

N Rate Trial (no topdress)

Zhang Zhenglin

Contents

Necessary libraries	2
Data Organisation	2
Read from excel	2
Initial visualisation	2
Check for most important variables	3
Subset data from the 3 years	4
Treatment and year effect	5
Quadratic models	7
Automating the process with functions	7
Quadratic regressions and optimal N rates	8
Data visualisation	9
Graphing dataframes	9
2021	10
2022	15
2023	20
Average	25
Combine all plots	32
Combine 0N yields into 1 excel file	34

Necessary libraries

Data Organisation

Read from excel

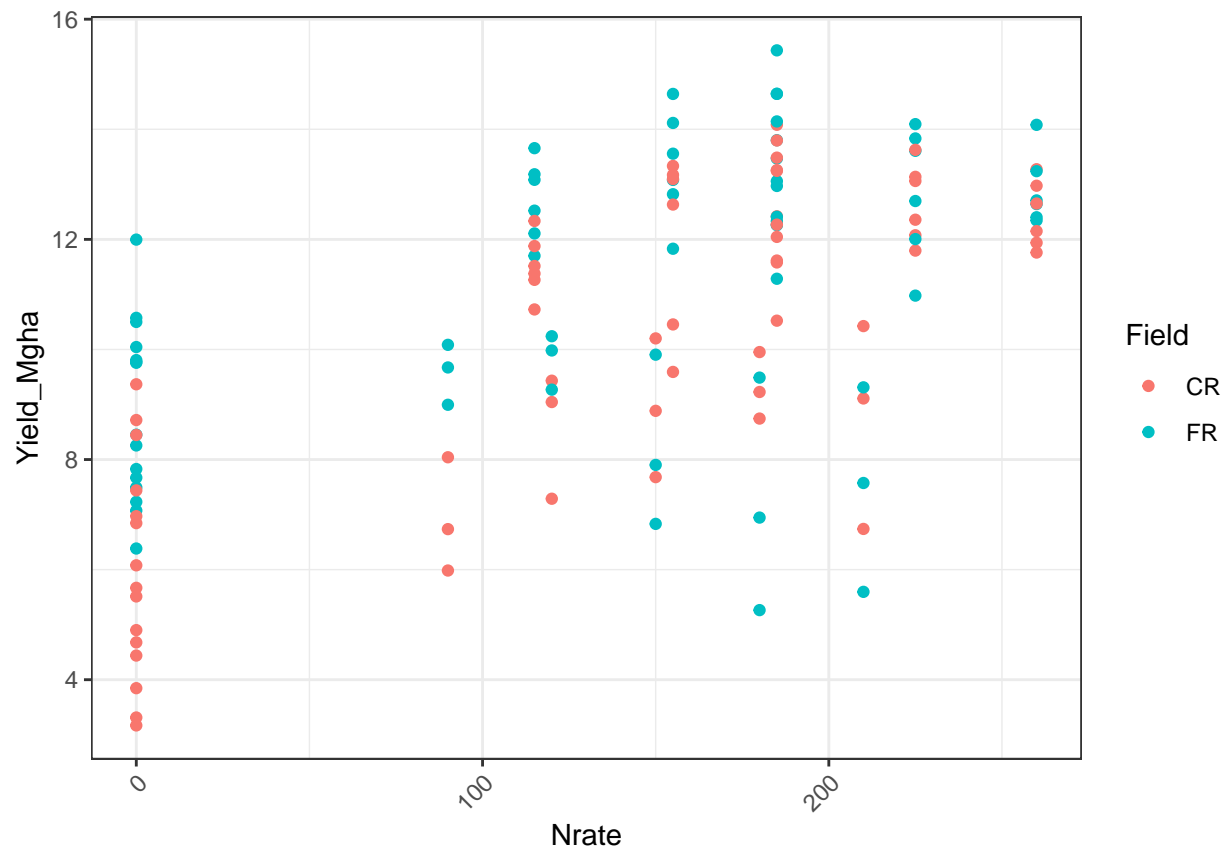
```
master <- read_excel("master_yields.xlsx", sheet = 1)
master$Treatment <- master$Field
master <- mutate_if(master, is.character, as.factor)
master$NrateF <- as.factor(master$Nrate)
master$Year <- as.factor(master$Year)
master$Yield_Mgha <- master$Yield_kgha/1000

str(master)
```

```
## tibble [132 x 11] (S3: tbl_df/tbl/data.frame)
##  $ Year          : Factor w/ 3 levels "2021","2022",...: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Plot          : Factor w/ 132 levels "101","102","103",...: 97 98 99 100 101 102 103 104 105 106 .
##  $ Blk           : num [1:132] 7 7 7 7 7 7 7 7 7 7 ...
##  $ Nrate         : num [1:132] 180 120 150 0 210 90 150 90 120 0 ...
##  $ Field         : Factor w/ 2 levels "CR","FR": 2 2 2 2 2 2 1 1 1 1 ...
##  $ Yield_kgha    : num [1:132] 5264 9982 7902 8451 5596 ...
##  $ Topdress_study: num [1:132] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Topdressed    : Factor w/ 3 levels "0","1","NIL": 3 3 3 3 3 3 3 3 3 3 ...
##  $ Treatment     : Factor w/ 2 levels "CR","FR": 2 2 2 2 2 2 1 1 1 1 ...
##  $ NrateF        : Factor w/ 11 levels "0","90","115",...: 7 4 5 1 9 2 5 2 4 1 ...
##  $ Yield_Mgha    : num [1:132] 5.26 9.98 7.9 8.45 5.6 ...
```

Initial visualisation

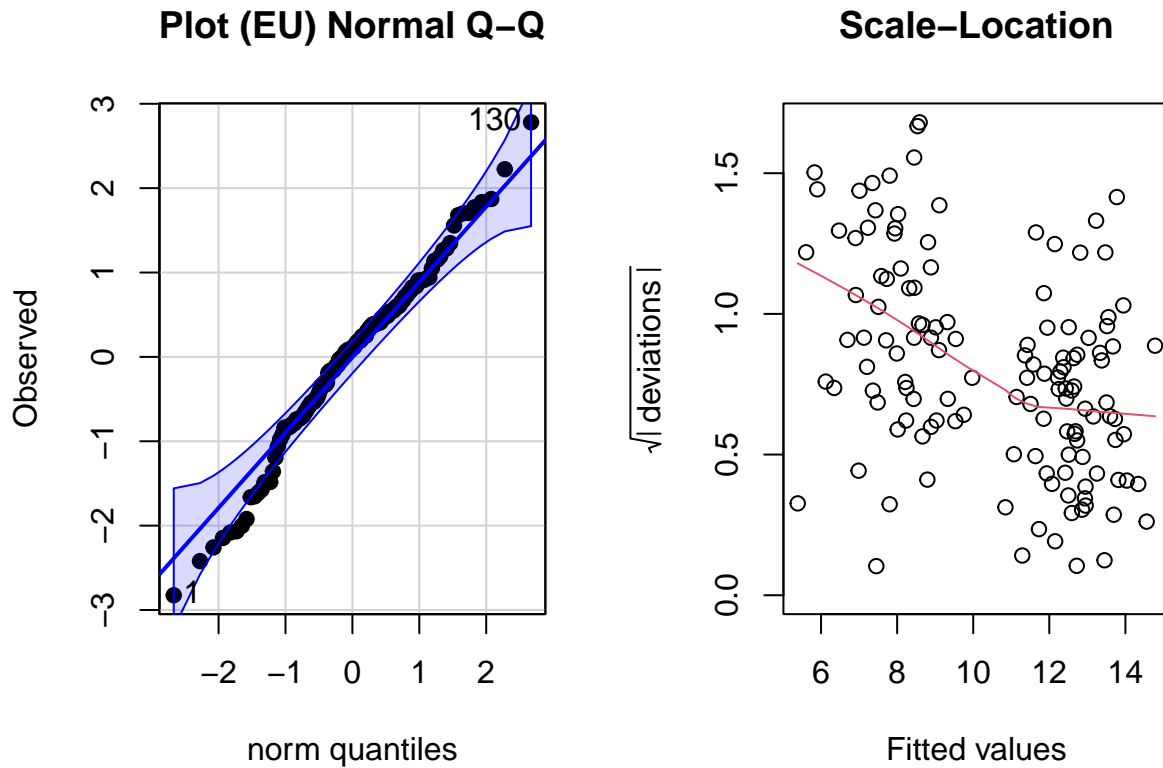
```
ggplot(master, aes(y=Yield_Mgha, x=Nrate, color=Field))+
  geom_point()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size= 9))
```



Check for most important variables

Year effect very significant. So analyse data by 3 different years.

```
all_factors_model <- lm(Yield_Mgha~Year+Blk+NrateF+Field+Topdressed, data = master)
pls205_diagnostics(all_factors_model)
```



```
anova(all_factors_model)
```

```
## Analysis of Variance Table
##
## Response: Yield_Mgha
##          Df Sum Sq Mean Sq F value    Pr(>F)
## Year       2 319.22  159.610  93.6799 < 2.2e-16 ***
## Blk        1   4.30    4.302   2.5248  0.114791
## NrateF     10 483.11   48.311  28.3552 < 2.2e-16 ***
## Field      1  38.87   38.870  22.8141 5.263e-06 ***
## Topdressed 1  14.73   14.734   8.6481 0.003953 **
## Residuals 116 197.64    1.704
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Subset data from the 3 years

```
master$main_plot <- paste(master$Blk, master$Treatment, sep="_")
master$modular <- paste(as.numeric(master$Blk)%3, master$Treatment, sep = "_")

notopdress <- master %>% filter (Topdressed != 1)
```

```

yield_average <- notopdress
yield_2021 <- notopdress %>% filter (Year == "2021")
yield_2022 <- notopdress %>% filter (Year == "2022")
yield_2023 <- notopdress %>% filter (Year == "2023")
#checked datasets, all looks good.

```

Treatment and year effect

```

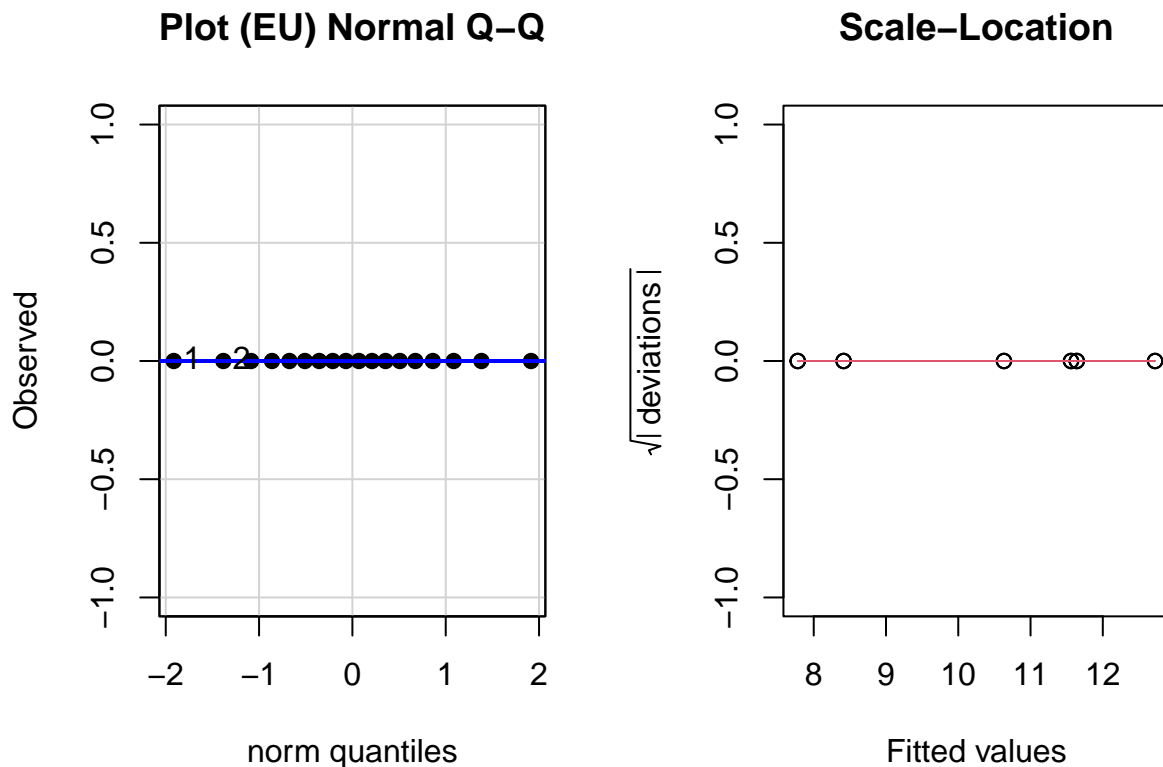
model_all_years <- lmer(Yield_Mgha ~ Treatment*Year+(1|Blk:Treatment), data = yield_average)

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

#lm(Yield_Mgha ~ Treatment*Year, data = yield_average)

pls205_diagnostics(model_all_years, EU = "Blk:Treatment")

```



```
anova(model_all_years)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)

```

```
## Treatment      23.61 23.605      1 102 4.2530 0.04172 *
## Year           319.43 159.717      2 102 28.7766 1.231e-10 ***
## Treatment:Year   1.33 0.664      2 102 0.1197 0.88732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
field_means_all_years = emmeans(model_all_years, spec = 'Treatment', by = 'Year')
field_effects_all_years = contrast(field_means_all_years, method = 'pairwise', adjust = "Tukey")
summary(field_effects_all_years)
```

```
## Year = 2021:
## contrast estimate SE df t.ratio p.value
## CR - FR      -1.163 0.785 12 -1.481 0.1643
##
## Year = 2022:
## contrast estimate SE df t.ratio p.value
## CR - FR      -1.007 0.785 12 -1.283 0.2238
##
## Year = 2023:
## contrast estimate SE df t.ratio p.value
## CR - FR      -0.634 0.785 12 -0.808 0.4349
##
## Degrees-of-freedom method: kenward-roger
```

```
cld(field_means_all_years)
```

```
## Year = 2021:
## Treatment emmean SE df lower.CL upper.CL .group
## CR        11.56 0.555 12 10.35 12.77 1
## FR        12.72 0.555 12 11.51 13.93 1
##
## Year = 2022:
## Treatment emmean SE df lower.CL upper.CL .group
## CR        10.63 0.555 12 9.42 11.84 1
## FR        11.64 0.555 12 10.43 12.85 1
##
## Year = 2023:
## Treatment emmean SE df lower.CL upper.CL .group
## CR         7.78 0.555 12 6.57 8.99 1
## FR         8.41 0.555 12 7.20 9.62 1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
##       then we cannot show them to be different.
##       But we also did not show them to be the same.
```

```
anova_results <- as.data.frame(anova(model_all_years))
```

Quadratic models

Automating the process with functions

We will be assuming that the yield-nrate curve will follow a quadratic curve. It will be messy to extract all the coefficients and then compute local maxima of each treatment. So here is a function to do that.

```
#this function automates the extraction of coefficients from quadratic models to calculate optimal N rate

optimal_N <- function(model) {
  # Extract the coefficients
  coefficients <- coef(model)

  # Get the names of the terms to make sure we handle your model's naming conventions
  term_names <- names(coefficients)

  # Identify the names for the intercept, linear, and quadratic terms
  intercept_term <- "(Intercept)"
  linear_term <- term_names[grepl("^Nrate", term_names)]
  quadratic_term <- term_names[grepl("I\\(Nrate\\^2\\)", term_names)]

  # Ensure both linear and quadratic terms are found
  if (length(linear_term) == 0 | length(quadratic_term) == 0) {
    stop("The model must include both Nrate and I(Nrate^2) terms.")
  }

  # Extract individual coefficients
  beta_0 <- coefficients[intercept_term]
  beta_1 <- coefficients[linear_term]
  beta_2 <- coefficients[quadratic_term]

  # Calculate the optimal N rate (x value) where the slope is zero
  x_optimal <- -beta_1 / (2 * beta_2)

  # Calculate the corresponding y value at the optimal N rate
  y_optimal <- beta_0 + beta_1 * x_optimal + beta_2 * x_optimal^2

  # Return both x and y values as a list
  return(list(x_optimal = x_optimal, y_optimal = y_optimal))
}

#this function automates the construction of quadratic models

run_all_models <- function() {
  # Create an empty data frame to store the results
  results <- data.frame(
    Year = character(),
    Treatment = character(),
    Nrate_optimal = numeric(),
    Yield_max = numeric(),
    stringsAsFactors = FALSE
  )

  # Define the years and treatments
```

```

years <- c(2021, 2022, 2023, "average")
treatments <- c("CR", "FR")

# Loop through each year and treatment
for (year in years) {
  for (treatment in treatments) {
    # Construct the model name dynamically
    model_formula <- as.formula(paste0("Yield_Mgha ~ Nrate + I(Nrate^2)"))
    data_name <- paste0("yield_", year)
    model_data <- get(data_name) %>% filter(Treatment == treatment)

    # Fit the model for the current year and treatment
    model <- lm(model_formula, data = model_data)

    # Calculate optimal Nrate and maximum yield using the optimal_N function
    optimal_values <- optimal_N(model)

    # Add the results to the data frame
    results <- rbind(results, data.frame(
      Year = year,
      Treatment = treatment,
      Nrate_optimal = optimal_values$x_optimal,
      Yield_max = optimal_values$y_optimal
    ))
  }
}

return(results)
}

```

Quadratic regressions and optimal N rates

```

#Here are all the optimal N rates

optimal_results <- run_all_models()

optimal_results

```

```

##      Year Treatment Nrate_optimal Yield_max
## Nrate    2021      CR    193.27222  13.408929
## Nrate1    2021      FR    188.42729  14.279893
## Nrate2    2022      CR    242.11177  12.336015
## Nrate3    2022      FR    188.44741  12.748488
## Nrate4    2023      CR    200.61488   9.067150
## Nrate5    2023      FR     85.28421   9.402874
## Nrate6 average      CR    280.86405  12.321429
## Nrate7 average      FR    281.63233  12.287842

```

```

optimal_results$Year <- as.factor(optimal_results$Year)

write_xlsx(optimal_results, "D:/Academics/UC Davis/School Work/Linquist Lab/Data/R stats/Agronomic paper

```


Data visualisation

Graphing dataframes

```
# 2021 dataframe
```

```
yield_2021_dataframe <- yield_2021 %>%  
  group_by(Treatment, Nrate) %>%  
  mutate(Yield_Mgha_se = sd(Yield_Mgha)/sqrt(3)) %>%  
  summarise(Yield_Mgha = mean(Yield_Mgha),  
            Yield_Mgha_se = mean(Yield_Mgha_se)) %>%  
  mutate(Year = "2021") %>%  
  left_join(optimal_results %>% select(Year, Treatment, Nrate_optimal), by = c("Year", "Treatment")) %>%  
  mutate(Nrate_optimal = round(Nrate_optimal))
```

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

```
# 2022 dataframe
```

```
yield_2022_dataframe <- yield_2022 %>%  
  group_by(Treatment, Nrate) %>%  
  mutate(Yield_Mgha_se = sd(Yield_Mgha)/sqrt(3)) %>%  
  summarise(Yield_Mgha = mean(Yield_Mgha),  
            Yield_Mgha_se = mean(Yield_Mgha_se)) %>%  
  mutate(Year = "2022") %>%  
  left_join(optimal_results %>% select(Year, Treatment, Nrate_optimal), by = c("Year", "Treatment")) %>%  
  mutate(Nrate_optimal = round(Nrate_optimal))
```

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

```
# 2023 dataframe
```

```
yield_2023_dataframe <- yield_2023 %>%  
  group_by(Treatment, Nrate) %>%  
  mutate(Yield_Mgha_se = sd(Yield_Mgha)/sqrt(3)) %>%  
  summarise(Yield_Mgha = mean(Yield_Mgha),  
            Yield_Mgha_se = mean(Yield_Mgha_se)) %>%  
  mutate(Year = "2023") %>%  
  left_join(optimal_results %>% select(Year, Treatment, Nrate_optimal), by = c("Year", "Treatment")) %>%  
  mutate(Nrate_optimal = round(Nrate_optimal))
```

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

```
# average dataframe
```

```
yield_average_dataframe <- yield_average %>%  
  group_by(Treatment, Nrate) %>%
```

```

mutate(Yield_Mgha_se = sd(Yield_Mgha)/sqrt(3)) %>%
summarise(Yield_Mgha = mean(Yield_Mgha),
          Yield_Mgha_se = mean(Yield_Mgha_se)) %>%
mutate(Year = "average")%>%
left_join(optimal_results %>% select(Year, Treatment, Nrate_optimal), by = c("Year", "Treatment"))%>%
mutate(Nrate_optimal = round(Nrate_optimal))

```

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

2021

Statistical test, comparisons by N rate and Treatment

```

model_2021 <- lmer(Yield_Mgha ~ NrateF*Treatment+(1|Treatment:Blk), data=yield_2021)

anova(model_2021)

```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##               Sum Sq Mean Sq NumDF DenDF  F value    Pr(>F)
## NrateF          251.041   50.208     5     20 150.6228 3.674e-15 ***
## Treatment        10.554   10.554     1      4  31.6627 0.004906 **
## NrateF:Treatment    2.236    0.447     5     20   1.3414 0.287657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

field_means_2021 = emmeans(model_2021, spec = 'Treatment', by = 'NrateF')
field_effects_2021 = contrast(field_means_2021, method = 'pairwise', adjust = "Tukey")

summary(field_effects_2021)

```

```

## NrateF = 0:
## contrast estimate SE df t.ratio p.value
## CR - FR      -2.246 0.477 23.9 -4.704 0.0001
##
## NrateF = 115:
## contrast estimate SE df t.ratio p.value
## CR - FR      -1.177 0.477 23.9 -2.466 0.0212
##
## NrateF = 155:
## contrast estimate SE df t.ratio p.value
## CR - FR      -0.904 0.477 23.9 -1.895 0.0703
##
## NrateF = 185:
## contrast estimate SE df t.ratio p.value
## CR - FR      -0.988 0.477 23.9 -2.069 0.0495
##
## NrateF = 225:
## contrast estimate SE df t.ratio p.value
## CR - FR      -0.806 0.477 23.9 -1.687 0.1045

```

```
##
## NrateF = 260:
## contrast estimate SE df t.ratio p.value
## CR - FR -0.858 0.477 23.9 -1.797 0.0849
##
## Degrees-of-freedom method: kenward-roger
```

```
cld(field_means_2021)
```

```
## NrateF = 0:
## Treatment emmean SE df lower.CL upper.CL .group
## CR 5.22 0.338 23.9 4.52 5.92 1
## FR 7.46 0.338 23.9 6.77 8.16 2
##
## NrateF = 115:
## Treatment emmean SE df lower.CL upper.CL .group
## CR 11.91 0.338 23.9 11.21 12.61 1
## FR 13.09 0.338 23.9 12.39 13.78 2
##
## NrateF = 155:
## Treatment emmean SE df lower.CL upper.CL .group
## CR 13.20 0.338 23.9 12.50 13.90 1
## FR 14.10 0.338 23.9 13.41 14.80 1
##
## NrateF = 185:
## Treatment emmean SE df lower.CL upper.CL .group
## CR 13.53 0.338 23.9 12.83 14.23 1
## FR 14.52 0.338 23.9 13.82 15.21 2
##
## NrateF = 225:
## Treatment emmean SE df lower.CL upper.CL .group
## CR 13.04 0.338 23.9 12.34 13.74 1
## FR 13.85 0.338 23.9 13.15 14.54 1
##
## NrateF = 260:
## Treatment emmean SE df lower.CL upper.CL .group
## CR 12.46 0.338 23.9 11.77 13.16 1
## FR 13.32 0.338 23.9 12.62 14.02 1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
## then we cannot show them to be different.
## But we also did not show them to be the same.
```

```
emmeans(model_2021, ~ Treatment + NrateF)
```

```
## Treatment NrateF emmean SE df lower.CL upper.CL
## CR 0 5.22 0.338 23.9 4.52 5.92
## FR 0 7.46 0.338 23.9 6.77 8.16
## CR 115 11.91 0.338 23.9 11.21 12.61
## FR 115 13.09 0.338 23.9 12.39 13.78
```

```
## CR      155      13.20 0.338 23.9      12.50      13.90
## FR      155      14.10 0.338 23.9      13.41      14.80
## CR      185      13.53 0.338 23.9      12.83      14.23
## FR      185      14.52 0.338 23.9      13.82      15.21
## CR      225      13.04 0.338 23.9      12.34      13.74
## FR      225      13.85 0.338 23.9      13.15      14.54
## CR      260      12.46 0.338 23.9      11.77      13.16
## FR      260      13.32 0.338 23.9      12.62      14.02
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
cld_2021 <- cld(emmeans(model_2021, ~ Treatment + NrateF), adjust = "Sidak")

top_yields_2021 <- cld_2021 %>%
  group_by(Treatment) %>%
  filter(emmean == max(emmean)) %>%
  ungroup()
```

Graph_2021

```
N_response_curve_2021 <-
ggplot(yield_2021_dataframe, aes(x=Nrate, y=Yield_Mgha, color=Treatment))+
  geom_point(data=yield_2021_dataframe, size=2.5)+ #this is the mean values
  geom_point(data=yield_2021, size=1, alpha=0.4)+ #this are the raw values
  scale_color_manual(values=c("#0072B2", "#FFCC66"), name = "Treatment", labels = c('Continuous Rice (CR)', 'Fertilizer Rice (FR)'))
  scale_x_continuous(name=expression("N rate (kg N ha"^{-1}*")"), limits = c(-20, 280), breaks = c(0, 115, 155, 185, 225, 260))
  scale_y_continuous(name= expression("Yield (Mg ha"^{-1}*")"), limits = c(0, 22), breaks = seq(0, 22, 1))
  #geom_errorbar(data=yield_2021_dataframe, aes(ymin=Yield_Mgha-Yield_Mgha_se, ymax=Yield_Mgha+Yield_Mgha_se))
  theme_classic()+
  geom_smooth(data = yield_2021, method = "lm", formula = y ~ poly(x, 2), se = FALSE)+
  #geom_vline(xintercept = c(0, 115, 155, 185, 225, 260), linetype = "twodash", color = "black") +
  stat_regline_equation(data=yield_2021,
    aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
      label = paste(..rr.label..)),
    formula = y ~ poly(x, 2, raw = TRUE),
    show.legend = FALSE, label.x = 0, label.y = c(16.5, 19))+
  stat_regline_equation(data=yield_2021,
    aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
      label = paste(..eq.label..)),
    formula = y ~ poly(x, 2, raw = TRUE),
    show.legend = FALSE, label.x = 0, label.y = c(17.5, 20))+
  theme(axis.text = element_text(size = 14), axis.title = element_text(size=16))+
  theme(legend.text = element_text(size = 12), legend.title = element_text(size = 14))+
  theme(plot.title = element_text(hjust = 0.5, size = 20, face = "bold"))+
  ggtitle(expression("Yield: 2021"))+
  #annotate(
  #  "text",
  #  x = c(0),
  #  y = yield_2021_dataframe %>%
  #    filter(Treatment == "FR", Nrate == 0) %>%
  #    mutate(Yield_Mgha_plus_higher = Yield_Mgha + 1) %>%
```

```

    # pull(Yield_Mgha_plus_higher),
    #label = "*",
    #size = 7,
    #vjust = 0
    #)+
# Vertical line for CR
geom_segment(
  aes(x = optimal_results %>% filter(Treatment == "CR", Year == "2021") %>% pull(Nrate_optimal),
      xend = optimal_results %>% filter(Treatment == "CR", Year == "2021") %>% pull(Nrate_optimal),
      y = 0, # Start exactly from 0 on y-axis
      yend = optimal_results %>% filter(Treatment == "CR", Year == "2021") %>% pull(Yield_max)),
  linetype = "twodash", color = "#0072B2", size = 0.6
) +
# Vertical line for FR
geom_segment(
  aes(x = optimal_results %>% filter(Treatment == "FR", Year == "2021") %>% pull(Nrate_optimal),
      xend = optimal_results %>% filter(Treatment == "FR", Year == "2021") %>% pull(Nrate_optimal),
      y = 0, # Start exactly from 0 on y-axis
      yend = optimal_results %>% filter(Treatment == "FR", Year == "2021") %>% pull(Yield_max)),
  linetype = "twodash", color = "#FFCC66", size = 0.6
) +
# Horizontal line for CR
geom_segment(
  aes(y = optimal_results %>% filter(Treatment == "CR", Year == "2021") %>% pull(Yield_max),
      yend = optimal_results %>% filter(Treatment == "CR", Year == "2021") %>% pull(Yield_max),
      x = -20, # Start exactly from 0 on x-axis
      xend = optimal_results %>% filter(Treatment == "CR", Year == "2021") %>% pull(Nrate_optimal)),
  linetype = "twodash", color = "#0072B2", size = 0.6
) +
# Horizontal line for FR
geom_segment(
  aes(y = optimal_results %>% filter(Treatment == "FR", Year == "2021") %>% pull(Yield_max),
      yend = optimal_results %>% filter(Treatment == "FR", Year == "2021") %>% pull(Yield_max),
      x = -20, # Start exactly from 0 on x-axis
      xend = optimal_results %>% filter(Treatment == "FR", Year == "2021") %>% pull(Nrate_optimal)),
  linetype = "twodash", color = "#FFCC66", size = 0.6
)+
  geom_curve(x = top_yields_2021 %>%
    filter(Treatment == "FR") %>%
    mutate(NrateF = as.numeric(paste(NrateF))) %>%
    pull(NrateF)-2,
    y = top_yields_2021 %>%
    filter(Treatment == "FR") %>%
    pull(emmean),
    xend = top_yields_2021 %>%
    filter(Treatment == "CR") %>%
    mutate(NrateF = as.numeric(paste(NrateF))) %>%
    pull(NrateF)-2,
    yend = top_yields_2021 %>%
    filter(Treatment == "CR") %>%
    pull(emmean),
    color = 1, size=1,

```

```

        curvature = 1.2,)+
  annotate(
    "text",
    x = (top_yields_2021 %>%
          filter(Treatment == "FR") %>%
          mutate(NrateF = as.numeric(paste(NrateF))) %>%
          pull(NrateF)+
          top_yields_2021 %>%
          filter(Treatment == "CR") %>%
          mutate(NrateF = as.numeric(paste(NrateF))) %>%
          pull(NrateF))/2,
    y = (top_yields_2021 %>%
          filter(Treatment == "FR") %>%
          pull(emmean)+
          top_yields_2021 %>%
          filter(Treatment == "CR") %>%
          pull(emmean))/2,
    label = "ns",
    size = 5,
    vjust = 0.3,
    hjust = +2.2 )

```

```

## 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.

```

```
N_response_curve_2021
```

```

## Warning: The dot-dot notation ('..rr.label..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(rr.label)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

```

```

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

```

```

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

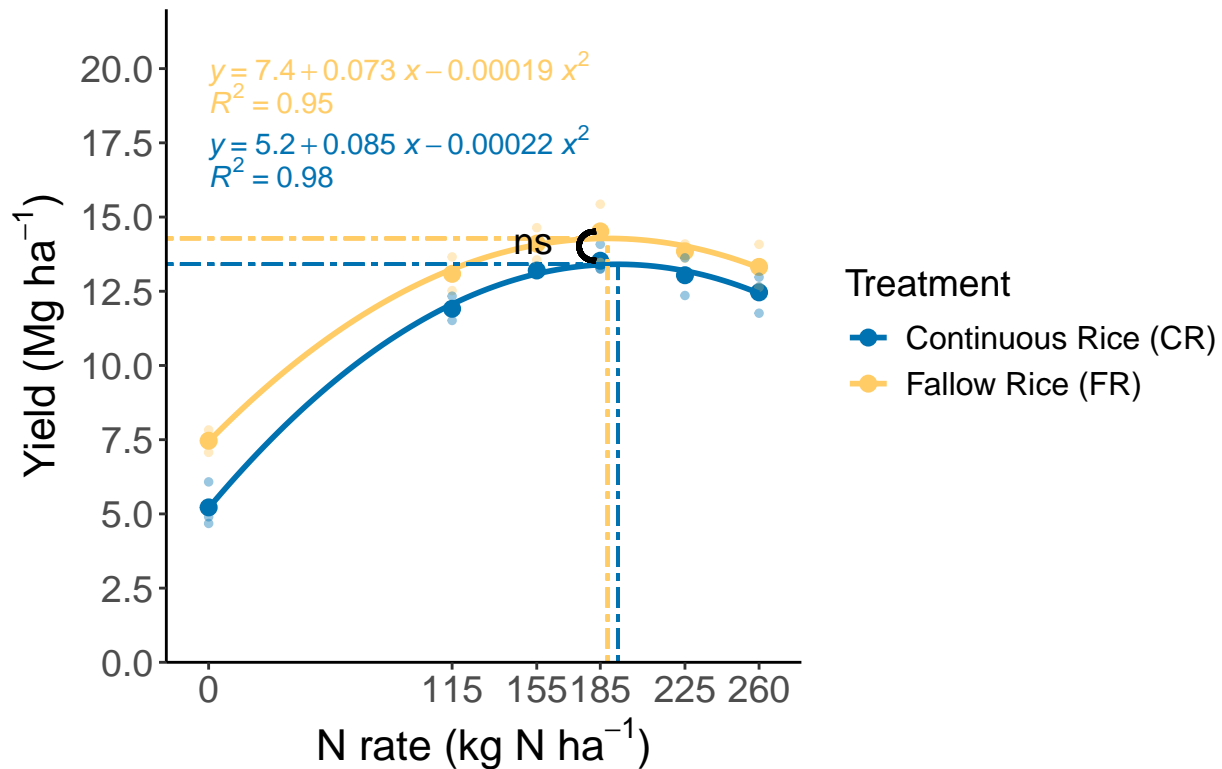
```

```

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

```

Yield: 2021



2022

Statistical test, comparisons by N rate and Treatment

```
model_2022 <- lmer(Yield_Mgha ~ NrateF*Treatment+(1|Treatment:Blk), data=yield_2022)
```

```
anova(model_2022)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##               Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## NrateF         155.079  31.0159     5     20 37.7791 1.601e-09 ***
## Treatment         3.156   3.1563     1      4  3.8445  0.1215
## NrateF:Treatment   8.633   1.7265     5     20  2.1030  0.1073
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
field_means_2022 = emmeans(model_2022, spec = 'Treatment', by = 'NrateF')
field_effects_2022 = contrast(field_means_2022, method = 'pairwise', adjust = "Tukey")
summary(field_effects_2022)
```

```
## NrateF = 0:
## contrast estimate    SE    df t.ratio p.value
```

```
## CR - FR    -2.5151 0.849 18.6  -2.964  0.0081
##
## NrateF = 115:
## contrast estimate    SE    df t.ratio p.value
## CR - FR    -1.2073 0.849 18.6  -1.423  0.1714
##
## NrateF = 155:
## contrast estimate    SE    df t.ratio p.value
## CR - FR    -1.6841 0.849 18.6  -1.985  0.0621
##
## NrateF = 185:
## contrast estimate    SE    df t.ratio p.value
## CR - FR    -1.0247 0.849 18.6  -1.208  0.2423
##
## NrateF = 225:
## contrast estimate    SE    df t.ratio p.value
## CR - FR     0.4174 0.849 18.6   0.492  0.6286
##
## NrateF = 260:
## contrast estimate    SE    df t.ratio p.value
## CR - FR    -0.0308 0.849 18.6  -0.036  0.9714
##
## Degrees-of-freedom method: kenward-roger
```

```
cld(field_means_2022)
```

```
## NrateF = 0:
## Treatment emmean SE    df lower.CL upper.CL .group
## CR          5.28 0.6 18.6     4.02     6.53  1
## FR          7.79 0.6 18.6     6.53     9.05  2
##
## NrateF = 115:
## Treatment emmean SE    df lower.CL upper.CL .group
## CR          11.12 0.6 18.6     9.87    12.38  1
## FR          12.33 0.6 18.6    11.07    13.59  1
##
## NrateF = 155:
## Treatment emmean SE    df lower.CL upper.CL .group
## CR          10.89 0.6 18.6     9.64    12.15  1
## FR          12.58 0.6 18.6    11.32    13.83  1
##
## NrateF = 185:
## Treatment emmean SE    df lower.CL upper.CL .group
## CR          11.74 0.6 18.6    10.48    13.00  1
## FR          12.76 0.6 18.6    11.50    14.02  1
##
## NrateF = 225:
## Treatment emmean SE    df lower.CL upper.CL .group
## FR          11.89 0.6 18.6    10.64    13.15  1
## CR          12.31 0.6 18.6    11.05    13.57  1
##
## NrateF = 260:
## Treatment emmean SE    df lower.CL upper.CL .group
## CR          12.45 0.6 18.6    11.20    13.71  1
```



```
## FR      12.48 0.6 18.6    11.23    13.74  1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
##       then we cannot show them to be different.
##       But we also did not show them to be the same.
```

```
emmeans(model_2022, ~ Treatment + NrateF)
```

```
## Treatment NrateF emmean SE df lower.CL upper.CL
## CR      0      5.28 0.6 18.6    4.02    6.53
## FR      0      7.79 0.6 18.6    6.53    9.05
## CR     115     11.12 0.6 18.6    9.87   12.38
## FR     115     12.33 0.6 18.6   11.07   13.59
## CR     155     10.89 0.6 18.6    9.64   12.15
## FR     155     12.58 0.6 18.6   11.32   13.83
## CR     185     11.74 0.6 18.6   10.48   13.00
## FR     185     12.76 0.6 18.6   11.50   14.02
## CR     225     12.31 0.6 18.6   11.05   13.57
## FR     225     11.89 0.6 18.6   10.64   13.15
## CR     260     12.45 0.6 18.6   11.20   13.71
## FR     260     12.48 0.6 18.6   11.23   13.74
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
cld_2022 <- cld(emmeans(model_2022, ~ Treatment + NrateF), adjust = "Sidak")
```

```
top_yields_2022 <- cld_2022 %>%
  group_by(Treatment) %>%
  filter(emmean == max(emmean)) %>%
  ungroup()
```

Graph_2022

```
N_response_curve_2022 <-
ggplot(yield_2022_dataframe, aes(x=Nrate, y=Yield_Mgha, color=Treatment))+
  geom_point(data=yield_2022_dataframe %>% filter(Treatment == "FR"),
    aes(x=Nrate, y=Yield_Mgha, color="FR"), size=2.5, shape="triangle") +
  geom_point(data=yield_2022_dataframe %>% filter(Treatment == "CR"),
    aes(x=Nrate, y=Yield_Mgha, color="CR"), size=2.5, shape="triangle") +
  #geom_point(data=yield_2022_dataframe, size=2.5, shape="triangle")+ #this is the mean values
  geom_point(data=yield_2022, size=1, alpha=0.4, shape="triangle")+ #this are the raw values
  scale_color_manual(values=c("#0072B2", "#FFCC66"), name = "Treatment", labels = c('Continuous Rice (CR)', 'Fertilizer (FR)'))
  scale_x_continuous(name=expression("N rate (kg N ha"^{-1}*")"), limits = c(-20, 280), breaks = c(0, 100, 200, 280))
  scale_y_continuous(name= expression("Yield (Mg ha"^{-1}*")"), limits = c(0, 22), breaks = seq(0, 22, 2))
  #geom_errorbar(data=yield_2022_dataframe, aes(ymin=Yield_Mgha-Yield_Mgha_se, ymax=Yield_Mgha+Yield_Mgha_se))
  theme_classic()+
  geom_smooth(data = yield_2022, method = "lm", formula = y ~ poly(x, 2), se = FALSE)+
```

```

#geom_vline(xintercept = c(0, 115, 155, 185, 225, 260), linetype = "twodash", color = "black") +
theme(axis.text = element_text(size = 14), axis.title = element_text(size=16))+
theme(legend.text = element_text(size = 12),legend.title = element_text(size = 14))+
theme(plot.title = element_text(hjust = 0.5, size = 20, face = "bold"))+
ggtitle(expression("Yield: 2022"))+

# Vertical line for CR
geom_segment(
  aes(x = optimal_results %>% filter(Treatment == "CR", Year == "2022") %>% pull(Nrate_optimal),
      xend = optimal_results %>% filter(Treatment == "CR", Year == "2022") %>% pull(Nrate_optimal),
      y = 0, # Start exactly from 0 on y-axis
      yend = optimal_results %>% filter(Treatment == "CR", Year == "2022") %>% pull(Yield_max)),
  linetype = "twodash", color = "#0072B2", size = 0.6
) +

# Vertical line for FR
geom_segment(
  aes(x = optimal_results %>% filter(Treatment == "FR", Year == "2022") %>% pull(Nrate_optimal),
      xend = optimal_results %>% filter(Treatment == "FR", Year == "2022") %>% pull(Nrate_optimal),
      y = 0, # Start exactly from 0 on y-axis
      yend = optimal_results %>% filter(Treatment == "FR", Year == "2022") %>% pull(Yield_max)),
  linetype = "twodash", color = "#FFCC66", size = 0.6
) +

# Horizontal line for CR
geom_segment(
  aes(y = optimal_results %>% filter(Treatment == "CR", Year == "2022") %>% pull(Yield_max),
      yend = optimal_results %>% filter(Treatment == "CR", Year == "2022") %>% pull(Yield_max),
      x = -20, # Start exactly from 0 on x-axis
      xend = optimal_results %>% filter(Treatment == "CR", Year == "2022") %>% pull(Nrate_optimal)),
  linetype = "twodash", color = "#0072B2", size = 0.6
) +

# Horizontal line for FR
geom_segment(
  aes(y = optimal_results %>% filter(Treatment == "FR", Year == "2022") %>% pull(Yield_max),
      yend = optimal_results %>% filter(Treatment == "FR", Year == "2022") %>% pull(Yield_max),
      x = -20, # Start exactly from 0 on x-axis
      xend = optimal_results %>% filter(Treatment == "FR", Year == "2022") %>% pull(Nrate_optimal)),
  linetype = "twodash", color = "#FFCC66", size = 0.6
)+

stat_regline_equation(data=yield_2022,
  aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
      label = paste(..rr.label..)),
  formula = y ~ poly(x, 2, raw = TRUE),
  show.legend = FALSE, label.x = 0, label.y = c(16.5, 19))+
stat_regline_equation(data=yield_2022,
  aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
      label = paste(..eq.label..)),
  formula = y ~ poly(x, 2, raw = TRUE),
  show.legend = FALSE, label.x = 0, label.y = c(17.5, 20))+
geom_curve(x = top_yields_2022 %>%
  filter(Treatment == "FR") %>%
  mutate(NrateF = as.numeric(paste(NrateF))) %>%
  pull(NrateF),

```

```

    y = top_yields_2022 %>%
    filter(Treatment == "FR") %>%
    pull(emmean),
xend = top_yields_2022 %>%
    filter(Treatment == "CR") %>%
    mutate(NrateF = as.numeric(paste(NrateF))) %>%
    pull(NrateF),
yend = top_yields_2022 %>%
    filter(Treatment == "CR") %>%
    pull(emmean),
color = 1, size=1,
curvature = -0.2)+
annotate(
  "text",
  x = (top_yields_2022 %>%
        filter(Treatment == "FR") %>%
        mutate(NrateF = as.numeric(paste(NrateF))) %>%
        pull(NrateF)+
        top_yields_2022 %>%
        filter(Treatment == "CR") %>%
        mutate(NrateF = as.numeric(paste(NrateF))) %>%
        pull(NrateF))/2,
  y = mean(top_yields_2022 %>%
            filter(Treatment == "FR") %>%
            pull(emmean),
            top_yields_2022 %>%
            filter(Treatment == "CR") %>%
            pull(emmean)),
  label = "ns",
  size = 5,
  vjust = -2,
  hjust = 0.1 )

```

N_response_curve_2022

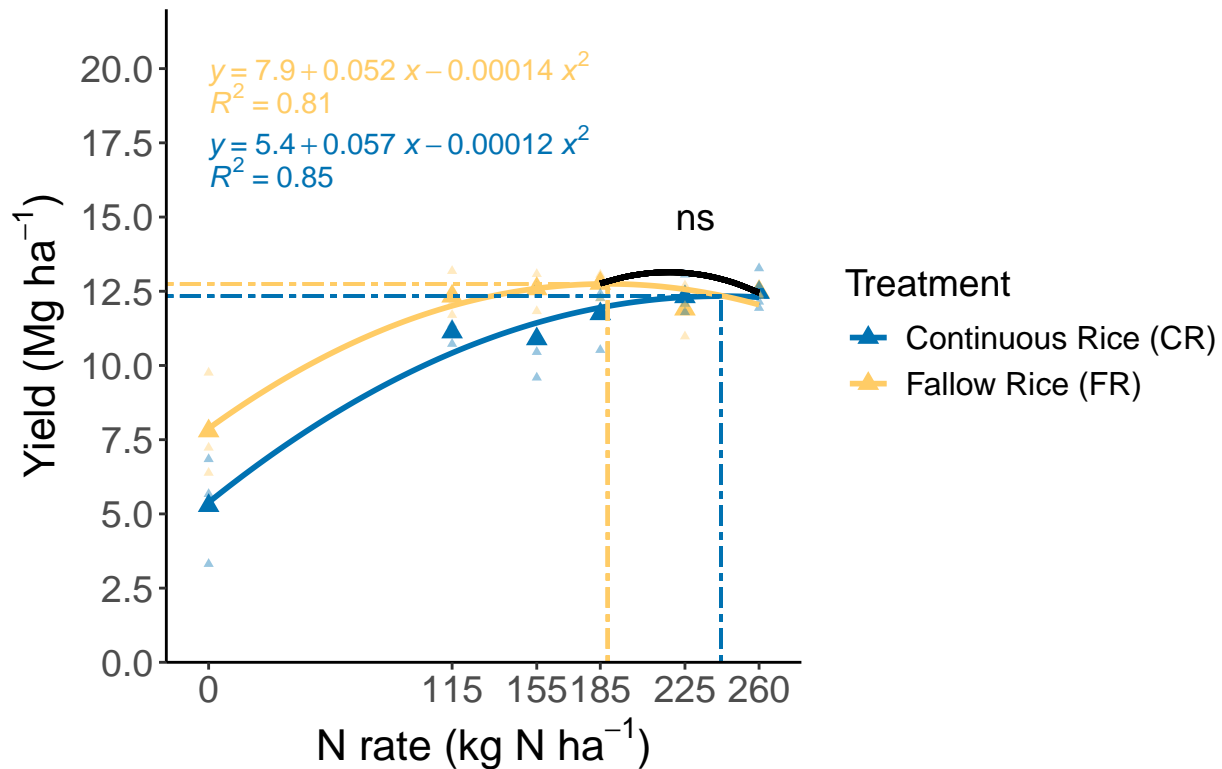
```
## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.
```

```
## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.
```

```
## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.
```

```
## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.
```

Yield: 2022



2023

Statistical test, comparisons by N rate and Treatment

```
model_2023 <- lmer(Yield_Mgha ~ NrateF*Treatment+(1|Treatment:Blk), data=yield_2023)
```

```
anova(model_2023)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##              Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## NrateF        31.805   6.3610     5    20  4.0251 0.01088 *
## Treatment       2.477   2.4772     1     4  1.5675 0.27878
## NrateF:Treatment 42.351   8.4702     5    20  5.3597 0.00276 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
field_means_2023 = emmeans(model_2023, spec = 'Treatment', by = 'NrateF')
field_effects_2023 = contrast(field_means_2023, method = 'pairwise', adjust = "Tukey")
```

```
summary(field_effects_2023)
```

```
## NrateF = 0:
## contrast estimate SE df t.ratio p.value
```

```
## CR - FR      -3.950 1.07 23.4  -3.708  0.0011
##
## NrateF = 90:
## contrast estimate SE df t.ratio p.value
## CR - FR      -2.664 1.07 23.4  -2.501  0.0198
##
## NrateF = 120:
## contrast estimate SE df t.ratio p.value
## CR - FR      -1.242 1.07 23.4  -1.166  0.2554
##
## NrateF = 150:
## contrast estimate SE df t.ratio p.value
## CR - FR       0.709 1.07 23.4   0.666  0.5120
##
## NrateF = 180:
## contrast estimate SE df t.ratio p.value
## CR - FR       2.076 1.07 23.4   1.948  0.0635
##
## NrateF = 210:
## contrast estimate SE df t.ratio p.value
## CR - FR       1.264 1.07 23.4   1.187  0.2473
##
## Degrees-of-freedom method: kenward-roger
```

```
cld(field_means_2023)
```

```
## NrateF = 0:
## Treatment emmean SE df lower.CL upper.CL .group
## CR          4.18 0.753 23.4    2.62    5.73    1
## FR          8.13 0.753 23.4    6.57    9.68    2
##
## NrateF = 90:
## Treatment emmean SE df lower.CL upper.CL .group
## CR          6.92 0.753 23.4    5.36    8.48    1
## FR          9.58 0.753 23.4    8.03   11.14    2
##
## NrateF = 120:
## Treatment emmean SE df lower.CL upper.CL .group
## CR          8.59 0.753 23.4    7.03   10.15    1
## FR          9.83 0.753 23.4    8.27   11.39    1
##
## NrateF = 150:
## Treatment emmean SE df lower.CL upper.CL .group
## FR          8.21 0.753 23.4    6.66    9.77    1
## CR          8.92 0.753 23.4    7.37   10.48    1
##
## NrateF = 180:
## Treatment emmean SE df lower.CL upper.CL .group
## FR          7.23 0.753 23.4    5.68    8.79    1
## CR          9.31 0.753 23.4    7.75   10.86    1
##
## NrateF = 210:
## Treatment emmean SE df lower.CL upper.CL .group
## FR          7.49 0.753 23.4    5.94    9.05    1
```

```
## CR      8.76 0.753 23.4      7.20      10.31  1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
##       then we cannot show them to be different.
##       But we also did not show them to be the same.
```

```
emmeans(model_2023, ~ Treatment + NrateF)
```

```
## Treatment NrateF emmean      SE    df lower.CL upper.CL
## CR        0      4.18 0.753 23.4      2.62      5.73
## FR        0      8.13 0.753 23.4      6.57      9.68
## CR       90      6.92 0.753 23.4      5.36      8.48
## FR       90      9.58 0.753 23.4      8.03     11.14
## CR      120      8.59 0.753 23.4      7.03     10.15
## FR      120      9.83 0.753 23.4      8.27     11.39
## CR      150      8.92 0.753 23.4      7.37     10.48
## FR      150      8.21 0.753 23.4      6.66      9.77
## CR      180      9.31 0.753 23.4      7.75     10.86
## FR      180      7.23 0.753 23.4      5.68      8.79
## CR      210      8.76 0.753 23.4      7.20     10.31
## FR      210      7.49 0.753 23.4      5.94      9.05
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
cld_2023 <- cld(emmeans(model_2023, ~ Treatment + NrateF), adjust = "Sidak")
```

```
top_yields_2023 <- cld_2023 %>%
  group_by(Treatment) %>%
  filter(emmean == max(emmean)) %>%
  ungroup()
```

Graph_2023

```
N_response_curve_2023 <-
ggplot(yield_2023_dataframe, aes(x=Nrate, y=Yield_Mgha, color=Treatment))+
  geom_point(data=yield_2023_dataframe, size=2.5, shape="square")+ #this is the mean values
  geom_point(data=yield_2023, size=1, alpha=0.4, shape="square")+ #this are the raw values
  scale_color_manual(values=c("#0072B2", "#FFCC66"), name = "Treatment", labels = c('Continuous Rice (CR)', 'Fertilizer Rice (FR)'))
  scale_x_continuous(name=expression("N rate (kg N ha"^{-1}*")"), limits = c(-20, 280), breaks = c(0, 90, 180, 270))
  scale_y_continuous(name= expression("Yield (Mg ha"^{-1}*")"), limits = c(0, 22), breaks = seq(0, 22, 1))
  #geom_errorbar(data=yield_2023_dataframe, aes(ymin=Yield_Mgha-Yield_Mgha_se, ymax=Yield_Mgha+Yield_Mgha_se))
  theme_classic()+
  geom_smooth(data = yield_2023, method = "lm", formula = y ~ poly(x, 2), se = FALSE)+
  #geom_vline(xintercept = c(0, 115, 155, 185, 225, 260), linetype = "twodash", color = "black") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size=16))+
  theme(legend.text = element_text(size = 12), legend.title = element_text(size = 14))+
  theme(plot.title = element_text(hjust = 0.5, size = 20, face = "bold"))+
```

```

ggtitle(expression("Yield: 2023"))+

# Vertical line for CR
geom_segment(
  aes(x = optimal_results %>% filter(Treatment == "CR", Year == "2023") %>% pull(Nrate_optimal),
      xend = optimal_results %>% filter(Treatment == "CR", Year == "2023") %>% pull(Nrate_optimal),
      y = 0, # Start exactly from 0 on y-axis
      yend = optimal_results %>% filter(Treatment == "CR", Year == "2023") %>% pull(Yield_max)),
  linetype = "twodash", color = "#0072B2", size = 0.6
) +

# Vertical line for FR
geom_segment(
  aes(x = optimal_results %>% filter(Treatment == "FR", Year == "2023") %>% pull(Nrate_optimal),
      xend = optimal_results %>% filter(Treatment == "FR", Year == "2023") %>% pull(Nrate_optimal),
      y = 0, # Start exactly from 0 on y-axis
      yend = optimal_results %>% filter(Treatment == "FR", Year == "2023") %>% pull(Yield_max)),
  linetype = "twodash", color = "#FFCC66", size = 0.6
) +

# Horizontal line for CR
geom_segment(
  aes(y = optimal_results %>% filter(Treatment == "CR", Year == "2023") %>% pull(Yield_max),
      yend = optimal_results %>% filter(Treatment == "CR", Year == "2023") %>% pull(Yield_max),
      x = -20, # Start exactly from 0 on x-axis
      xend = optimal_results %>% filter(Treatment == "CR", Year == "2023") %>% pull(Nrate_optimal)),
  linetype = "twodash", color = "#0072B2", size = 0.6
) +

# Horizontal line for FR
geom_segment(
  aes(y = optimal_results %>% filter(Treatment == "FR", Year == "2023") %>% pull(Yield_max),
      yend = optimal_results %>% filter(Treatment == "FR", Year == "2023") %>% pull(Yield_max),
      x = -20, # Start exactly from 0 on x-axis
      xend = optimal_results %>% filter(Treatment == "FR", Year == "2023") %>% pull(Nrate_optimal)),
  linetype = "twodash", color = "#FFCC66", size = 0.6
)+

geom_curve(x = top_yields_2023 %>%
  filter(Treatment == "FR") %>%
  mutate(NrateF = as.numeric(paste(NrateF))) %>%
  pull(NrateF),
  y = top_yields_2023 %>%
  filter(Treatment == "FR") %>%
  pull(emmean)+0.1,
  xend = top_yields_2023 %>%
  filter(Treatment == "CR") %>%
  mutate(NrateF = as.numeric(paste(NrateF))) %>%
  pull(NrateF),
  yend = top_yields_2023 %>%
  filter(Treatment == "CR") %>%
  pull(emmean)+0.1,
  color = 1, size=1,
  curvature = -0.2)+
stat_regline_equation(data=yield_2023,
  aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,

```

```

        label = paste(..rr.label..)),
        formula = y ~ poly(x, 2, raw = TRUE),
        show.legend = FALSE, label.x = 0, label.y = c(16.5, 19))+
  stat_regline_equation(data=yield_2023,
    aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
      label = paste(..eq.label..)),
    formula = y ~ poly(x, 2, raw = TRUE),
    show.legend = FALSE, label.x = 0, label.y = c(17.5, 20))+
  annotate(
    "text",
    x = (top_yields_2023 %>%
      filter(Treatment == "FR") %>%
      mutate(NrateF = as.numeric(paste(NrateF))) %>%
      pull(NrateF))+
      top_yields_2023 %>%
      filter(Treatment == "CR") %>%
      mutate(NrateF = as.numeric(paste(NrateF))) %>%
      pull(NrateF))/2,
    y = mean(top_yields_2023 %>%
      filter(Treatment == "FR") %>%
      pull(emmean),
      top_yields_2023 %>%
      filter(Treatment == "CR") %>%
      pull(emmean)),
    label = "ns",
    size = 5,
    vjust = -1.3,
    hjust = 0.6)

```

N_response_curve_2023

```

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

```

```

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

```

```

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

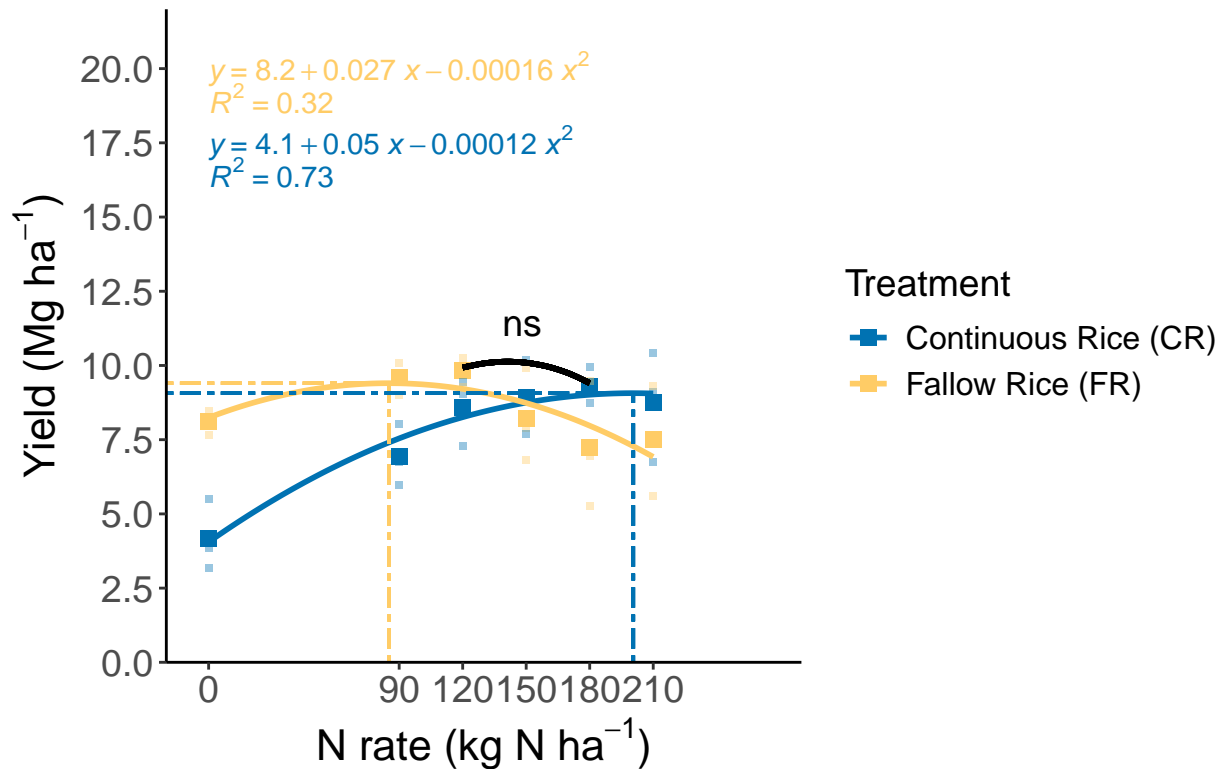
```

```

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

```


Yield: 2023



Average

Statistical test, comparisons by N rate and Treatment

```
model_average <- lmer(Yield_Mgha ~ NrateF*Treatment+(1|Treatment:Blk), data=yield_average)
anova(model_average)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##               Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## NrateF         531.51   53.151    10  66.280 55.5771 < 2.2e-16 ***
## Treatment         2.72    2.724     1  15.504  2.8482  0.1115
## NrateF:Treatment  49.73    4.973    10  66.280  5.2001  1.35e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
field_means_average = emmeans(model_average, spec = 'Treatment', by = 'NrateF')
field_effects_average = contrast(field_means_average, method = 'pairwise', adjust = "Tukey")
summary(field_effects_average)
```

```
## NrateF = 0:
```

```

## contrast estimate SE df t.ratio p.value
## CR - FR -2.904 0.529 67.2 -5.489 <.0001
##
## NrateF = 90:
## contrast estimate SE df t.ratio p.value
## CR - FR -2.412 0.907 79.8 -2.661 0.0094
##
## NrateF = 115:
## contrast estimate SE df t.ratio p.value
## CR - FR -1.318 0.644 73.8 -2.045 0.0444
##
## NrateF = 120:
## contrast estimate SE df t.ratio p.value
## CR - FR -0.990 0.907 79.8 -1.093 0.2779
##
## NrateF = 150:
## contrast estimate SE df t.ratio p.value
## CR - FR 0.961 0.907 79.8 1.060 0.2923
##
## NrateF = 155:
## contrast estimate SE df t.ratio p.value
## CR - FR -1.420 0.644 73.8 -2.204 0.0307
##
## NrateF = 180:
## contrast estimate SE df t.ratio p.value
## CR - FR 2.327 0.907 79.8 2.567 0.0121
##
## NrateF = 185:
## contrast estimate SE df t.ratio p.value
## CR - FR -1.132 0.644 73.8 -1.757 0.0831
##
## NrateF = 210:
## contrast estimate SE df t.ratio p.value
## CR - FR 1.516 0.907 79.8 1.672 0.0985
##
## NrateF = 225:
## contrast estimate SE df t.ratio p.value
## CR - FR -0.320 0.644 73.8 -0.496 0.6211
##
## NrateF = 260:
## contrast estimate SE df t.ratio p.value
## CR - FR -0.570 0.644 73.8 -0.885 0.3791
##
## Degrees-of-freedom method: kenward-roger

```

```
cld(field_means_average)
```

```

## NrateF = 0:
## Treatment emmean SE df lower.CL upper.CL .group
## CR 4.89 0.374 67.2 4.14 5.64 1
## FR 7.79 0.374 67.2 7.05 8.54 2
##
## NrateF = 90:
## Treatment emmean SE df lower.CL upper.CL .group

```

```

## CR          7.09 0.641 79.8      5.82      8.37  1
## FR          9.50 0.641 79.8      8.23      10.78  2
##
## NrateF = 115:
## Treatment emmean      SE    df lower.CL upper.CL .group
## CR          11.43 0.456 73.8     10.52     12.34  1
## FR          12.75 0.456 73.8     11.84     13.66  2
##
## NrateF = 120:
## Treatment emmean      SE    df lower.CL upper.CL .group
## CR          8.76 0.641 79.8      7.48     10.04  1
## FR          9.75 0.641 79.8      8.47     11.03  1
##
## NrateF = 150:
## Treatment emmean      SE    df lower.CL upper.CL .group
## FR          8.13 0.641 79.8      6.86      9.41  1
## CR          9.09 0.641 79.8      7.82     10.37  1
##
## NrateF = 155:
## Treatment emmean      SE    df lower.CL upper.CL .group
## CR          11.96 0.456 73.8     11.05     12.87  1
## FR          13.38 0.456 73.8     12.47     14.29  2
##
## NrateF = 180:
## Treatment emmean      SE    df lower.CL upper.CL .group
## FR          7.15 0.641 79.8      5.88      8.43  1
## CR          9.48 0.641 79.8      8.20     10.76  2
##
## NrateF = 185:
## Treatment emmean      SE    df lower.CL upper.CL .group
## CR          12.55 0.456 73.8     11.64     13.46  1
## FR          13.68 0.456 73.8     12.77     14.59  1
##
## NrateF = 210:
## Treatment emmean      SE    df lower.CL upper.CL .group
## FR          7.41 0.641 79.8      6.14      8.69  1
## CR          8.93 0.641 79.8      7.65     10.21  1
##
## NrateF = 225:
## Treatment emmean      SE    df lower.CL upper.CL .group
## CR          12.59 0.456 73.8     11.68     13.50  1
## FR          12.91 0.456 73.8     12.00     13.82  1
##
## NrateF = 260:
## Treatment emmean      SE    df lower.CL upper.CL .group
## CR          12.37 0.456 73.8     11.46     13.28  1
## FR          12.94 0.456 73.8     12.03     13.85  1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
##       then we cannot show them to be different.
##       But we also did not show them to be the same.

```

```
emmeans(model_average, ~ Treatment + NrateF)
```

```
## Treatment NrateF emmean SE df lower.CL upper.CL
## CR 0 4.89 0.374 67.2 4.14 5.64
## FR 0 7.79 0.374 67.2 7.05 8.54
## CR 90 7.09 0.641 79.8 5.82 8.37
## FR 90 9.50 0.641 79.8 8.23 10.78
## CR 115 11.43 0.456 73.8 10.52 12.34
## FR 115 12.75 0.456 73.8 11.84 13.66
## CR 120 8.76 0.641 79.8 7.48 10.04
## FR 120 9.75 0.641 79.8 8.47 11.03
## CR 150 9.09 0.641 79.8 7.82 10.37
## FR 150 8.13 0.641 79.8 6.86 9.41
## CR 155 11.96 0.456 73.8 11.05 12.87
## FR 155 13.38 0.456 73.8 12.47 14.29
## CR 180 9.48 0.641 79.8 8.20 10.76
## FR 180 7.15 0.641 79.8 5.88 8.43
## CR 185 12.55 0.456 73.8 11.64 13.46
## FR 185 13.68 0.456 73.8 12.77 14.59
## CR 210 8.93 0.641 79.8 7.65 10.21
## FR 210 7.41 0.641 79.8 6.14 8.69
## CR 225 12.59 0.456 73.8 11.68 13.50
## FR 225 12.91 0.456 73.8 12.00 13.82
## CR 260 12.37 0.456 73.8 11.46 13.28
## FR 260 12.94 0.456 73.8 12.03 13.85
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
cld_average <- cld(emmeans(model_average, ~ Treatment + NrateF), adjust = "Sidak")

top_yields_average <- cld_average %>%
  group_by(Treatment) %>%
  filter(emmean == max(emmean)) %>%
  ungroup()
```

Graph_average_with_equations

```
N_response_curve_average_v1 <-
ggplot(yield_average_dataframe, aes(x=Nrate, y=Yield_Mgha, color=Treatment))+
  geom_point(data=yield_average_dataframe, size=2.5)+ #this is the mean values
  geom_point(data=yield_average, size=1, alpha=0.4, aes(shape=Year))+ #this are the raw values
  scale_color_manual(values=c("#0072B2", "#FFCC66"), name = "Treatment", labels = c('Continuous Rice (CR)', 'Fertilizer (FR)'))
  scale_x_continuous(name=expression("N Rate (kg N ha"^{-1}*")"), limits = c(-5, 290), breaks = c(0, 115, 155, 185, 225, 260))
  scale_y_continuous(name= expression("Yield (Mg ha"^{-1}*")"), limits = c(0, 20), breaks = seq(0, 20, 1))
  #geom_errorbar(data=yield_average_dataframe, aes(ymin=Yield_Mgha-Yield_Mgha_se, ymax=Yield_Mgha+Yield_Mgha_se))
  theme_classic()+
  theme(axis.text = element_text(size = 10), axis.title = element_text(size=10))+
  geom_smooth(data = yield_average, method = "lm", formula = y ~ poly(x, 2), se = FALSE)+
  #geom_vline(xintercept = c(0, 115, 155, 185, 225, 260), linetype = "twodash", color = "black") +
  theme(legend.text = element_text(size = 10), legend.title = element_text(size = 14))+
```

```

stat_regline_equation(data=yield_average,
  aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
    label = paste(..rr.label..)),
  formula = y ~ poly(x, 2, raw = TRUE),
  show.legend = FALSE, label.x = 0, label.y = c(16.5, 19))+
stat_regline_equation(data=yield_average,
  aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
    label = paste(..eq.label..)),
  formula = y ~ poly(x, 2, raw = TRUE),
  show.legend = FALSE, label.x = 0, label.y = c(17.5, 20))+

theme(plot.title = element_text(hjust = 0.5, size = 15))+
ggtitle(expression("average"))+
annotate(
  "text",
  x = c(0),
  y = yield_average_dataframe %>%
    filter(Treatment == "FR", Nrate == 0) %>%
    mutate(Yield_Mgha_plus_higher = Yield_Mgha + 1) %>%
    pull(Yield_Mgha_plus_higher),
  label = "*",
  size = 7,
  vjust = 0
)+
geom_vline(
  aes(xintercept = optimal_results %>% filter(Treatment == "CR") %>% filter(Year == "average") %>% pull(optimal_results),
    linetype = "twodash", color = "#0072B2", size = 1
)+
geom_vline(
  aes(xintercept = optimal_results %>% filter(Treatment == "FR") %>% filter(Year == "average") %>% pull(optimal_results),
    linetype = "twodash", color = "#FFCC66", size = 1
)+
geom_curve(x = top_yields_average %>%
  filter(Treatment == "FR") %>%
  mutate(NrateF = as.numeric(paste(NrateF))) %>%
  pull(NrateF),
  y = top_yields_average %>%
  filter(Treatment == "FR") %>%
  pull(emmean)+0.2,
  xend = top_yields_average %>%
  filter(Treatment == "CR") %>%
  mutate(NrateF = as.numeric(paste(NrateF))) %>%
  pull(NrateF),
  yend = top_yields_average %>%
  filter(Treatment == "CR") %>%
  pull(emmean)+0.2,
  color = 1, size=1,
  curvature = -0.2,
  linetype = "twodash")+
annotate(
  "text",
  x = (top_yields_average %>%
    filter(Treatment == "FR") %>%

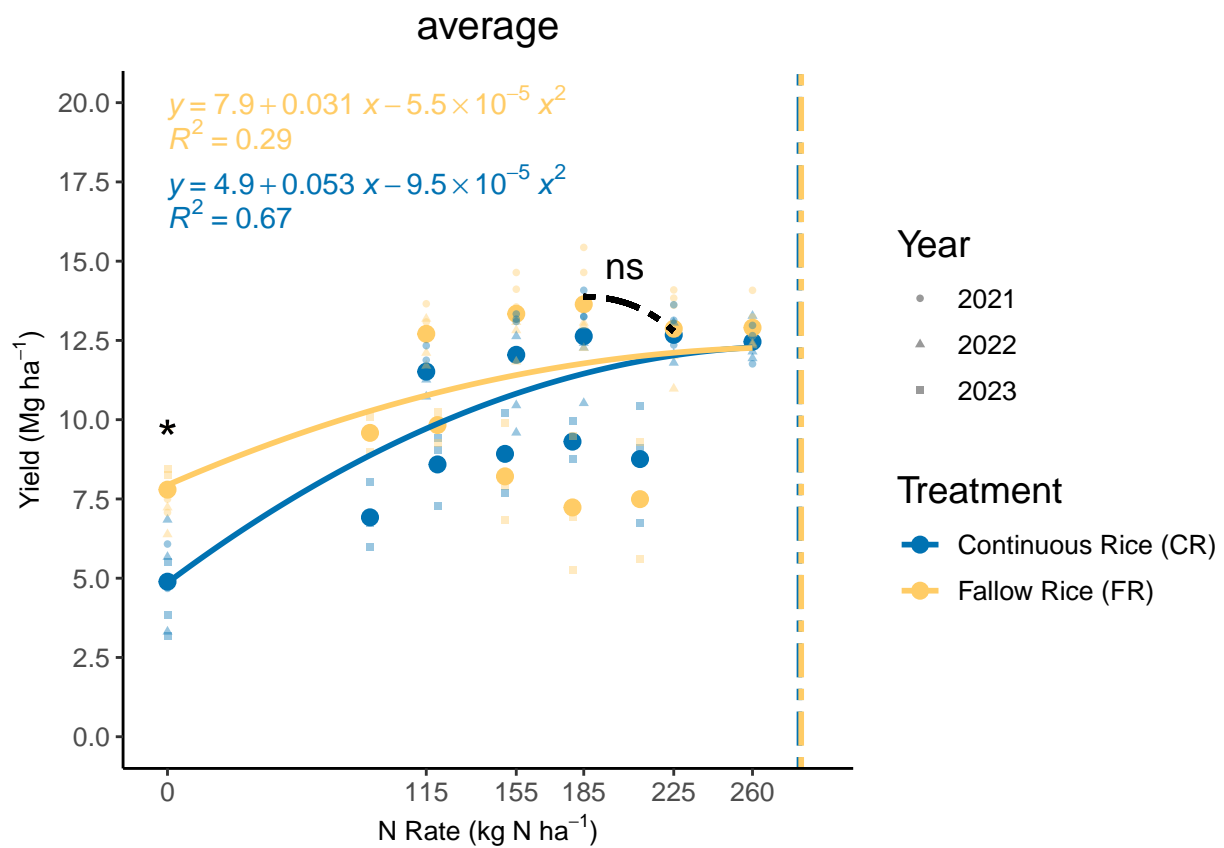
```

```

mutate(NrateF = as.numeric(paste(NrateF))) %>%
pull(NrateF)+
top_yields_average %>%
filter(Treatment == "CR") %>%
mutate(NrateF = as.numeric(paste(NrateF))) %>%
pull(NrateF))/2,
y = mean(top_yields_average %>%
filter(Treatment == "FR") %>%
pull(emmean),
top_yields_average %>%
filter(Treatment == "CR") %>%
pull(emmean)),
label = "ns",
size = 5,
vjust = -0.9,
hjust = 0.6)

```

N_response_curve_average_v1



Graph_average

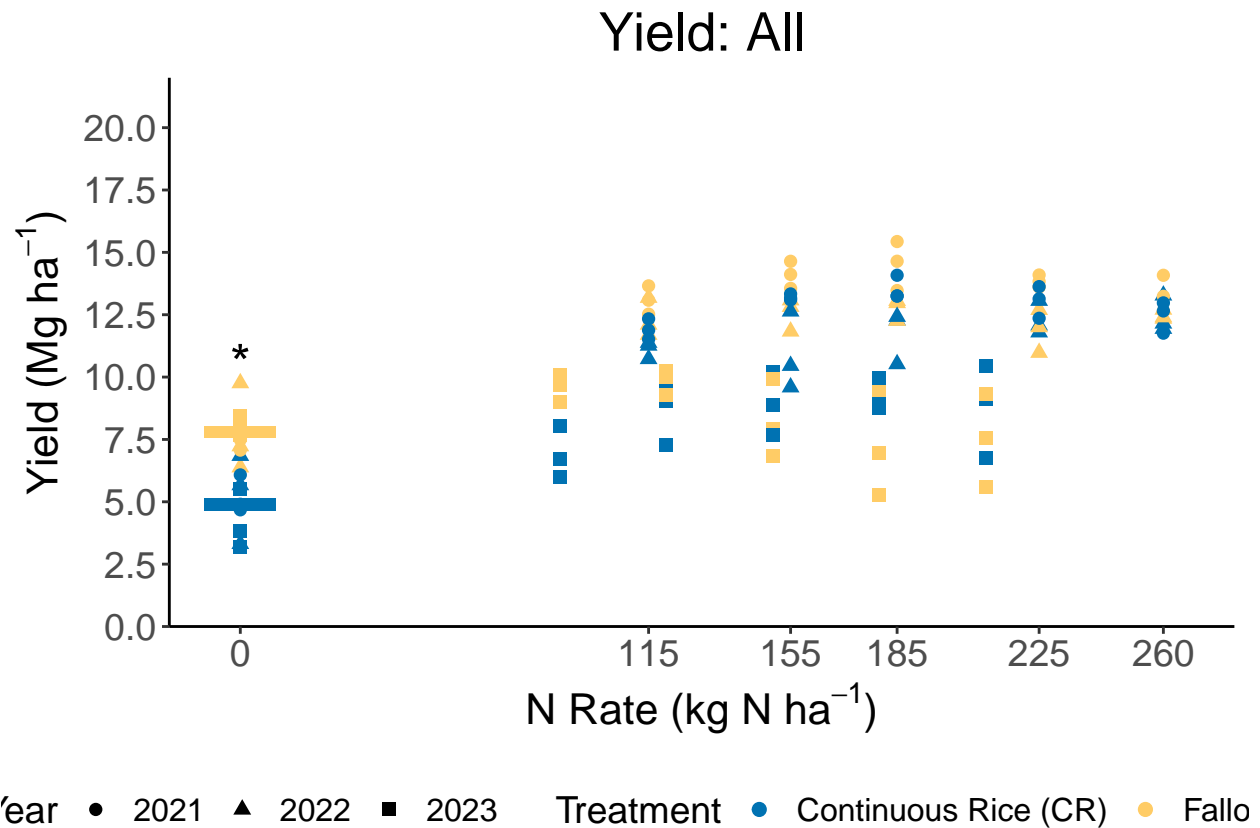
```

N_response_curve_average <-
ggplot(yield_average, aes(x=Nrate, y=Yield_Mgha, color=Treatment, shape=Year))+
  geom_point(data=yield_average, size=2)+ #this is the mean values+
  geom_segment(x = -10,
    y = yield_average_dataframe %>%
      filter(Nrate==0) %>%
      filter(Treatment == "FR") %>%
      pull(Yield_Mgha),
    xend = 10,
    yend = yield_average_dataframe %>%
      filter(Nrate==0) %>%
      filter(Treatment == "FR") %>%
      pull(Yield_Mgha),
    color = "#FFCC66", size=2)+
  geom_segment(x = -10,
    y = yield_average_dataframe %>%
      filter(Nrate==0) %>%
      filter(Treatment == "CR") %>%
      pull(Yield_Mgha),
    xend = 10,
    yend = yield_average_dataframe %>%
      filter(Nrate==0) %>%
      filter(Treatment == "CR") %>%
      pull(Yield_Mgha),
    color = "#0072B2", size=2)+
  #geom_point(data=yield_average, size=1, alpha=0.4)+ #this are the raw values
  scale_color_manual(values=c("#0072B2", "#FFCC66"), name = "Treatment", labels = c('Continuous Rice (CR)', 'Fertilizer (FR)'))+
  scale_x_continuous(name=expression("N Rate (kg N ha"-1)*"), limits = c(-20, 280), breaks = c(0, 115, 155, 185, 225, 260), labels = c(0, 115, 155, 185, 225, 260))+
  scale_y_continuous(name= expression("Yield (Mg ha"-1)*"), limits = c(0, 22), breaks = seq(0, 22, 2), labels = c(0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22))+
  theme_classic()+
  #geom_smooth(data = yield_average, aes(group = Treatment), method = "lm", formula = y ~ poly(x, 2), size = 1.5, color = "black") +
  #geom_vline(xintercept = c(0, 115, 155, 185, 225, 260), linetype = "twodash", color = "black") +
  #stat_regline_equation(data=yield_average,
    # aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
    # label = paste(..rr.label..)),
    # formula = y ~ poly(x, 2, raw = TRUE),
    # show.legend = FALSE, label.x = 0, label.y = c(16.5, 19))+
  #stat_regline_equation(data=yield_average,
    # aes(x=Nrate, y=Yield_Mgha, color=Treatment, group = Treatment,
    # label = paste(..eq.label..)),
    # formula = y ~ poly(x, 2, raw = TRUE),
    # show.legend = FALSE, label.x = 0, label.y = c(17.5, 20))+
  theme(axis.text = element_text(size = 14), axis.title = element_text(size=16))+
  theme(legend.text = element_text(size = 12), legend.title = element_text(size = 14))+
  theme(plot.title = element_text(hjust = 0.5, size = 20, face = "bold"))+
  ggtitle(expression("Yield: All"))+
  annotate(
    "text",
    x = c(0),
    y = yield_average_dataframe %>%
      filter(Treatment == "FR", Nrate == 0) %>%
      mutate(Yield_Mgha_plus_higher = Yield_Mgha + 2) %>%
      pull(Yield_Mgha_plus_higher),

```

```
label = "*",
size = 7,
vjust = 0
)+
theme(legend.position = "bottom")
```

N_response_curve_average



Combine all plots

```
Yields <- ggarrange(N_response_curve_2021,
  N_response_curve_2022,
  N_response_curve_2023,
  #N_response_curve_average,
  nrow = 1,
  ncol = 3,
  common.legend = TRUE,
  #legend.grob = get_legend(N_response_curve_average),
  legend= "bottom")
```

```
## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
```



```

## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
##   a single row.

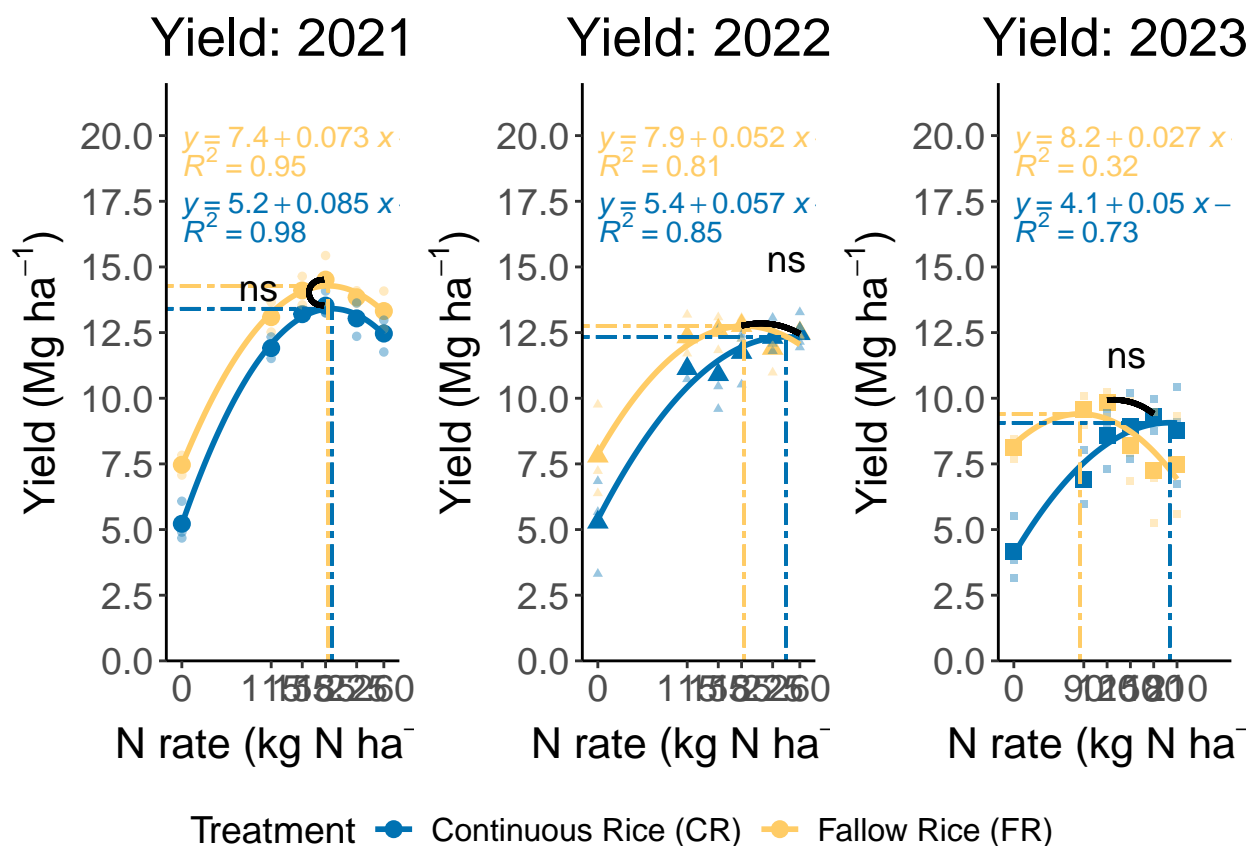
```

```
## Warning in geom_segment(aes(x = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
## a single row.
```

```
## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
## a single row.
```

```
## Warning in geom_segment(aes(y = optimal_results %>% filter(Treatment == : All aesthetics have length
## i Please consider using 'annotate()' or provide this layer with data containing
## a single row.
```

Yields



```
ggsave(filename = "Yields_no_avg.jpg", # Include the file extension here
        plot = Yields,                  # Specify the plot
        path = "D:/Academics/UC Davis/School Work/Linquist Lab/Data/R stats/Agronomic paper/Figures",
        dpi = 400,
        height = 20, width = 50, units = "cm")
```

Combine 0N yields into 1 excel file

```

all_means <- rbind(as.data.frame(field_means_2021) %>% mutate(Year = 2021),
                  as.data.frame(field_means_2022) %>% mutate(Year = 2022),
                  as.data.frame(field_means_2023) %>% mutate(Year = 2023),
                  as.data.frame(field_means_average) %>% mutate(Year = "Avg")
                  ) %>% arrange(Year) %>% filter(NrateF == "0") %>%
  mutate(yield = round(emmean, 1))

write_xlsx(all_means, "D:/Academics/UC Davis/School Work/Linguist Lab/Data/R stats/Agronomic paper/Yield")

```