ECOL 610: Analysis of NEON Data - Part 1

Group - Santa Rita Experimental Range (SRER)

23 September, 2022

Contents

Group Members **1**

Setup **1**

Introduction **2**

Load Data **2**

Assignment **4**

Carbon fluxes: Plots of GPP, Re, and NEE over time 4

Environmental variables 8

Plots of C fluxes vs. Environmental variables 10

Scatter Plot Matrices (SPLOMS) 14

Group Members

- Emily Swartz
- Shahriar Shah Heydari
- Stephanie Cardinalli
- George Woolsey

Setup

First, load in the needed packages. Install the packages if needed.

```
library(tidyverse)
library(lubridate)
library(viridis)
library(RColorBrewer)
library(scales)
library(latex2exp)
library(psych)
```

```
remove(list=ls())
# set year range
min_yr <- 2020
max_yr <- 2021
# what is your site name?
site <- "Santa Rita Experimental Range"
```

Introduction

Net ecosystem exchange (NEE) is defined, by convention, as CO_2 flux from the ecosystem to the atmosphere. It corresponds to a negative carbon input to ecosystems. NEE is defined in this way because atmospheric scientists, who originated the term, seek to document net sources of CO_2 to the atmosphere (i.e., NEE) that account for rising atmospheric CO_2 concentration. Therefore, CO_2 input to the ecosystem is a negative NEE.

Chapin, F. S., Matson, P. A., Mooney, H. A., & Vitousek, P. M. (2002). Principles of terrestrial ecosystem ecology. p.208

Key point: CO_2 **input to the ecosystem is a negative NEE**

$$NEE = R_E - GPP$$

$$GPP = R_E - NEE$$

$$NEP = GPP - R_E$$

Load Data

Information about these variables can be found in the metadata file

```
# create title date range
if(min_yr==max_yr){
  yr_lab <- paste0(min_yr)
}else{
  yr_lab <- paste0(min_yr, "-", max_yr)
}

# load 30 min data
  # I put both CPER and SRER data in this directory
  # all data will be loaded in the same R data set
f_list <- list.files(path = "../data/", pattern="*30 min.csv")
for (i in 1:length(f_list)){
  nm <- stringr::word(f_list[i], start = 1, sep = "30 min") %>%
    stringr::str_replace_all(pattern = "[[:punct:]]", replacement = "") %>%
    stringr::str_trim() %>%
    stringr::str_squish()
  temp <- read.csv(paste0("../data/", f_list[i])) %>%
    dplyr::mutate(neon_site_name = nm) %>%
    dplyr::relocate(neon_site_name) %>%
```

```
    dplyr::rename_with(~ tolower(
      gsub(" ", "_",
         str_trim(gsub("\\s+", " ", .x))
      )
    ))
  if(i==1){
    dta_30min <- temp
  }else{
    dta_30min <- dplyr::union_all(dta_30min, temp)
  }
  remove(temp)
}
# load daily data
f_list <- list.files(path = "../data/", pattern="*daily.csv")
for (i in 1:length(f_list)){
  nm <- stringr::word(f_list[i], start = 1, sep = "daily") %>%
    stringr::str_replace_all(pattern = "[[:punct:]]", replacement = "") %>%
    stringr::str_trim() %>%
    stringr::str_squish()
  temp <- read.csv(paste0("../data/", f_list[i])) %>%
    dplyr::mutate(neon_site_name = nm) %>%
    dplyr::relocate(neon_site_name) %>%
    dplyr::rename_with(~ tolower(
      gsub(" ", "_",
         str_trim(gsub("\\s+", " ", .x))
      )
    ))
  if(i==1){
    dta_1day <- temp
  }else{
    dta_1day <- dplyr::union_all(dta_1day, temp)
  }
  remove(temp)
}
# create dates and record counts
# 1-day
dta_1day <- dta_1day %>%
  dplyr::mutate(
    date_id = lubridate::make_date(year = year, month = month, day = day)
    , week = lubridate::week(date_id)
    , has_gpp = ifelse(!is.na(gpp), 1, 0)
    , season =
        dplyr::case_when(
          month %in% c(1:3, 12) ~ "Winter"
          , month %in% c(4:5) ~ "Spring"
          , month %in% c(6:8) ~ "Summer"
          , month %in% c(9:11) ~ "Autumn"
          , TRUE  ~ "Other")
  ) %>%
  dplyr::group_by(neon_site_name, week, year) %>%
  dplyr::mutate(is_full_week = sum(has_gpp)==7) %>%
  dplyr::ungroup()
# 30-min
```

```
dta_30min <- dta_30min %>%
  dplyr::mutate(
    date_id = lubridate::make_date(year = year, month = month, day = day)
    , time_id = lubridate::make_datetime(year = year, month = month, day = day
                          , hour = floor(hour)
                          , min = (hour-floor(hour))*60
                )
    , week = lubridate::week(date_id)
    , has_gpp = ifelse(!is.na(gpp), 1, 0)
    , season =
        dplyr::case_when(
          month %in% c(1:3, 12) ~ "Winter"
          , month %in% c(4:5) ~ "Spring"
          , month %in% c(6:8) ~ "Summer"
          , month %in% c(9:11) ~ "Autumn"
          , TRUE  ~ "Other")
  ) %>%
  dplyr::group_by(neon_site_name, week, year) %>%
  dplyr::mutate(is_full_week = sum(has_gpp)==24*2*7) %>%
  dplyr::ungroup()
```

Assignment

For one year of data (your choice!) for CPER and your site please create the following plots and use them to answer the below questions:

- Carbon fluxes: Plots of GPP, Re, and NEE over time
- Environmental variables: Plots of air temperature, soil temperature, soil moisture and PAR/PPFD over time
- Plots of GPP, Re, NEE vs. Environmental variables and color by day of year (ask Katie or your group mates if you need help!)

Then answer the following questions about each graph:

Identify outliers (make sure data look ok) Describe patterns and hypothesize what is driving them.

Submit your code, plots, and the answers to these questions in an RMarkdown PDF. As always, let us know if you have questions. This is due Tuesday, September 27th at 9:30am (class start).

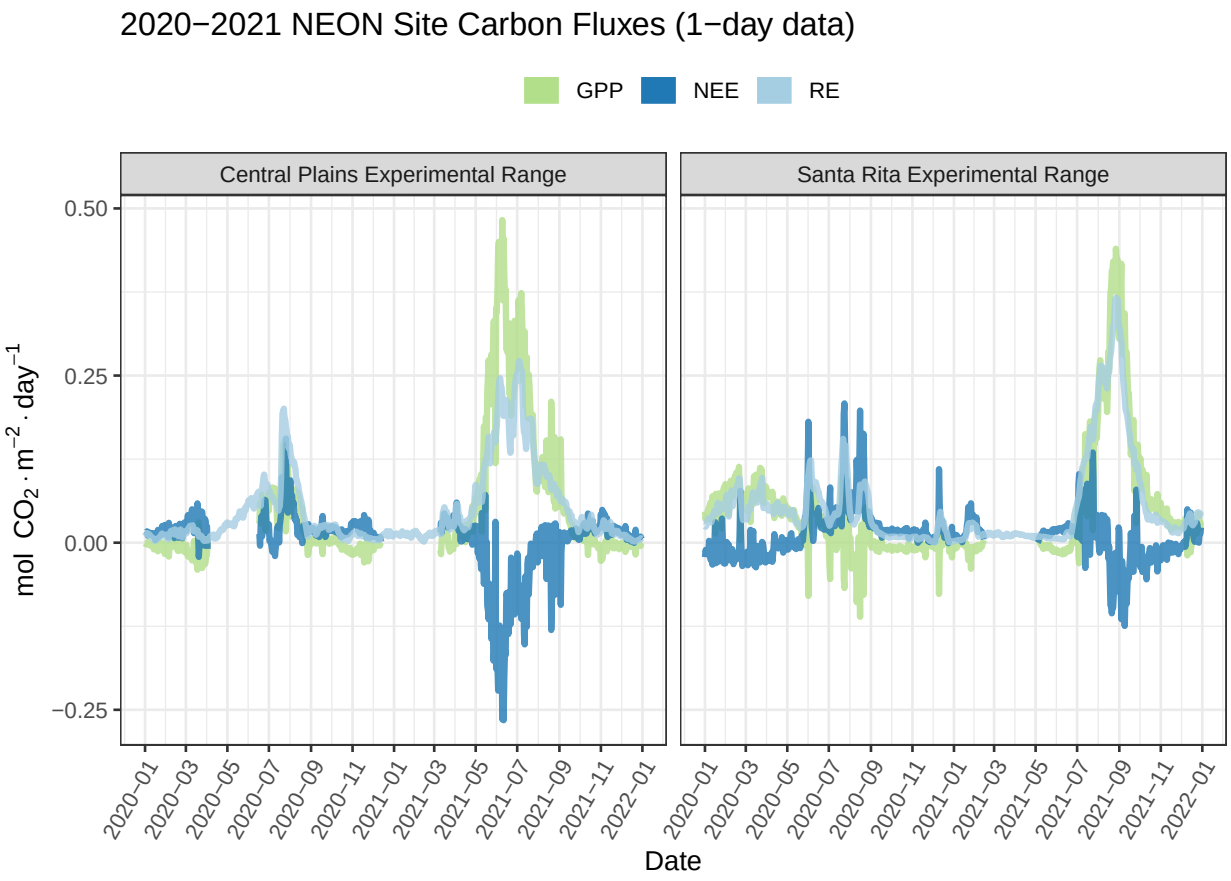
Carbon fluxes: Plots of GPP, Re, and NEE over time

Compare annual time-trend of GPP, NEE, R_E for sites:

Daily

```
# plot daily gpp, nee, re
dta_1day %>%
  dplyr::filter(
    # neon_site_name == site
```

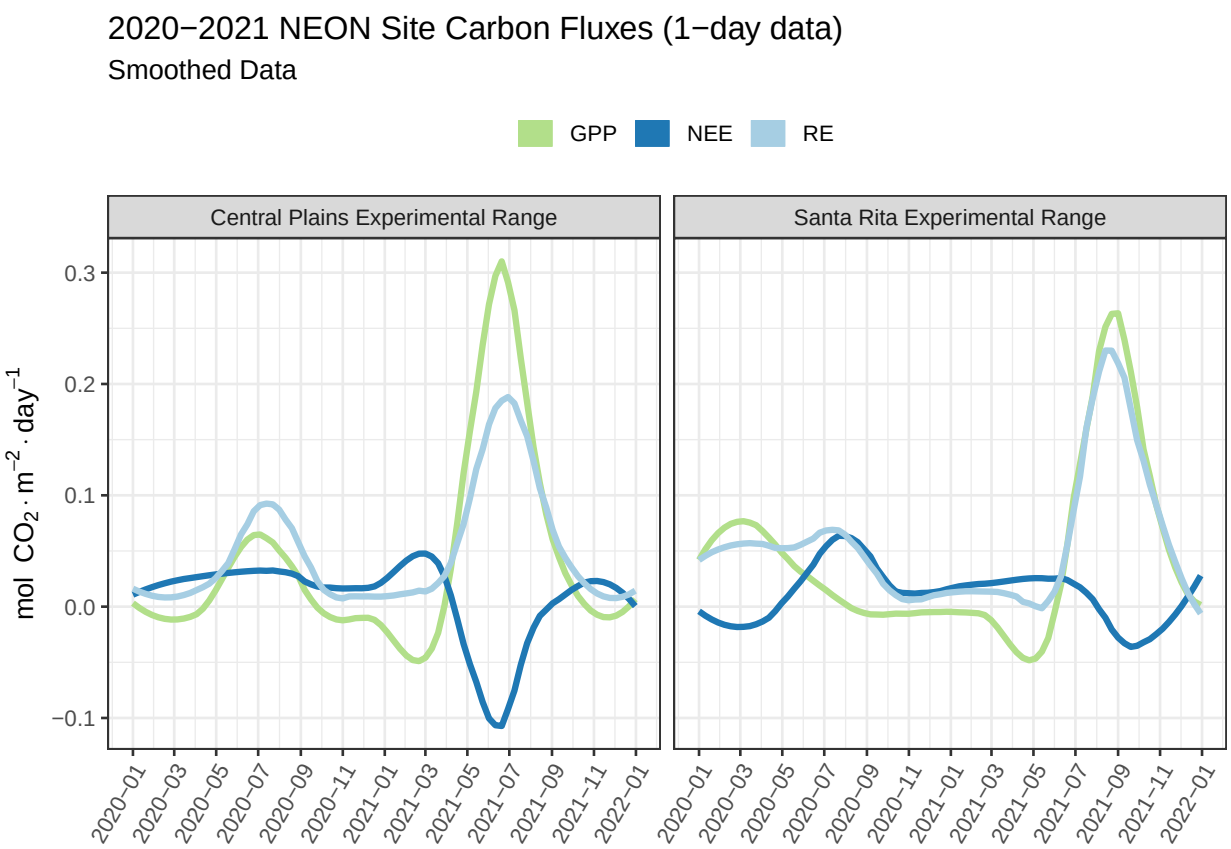
```
    year %in% c(min_yr:max_yr)
  ) %>%
ggplot(., aes(x = date_id)) +
  geom_line(
    aes(y = gpp, color = "GPP")
    , lwd = 1.2
    , alpha = 0.8
  ) +
  geom_line(
    aes(y = nee, color = "NEE")
    , lwd = 1.2
    , alpha = 0.8
  ) +
  geom_line(
    aes(y = re, color = "RE")
    , lwd = 1.2
    , alpha = 0.8
  ) +
  facet_wrap(~neon_site_name) +
  scale_color_brewer(type = "qual", palette = "Paired", direction = -1) +
  scale_x_date(date_breaks = "2 month", date_labels = "%Y-%m") +
  xlab("Date") +
  ylab(latex2exp::TeX("$mol \\; CO_{2} \\cdot m^{-2} \\cdot day^{-1}$")) +
  labs(
    title = paste0(yr_lab, " NEON Site Carbon Fluxes (1-day data)")
  ) +
  theme_bw() +
  theme(
    legend.position = "top"
    , legend.direction = "horizontal"
    , legend.title = element_blank()
    , axis.text.x = element_text(angle = 60, hjust=1)
  ) +
  guides(color = guide_legend(override.aes = list(size = 5)))
```



Smoothed Daily

```
# plot daily gpp, nee, re
dta_1day %>%
  dplyr::filter(
    # neon_site_name == site
    year %in% c(min_yr:max_yr)
  ) %>%
ggplot(., aes(x = date_id)) +
  geom_smooth(
    aes(y = gpp, color = "GPP")
    , method = "loess"
    , span = 0.3
    , se = FALSE
    , lwd = 1.2
    , alpha = 0.8
  ) +
  geom_smooth(
    aes(y = nee, color = "NEE")
    , method = "loess"
    , span = 0.3
    , se = FALSE
    , lwd = 1.2
```

```
  , alpha = 0.8
) +
geom_smooth(
  aes(y = re, color = "RE")
  , method = "loess"
  , span = 0.3
  , se = FALSE
  , lwd = 1.2
  , alpha = 0.8
) +
facet_wrap(~neon_site_name) +
scale_color_brewer(type = "qual", palette = "Paired", direction = -1) +
scale_x_date(date_breaks = "2 month", date_labels = "%Y-%m") +
xlab("") +
ylab(latex2exp::TeX("$mol \\; CO_{2} \\cdot m^{-2} \\cdot day^{-1}$")) +
labs(
  title = paste0(yr_lab, " NEON Site Carbon Fluxes (1-day data)")
  , subtitle = "Smoothed Data"
) +
theme_bw() +
theme(
  legend.position = "top"
  , legend.direction = "horizontal"
  , legend.title = element_blank()
  , axis.text.x = element_text(angle = 60, hjust=1)
) +
guides(color = guide_legend(override.aes = list(size = 5)))
```



Environmental variables

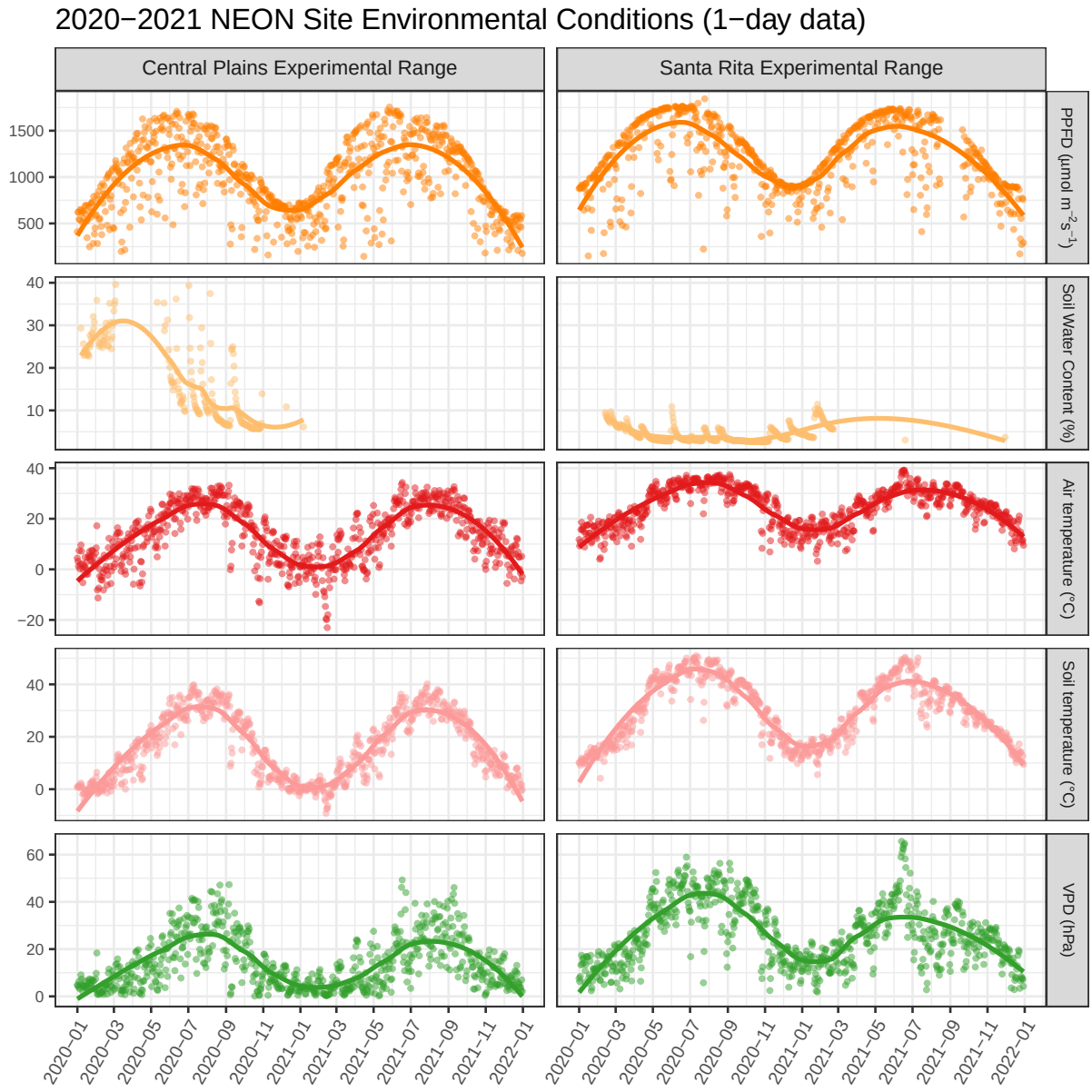
Plots of air temperature, soil temperature, soil moisture and PAR/PPFD over time

```
env_vars <- c("swc", "ta", "ts", "ppfd_in", "vpd")

# filter data
    dta_1day %>%
      dplyr::filter(
        year %in% c(min_yr:max_yr)
      ) %>%
      dplyr::select(
        neon_site_name
        , date_id
        , tidyselect::all_of(env_vars)
      ) %>%
  # pivot dependent vars
    tidyr::pivot_longer(
      cols = tidyselect::all_of(env_vars)
      , names_to = "var_name"
      , values_to = "var_value"
      , values_drop_na = FALSE
    ) %>%
    dplyr::mutate(
```

```
    var_name_long = factor(var_name, labels = c(
      expression(paste("PPFD (", mu, "mol ", m^{-2}, s^{-1}, ")"))
      , expression(paste("Soil Water Content (%)"))
      , expression(paste("Air temperature (\u00B0C)"))
      , expression(paste("Soil temperature (\u00B0C)"))
      , expression(paste("VPD (hPa)"))
      )
    )
    , site_lab = factor(neon_site_name, labels = c(
        expression(paste("Central Plains Experimental Range"))
        , expression(paste("Santa Rita Experimental Range"))
      )
    )
  ) %>%
  dplyr::arrange(
    neon_site_name
    , date_id
    , var_name
  ) %>%
# plot
  ggplot(
    data = . # "." means the data that is passed to ggplot via the pipe "%>%"
    , mapping = aes(
      x = date_id
      , y = var_value
      , color = var_name_long
      , group = neon_site_name
    )
  ) +
  geom_point(
    alpha = 0.5
    , size = 0.8
  ) +
  geom_smooth(
    method = "loess"
    , se = FALSE
    , span = 0.5
    , lwd = 1
    , alpha = 0.8
  ) +
  facet_grid(var_name_long~site_lab, scales = "free_y", labeller = label_parsed) +
  scale_color_manual(values = RColorBrewer::brewer.pal(n = 8, "Paired")[8:4]) +
  scale_x_date(date_breaks = "2 month", date_labels = "%Y-%m") +
  xlab("") +
  ylab("") +
  labs(
    title = paste0(yr_lab, " NEON Site Environmental Conditions (1-day data)")
    # , subtitle = "Smoothed Data"
  ) +
  theme_bw() +
  theme(
    legend.position = "none"
    , axis.text.x = element_text(angle = 60, hjust=1, size = 8)
```

```
    , axis.text.y = element_text(size = 7)
    , strip.text.y = element_text(size = 7)
  )
```



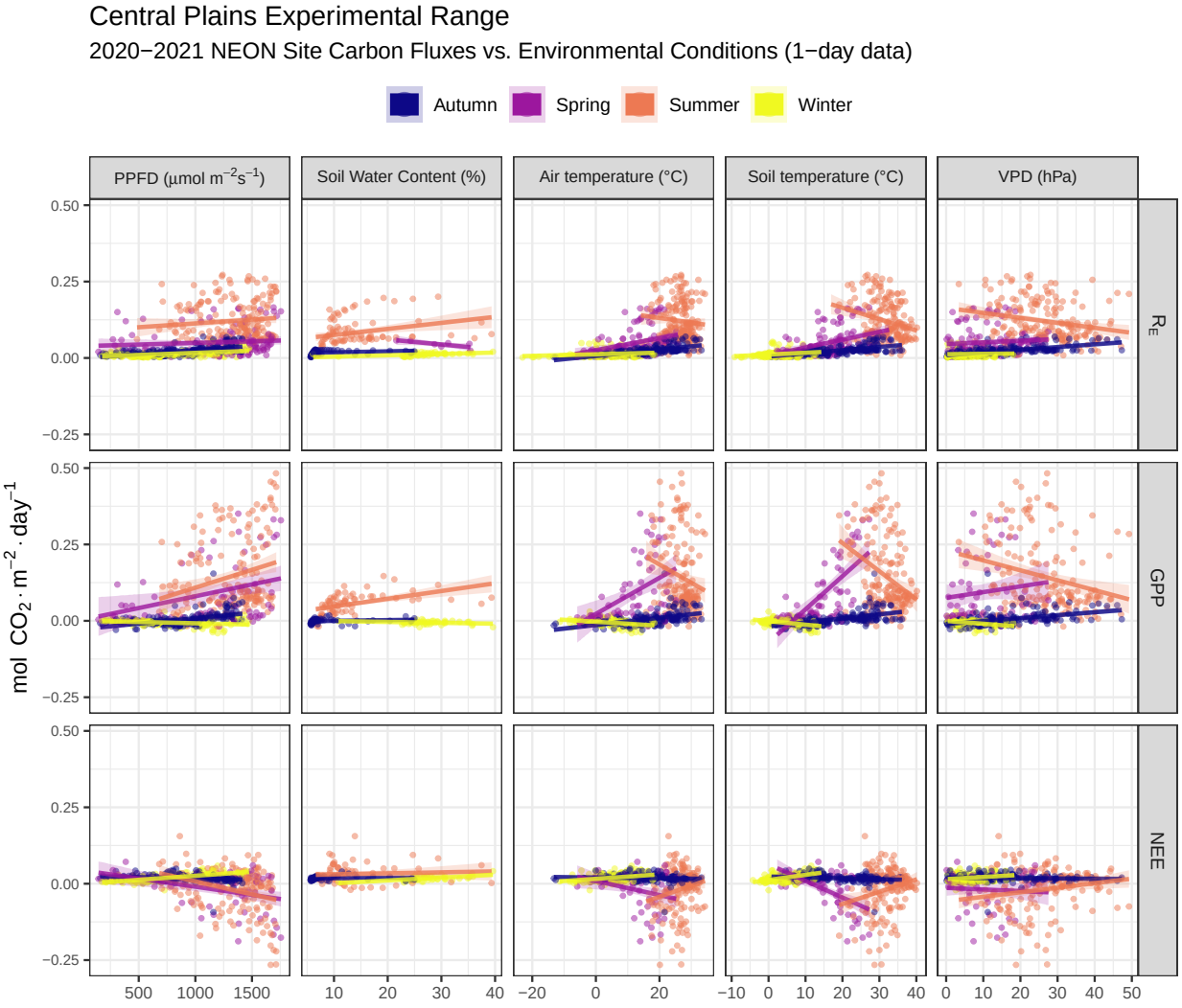
Plots of C fluxes vs. Environmental variables

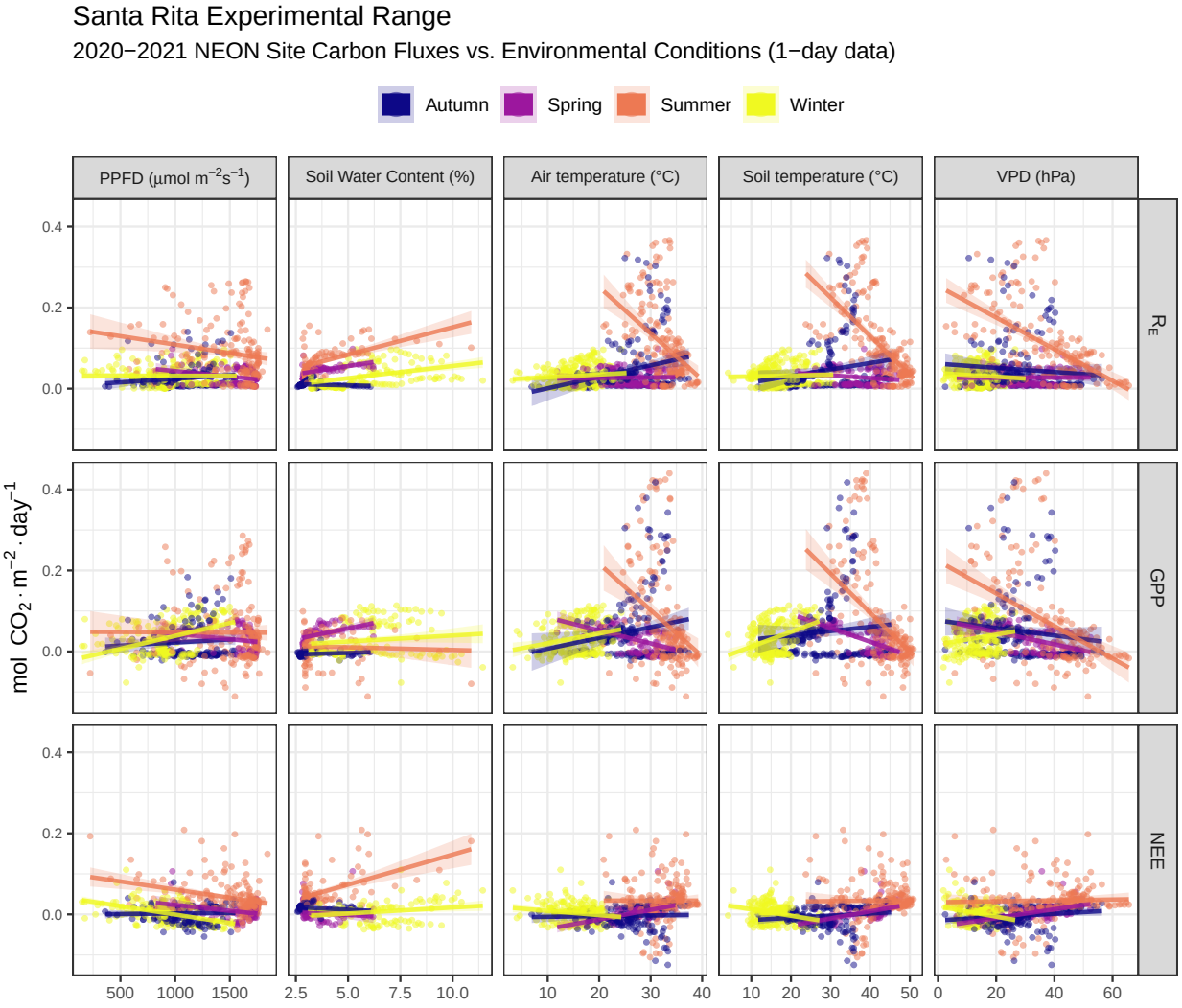
Plots of GPP, Re, NEE vs. Environmental variables and color by ~~day of year~~ **season**

I personally found the coloring of the points by day of year to be difficult to interpret and instead colored the points by season so that high and low days of the year (i.e. winter) are colored the same.

```r
env_vars <- c("swc", "ta", "ts", "ppfd_in", "vpd")
dep_vars <- c("nee", "re", "gpp")
plt_flx_env <- function(my_site){(
# filter data
    dta_1day %>%
      dplyr::filter(
        year %in% c(min_yr:max_yr)
        , neon_site_name %in% my_site
      ) %>%
      dplyr::select(
        neon_site_name
        , date_id
        , season
        , tidyselect::all_of(dep_vars)
        , tidyselect::all_of(env_vars)
      ) %>%
  # pivot dependent vars
    tidyr::pivot_longer(
      cols = tidyselect::all_of(dep_vars)
      , names_to = "dep_var_name"
      , values_to = "dep_var_value"
      , values_drop_na = FALSE
    ) %>%
  # pivot independent vars
    tidyr::pivot_longer(
      cols = tidyselect::all_of(env_vars)
      , names_to = "var_name"
      , values_to = "var_value"
      , values_drop_na = FALSE
    ) %>%
    dplyr::mutate(
      var_name_long = factor(var_name, labels = c(
        expression(paste("PPFD (", mu, "mol ", m^{-2}, s^{-1}, ")"))
        , expression(paste("Soil Water Content (%)"))
        , expression(paste("Air temperature (\u00B0C)"))
        , expression(paste("Soil temperature (\u00B0C)"))
        , expression(paste("VPD (hPa)"))
        )
      )
      , dep_var_lab = ordered(dep_var_name, levels = c("re", "gpp", "nee"), labels = c(
          expression(paste(R[E]))
          , expression(paste("GPP"))
          , expression(paste("NEE"))
        )
      )
    ) %>%
    dplyr::arrange(
      neon_site_name
      , date_id
      , dep_var_name
      , var_name
    ) %>%
  # plot
```

```
    ggplot(
      data = . # "." means the data that is passed to ggplot via the pipe "%>%"
      , mapping = aes(
        x = var_value
        , y = dep_var_value
        , color = season
        , fill = season
        # , group = season
      )
    ) +
    geom_point(
      alpha = 0.5
      , size = 0.8
    ) +
    geom_smooth(
      method = "lm"
      , se = TRUE
      # , span = 3
      , lwd = 1
      , alpha = 0.2
    ) +
    facet_grid(dep_var_lab~var_name_long, scales = "free_x", labeller = label_parsed) +
    scale_color_viridis_d(option = "plasma", alpha = 0.8) +
    scale_fill_viridis_d(option = "plasma") +
    # scale_x_date(date_breaks = "2 month", date_labels = "%Y-%m") +
    xlab("") +
    ylab(latex2exp::TeX("$mol \\; CO_{2} \\cdot m^{-2} \\cdot day^{-1}$")) +
    labs(
      title = my_site
      , subtitle = paste0(yr_lab, " NEON Site Carbon Fluxes vs. Environmental Conditions (1-day data)")
    ) +
    theme_bw() +
    theme(
      legend.position = "top"
      , legend.direction = "horizontal"
      , legend.title = element_blank()
      , axis.text.x = element_text(size = 8)
      , axis.text.y = element_text(size = 7)
      , strip.text.x = element_text(size = 8)
      , strip.text.y = element_text(size = 9)
    ) +
    guides(color = guide_legend(override.aes = list(size = 5)))
)}
# sites
sts <- unique(dta_1day$neon_site_name)
for (i in 1:length(sts)) {
  print(plt_flx_env(my_site = sts[i]))
}
```


Central Plains Experimental Range
2020−2021 NEON Site Carbon Fluxes vs. Environmental Conditions (1−day data)
Autumn
Spring
Summer
Winter
PPFD (µmol m−2s−1)
Soil Water Content (%)
Air temperature (°C)
Soil temperature (°C)
VPD (hPa)
RE
GPP
NEE
mol CO2 · m−2 · day−1
0.50
0.25
0.00
−0.25
500
1000
1500
10
20
30
40
−20
0
20
−10
0
10
20
30
40
0
10
20
30
40
50

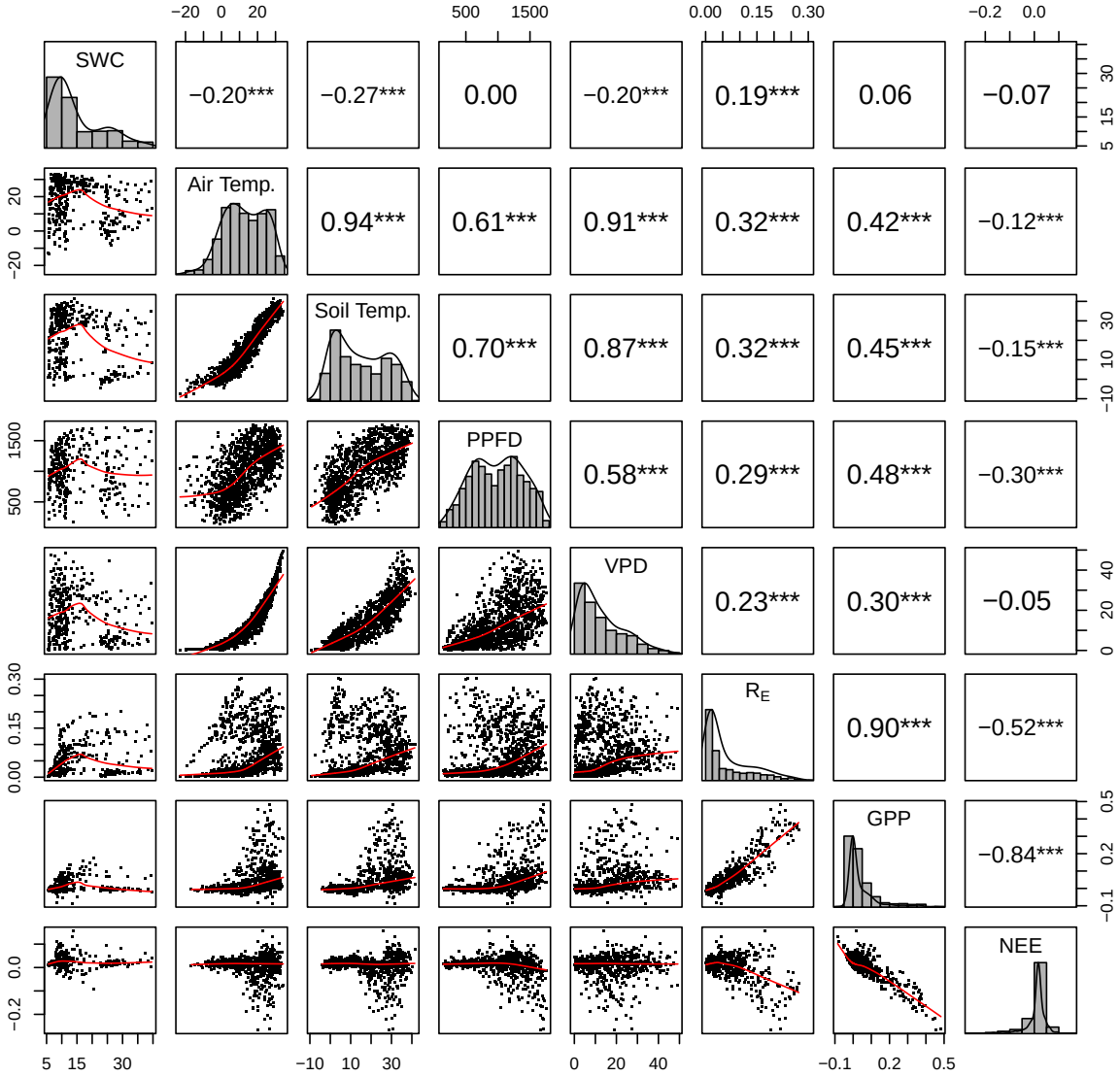


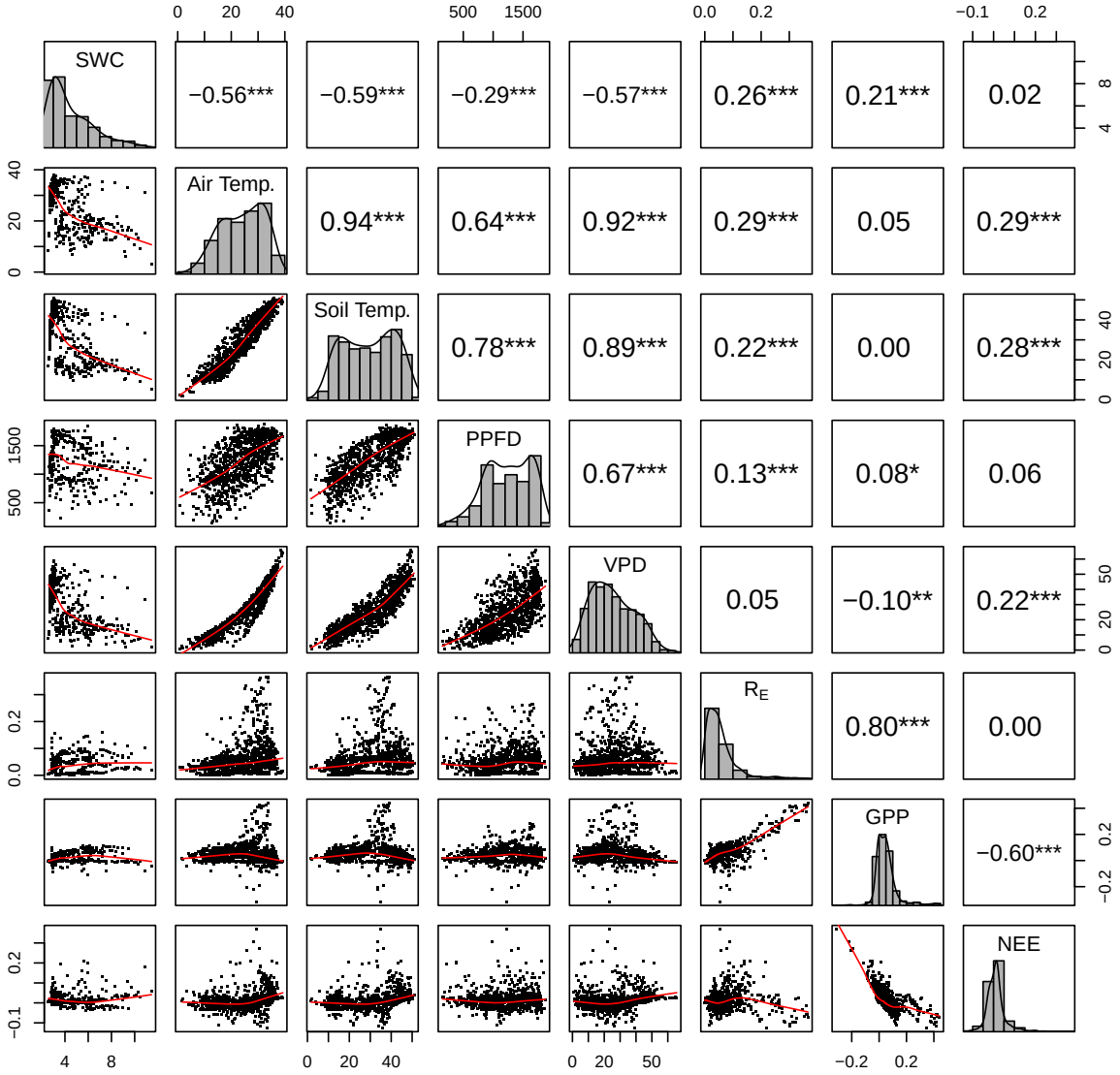
Scatter Plot Matrices (SPLOMS)

```
my_corr_plot_fn <- function(my_site) {
  (
    psych::pairs.panels(
        dta_1day %>%
        dplyr::filter(neon_site_name == my_site) %>%
        dplyr::select(
          swc, ta, ts, ppfd_in
          , vpd
          , re
          , gpp
          , nee
        )
      , pch = "."  # 21 for color by group in line above # "." for no color
      , labels = c(
          "SWC"
```

```
          , "Air Temp."
          , "Soil Temp."
          , "PPFD"
          , "VPD"
          , latex2exp::TeX("$R_E$")
          , "GPP"
          , "NEE"
      )
      , method = "pearson" # correlation method
      , hist.col = "gray70"
      , density = TRUE  # show density plots
      , ellipses = FALSE # show correlation ellipses
      , rug = FALSE
      , stars = TRUE
      , main = my_site
    )
  )
}

# sites
sts <- unique(dta_1day$neon_site_name)
for (i in 1:length(sts)) {
  my_corr_plot_fn(my_site = sts[i])
}
```


Central Plains Experimental Range
SWC
Air Temp.
Soil Temp.
PPFD
VPD
R_E
GPP
NEE
−0.20***
−0.27***
0.00
−0.20***
0.19***
0.06
−0.07
0.94***
0.61***
0.91***
0.32***
0.42***
−0.12***
0.70***
0.87***
0.32***
0.45***
−0.15***
0.58***
0.29***
0.48***
−0.30***
0.23***
0.30***
−0.05
0.90***
−0.52***
−0.84***


Santa Rita Experimental Range
SWC
Air Temp.
Soil Temp.
PPFD
VPD
R_E
GPP
NEE
−0.56***
−0.59***
−0.29***
−0.57***
0.26***
0.21***
0.02
0.94***
0.64***
0.92***
0.29***
0.05
0.29***
0.78***
0.89***
0.22***
0.00
0.28***
0.67***
0.13***
0.08*
0.06
0.05
−0.10**
0.22***
0.80***
0.00
−0.60***