Emissions Over Time All

Zhang Zhenglin

Contents

Necessary libraries

```
library(knitr)
library(ggplot2)
theme_set(theme_bw())
library(emmeans)
library(multcomp)
library(PLS205)
library(lme4)
library(lmerTest)
library(multcompView)
library(car)
library(Rmisc)
library(dplyr) #https://r4ds.had.co.nz/ (Chapter 3, Chapter 5, look at filter and select)
```

```
# https://bookdown.org/ansellbr/WEHI_tidyR_course_book/
library(stringr)
library(data.table)
library(GGally)
library(formatR)
library(readxl)
library(ggpubr)
library(lubridate)
library(paletteer)
library(forcats)
```

Data Organisation

Read from excel

```
#read directly from the directory where I generated the files from
master <- read_excel("D:/Academics/UC Davis/School Work/Linquist Lab/Data/R stats/GHG and MAOM POM/Daily
str(master)
```

```
## tibble [746 x 4] (S3: tbl_df/tbl/data.frame)
##  $ Plot        : chr [1:746] "106" "107" "204" "209" ...
##  $ Date        : POSIXct[1:746], format: "2021-05-11" "2021-05-11" ...
##  $ CH4_g_ha_day: num [1:746] 0 0 0 0 0 ...
##  $ N2O_g_ha_day: num [1:746] 0 0 0 0 0 ...
```

2021/2022

```
CRFRF_2021 <- master %>%
  filter(Date >= as.POSIXct("2021-05-10") & Date <= as.POSIXct("2022-05-12"))%>%
  mutate(Treatment = case_when(
    Plot %in% c("106", "204", "302") ~ "FR",
    Plot %in% c("107", "209", "307") ~ "CR",
    Plot %in% c("402", "505", "601") ~ "F",
    TRUE ~ "Other"  # This line handles cases where plot is not listed
  ))

CRFRF_2021 %>%
  group_by(Date, Treatment) %>%
  mutate(CH4_g_ha_day_se = sd(CH4_g_ha_day)/sqrt(3)) %>%
  summarise(CH4_g_ha_day = mean(CH4_g_ha_day),
            CH4_g_ha_day_se = mean(CH4_g_ha_day_se)) %>%
  ungroup() %>%
  group_by(Treatment)%>%
  summarise(max(CH4_g_ha_day))
```

```
## `summarise()` has grouped output by 'Date'. You can override using the
## `.groups` argument.
## # A tibble: 3 x 2
##   Treatment ‘max(CH4_g_ha_day)‘
##   <chr>                   <dbl>
## 1 CR                    7731.
## 2 F                        4.89
## 3 FR                    4628.
```

2021/22 CH4

```
CH4_g_ha_day_2021_graphing <- CRFRF_2021 %>% group_by(Date, Treatment) %>%
mutate(CH4_g_ha_day_se = sd(CH4_g_ha_day)/sqrt(3)) %>%
  summarise(CH4_g_ha_day = mean(CH4_g_ha_day),
            CH4_g_ha_day_se = mean(CH4_g_ha_day_se))
```

```
## ‘summarise()‘ has grouped output by ’Date’. You can override using the
## ‘.groups‘ argument.
```

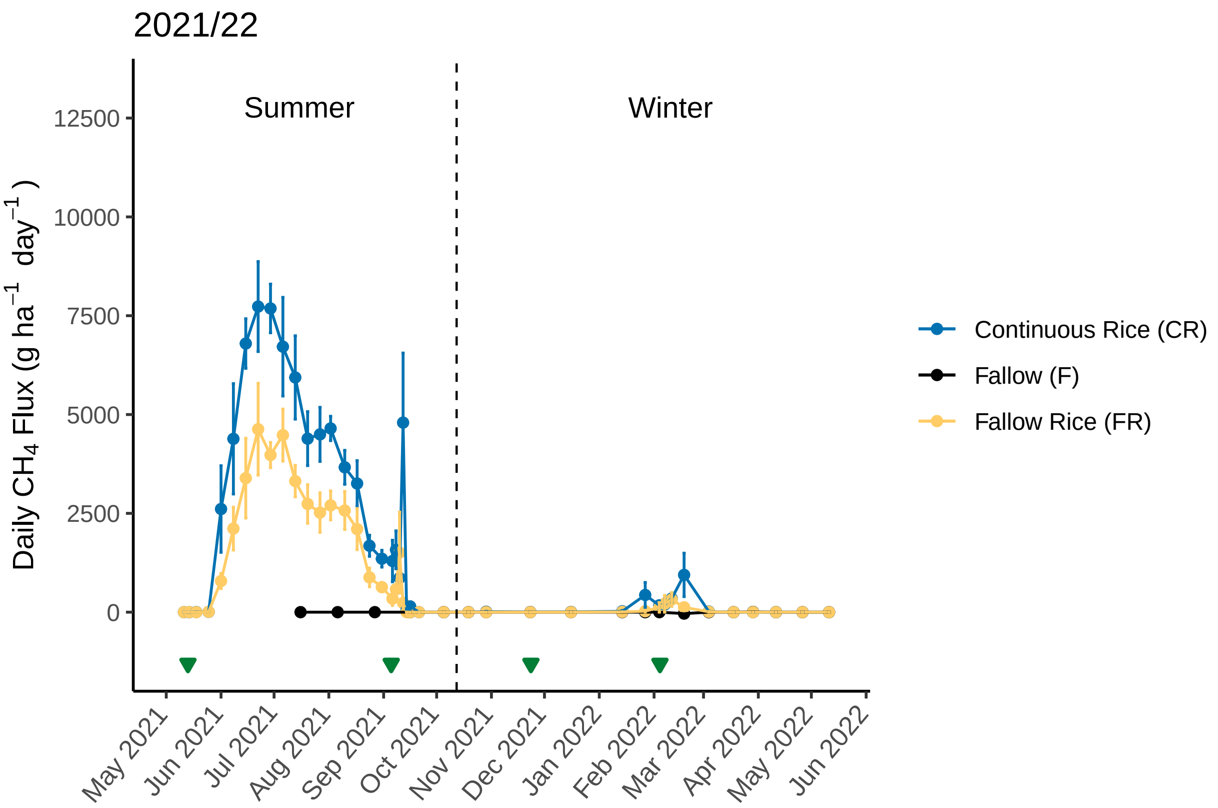
```
CH4_2021_graph <-
  ggplot(CH4_g_ha_day_2021_graphing, aes(x= Date, y = CH4_g_ha_day, color = Treatment))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=CH4_g_ha_day-CH4_g_ha_day_se, ymax=CH4_g_ha_day+CH4_g_ha_day_se))+
  scale_color_manual(values = c("CR" = "#0072B2", "FR" = "#FFCC66", "F" = "black"),
                     labels = c(expression("Continuous Rice (CR)"),
                                expression("Fallow (F)"),
                                expression("Fallow Rice (FR)")
                                )
                     )+
  labs(y = expression("Daily CH"[4]~"Flux (g ha"^-1~" day"^-1~")")) +
  scale_y_continuous(limits = c(-2000, 14000),
                     expand = c(0, 0),
                     breaks = seq(0, 14000, by = 2500))+
  scale_x_datetime(name = "",
                   limits = as.POSIXct(c("2021-05-01", "2022-05-15")),  # Set x-axis limits
                   date_labels = "%b %Y",
                   date_breaks = "1 month")+
  ggtitle("2021/22") +
  theme_classic()+
  theme(legend.title = element_blank(),
        axis.text.x = element_text(size=10,angle=50, hjust=1.2, vjust=1.2))+
  geom_vline(xintercept = as.POSIXct(c("2021-10-12")), linetype = "dashed", color = "black", size =0.4)
  annotate("point", x = as.POSIXct(c("2021-05-13",
                                       "2021-11-23")), y = -1300, shape = 25, color = "#017d35", fill =
    annotate("point", x = as.POSIXct(c("2021-09-05",
                                       "2022-02-04")), y = -1300, shape = 25, color = "#017d35", stroke
  annotate(
  "text",
  x = as.POSIXct(c("2021-07-15")),  # X-axis positions for annotations
  y = c(12500),  # Y-axis positions for annotations
  label = expression("Summer"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)+
  annotate(
  "text",
  x = as.POSIXct(c("2022-02-10")),  # X-axis positions for annotations
  y = c(12500),  # Y-axis positions for annotations
  label = expression("Winter"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)
```

```
## 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.
```

```
CH4_2021_graph
```

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

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



2021/22 N2O

```
N2O_g_ha_day_2021_graphing <- CRFRF_2021 %>% group_by(Date, Treatment) %>%
mutate(N2O_g_ha_day_se = sd(N2O_g_ha_day)/sqrt(3)) %>%
  summarise(N2O_g_ha_day = mean(N2O_g_ha_day),
            N2O_g_ha_day_se = mean(N2O_g_ha_day_se))
```

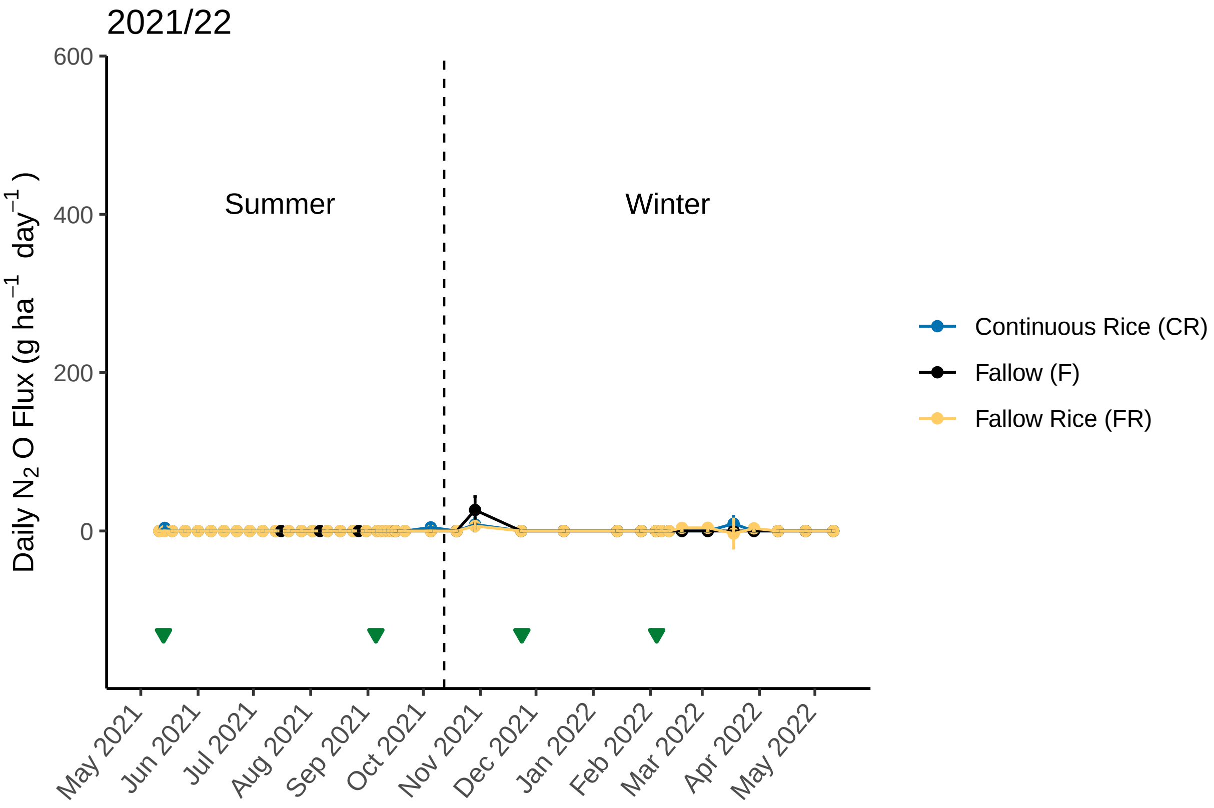
```
## `summarise()` has grouped output by 'Date'. You can override using the
## `.groups` argument.
```

```
N2O_2021_graph <-
  ggplot(N2O_g_ha_day_2021_graphing, aes(x= Date, y = N2O_g_ha_day, color = Treatment))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=N2O_g_ha_day-N2O_g_ha_day_se, ymax=N2O_g_ha_day+N2O_g_ha_day_se))+
  scale_color_manual(values = c("CR" = "#0072B2", "FR" = "#FFCC66", "F" = "black"),
                     labels = c(expression("Continuous Rice (CR)"),
                                expression("Fallow (F)"),
                                expression("Fallow Rice (FR)")
                                )
                     )+
  labs(y = expression("Daily N"[2]~"O Flux (g ha"^-1~" day"^-1~")")) +
  scale_y_continuous(limits = c(-199, 600),
                     expand = c(0, 0),
                     )+
  scale_x_datetime(name = "",
                   limits = as.POSIXct(c("2021-05-01", "2022-05-12")),  # Set x-axis limits
                   date_labels = "%b %Y",
                   date_breaks = "1 month")+
  ggtitle("2021/22") +
  theme_classic()+
  theme(legend.title = element_blank(),
      axis.text.x = element_text(size=10,angle=50, hjust=1.2, vjust=1.2))+
  geom_vline(xintercept = as.POSIXct(c("2021-10-12")), linetype = "dashed", color = "black", size =0.4)+
  annotate("point", x = as.POSIXct(c("2021-05-13",
                                       "2021-11-23")), y = -130, shape = 25, color = "#017d35", fill ="#
    annotate("point", x = as.POSIXct(c("2021-09-05",
                                       "2022-02-04")), y = -130, shape = 25, color = "#017d35", fill ="#
  annotate(
  "text",
  x = as.POSIXct(c("2021-07-15")),  # X-axis positions for annotations
  y = c(400),  # Y-axis positions for annotations
  label = expression("Summer"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)+
  annotate(
  "text",
  x = as.POSIXct(c("2022-02-10")),  # X-axis positions for annotations
  y = c(400),  # Y-axis positions for annotations
  label = expression("Winter"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)


N2O_2021_graph
```

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

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



2022/23

```
CRFRF_2022 <- master %>%
  filter(Date >= as.POSIXct("2022-05-15") & Date <= as.POSIXct("2023-04-19"))%>%
  filter (Plot %in% c(402, 409, 505, 512, 601, 608, 209, 307, "K1"))%>%
  mutate(Treatment = case_when(
    Plot %in% c("402", "505", "601") ~ "FR",
    Plot %in% c("409", "512", "608") ~ "CR",
    Plot %in% c("209", "307", "K1") ~ "F",
    TRUE ~ "Other"  # This line handles cases where plot is not listed
  ))

CRFRF_2022 %>%
  group_by(Date, Treatment) %>%
  mutate(CH4_g_ha_day_se = sd(CH4_g_ha_day)/sqrt(3)) %>%
  summarise(CH4_g_ha_day = mean(CH4_g_ha_day),
            CH4_g_ha_day_se = mean(CH4_g_ha_day_se)) %>%
  ungroup() %>%
  group_by(Treatment)%>%
  summarise(max(CH4_g_ha_day))
```

```
## ‘summarise()‘ has grouped output by ’Date’. You can override using the
## ‘.groups‘ argument.
```

```
## # A tibble: 3 x 2
##   Treatment ‘max(CH4_g_ha_day)‘
##   <chr>                   <dbl>
## 1 CR                     6077.
## 2 F                        15.7
## 3 FR                     7558.
```

2022/2023 CH4

```
CH4_g_ha_day_2022_graphing <- CRFRF_2022 %>% group_by(Date, Treatment) %>%
mutate(CH4_g_ha_day_se = sd(CH4_g_ha_day)/sqrt(3)) %>%
  summarise(CH4_g_ha_day = mean(CH4_g_ha_day),
            CH4_g_ha_day_se = mean(CH4_g_ha_day_se))
```

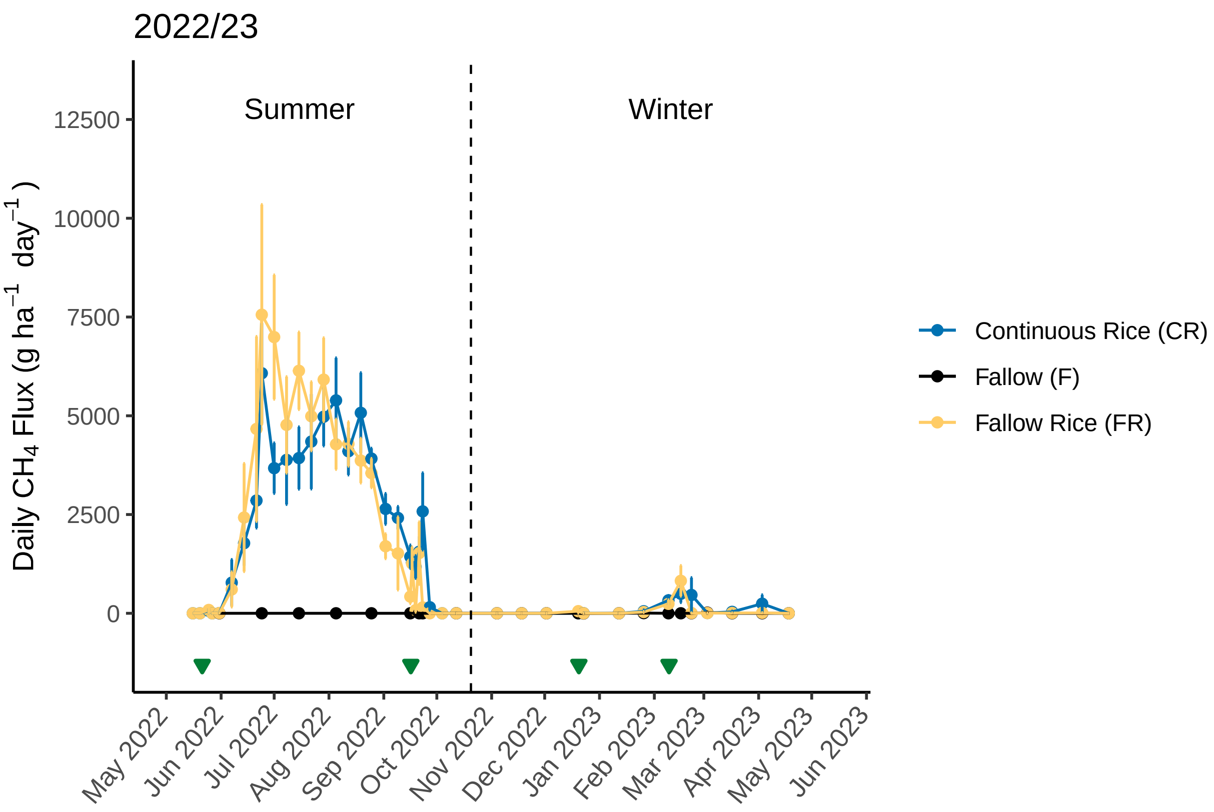
```
## ‘summarise()‘ has grouped output by ’Date’. You can override using the
## ‘.groups‘ argument.
```

```
CH4_2022_graph <-
  ggplot(CH4_g_ha_day_2022_graphing, aes(x= Date, y = CH4_g_ha_day, color = Treatment))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=CH4_g_ha_day-CH4_g_ha_day_se, ymax=CH4_g_ha_day+CH4_g_ha_day_se))+
  scale_color_manual(values = c("CR" = "#0072B2", "FR" = "#FFCC66", "F" = "black"),
                     labels = c(expression("Continuous Rice (CR)"),
                                expression("Fallow (F)"),
                                expression("Fallow Rice (FR)")
                                )
                     )+
  labs(y = expression("Daily CH"[4]~"Flux (g ha"^-1~" day"^-1~")")) +
  scale_y_continuous(limits = c(-2000, 14000),
                     expand = c(0, 0),
                     breaks = seq(0, 14000, by = 2500))+
  scale_x_datetime(name = "",
                   limits = as.POSIXct(c("2022-05-01", "2023-05-15")),  # Set x-axis limits
                   date_labels = "%b %Y",
                   date_breaks = "1 month")+
  ggtitle("2022/23") +
  theme_classic()+
  theme(legend.title = element_blank(),
        axis.text.x = element_text(size=10,angle=50, hjust=1.2, vjust=1.2))+
  geom_vline(xintercept = as.POSIXct(c("2022-10-20")), linetype = "dashed", color = "black", size =0.4)+
  annotate("point", x = as.POSIXct(c("2022-05-21",
                                       "2022-12-20")), y = -1300, shape = 25, color = "#017d35", fill =
    annotate("point", x = as.POSIXct(c("2022-09-16",
                                       "2023-02-09")), y = -1300, shape = 25, color = "#017d35", fill =
  annotate(
  "text",
  x = as.POSIXct(c("2022-07-15")),  # X-axis positions for annotations
  y = c(12500),  # Y-axis positions for annotations
  label = expression("Summer"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)+
  annotate(
  "text",
  x = as.POSIXct(c("2023-02-10")),  # X-axis positions for annotations
  y = c(12500),  # Y-axis positions for annotations
  label = expression("Winter"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)

CH4_2022_graph
```

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

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



2022/2023 N2O

```
N2O_g_ha_day_2022_graphing <- CRFRF_2022 %>% group_by(Date, Treatment) %>%
mutate(N2O_g_ha_day_se = sd(N2O_g_ha_day)/sqrt(3)) %>%
  summarise(N2O_g_ha_day = mean(N2O_g_ha_day),
            N2O_g_ha_day_se = mean(N2O_g_ha_day_se))
```

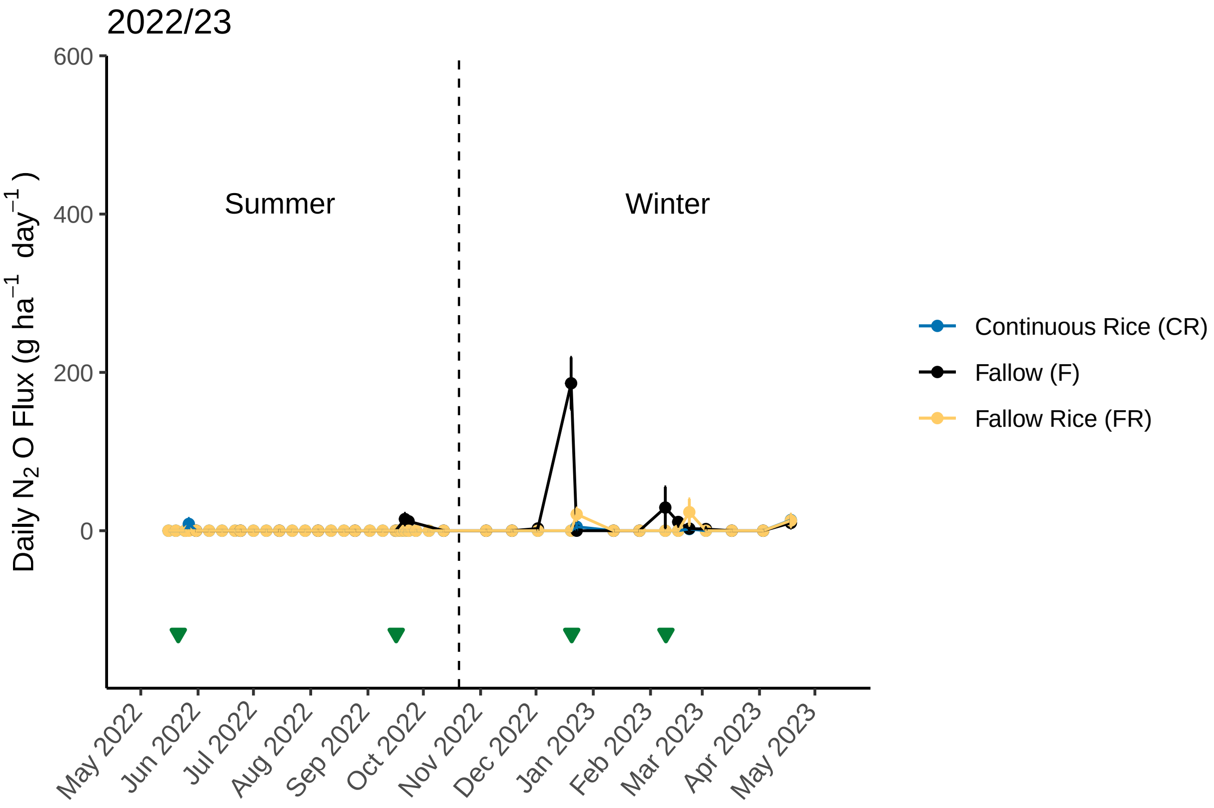
```
## ‘summarise()‘ has grouped output by ’Date’. You can override using the
## ‘.groups‘ argument.
```

```
N2O_2022_graph <-
  ggplot(N2O_g_ha_day_2022_graphing, aes(x= Date, y = N2O_g_ha_day, color = Treatment))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=N2O_g_ha_day-N2O_g_ha_day_se, ymax=N2O_g_ha_day+N2O_g_ha_day_se))+
  scale_color_manual(values = c("CR" = "#0072B2", "FR" = "#FFCC66", "F" = "black"),
                     labels = c(expression("Continuous Rice (CR)"),
                                expression("Fallow (F)"),
                                expression("Fallow Rice (FR)")
                                )
                     )+
  labs(y = expression("Daily N"[2]~"O Flux (g ha"^-1~" day"^-1~")")) +
  scale_y_continuous(limits = c(-199, 600),
                     expand = c(0, 0),
                     )+
  scale_x_datetime(name = "",
                   limits = as.POSIXct(c("2022-05-01", "2023-05-12")),  # Set x-axis limits
                   date_labels = "%b %Y",
                   date_breaks = "1 month")+
  ggtitle("2022/23") +
  theme_classic()+
  theme(legend.title = element_blank(),
        axis.text.x = element_text(size=10,angle=50, hjust=1.2, vjust=1.2))+
  geom_vline(xintercept = as.POSIXct(c("2022-10-20")), linetype = "dashed", color = "black", size =0.4)+
  annotate("point", x = as.POSIXct(c("2022-05-21",
                                       "2022-12-20")), y = -130, shape = 25, color = "#017d35", fill ="#
    annotate("point", x = as.POSIXct(c("2022-09-16",
                                       "2023-02-09")), y = -130, shape = 25, color = "#017d35",fill ="#0
  annotate(
  "text",
  x = as.POSIXct(c("2022-07-15")),  # X-axis positions for annotations
  y = c(400),  # Y-axis positions for annotations
  label = expression("Summer"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)+
  annotate(
  "text",
  x = as.POSIXct(c("2023-02-10")),  # X-axis positions for annotations
  y = c(400),  # Y-axis positions for annotations
  label = expression("Winter"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)

N2O_2022_graph
```

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

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



2023

```
CRFRF_2023 <- master %>%
  filter(Date >= as.POSIXct("2023-05-21") & Date <= as.POSIXct("2023-10-2"))%>%
  mutate(Treatment = case_when(
    Plot %in% c("701", "805", "903") ~ "FR",
    Plot %in% c("711", "812", "909") ~ "CR",
    TRUE ~ "Other"  # This line handles cases where plot is not listed
  ))

CRFRF_2023 %>%
  group_by(Date, Treatment) %>%
  mutate(CH4_g_ha_day_se = sd(CH4_g_ha_day)/sqrt(3)) %>%
  summarise(CH4_g_ha_day = mean(CH4_g_ha_day),
            CH4_g_ha_day_se = mean(CH4_g_ha_day_se)) %>%
  ungroup() %>%
  group_by(Treatment)%>%
  summarise(max(CH4_g_ha_day))
```

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

```
## # A tibble: 2 x 2
##   Treatment ‘max(CH4_g_ha_day)‘
##   <chr>                     <dbl>
## 1 CR                        8207.
## 2 FR                        4416.
```

2023 CH4

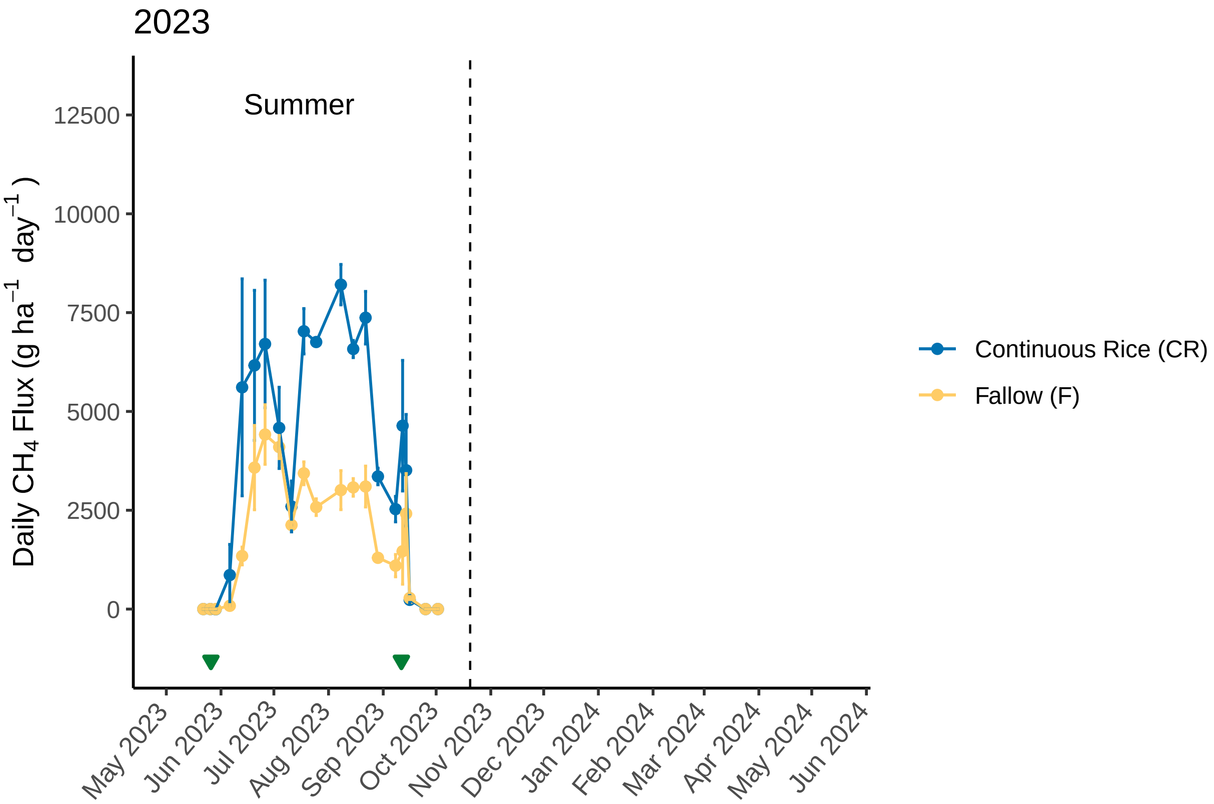
```
CH4_g_ha_day_2023_graphing <- CRFRF_2023 %>% group_by(Date, Treatment) %>%
mutate(CH4_g_ha_day_se = sd(CH4_g_ha_day)/sqrt(3)) %>%
  summarise(CH4_g_ha_day = mean(CH4_g_ha_day),
            CH4_g_ha_day_se = mean(CH4_g_ha_day_se))
```

```
## ‘summarise()‘ has grouped output by ’Date’. You can override using the
## ‘.groups‘ argument.
```

```
CH4_2023_graph <-
  ggplot(CH4_g_ha_day_2023_graphing, aes(x= Date, y = CH4_g_ha_day, color = Treatment))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=CH4_g_ha_day-CH4_g_ha_day_se, ymax=CH4_g_ha_day+CH4_g_ha_day_se))+
  scale_color_manual(values = c("CR" = "#0072B2", "FR" = "#FFCC66", "F" = "black"),
                     labels = c(expression("Continuous Rice (CR)"),
                                expression("Fallow (F)"),
                                expression("Fallow Rice (FR)")
                                )
                     )+
  labs(y = expression("Daily CH"[4]~"Flux (g ha"^-1~" day"^-1~")")) +
  scale_y_continuous(limits = c(-2000, 14000),
                     expand = c(0, 0),
                     breaks = seq(0, 14000, by = 2500))+
  scale_x_datetime(name = "",
                   limits = as.POSIXct(c("2023-05-01", "2024-05-15")),  # Set x-axis limits
                   date_labels = "%b %Y",
                   date_breaks = "1 month")+
  ggtitle("2023") +
  theme_classic()+
  theme(legend.title = element_blank(),
        axis.text.x = element_text(size=10,angle=50, hjust=1.2, vjust=1.2))+
  geom_vline(xintercept = as.POSIXct(c("2023-10-20")), linetype = "dashed", color = "black", size =0.4)+
  annotate("point", x = as.POSIXct(c("2023-05-26")), y = -1300, shape = 25, color = "#017d35", fill ="#0
    annotate("point", x = as.POSIXct(c("2023-09-11")), y = -1300, shape = 25, color = "#017d35", fill =
  annotate(
  "text",
  x = as.POSIXct(c("2023-07-15")),  # X-axis positions for annotations
  y = c(12500),  # Y-axis positions for annotations
  label = expression("Summer"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)

CH4_2023_graph
```

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



2023 N2O

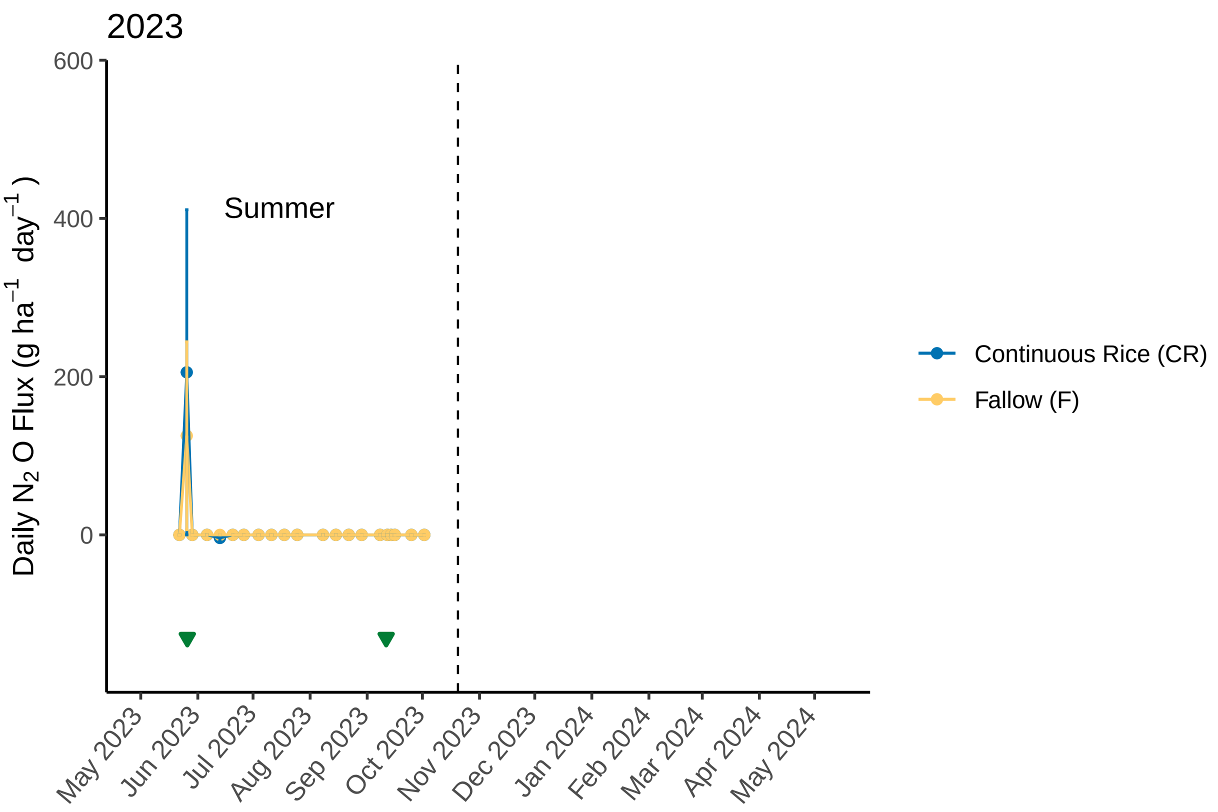
```
N2O_g_ha_day_2023_graphing <- CRFRF_2023 %>% group_by(Date, Treatment) %>%
mutate(N2O_g_ha_day_se = sd(N2O_g_ha_day)/sqrt(3)) %>%
  summarise(N2O_g_ha_day = mean(N2O_g_ha_day),
            N2O_g_ha_day_se = mean(N2O_g_ha_day_se))
```

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

```
N2O_2023_graph <-
  ggplot(N2O_g_ha_day_2023_graphing, aes(x= Date, y = N2O_g_ha_day, color = Treatment))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=N2O_g_ha_day-N2O_g_ha_day_se, ymax=N2O_g_ha_day+N2O_g_ha_day_se))+
  scale_color_manual(values = c("CR" = "#0072B2", "FR" = "#FFCC66", "F" = "black"),
                     labels = c(expression("Continuous Rice (CR)"),
                                expression("Fallow (F)"),
                                expression("Fallow Rice (FR)")
                                )
                     )+
  labs(y = expression("Daily N"[2]~"O Flux (g ha"^-1~" day"^-1~")")) +
  scale_y_continuous(limits = c(-199, 600),
                     expand = c(0, 0),
                     )+
  scale_x_datetime(name = "",
                   limits = as.POSIXct(c("2023-05-01", "2024-05-12")),  # Set x-axis limits
                   date_labels = "%b %Y",
                   date_breaks = "1 month")+
  ggtitle("2023") +
  theme_classic()+
  theme(legend.title = element_blank(),
        axis.text.x = element_text(size=10,angle=50, hjust=1.2, vjust=1.2))+
  geom_vline(xintercept = as.POSIXct(c("2023-10-20")), linetype = "dashed", color = "black", size =0.4)
  annotate("point", x = as.POSIXct(c("2023-05-26")), y = -130, shape = 25, color = "#017d35", fill ="#0
    annotate("point", x = as.POSIXct(c("2023-09-11")), y = -130, shape = 25, color = "#017d35", fill ="
  annotate(
  "text",
  x = as.POSIXct(c("2023-07-15")),  # X-axis positions for annotations
  y = c(400),  # Y-axis positions for annotations
  label = expression("Summer"),
  size = 4,
  vjust = 0  # Adjust vertical position of asterisks
)

N2O_2023_graph
```

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



Combine CR/FRsummer_winter into one figure

```
all_years_CH4 <- ggarrange(CH4_2021_graph, CH4_2022_graph, CH4_2023_graph,
                           nrow = 3,
                           common.legend = TRUE,
                           legend= "bottom")
```

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

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’
```

```
ggsave(all_years_CH4, filename = "all_years_CH4.jpg", height = 25, width = 25, units = "cm", dpi=400)

all_years_N2O <- ggarrange(N2O_2021_graph, N2O_2022_graph, N2O_2023_graph,
                 nrow = 3,
                 common.legend = TRUE,
                 legend= "bottom")
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## ’expression’
```

```
ggsave(all_years_N2O, filename = "all_years_N2O.jpg", height = 25, width = 25, units = "cm", dpi=400)
```