Climate Water Loss Experiment - Weather Over Time

Savannah Weaver

2021

Contents

Packages

```
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse") # workflow and plots
if (!require("weathermetrics")) install.packages("weathermetrics")
library("weathermetrics") # F to C conversion
```

Background

Our analyses of baseline variation in CEWL and plasma osmolality yielded VERY different models for this current project/dataset compared to our last iteration of data collection. In this Rmd, I investigate the weather trends across both studies to assess whether variation, or lack of, in weather may have affected the very different results we got.

Load Data

```
weather_dat <- read.csv("./data/weather_long_term.csv", sep = ';') %>%
  mutate(temperature_F = as.numeric(temperature_F),
         wind_speed_mph = as.numeric(wind_speed_mph),
         RH_percent = as.numeric(RH_percent),
         solar_radiation_W_m2 = as.numeric(solar_radiation_W_m2),
```

```
        date_time = as.POSIXct(paste(date, time),
                               format = "%m/%d/%y %I:%M %p"),
        date = as.Date(date, format = "%m/%d/%y"),
        temp_C = fahrenheit.to.celsius(temperature_F, round = 2),
        temp_K = temp_C + 273.15,
        e_s_kPa = 0.611*exp((2500000/461.5)*
                              ((1/273)-(1/temp_K))),
        e_a_kPa = e_s_kPa * (RH_percent/100),
        VPD_kPa = e_s_kPa - e_a_kPa
        ) %>%
  dplyr::filter(complete.cases(temp_C)) %>%
  dplyr::filter(date_time >= "2021-04-01 00:00:00" &
                  date_time <= "2021-09-01 00:00:00")
```

```
## Warning: Problem with `mutate()` input `temperature_F`.
## i NAs introduced by coercion
## i Input `temperature_F` is `as.numeric(temperature_F)`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
## Warning: Problem with `mutate()` input `wind_speed_mph`.
## i NAs introduced by coercion
## i Input `wind_speed_mph` is `as.numeric(wind_speed_mph)`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
## Warning: Problem with `mutate()` input `RH_percent`.
## i NAs introduced by coercion
## i Input `RH_percent` is `as.numeric(RH_percent)`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
## Warning: Problem with `mutate()` input `solar_radiation_W_m2`.
## i NAs introduced by coercion
## i Input `solar_radiation_W_m2` is `as.numeric(solar_radiation_W_m2)`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
summary(weather_dat)
```

```
##       date                time           temperature_F    wind_speed_mph
##  Min.   :2021-04-01   Length:14689       Min.   : 38.20   Min.   : 0.100
##  1st Qu.:2021-05-09   Class :character   1st Qu.: 54.00   1st Qu.: 0.100
##  Median :2021-06-16   Mode  :character   Median : 58.60   Median : 2.000
##  Mean   :2021-06-16                      Mean   : 60.65   Mean   : 2.304
##  3rd Qu.:2021-07-24                      3rd Qu.: 66.90   3rd Qu.: 4.200
##  Max.   :2021-09-01                      Max.   :101.00   Max.   :16.700
##    RH_percent     solar_radiation_W_m2 precip_inches
##  Min.   : 14.70   Min.   :   0.0       Length:14689
##  1st Qu.: 67.90   1st Qu.:   0.0       Class :character
##  Median : 90.20   Median :  37.2       Mode  :character
##  Mean   : 82.91   Mean   : 296.7
##  3rd Qu.:100.00   3rd Qu.: 637.9
##  Max.   :100.00   Max.   :1156.4
##    date_time                      temp_C          temp_K         e_s_kPa
##  Min.   :2021-04-01 00:00:00   Min.   : 3.44   Min.   :276.6   Min.   :0.7905
##  1st Qu.:2021-05-09 06:00:00   1st Qu.:12.22   1st Qu.:285.4   1st Qu.:1.4441
##  Median :2021-06-16 12:00:00   Median :14.78   Median :287.9   Median :1.7096
```

```
##  Mean   :2021-06-16 12:00:00   Mean   :15.92   Mean   :289.1   Mean   :1.9401
##  3rd Qu.:2021-07-24 18:00:00   3rd Qu.:19.39   3rd Qu.:292.5   3rd Qu.:2.2996
##  Max.   :2021-09-01 00:00:00   Max.   :38.33   Max.   :311.5   Max.   :7.0903
##     e_a_kPa          VPD_kPa
##  Min.   :0.6525   Min.   :0.0000
##  1st Qu.:1.2723   1st Qu.:0.0000
##  Median :1.5191   Median :0.1551
##  Mean   :1.4972   Mean   :0.4429
##  3rd Qu.:1.7219   3rd Qu.:0.7345
##  Max.   :2.1944   Max.   :5.8841
```

get average daily values:

```
weather_dat_daily <- weather_dat %>%
  group_by(date) %>%
  summarise(temp_C = mean(temp_C),
            VPD_kPa = mean(VPD_kPa),
            RH_percent = mean(RH_percent),
            wind_speed_mph = mean(wind_speed_mph),
            solar_radiation_W_m2 = mean(solar_radiation_W_m2)
            )
```

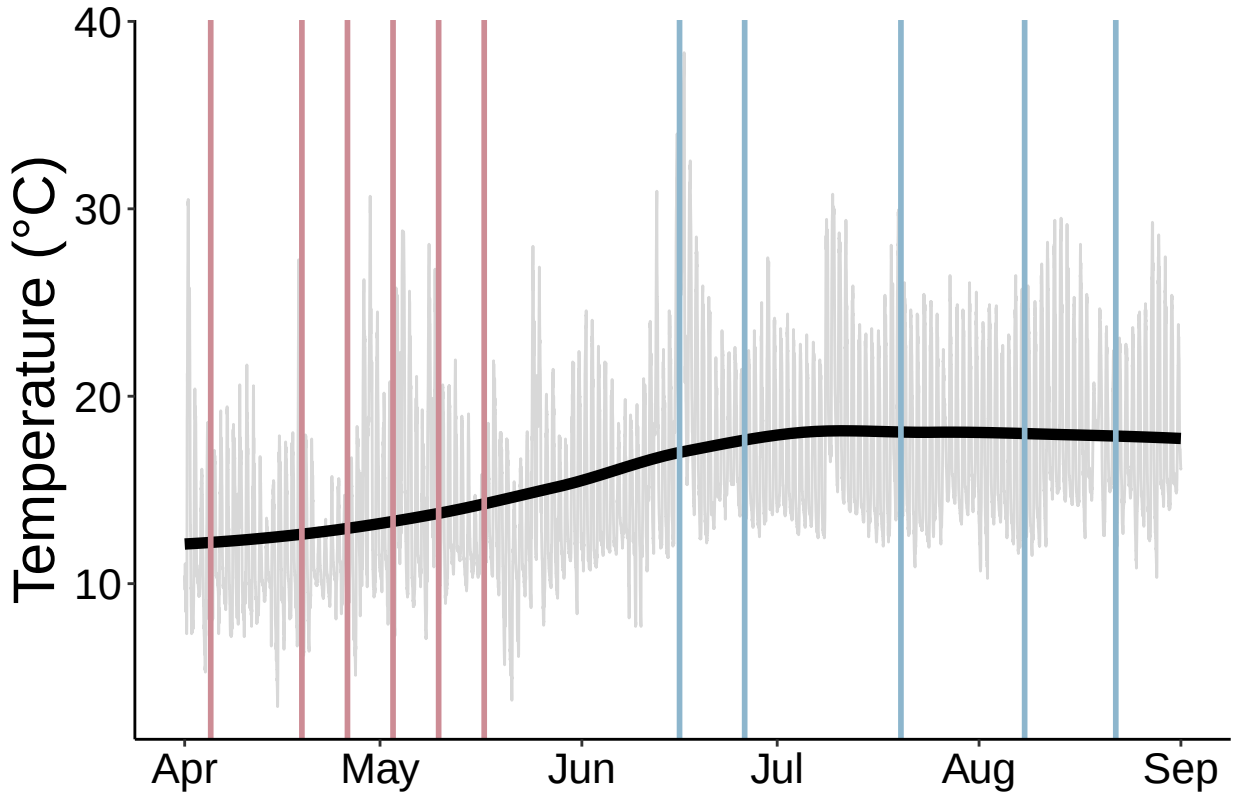
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

Visualize

Temperature

```
ggplot(data = weather_dat,
       aes(x = date_time,
                 y = temp_C)) +
  # all data
  geom_line(alpha = 0.6,
            color = "gray") +
  # trend over time
  stat_smooth(formula = y ~ x,
              method = "loess",
              color = "black",
              se = F,
              size = 2,
              alpha = 1 ) +
  # sampling dates for BOTH studies
  geom_vline(xintercept = as.POSIXct("2021-04-05 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-04-19 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-04-26 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-03 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-10 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-17 00:00:00"), size = 1,
             color = "lightpink3") +
```

```
  geom_vline(xintercept = as.POSIXct("2021-06-16 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-06-26 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-07-20 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-08 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-22 00:00:00"), size = 1,
             color = "lightskyblue3") +
  # rest is formatting
  theme_classic() +
  xlab("") +
  ylab("Temperature (°C)") +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 14),
        legend.text.align = 0,
        legend.position = "right"
        ) -> temp_fig
temp_fig
```

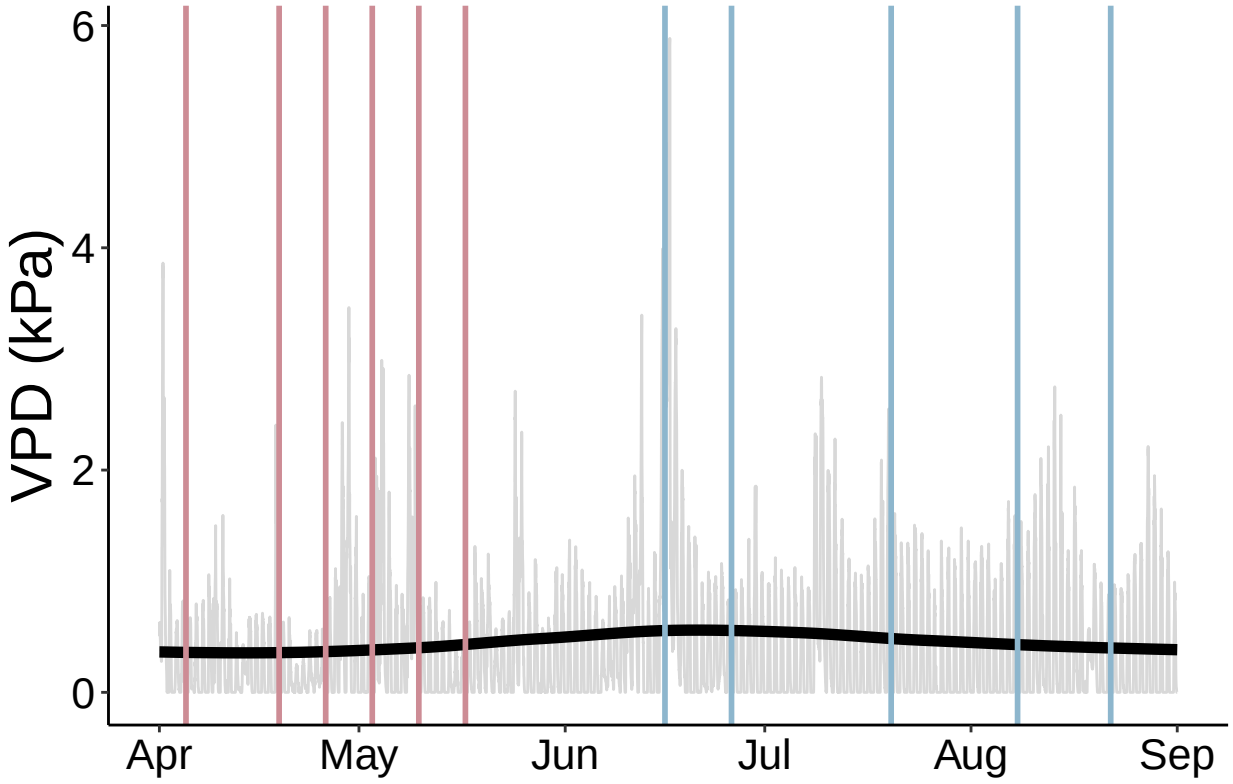


```
# export figure
ggsave(filename = "weather_long_trends_temp.jpeg",
       plot = temp_fig,
       path = "./results_figures",
       device = "jpeg",
       dpi = 1200,
       width = 6, height = 4)
```

VPD

```
ggplot(data = weather_dat,
       aes(x = date_time,
                y = VPD_kPa)) +
  # all data
  geom_line(alpha = 0.6,
            color = "gray") +
  # trend over time
  stat_smooth(formula = y ~ x,
              method = "loess",
              color = "black",
              se = F,
              size = 2,
              alpha = 1 ) +
  # sampling dates for BOTH studies
  geom_vline(xintercept = as.POSIXct("2021-04-05 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-04-19 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-04-26 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-03 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-10 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-17 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-06-16 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-06-26 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-07-20 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-08 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-22 00:00:00"), size = 1,
             color = "lightskyblue3") +
  # rest is formatting
  theme_classic() +
  xlab("") +
  ylab("VPD (kPa)") +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
```

```
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 14),
        legend.text.align = 0,
        legend.position = "right"
        ) -> VPD_fig
VPD_fig
```

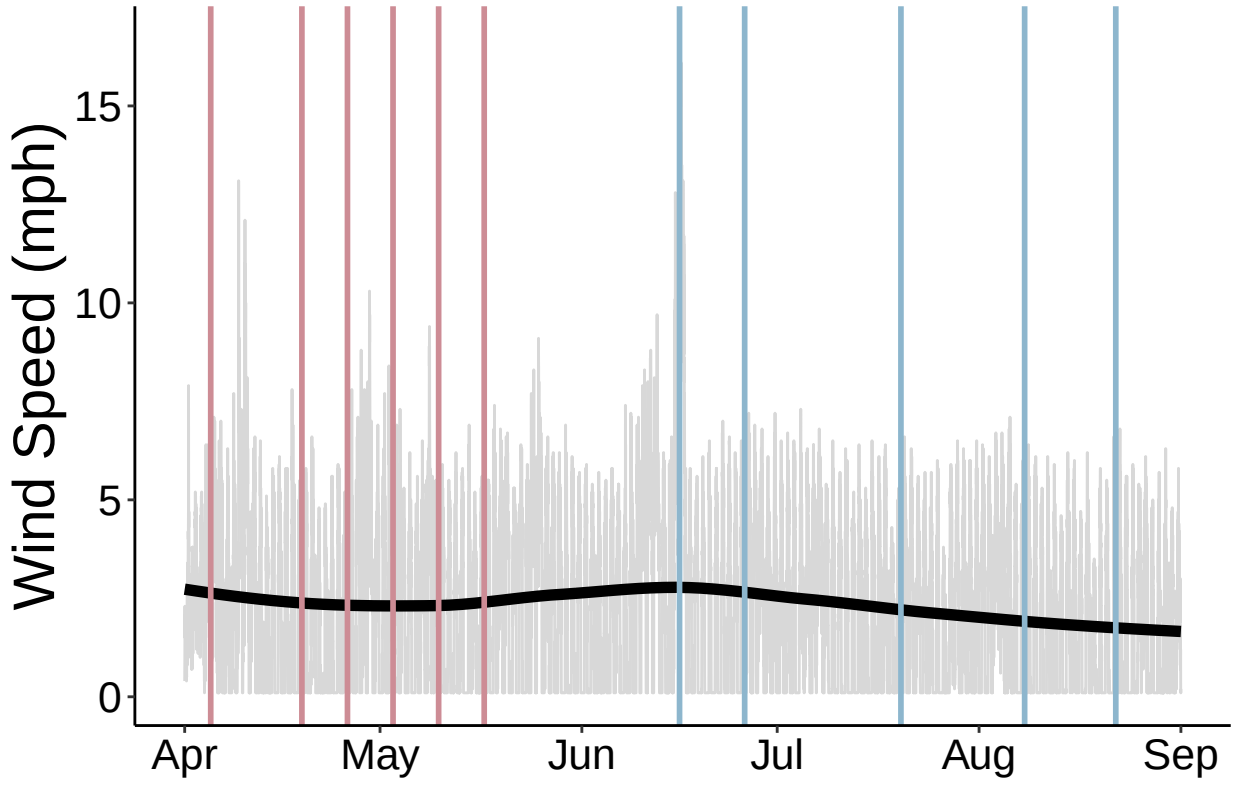



```
# export figure
ggsave(filename = "weather_long_trends_VPD.jpeg",
       plot = VPD_fig,
       path = "./results_figures",
       device = "jpeg",
       dpi = 1200,
       width = 6, height = 4)
```

Wind Speed

```
ggplot(data = weather_dat,
       aes(x = date_time,
                 y = wind_speed_mph)) +
  # all data
  geom_line(alpha = 0.6,
            color = "gray") +
  # trend over time
```

```
  stat_smooth(formula = y ~ x,
              method = "loess",
              color = "black",
              se = F,
              size = 2,
              alpha = 1 ) +
  # sampling dates for BOTH studies
  geom_vline(xintercept = as.POSIXct("2021-04-05 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-04-19 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-04-26 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-03 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-10 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-17 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-06-16 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-06-26 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-07-20 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-08 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-22 00:00:00"), size = 1,
             color = "lightskyblue3") +
  # rest is formatting
  theme_classic() +
  xlab("") +
  ylab("Wind Speed (mph)") +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 14),
        legend.text.align = 0,
        legend.position = "right"
        ) -> wind_fig
wind_fig
```

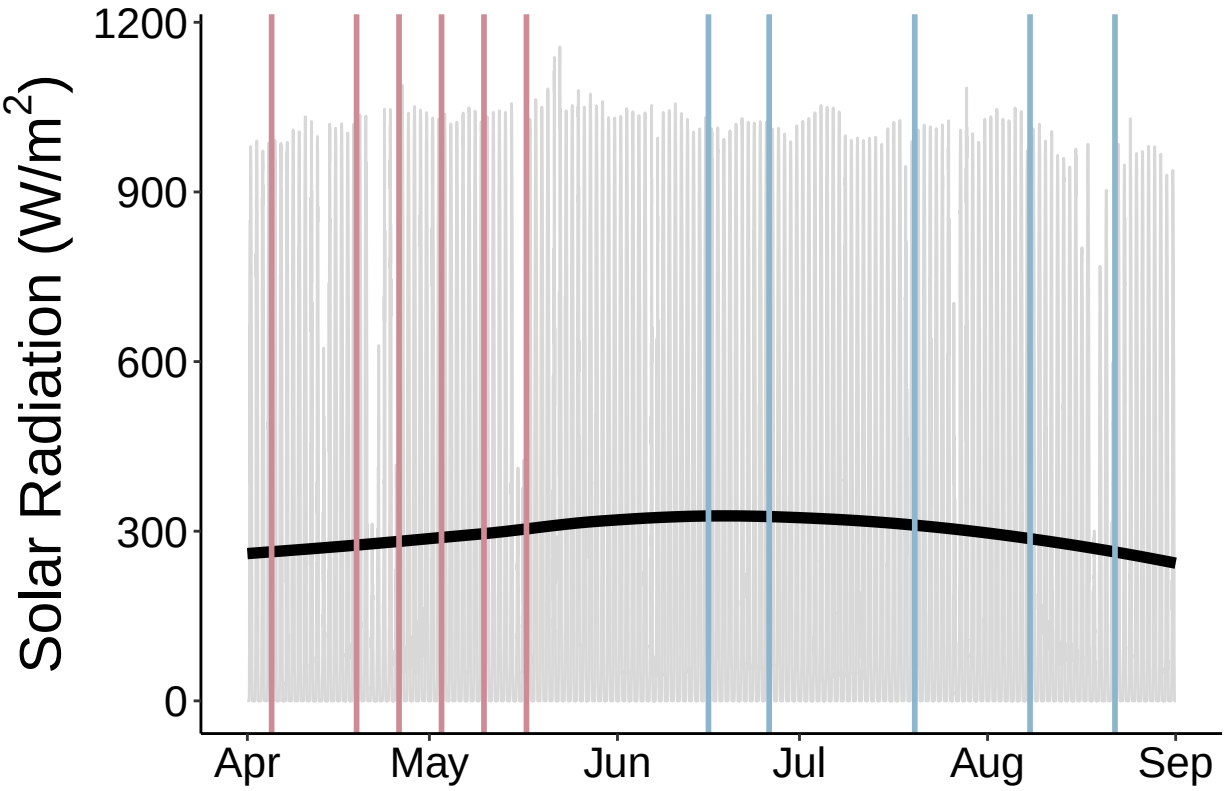



```
# export figure
ggsave(filename = "weather_long_trends_wind.jpeg",
       plot = wind_fig,
       path = "./results_figures",
       device = "jpeg",
       dpi = 1200,
       width = 6, height = 4)
```

Solar Radiation

```
ggplot(data = weather_dat,
       aes(x = date_time,
                 y = solar_radiation_W_m2)) +
  # all data
  geom_line(alpha = 0.6,
            color = "gray") +
  # trend over time
  stat_smooth(formula = y ~ x,
              method = "loess",
              color = "black",
              se = F,
              size = 2,
              alpha = 1 ) +
  # sampling dates for BOTH studies
  geom_vline(xintercept = as.POSIXct("2021-04-05 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-04-19 00:00:00"), size = 1,
             color = "lightpink3") +
```

```
  geom_vline(xintercept = as.POSIXct("2021-04-26 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-03 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-10 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-05-17 00:00:00"), size = 1,
             color = "lightpink3") +
  geom_vline(xintercept = as.POSIXct("2021-06-16 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-06-26 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-07-20 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-08 00:00:00"), size = 1,
             color = "lightskyblue3") +
  geom_vline(xintercept = as.POSIXct("2021-08-22 00:00:00"), size = 1,
             color = "lightskyblue3") +
  # rest is formatting
  theme_classic() +
  xlab("") +
  ylab(bquote('Solar Radiation (W/'*m^2*')')) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 14),
        legend.text.align = 0,
        legend.position = "right"
        ) -> sorad_fig
sorad_fig
```




```
# export figure
ggsave(filename = "weather_long_trends_sorad.jpeg",
       plot = sorad_fig,
       path = "./results_figures",
       device = "jpeg",
       dpi = 1200,
       width = 6, height = 4)
```

Stats

Variability

```
variation <- weather_dat %>%
  mutate(study = as.factor(case_when(date_time >= "2021-04-05 00:00:00" &
                                       date_time <= "2021-05-18 00:00:00" ~
                                       "spring",
                                     date_time >= "2021-06-16 00:00:00" &
                                       date_time <= "2021-08-23 00:00:00" ~
                                       "summer"
                                       ))) %>%
  dplyr::filter(complete.cases(study)) %>%
  group_by(study) %>%
  summarise(temp_CV = (sd(temp_C)/mean(temp_C)) *100,
            temp_range = max(temp_C) - min(temp_C),
            VPD_CV = (sd(VPD_kPa)/mean(VPD_kPa)) *100,
            VPD_range = max(VPD_kPa) - min(VPD_kPa),
            wind_CV = (sd(wind_speed_mph)/mean(wind_speed_mph)) *100,
            wind_range = max(wind_speed_mph) - min(wind_speed_mph),
```

```
            sorad_CV = (sd(solar_radiation_W_m2)/mean(solar_radiation_W_m2)) *100,
            sorad_range = max(solar_radiation_W_m2) - min(solar_radiation_W_m2)
            )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
variation
```

```
## # A tibble: 2 x 9
##   study  temp_CV temp_range VPD_CV VPD_range wind_CV wind_range sorad_CV
##   <fct>    <dbl>      <dbl>  <dbl>     <dbl>   <dbl>      <dbl>    <dbl>
## 1 spring    34.6       27.2   144.      3.46    98.3       13       133.
## 2 summer    26.5       28.0   140.      5.88   104.        16.1     125.
## # ... with 1 more variable: sorad_range <dbl>
```

Weather variability (CV): temp: spring »> summer VPD: spring > summer wind: spring < summer sorad: spring »> summer

Range differences are pretty negligible.

SLR ~ Date

Try a simple linear model of weather ~ date:

```
summary(lm(data = weather_dat_daily, temp_C ~ date))
```

```
##
## Call:
## lm(formula = temp_C ~ date, data = weather_dat_daily)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6950 -1.3703 -0.4465  0.6827 12.8540
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.355e+02  7.521e+01  -11.11   <2e-16 ***
## date         4.530e-02  4.002e-03   11.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.208 on 152 degrees of freedom
## Multiple R-squared:  0.4575, Adjusted R-squared:  0.4539
## F-statistic: 128.2 on 1 and 152 DF,  p-value: < 2.2e-16
```

```
summary(lm(data = weather_dat_daily, VPD_kPa ~ date))
```

```
##
## Call:
## lm(formula = VPD_kPa ~ date, data = weather_dat_daily)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46985 -0.17400 -0.10019  0.03455  2.78020
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -6.8706941 12.1015078  -0.568    0.571
## date         0.0003890  0.0006439   0.604    0.547
##
## Residual standard error: 0.3552 on 152 degrees of freedom
## Multiple R-squared:  0.002395,   Adjusted R-squared:  -0.004168
## F-statistic: 0.365 on 1 and 152 DF,  p-value: 0.5467
```

```
summary(lm(data = weather_dat_daily, wind_speed_mph ~ date))
```

```
##
## Call:
## lm(formula = wind_speed_mph ~ date, data = weather_dat_daily)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7055 -0.4760 -0.2113  0.2062  7.8177
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 96.794848  34.660874   2.793  0.00590 **
## date        -0.005028   0.001844  -2.727  0.00715 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.017 on 152 degrees of freedom
## Multiple R-squared:  0.04663,    Adjusted R-squared:  0.04036
## F-statistic: 7.434 on 1 and 152 DF,  p-value: 0.007152
```

```
summary(lm(data = weather_dat_daily, solar_radiation_W_m2 ~ date))
```

```
##
## Call:
## lm(formula = solar_radiation_W_m2 ~ date, data = weather_dat_daily)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -290.242   -5.523   18.381   37.258   63.117
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1384.13275 2182.97056   0.634    0.527
## date          -0.05796    0.11615  -0.499    0.618
##
## Residual standard error: 64.08 on 152 degrees of freedom
## Multiple R-squared:  0.001636,   Adjusted R-squared:  -0.004933
## F-statistic: 0.249 on 1 and 152 DF,  p-value: 0.6185
```

Of the 4 weather variables, mean daily values correlated with date for only temperature (estimate = 0.05, SE = 0.004, df = 152, $p < 0.0001$) and wind speed (estimate = -0.005, SE = 0.002, df = 152, p = 0.007).

Daily Humidity

```
mean_humid_VPD <- weather_dat %>%
  # filter to be only for study 2
  dplyr::filter(date > "2021-06-15") %>%
```

```
  # get daytime values only, 6 am to 8 pm
  dplyr::filter(as.numeric(substr(date_time, 12, 13)) < 20 &
                  as.numeric(substr(date_time, 12, 13)) > 6) %>%
  # get by date first
  group_by(date) %>%
  summarise(VPD_kPa = mean(VPD_kPa),
            RH_percent = mean(RH_percent)) %>%
  # mean across dates
  summarise(mean_VPD = mean(VPD_kPa),
            sd_VPD = sd(VPD_kPa),
            mean_RH = mean(RH_percent),
            sd_RH = sd(RH_percent))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
mean_humid_VPD
```

```
## # A tibble: 1 x 4
##   mean_VPD sd_VPD mean_RH sd_RH
##      <dbl>  <dbl>   <dbl> <dbl>
## 1    0.824  0.504    73.1  9.28
```

Mean daily relative humidity during the CWL 2021 study was 73±9%!