ECOL 610: NEON Data - Summary Data

Group - Santa Rita Experimental Range (SRER)

01 October, 2022

Contents

Group Members

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

Setup

Load in the needed packages

```
library(tidyverse)
library(lubridate)
library(viridis)
library(RColorBrewer)
library(scales)
library(latex2exp)
library(kableExtra)
remove(list=ls())
# what is your site name?
site <- "Santa Rita Experimental Range"
```

Load Data

Information about these variables can be found in the metadata file

All daily data files can be found here.

```
# put all of the daily data you want to load in the directory up one level titled "data"
# 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_replace_all(pattern = "-", 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(1:2, 11:12) ~ "Winter"
          , month %in% c(3:5) ~ "Spring"
          , month %in% c(6:8) ~ "Summer"
          , month %in% c(9:10) ~ "Fall"
          , TRUE  ~ "Other")
  ) %>%
  dplyr::group_by(neon_site_name, week, year) %>%
  dplyr::mutate(is_full_week = sum(has_gpp)==7) %>%
  dplyr::ungroup() %>%
  dplyr::rename(ppfd = ppfd_in)
# count rows
dta_1day %>% dplyr::count(neon_site_name) %>%
  kableExtra::kable(
    caption = "Count of daily records by NEON Site"
    , col.names = c(
```

```
      "Site"
      , "N"
      )
  ) %>%
  kableExtra::kable_styling(font_size = 11) %>%
  kableExtra::column_spec(1, bold = TRUE, width = "18em") %>%
  kableExtra::kable_styling(latex_options = "HOLD_position")
```

Table 1: Count of daily records by NEON Site

Site	N
Central Plains Experimental Range	1276
Disney Wilderness	911
Harvard Forest	1460
Konza Prairie	1095
Niwot Ridge	364
Pu'u Maka'ala Natural Area Reserve	699
Santa Rita Experimental Range	1095
Toolik Field Station	1064
Wind River Experimental Forest	1095

Assignment

Part 1

One graph each with time on x-axis and each data type on the y-axis with fixed y-axis scale - turn in as a RMarkdown PDF

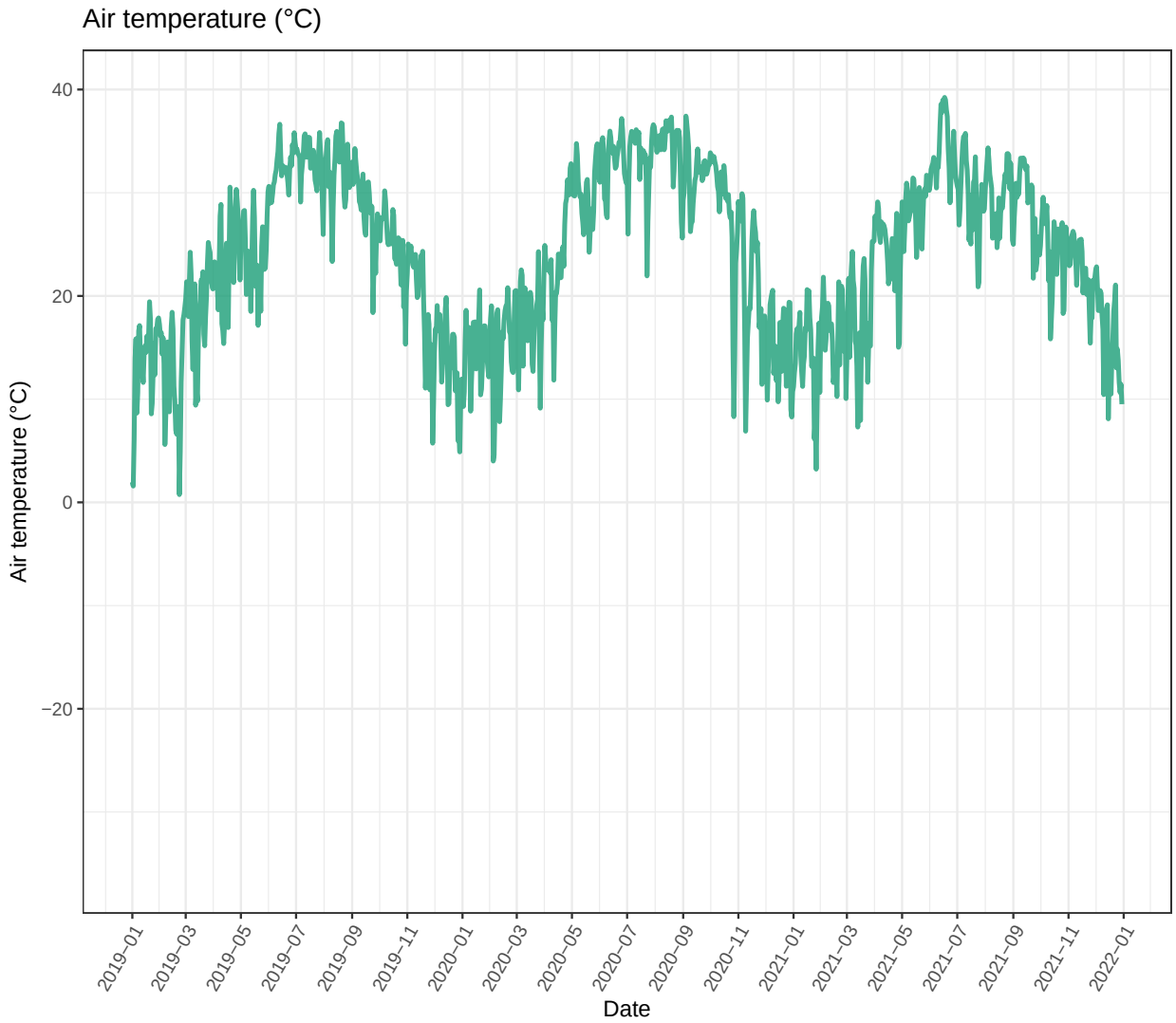
For each variable use the following y-scales (Max, Min):

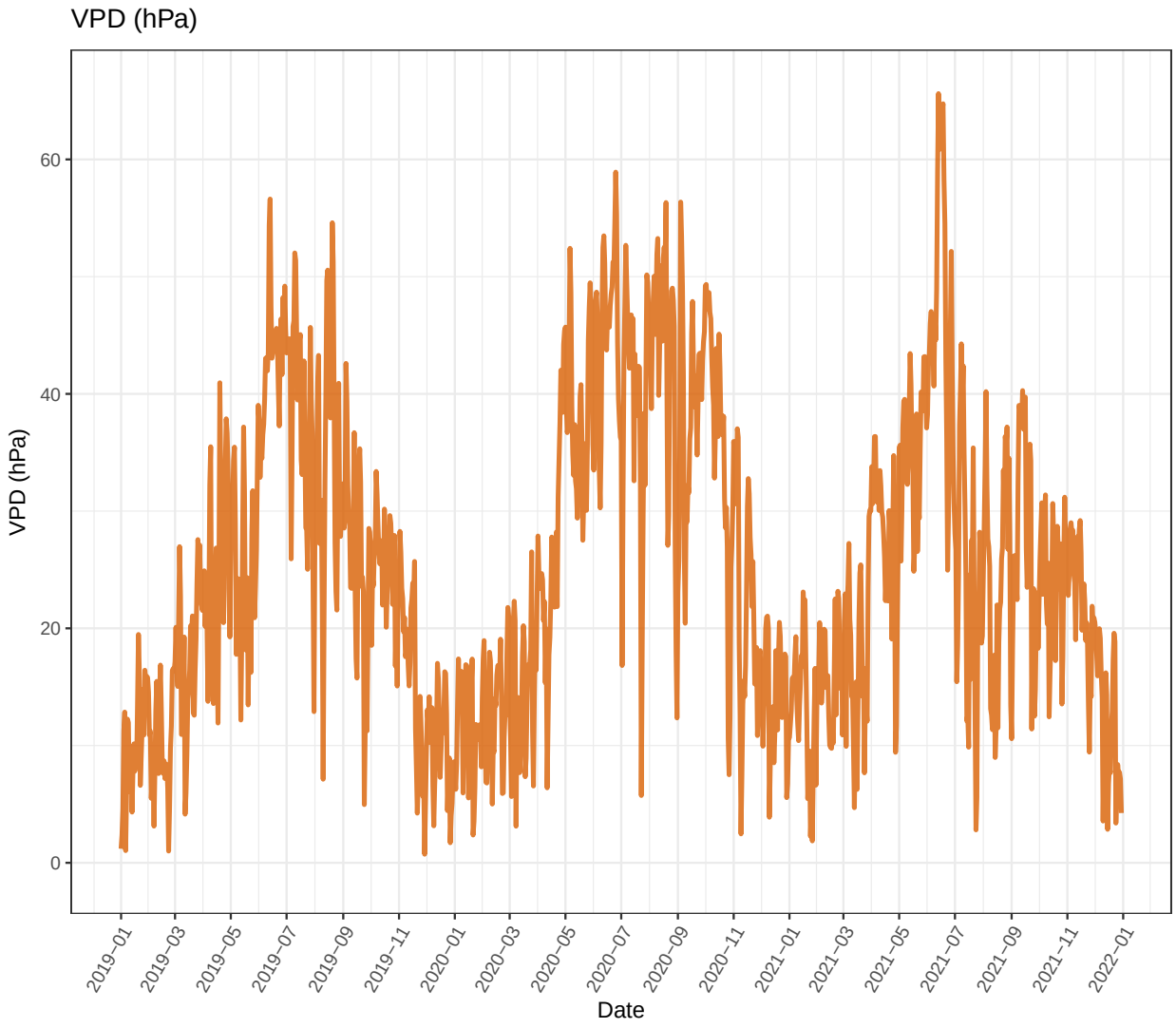
- TA (40, -36)
- VPD (66, -1)
- PPFD_IN (1898, 0.5)
- SWC (41, 2.6)
- TS (51, -11)
- NEE (1.1, -1)
- GPP (1.6, -0.5)
- RE (1.2, 0)

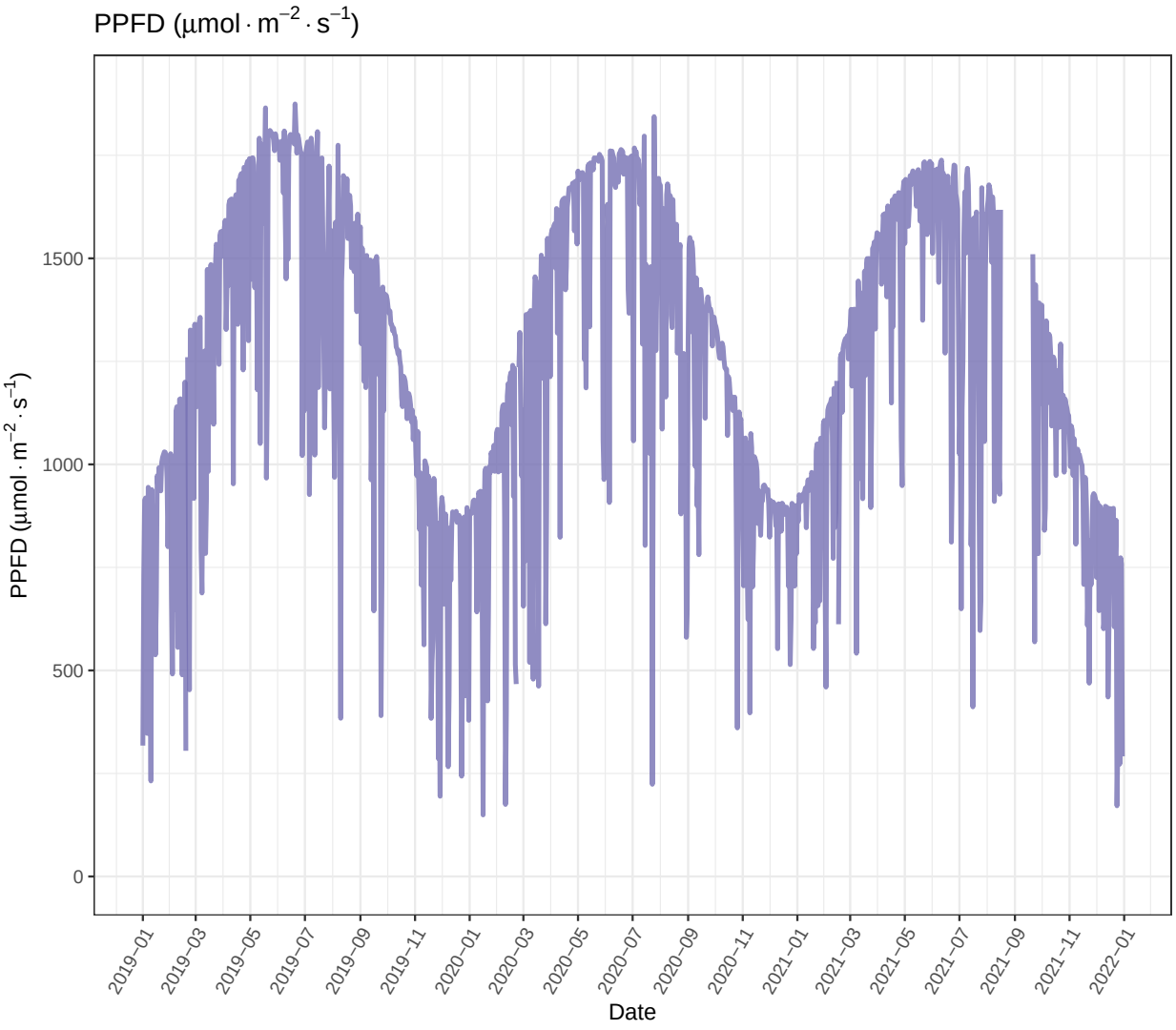
```
# set up vectors to use by function
  # vars
  vars <- c(
    "ta"
    , "vpd"
    , "ppfd"
    , "swc"
    , "ts"
    , "nee"
    , "gpp"
```

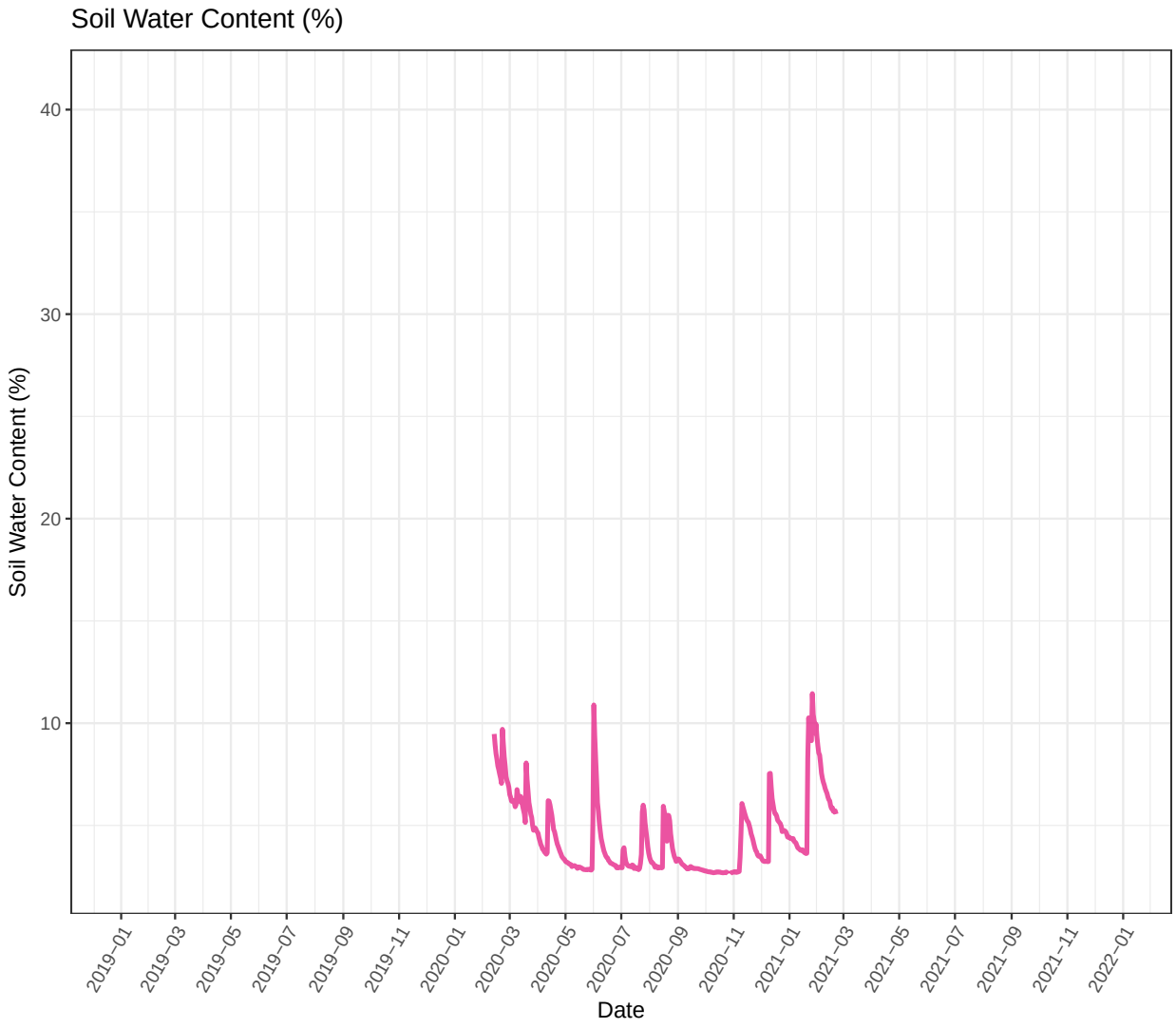
```
    , "re"
  )
  vars_title <- c(
    "Air temperature (\u00B0C)"
    , "VPD (hPa)"
    , latex2exp::TeX("PPFD ($\\mu mol  \\cdot  m^{-2}  \\cdot  s^{-1}$)")
    , "Soil Water Content (%)"
    , "Soil Temperature (\u00B0C)"
    , latex2exp::TeX("NEE $( mol \\; CO_{2} \\cdot m^{-2} \\cdot day^{-1})$")
    , latex2exp::TeX("GPP $( mol \\; CO_{2} \\cdot m^{-2} \\cdot day^{-1})$")
    , latex2exp::TeX("$R_E \\;( mol \\; CO_{2} \\cdot m^{-2} \\cdot day^{-1})$")
  )
  var_y_min <- c(
    -36
    , -1
    , 0.5
    , 2.6
    , -11
    , -1
    , -0.5
    , 0
  )
  var_y_max <- c(
    40
    , 66
    , 1898
    , 41
    , 51
    , 1.1
    , 1.6
    , 1.2
  )
my_site <- "Santa Rita Experimental Range"
# define plot function
  p_fn <- function(my_var) {
  #plot
  (
    dta_1day %>%
    dplyr::filter(
      neon_site_name == my_site
    ) %>%
    ggplot(., aes_string(x = "date_id", y = my_var)) +
      geom_line(
        lwd = 1.1
        , alpha = 0.8
        , color = RColorBrewer::brewer.pal(name = "Dark2", n = length(vars))[which(vars==my_var)]
      ) +
      xlab("Date") +
      ylab(vars_title[which(vars == my_var)]) +
      labs(
        title = vars_title[which(vars == my_var)]
      ) +
      scale_y_continuous(limits = c(var_y_min[which(vars == my_var)], var_y_max[which(vars == my_var)]))
```

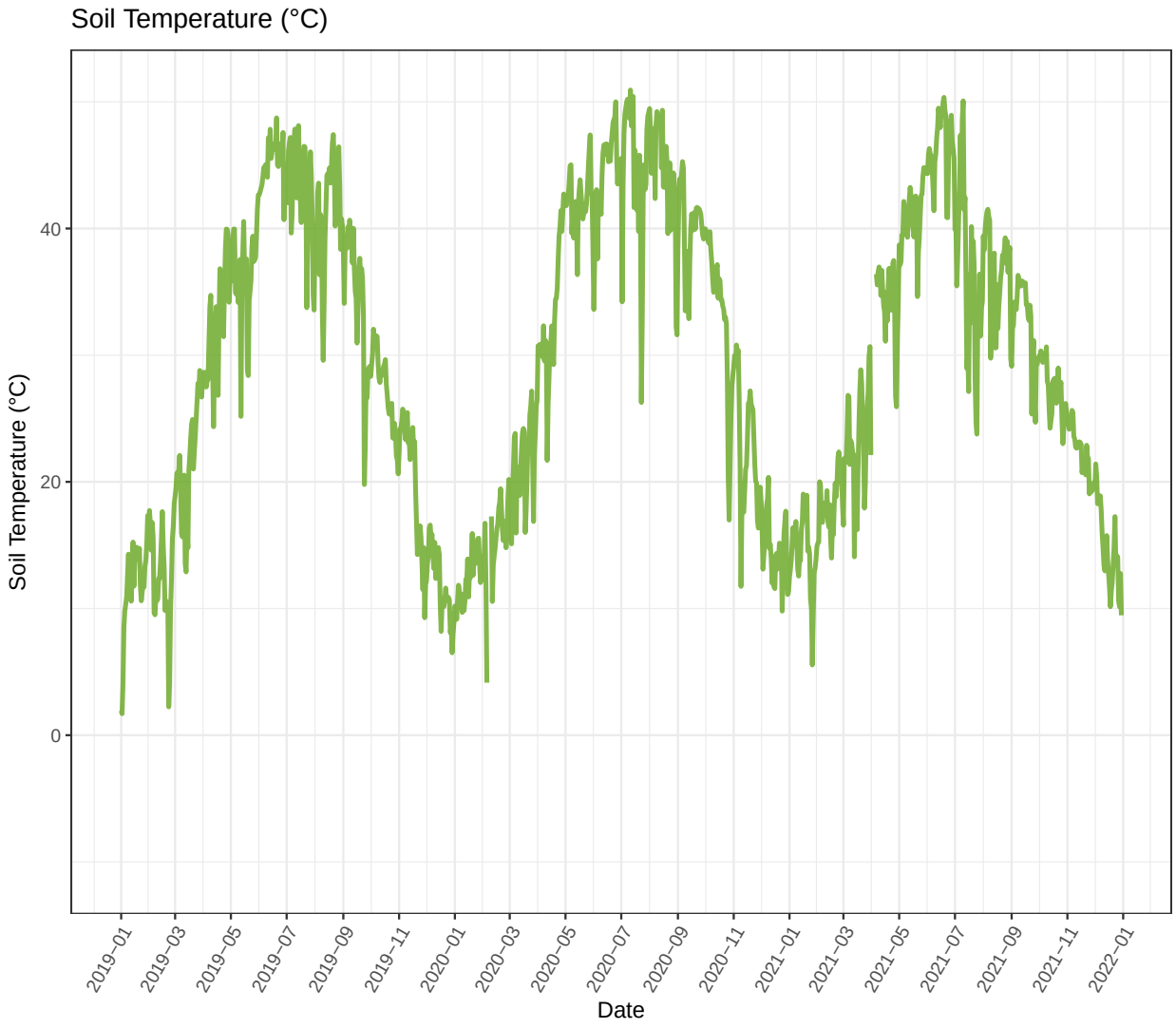
```
      scale_x_date(date_breaks = "2 month", date_labels = "%Y-%m") +
      theme_bw() +
      theme(
        legend.position = "none"
        , axis.text.x = element_text(angle = 60, hjust=1)
      )
  )

}
# call function
vars %>%
  purrr::map(p_fn)
```

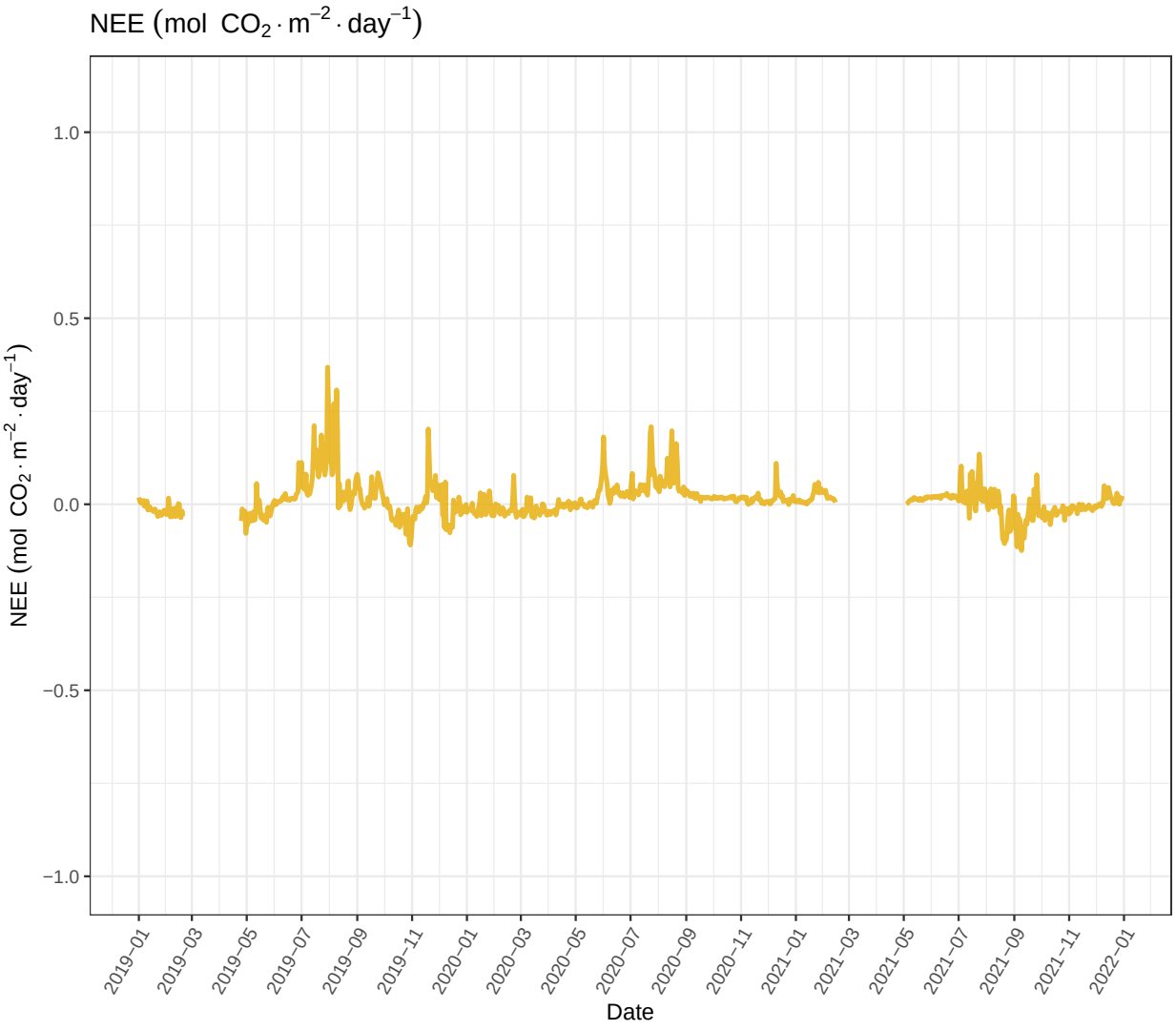


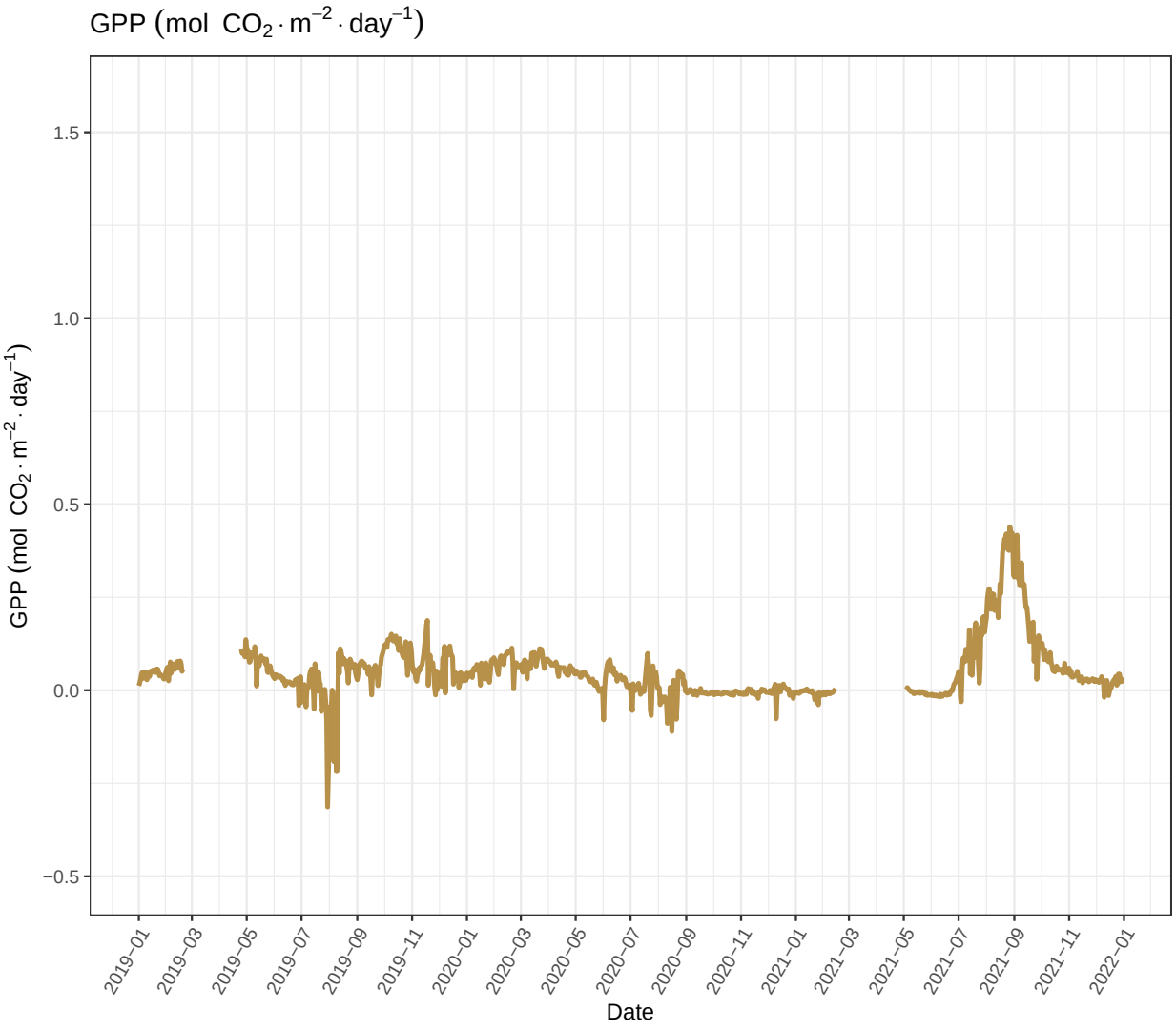

VPD (hPa)
60
40
20
0
VPD (hPa)
2019-01
2019-03
2019-05
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
Date

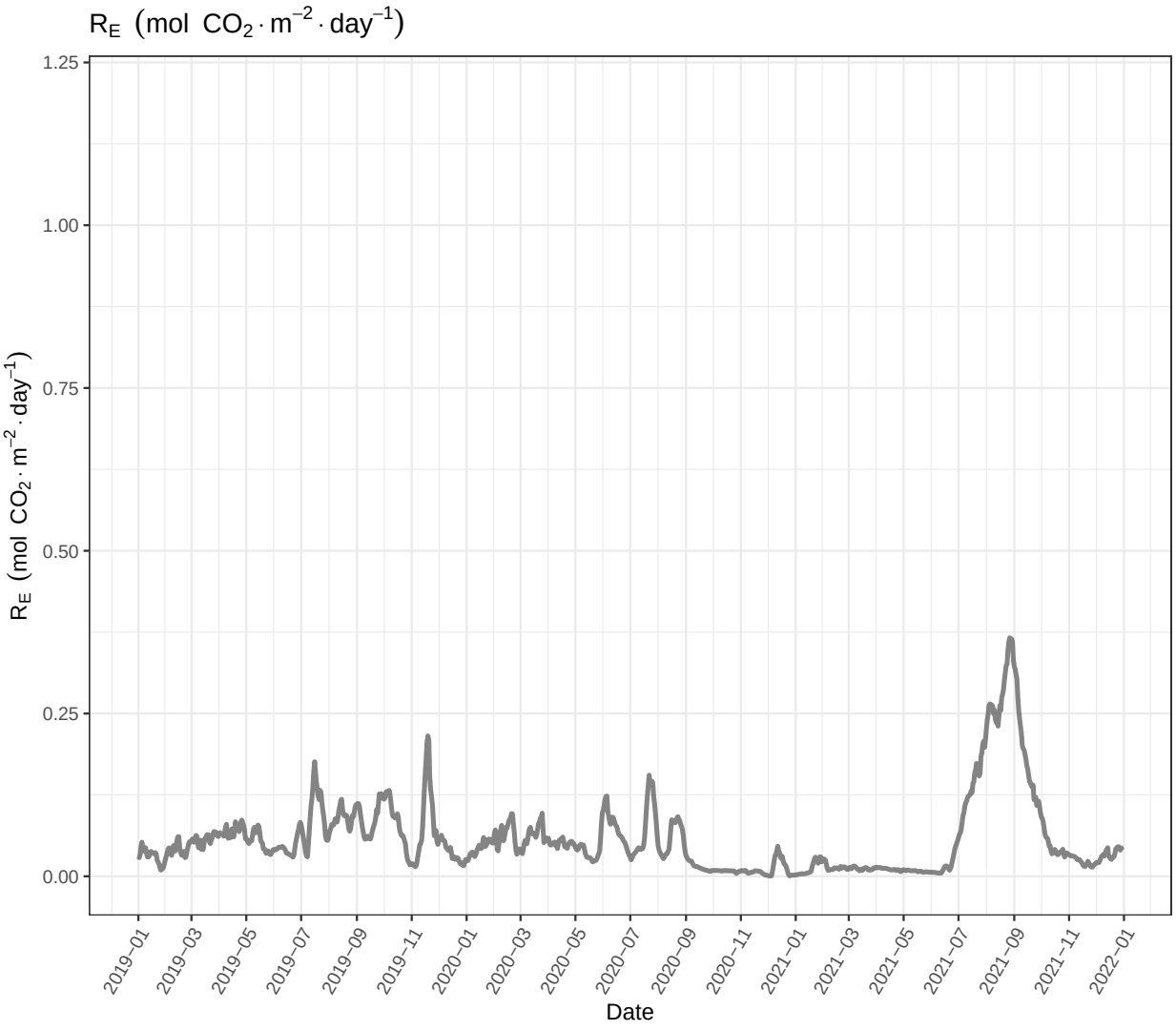

PPFD (µmol · m^{-2} · s^{-1})
1500
1000
500
0
2019–01
2019–03
2019–05
2019–07
2019–09
2019–11
2020–01
2020–03
2020–05
2020–07
2020–09
2020–11
2021–01
2021–03
2021–05
2021–07
2021–09
2021–11
2022–01
Date


Soil Water Content (%)
Soil Water Content (%)
40
30
20
10
2019-01
2019-03
2019-05
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
Date


Soil Temperature (°C)
Soil Temperature (°C)
40
20
0
2019-01
2019-03
2019-05
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
Date


NEE (mol $CO_2 \cdot m^{-2} \cdot day^{-1}$)
1.0
0.5
0.0
−0.5
−1.0
NEE (mol $CO_2 \cdot m^{-2} \cdot day^{-1}$)
2019−01
2019−03
2019−05
2019−07
2019−09
2019−11
2020−01
2020−03
2020−05
2020−07
2020−09
2020−11
2021−01
2021−03
2021−05
2021−07
2021−09
2021−11
2022−01
Date


GPP (mol $CO_2 \cdot m^{-2} \cdot day^{-1}$)
1.5
1.0
0.5
0.0
−0.5
GPP (mol $CO_2 \cdot m^{-2} \cdot day^{-1}$)
2019−01
2019−03
2019−05
2019−07
2019−09
2019−11
2020−01
2020−03
2020−05
2020−07
2020−09
2020−11
2021−01
2021−03
2021−05
2021−07
2021−09
2021−11
2022−01
Date



Part 2

The next item will be data you fill in on this templateDownload this template

Template key: sum = summer/ spring = spring/ fall = fall/ winter = winter; data types as above; metrics defined below

2. Some summary data broken out by time period -> average values for season across years (e.g., average across spring 2018, 2019, 2020)

For each Season and for each data type, provide the following metrics (code in template):

mean value for the season (mean) standard deviation (sd) 5th percentile value (5per) 95th percentile value (95per) number of days included/number of observations in your mean (N)

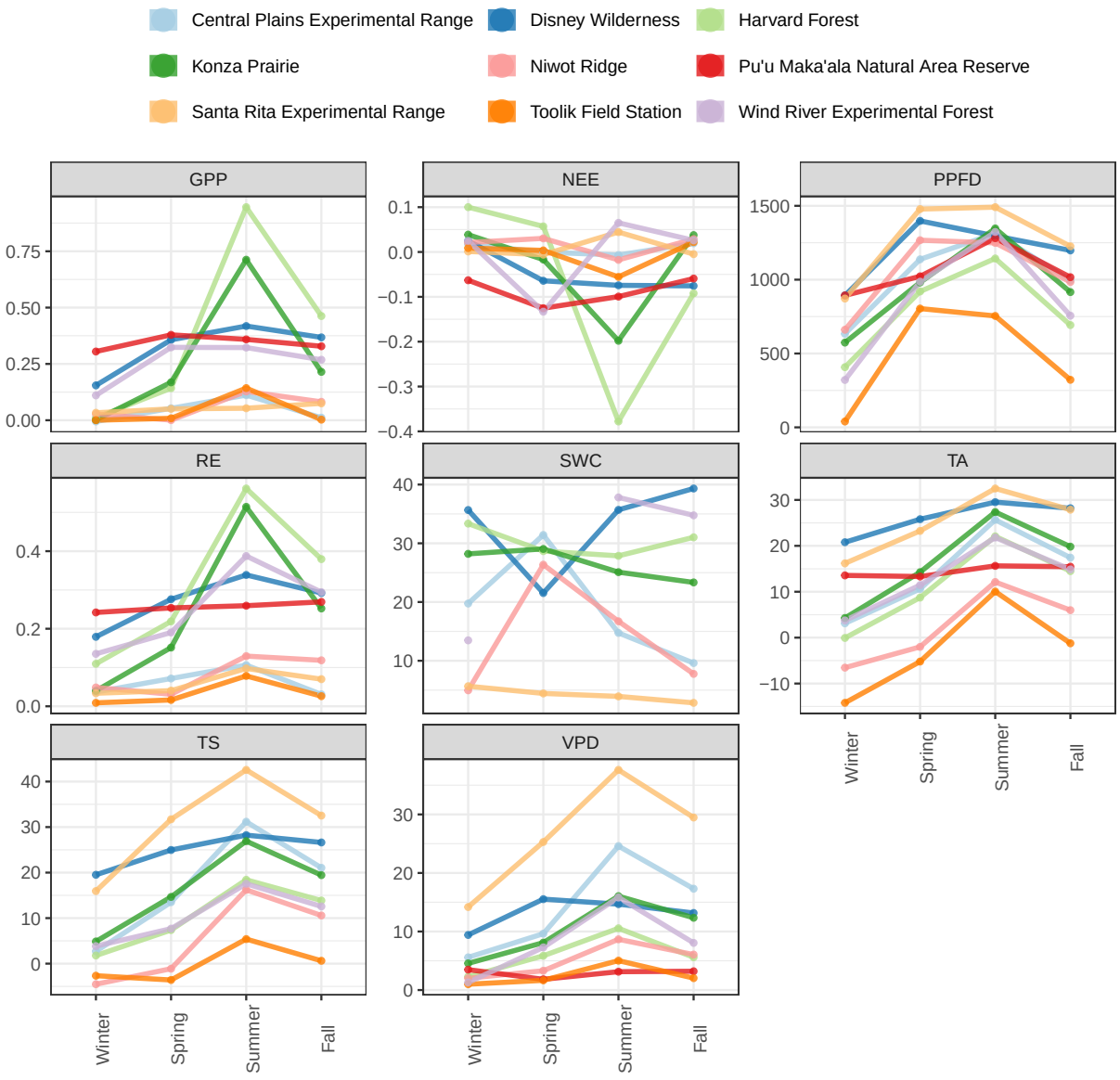
```
# list columns to summarize
summary_vars <- c(
  "gpp"
```

```
  , "nee"
  , "re"
  , "ta"
  , "vpd"
  , "ppfd"
  , "swc"
  , "ts"
)
# named list of functions
summary_fns <- list(
  mean = ~mean(.x, na.rm = TRUE)
  , sd = ~sd(.x, na.rm = TRUE)
  , quantile5 = ~quantile(.x, probs = 0.05, na.rm = TRUE)
  , quantile95 = ~quantile(.x, probs = 0.95, na.rm = TRUE)
  , N = ~sum(ifelse(is.na(.x), 0, 1))
)
# can't start a var name with a number so need rename fn
rn_fn <- function(x) paste0(gsub("quantile", "", x), "per")
# aggregate data
dta_summary_long <- dta_1day %>%
  dplyr::rename_at(summary_vars, toupper) %>%
  dplyr::group_by(neon_site_name, season) %>%
  dplyr::summarise(
    dplyr::across(
      toupper(summary_vars)
      , summary_fns
      , .names = "{.col}_{.fn}"
    )
  ) %>%
  dplyr::ungroup() %>%
  dplyr::rename_at(
    vars(tidyselect::contains("quantile"))
    , rn_fn
  )
# reshape long to wide
dta_summary_wide <-
  dta_summary_long %>%
  dplyr::mutate(season = tolower(season)) %>%
  tidyr::pivot_wider(
    names_from = season
    , values_from = -tidyselect::all_of(c("neon_site_name", "season"))
    , names_glue = "{.value}_{season}"
  )
# here's a long vector of variable names so that they are in the same order as requested...yay!
vars_order <- c("Site", "GPP_mean_sum", "GPP_mean_spring", "GPP_mean_fall", "GPP_mean_winter", "GPP_sd_

# export data to csv
dta_temp <- dta_summary_wide %>%
    # we can have the season names "autumn" and "winter" (both 6 letters)
      # but "summer" needs to be shortened to "sum" ... not to be confused with "sum" as in summation
    dplyr::rename_at(
      vars(tidyselect::contains("summer"))
      , function(x) gsub("summer", "sum", x)
```

```
    ) %>%
    dplyr::rename(
      Site = neon_site_name
    ) %>%
    dplyr::select(
      tidyselect::all_of(vars_order)
    )
# write to csv
write.csv(dta_temp, file = "../data/summary_data_ALL.csv", append = FALSE, row.names = FALSE)
write.csv(
  dta_temp %>% dplyr::filter(Site == "Santa Rita Experimental Range")
  , file = "../data/summary_data_SRER.csv"
  , append = FALSE
  , row.names = FALSE
)
remove(dta_temp)
```

Quick plot of season means

```
dta_summary_long %>%
  dplyr::select(tidyselect::contains("mean"), neon_site_name, season) %>%
  tidyr::pivot_longer(
    cols = -tidyselect::all_of(c("neon_site_name", "season"))
    , names_to = "var_name"
    , values_to = "var_value"
    , values_drop_na = FALSE
  ) %>%
  dplyr::mutate(
    var_name = gsub("_mean", "", var_name)
    , season = ordered(season, levels = c("Winter", "Spring", "Summer", "Fall"))
  ) %>%
ggplot(data = ., mapping = aes(x = season, y = var_value, group = neon_site_name, color = neon_site_nam
  geom_line(lwd = 1.2, alpha = 0.8) +
  geom_point(size = 1.2, alpha = 0.8) +
  facet_wrap(~var_name, scales = "free_y") +
  scale_color_brewer(type = "qual", palette = "Paired") +
  xlab("") +
  ylab("") +
  labs(
    caption = "*seasonal mean values shown"
  ) +
  theme_bw() +
  theme(
    legend.title = element_blank()
    , legend.position = "top"
    , axis.text.x = element_text(angle = 90)
  ) +
  guides(
    color = guide_legend(override.aes = list(size = 5), nrow = 3, byrow = TRUE)
  )
```



*seasonal mean values shown