kgha = c(34 , 0))

benefit_without_si <- emmeans(gs_ndvi_model , ~TopDress_kgha , at = mylist)
```

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

```
benefit_without_si <- as.data.frame(summary(contrast(benefit_without_si , "pairwise" , side = ">" )))

benefit_without_si <- benefit_without_si %>%
```

```
  mutate(Platform = factor( "Without SI" ) ,
         Yield_Mg = round(estimate , digits = 2),
         SE_Mg = round(SE , digits = 2),
         Sufficiency_Index = factor(0) ,
         Probability = round(((1 - p.value)*100), digits = 2)) %>%
  dplyr::select(Platform , Yield_Mg , SE_Mg , Sufficiency_Index , Probability)

print(benefit_without_si$Yield_Mg)
```

```
## [1] 0.52
```

```
print(benefit_without_si$SE_Mg)
```

```
## [1] 0.11
```

```
benefit_data1 <- rbind(benefit_gs_ndvi , benefit_uas_NDRE , benefit_without_si)
benefit_data2 <- rbind( benefit_uas_NDRE , benefit_gs_ndvi  )

benefit_data2 <- benefit_data2 %>%
  mutate(Yield_Mg_round = round(Yield_Mg , digits = 2),
         SE_Mg = round(SE_Mg , digits = 2)) %>%
  dplyr::select(Platform , Yield_Mg , Yield_Mg_round , SE_Mg , Sufficiency_Index , Probability)
```

Results for Paper

```
benefit_data3 <- benefit_data1 %>%
  mutate(Yield_Mg = round(Yield_Mg , digits = 2),
         SE_Mg = round(SE_Mg , digits = 2))

benefit_gs_ndvi_percent <- ( ( (benefit_data3$Yield_Mg[1] - benefit_data3$Yield_Mg[4]) / benefit_data3$

print( round( benefit_gs_ndvi_percent , digits = 0 ) )
```

```
## [1] 69
```

```
benefit_uas_ndre_percent <- ( ( (benefit_data3$Yield_Mg[5] - benefit_data3$Yield_Mg[8]) / benefit_data3

print( round( benefit_uas_ndre_percent , digits = 0 ) )
```

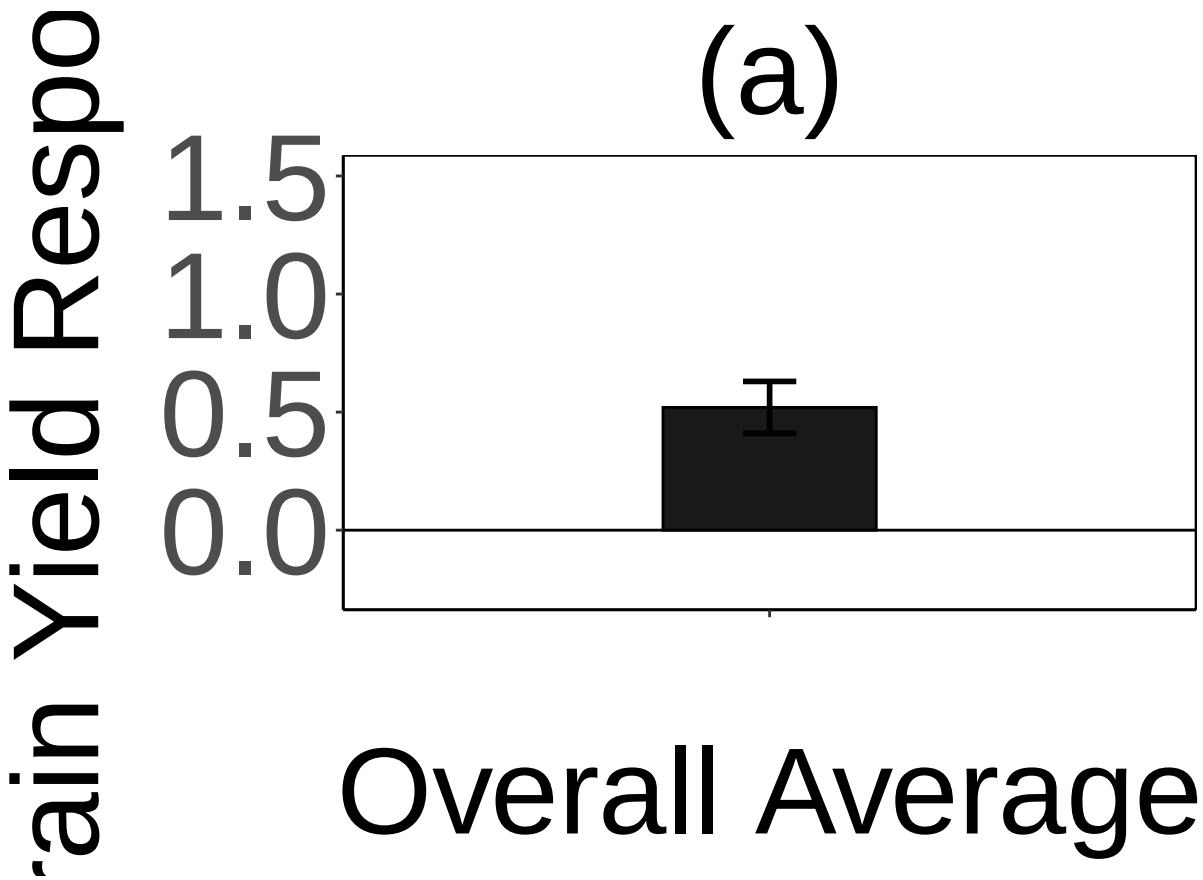
```
## [1] 113
```

Plots

overall average

```
Fig6.1 <- ggplot(data = benefit_without_si , aes ( x = Sufficiency_Index  , y = Yield_Mg)) +
  geom_bar(stat = "identity" , width = 0.30 , fill = "black" , color = "black" , alpha = 0.9) +
  geom_errorbar( data = benefit_without_si , aes( y = Yield_Mg , ymax = (Yield_Mg + SE_Mg) , ymin = (Yi
  labs(title = "(a)" , x = "Overall Average" , y = "Estimated Grain Yield Response ( Mg ha"^-1~")") +
  theme_classic() +
  theme(axis.text = element_text(size = 46),
        axis.title = element_text(size = 46),
        axis.text.x = element_text(color = "white"),
        panel.background = element_rect(fill = "white", color = "grey0"),
        plot.title = element_text(size = 46 , hjust = 0.5)) +
```

```
  coord_cartesian(ylim = c(-0.25 , 1.5 )) +
  scale_y_continuous(breaks = seq(-0.5 , 1.5 , by = .5 )) +
  geom_hline(yintercept = 0)

Fig6.1
```

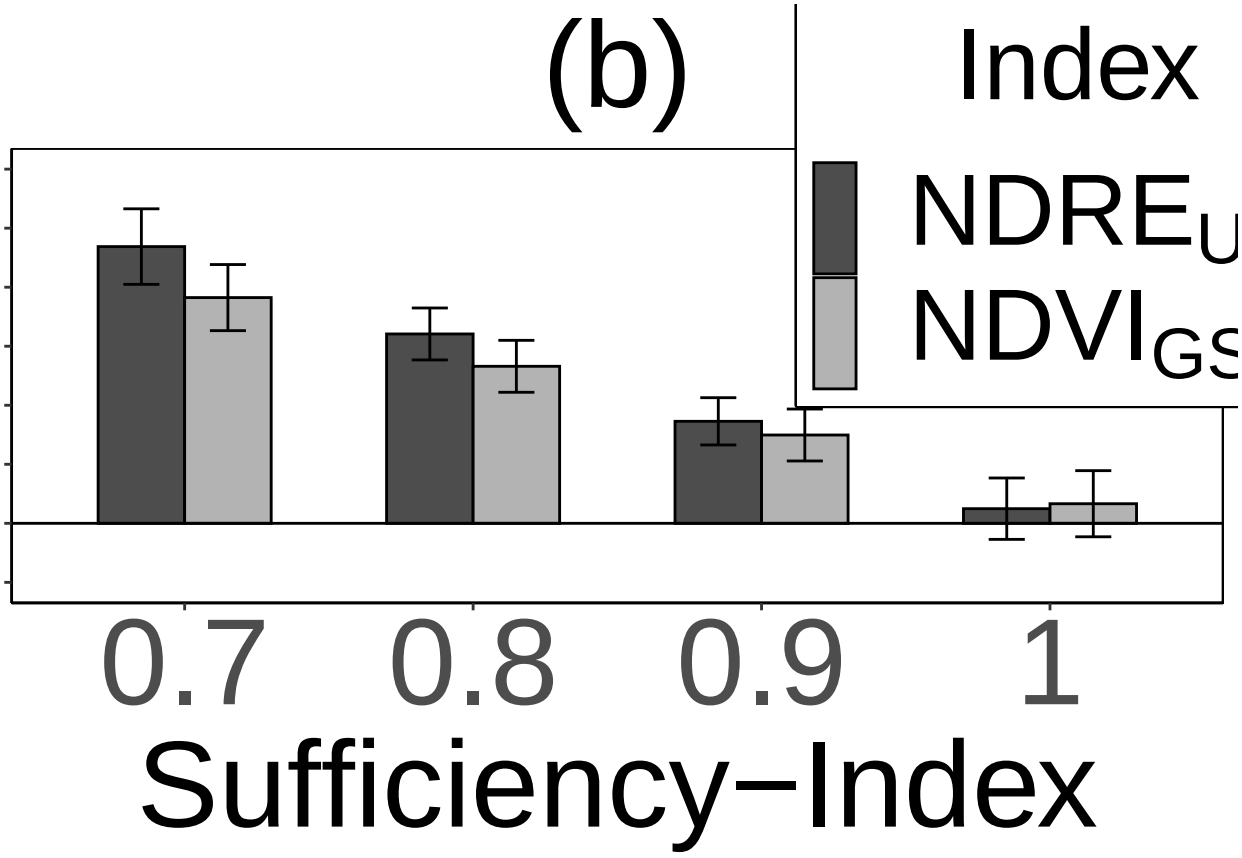



```
ggsave("FIGURES/NOT_IN_PUBLICATION/Fig6.1.pdf" , Fig6.1 , width = 14 , height = 13, dpi = 10000)
```

with SI

```
Fig6.2 <- ggplot(data = benefit_data2 , aes ( x = Sufficiency_Index  , y = Yield_Mg , fill = Platform ,
  geom_bar(position = "dodge" , stat = "identity" , width = 0.60) +
  geom_errorbar( data = benefit_data2 , aes( y = Yield_Mg , ymax = (Yield_Mg + SE_Mg) , ymin = (Yield_Mg
  labs(title = "(b)" ,
       x = "Sufficiency-Index" ,
       y = NULL ,
       color = "Index" ,
       fill = "Index" ,
       alpha = "Index") +
  theme_classic() +
  theme(axis.text = element_text(size = 46),
        axis.title = element_text(size = 46),
        panel.background = element_rect(fill = "white", color = "grey0"),
        plot.title = element_text(size = 46 , hjust = 0.5),
        axis.text.y = element_blank(),
```

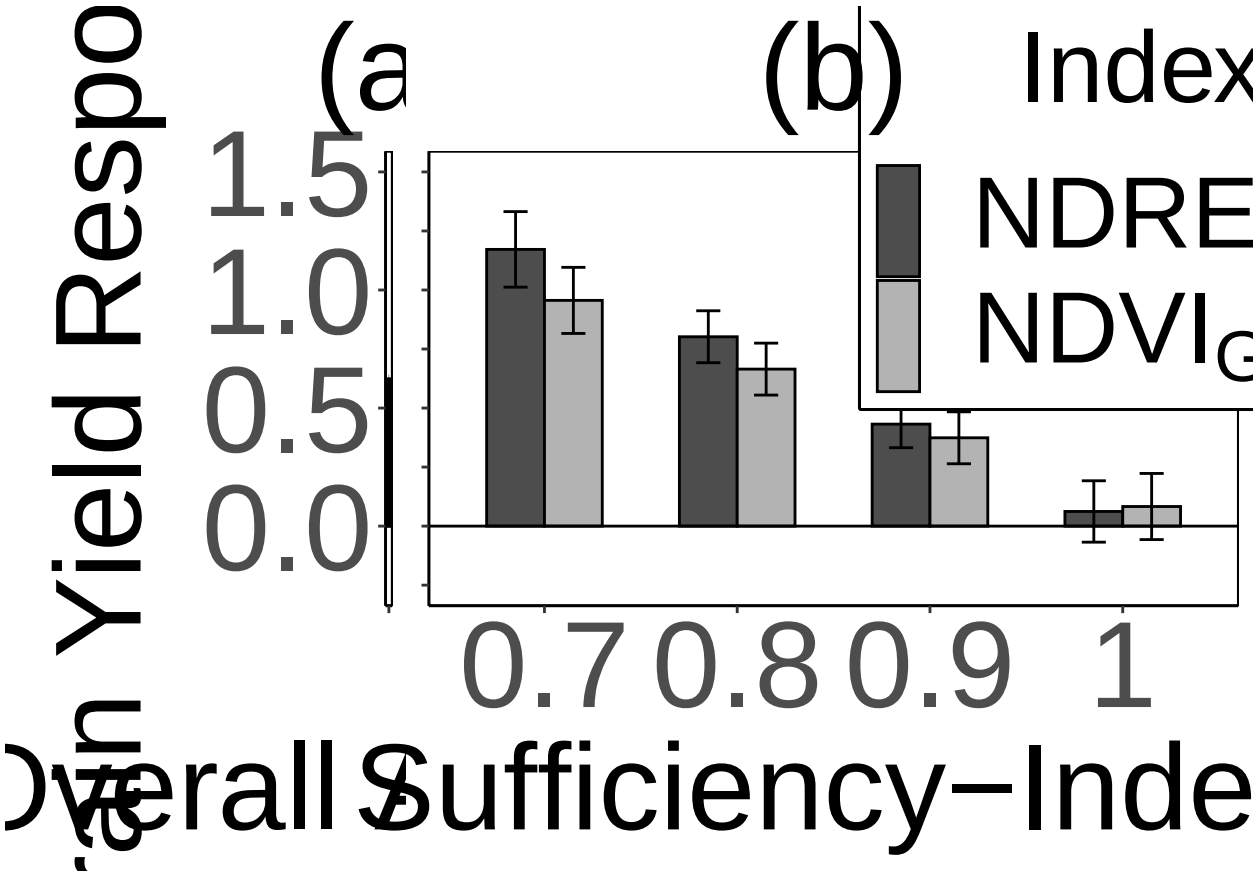
```
        legend.title = element_text(size = 38 , hjust = 0.5),
        legend.text = element_text(size = 38 , hjust = 0),
        legend.position = c(0.88 , 0.88 ),
        legend.box.background = element_rect(size = 1)) +
  coord_cartesian(ylim = c(-0.25 , 1.5)) +
  scale_y_continuous(breaks = seq(-0.5 , 1.5 , by = .25 )) +
  geom_hline(yintercept = 0) +
  scale_color_manual("Index" , breaks = c( "NDRE[UAS]" ,  "NDVI[GS]" ) , values = c( "black" ,  "black")
  scale_fill_manual("Index" , breaks = c(  "NDRE[UAS]" , "NDVI[GS]" ) , values = c( "grey30" ,  "grey70
    scale_alpha_manual("Index" , breaks = c(  "NDRE[UAS]" , "NDVI[GS]" ) , values = c( 1 , 1) , labels =

Fig6.2
```




```
ggsave("FIGURES/NOT_IN_PUBLICATION/Fig6.2.pdf" , Fig6.2 , width = 14 , height = 13, dpi = 10000)
```

FIG 6

```
Fig6 <- grid.arrange(arrangeGrob(Fig6.1 , Fig6.2 , nrow = 1 , ncol = 2 , widths = c(1.9 , 4)))
```




```
Fig6
```

```
## TableGrob (1 x 1) "arrange": 1 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[arrange]
```

```
ggsave("FIGURES/Fig6.pdf" , Fig6 , width = 25 , height = 13, dpi = 1000)
```

Results for Paper

```
print(round(min(paper2_data_OR$gs_NDVI_Sufficiency_Index) , digits = 2)) #min and max ndvi gs SI for ta
```

```
## [1] 0.25
```

```
print(round(max(paper2_data_OR$gs_NDVI_Sufficiency_Index) , digits = 2))
```

```
## [1] 1.05
```

```
print(round(min(paper2_data_OR$uas_NDRE_Sufficiency_Index) , digits = 2)) #min and max ndre uas SI for
```

```
## [1] 0.52
```

```
print(round(max(paper2_data_OR$uas_NDRE_Sufficiency_Index) , digits = 2))
```

```
## [1] 1.03
```

```
nrow(paper2_data_OR) #number of observations for table 3
```

```
## [1] 336
```

```
unique(paper2_data_OR$site_year)
```

```
## [1] Nicolaus-17   Williams-17   Arbuckle-19   Davis-19      Marysville-19
## [6] RES-19
## 6 Levels: Nicolaus-17 Williams-17 Arbuckle-19 Davis-19 ... RES-19
```