ThermalBuffer_Script

#Hourly Data ###Data Wrangling

```
#Read in data
quarterHour_condensed <- readRDS('quarterHour_condensed.RDS')
quarterHour_condensed <- as.data.frame(t(quarterHour_condensed))
quarterHour_condensed <- quarterHour_condensed %>%
mutate_all(~ifelse(is.nan(.), NA, .))

#Create 15 min sequence
quarterHour <- seq(mdy_h('7/1/2020 00'),mdy_h('7/13/2021 00'),by = "15 mins")
quarterHour <- as.data.frame(quarterHour)

#Attach to data frame
quarterHour_condensed <- cbind(quarterHour_condensed,quarterHour)

#Format data long
quarterHour_long <- gather(quarterHour_condensed, site, value, -quarterHour)
```

#Seasonal buffering ##Summer ###Data wrangling

```
#Read in data
daily_maxs_condensed <- readRDS('daily_maxs_condensed.RDS')
daily_maxs_condensed <- as.data.frame(t(daily_maxs_condensed))
daily_maxs_condensed <- daily_maxs_condensed %>%
mutate_all(~ifelse(is.nan(.), NA, .))

#Create "day" sequence
day <- seq(mdy('7/1/2020'),mdy('7/13/2021'),by = "day")
day <- as.data.frame(day) #convert "day" to data frame

#Attach day to data frame
daily_maxs_condensed <- cbind(daily_maxs_condensed,day)

#Format data long
daily_max_long <- gather(daily_maxs_condensed, site, value, -day)
data_summer <- daily_max_long %>%filter(day>'2020-07-15'& day < '2020-09-15')
```

###Seiad Creek

```
#Boxplot
b1 <- ggplot(subset(data_summer, site %in%
c("AP","SP","Durazo","LS","May","SC")))+
  geom_boxplot(aes(x = site, value, col = day))+
  labs(title = "(a) Seiad Creek, 15 July - 15 September")+
  theme_classic()+
  theme(text=element_text(size=16), axis.title.x=element_blank())+
```

```r
  scale_x_discrete(name ="Site",
                   limits = c("AP","SP","Durazo","LS","May","SC"),
                    labels=c("AP" = "Alexander","SP"= "Stender","Durazo"=
                               "Durazo","LS" = "LowerSeiad","May"="May","SC"
= "SeiadCreek"))+
  scale_y_continuous(name="Daily Maximum Temp (C)",limits=c(12.5, 22.5))


#ANOVA
data_summer_SC <- subset(data_summer, site %in%
c("AP","SP","Durazo","LS","May","SC"))
shapiro.test(data_summer_SC$value) #check for normality

##
##  Shapiro-Wilk normality test
##
## data:  data_summer_SC$value
## W = 0.97156, p-value = 1.378e-06

leveneTest(value ~ site, data = data_summer_SC) #check for equal variances

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   5  12.513 3.266e-11 ***
##       360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova_SC <- aov(value ~ site, data = data_summer_SC) #run one way anova
summary.aov(anova_SC)

##              Df Sum Sq Mean Sq F value Pr(>F)
## site          5 1653.6   330.7   146.6 <2e-16 ***
## Residuals   360  812.4     2.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(anova_SC) #posthoc analysis

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ site, data = data_summer_SC)
##
## $site
##                 diff        lwr        upr     p adj
## Durazo-AP  -4.0572131 -4.8365556 -3.2778706 0.0000000
## LS-AP      -1.4045902 -2.1839326 -0.6252477 0.0000059
```

```
## May-AP     -6.3854098 -7.1647523 -5.6060674 0.0000000
## SC-AP      -1.0391803 -1.8185228 -0.2598379 0.0021580
## SP-AP      -2.7762295 -3.5555720 -1.9968870 0.0000000
## LS-Durazo   2.6526230  1.8732805  3.4319654 0.0000000
## May-Durazo -2.3281967 -3.1075392 -1.5488543 0.0000000
## SC-Durazo   3.0180328  2.2386903  3.7973753 0.0000000
## SP-Durazo   1.2809836  0.5016411  2.0603261 0.0000520
## May-LS     -4.9808197 -5.7601621 -4.2014772 0.0000000
## SC-LS       0.3654098 -0.4139326  1.1447523 0.7605330
## SP-LS      -1.3716393 -2.1509818 -0.5922969 0.0000108
## SC-May      5.3462295  4.5668870  6.1255720 0.0000000
## SP-May      3.6091803  2.8298379  4.3885228 0.0000000
## SP-SC      -1.7370492 -2.5163917 -0.9577067 0.0000000
```

```
report(anova_SC)
```

```
## For one-way between subjects designs, partial eta squared is equivalent to
eta squared.
## Returning eta squared.
```

```
## The ANOVA (formula: value ~ site) suggests that:
##
##   - The main effect of site is statistically significant and large (F(5,
360) =
## 146.55, p < .001; Eta2 = 0.67, 95% CI [0.63, 1.00])
##
## Effect sizes were labelled following Field's (2013) recommendations.
```

###Horse Creek

```
#Boxplot
b2 <- ggplot(subset(data_summer, site %in% c("FG","UL","LL","GP","HC")))+
  geom_boxplot(aes(x = site, value, col = day))+
  labs(title = "(b) Horse Creek, 15 July - 15 September")+
  theme_classic()+
  theme(text=element_text(size=16), axis.title.x=element_blank())+
  scale_x_discrete(name ="Site",
                   limits = c("FG","UL","LL","GP","HC"),
                    labels=c("FG" = "FishGulch","UL"= "UpperLaw","LL"=
                               "LowerLaw","GP" =
"Goodman","HC"="HorseCreek"))+
  scale_y_continuous(name="Daily Maximum Temp (C)",limits=c(12.5, 22.5))

#ANOVA
data_summer_HC <- subset(data_summer, site %in% c("FG","UL","LL","GP","HC"))
shapiro.test(data_summer_HC$value) #check for normality
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  data_summer_HC$value
## W = 0.96968, p-value = 5.183e-06
```

```
leveneTest(value ~ site, data = data_summer_HC) #check for equal variances
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   4  21.089 2.481e-15 ***
##       299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova_HC <- aov(value ~ site, data = data_summer_HC) #run one way anova
summary.aov(anova_HC)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## site          4  834.0  208.51   300.9 <2e-16 ***
## Residuals   299  207.2    0.69
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

```
TukeyHSD(anova_HC) #posthoc analysis
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ site, data = data_summer_HC)
##
## $site
##             diff        lwr          upr     p adj
## GP-FG -2.6355984 -3.0510175 -2.220179202 0.0000000
## HC-FG  2.4490738  2.0336546  2.864492929 0.0000000
## LL-FG -0.4217459 -0.8371651 -0.006326743 0.0446373
## UL-FG  0.7445656  0.3291464  1.159984733 0.0000142
## HC-GP  5.0846721  4.6709731  5.498371121 0.0000000
## LL-GP  2.2138525  1.8001535  2.627551448 0.0000000
## UL-GP  3.3801639  2.9664649  3.793862924 0.0000000
## LL-HC -2.8708197 -3.2845187 -2.457120683 0.0000000
## UL-HC -1.7045082 -2.1182072 -1.290809207 0.0000000
## UL-LL  1.1663115  0.7526125  1.580010465 0.0000000
```

```
report(anova_HC)
```

```
## For one-way between subjects designs, partial eta squared is equivalent to
eta squared.
## Returning eta squared.
```

```
## The ANOVA (formula: value ~ site) suggests that:
##
##   - The main effect of site is statistically significant and large (F(4,
299) =
## 300.88, p < .001; Eta2 = 0.80, 95% CI [0.77, 1.00])
##
## Effect sizes were labelled following Field's (2013) recommendations.
```

##Winter ###Data wrangling

```
#Read in data
daily_mins_condensed <- readRDS('daily_mins_condensed.RDS')
daily_mins_condensed <- as.data.frame(t(daily_mins_condensed))
daily_mins_condensed <- daily_mins_condensed %>%
mutate_all(~ifelse(is.nan(.), NA, .))

#Create "day" sequence
day <- seq(mdy('7/1/2020'),mdy('7/13/2021'),by = "day")
day <- as.data.frame(day) #convert "day" to data frame

#Attach day to data frame
daily_mins_condensed <- cbind(daily_mins_condensed,day)

#Format data long
daily_min_long <- gather(daily_mins_condensed, site, value, -day)
data_winter <- daily_min_long %>%filter(day>'2020-12-15'& day < '2021-02-15')
```

###Seiad Creek

```
#Boxplot
b3 <- ggplot(subset(data_winter, site %in%
c("AP","SP","Durazo","LS","May","SC")))+
  geom_boxplot(aes(x = site, value, col = day))+
  labs(title = "(c) Seiad Creek, 15 December - 15 February")+
  theme_classic()+
  theme(text=element_text(size=16))+
  scale_x_discrete(name ="Site",
                   limits = c("AP","SP","Durazo","LS","May","SC"),
                    labels=c("AP" = "Alexander","SP"= "Stender","Durazo"=
                               "Durazo","LS" = "LowerSeiad","May"="May","SC"
= "SeiadCreek"))+
  scale_y_continuous(name="Daily Minimum Temp (C)",limits=c(4, 12))

#ANOVA
data_winter_SC <- subset(data_winter, site %in%
c("AP","SP","Durazo","LS","May","SC"))
shapiro.test(data_winter_SC$value) #check for normality

##
##  Shapiro-Wilk normality test
##
```

```
## data:  data_winter_SC$value
## W = 0.97899, p-value = 3.577e-05
```

```
leveneTest(value ~ site, data = data_winter_SC) #check for equal variances
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   5  2.0719 0.06831 .
##       360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#variances not equal, need to transform
data_winter_SC$trans <- log(data_winter_SC$value)
leveneTest(trans ~ site, data = data_winter_SC)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   5  6.4079 1.017e-05 ***
##       360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#variances equal with log transformation

anova_SC2 <- aov(trans ~ site, data = data_winter_SC) #run one way anova
summary.aov(anova_SC2)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## site          5 11.625  2.3250   131.1 <2e-16 ***
## Residuals   360  6.383  0.0177
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(anova_SC2) #posthoc analysis
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = trans ~ site, data = data_winter_SC)
##
## $site
##                   diff         lwr         upr     p adj
## Durazo-AP   0.241037424  0.17195420  0.31012065 0.0000000
## LS-AP      -0.102090413 -0.17117364 -0.03300719 0.0004165
## May-AP      0.373473971  0.30439074  0.44255720 0.0000000
```

```
## SC-AP      -0.110127715 -0.17921094 -0.04104449 0.0000988
## SP-AP       0.022372196 -0.04671103  0.09145542 0.9391630
## LS-Durazo  -0.343127837 -0.41221106 -0.27404461 0.0000000
## May-Durazo  0.132436547  0.06335332  0.20151977 0.0000011
## SC-Durazo  -0.351165138 -0.42024837 -0.28208191 0.0000000
## SP-Durazo  -0.218665228 -0.28774845 -0.14958200 0.0000000
## May-LS      0.475564383  0.40648116  0.54464761 0.0000000
## SC-LS      -0.008037302 -0.07712053  0.06104593 0.9994529
## SP-LS       0.124462609  0.05537938  0.19354584 0.0000060
## SC-May     -0.483601685 -0.55268491 -0.41451846 0.0000000
## SP-May     -0.351101775 -0.42018500 -0.28201855 0.0000000
## SP-SC       0.132499910  0.06341668  0.20158314 0.0000011
```

```
report(anova_SC2)
```

```
## For one-way between subjects designs, partial eta squared is equivalent to
eta squared.
## Returning eta squared.
```

```
## The ANOVA (formula: trans ~ site) suggests that:
##
##   - The main effect of site is statistically significant and large (F(5,
360) =
## 131.12, p < .001; Eta2 = 0.65, 95% CI [0.60, 1.00])
##
## Effect sizes were labelled following Field's (2013) recommendations.
```

###Horse Creek

```
#Boxplot
b4 <- ggplot(subset(data_winter, site %in% c("FG","UL","LL","GP","HC")))+
  geom_boxplot(aes(x = site, value, col = day))+
  labs(title = "(d) Horse Creek, 15 December - 15 February")+
  theme_classic()+
  theme(text=element_text(size=16))+
  scale_x_discrete(name ="Site",
                   limits = c("FG","UL","LL","GP","HC"),
                    labels=c("FG" = "FishGulch","UL"= "UpperLaw","LL"=
                               "LowerLaw","GP" =
"Goodman","HC"="HorseCreek"))+
  scale_y_continuous(name="Daily Minimum Temp (C)",limits=c(4, 12))

#ANOVA
data_winter_HC <- subset(data_winter, site %in% c("FG","UL","LL","GP","HC"))
shapiro.test(data_winter_HC$value) #check for normality
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_winter_HC$value
## W = 0.97587, p-value = 5.203e-05
```

```
leveneTest(value ~ site, data = data_winter_HC) #check for equal variances
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   4   6.604 4.179e-05 ***
##       300
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova_HC2 <- aov(value ~ site, data = data_winter_HC) #run one way anova
summary.aov(anova_HC2)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## site          4  461.0  115.24     170 <2e-16 ***
## Residuals   300  203.4    0.68
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(anova_HC2) #posthoc analysis
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ site, data = data_winter_HC)
##
## $site
##              diff        lwr         upr     p adj
## GP-FG  2.05360656  1.6444688  2.46274436 0.0000000
## HC-FG -1.12196721 -1.5311050 -0.71282941 0.0000000
## LL-FG  1.69054645  1.2814086  2.09968425 0.0000000
## UL-FG  1.74040984  1.3312720  2.14954764 0.0000000
## HC-GP -3.17557377 -3.5847116 -2.76643597 0.0000000
## LL-GP -0.36306011 -0.7721979  0.04607769 0.1088229
## UL-GP -0.31319672 -0.7223345  0.09594108 0.2223688
## LL-HC  2.81251366  2.4033759  3.22165146 0.0000000
## UL-HC  2.86237705  2.4532392  3.27151485 0.0000000
## UL-LL  0.04986339 -0.3592744  0.45900119 0.9972919
```

```
report(anova_HC2)
```

```
## For one-way between subjects designs, partial eta squared is equivalent to
eta squared.
## Returning eta squared.
```

```
## The ANOVA (formula: value ~ site) suggests that:
##
##   - The main effect of site is statistically significant and large (F(4,
300) =
## 170.01, p < .001; Eta2 = 0.69, 95% CI [0.65, 1.00])
```

```
##
## Effect sizes were labelled following Field's (2013) recommendations.
```

##Boxplot Figure

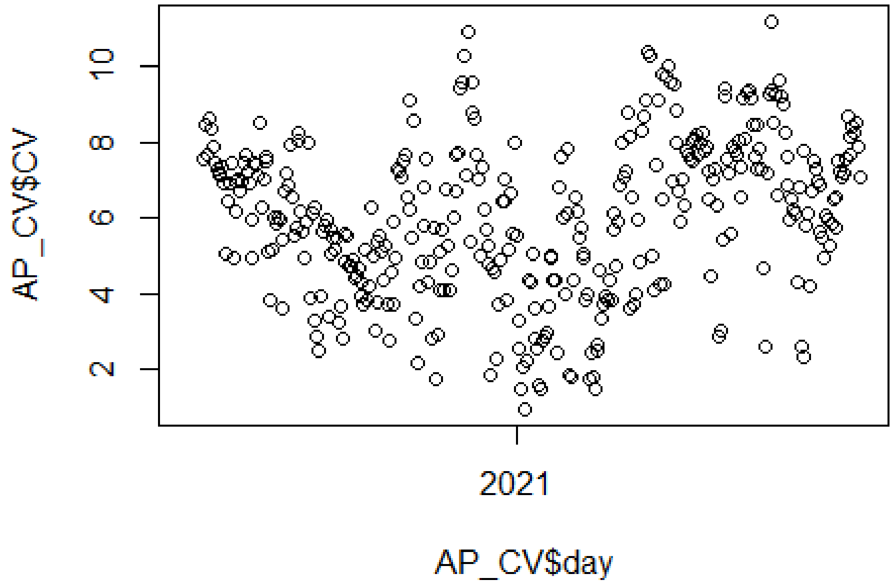
```
png("Fig_Boxplots.png",width = 1000, height = 600)
(b1 + b2) / (b3 + b4)

## Warning: Removed 12 rows containing non-finite values (stat_boxplot).

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).

## Warning: Removed 2 rows containing non-finite values (stat_boxplot).

## Warning: Removed 21 rows containing non-finite values (stat_boxplot).

dev.off()

## png
##   2
```

#Daily Buffering ##Data wrangling

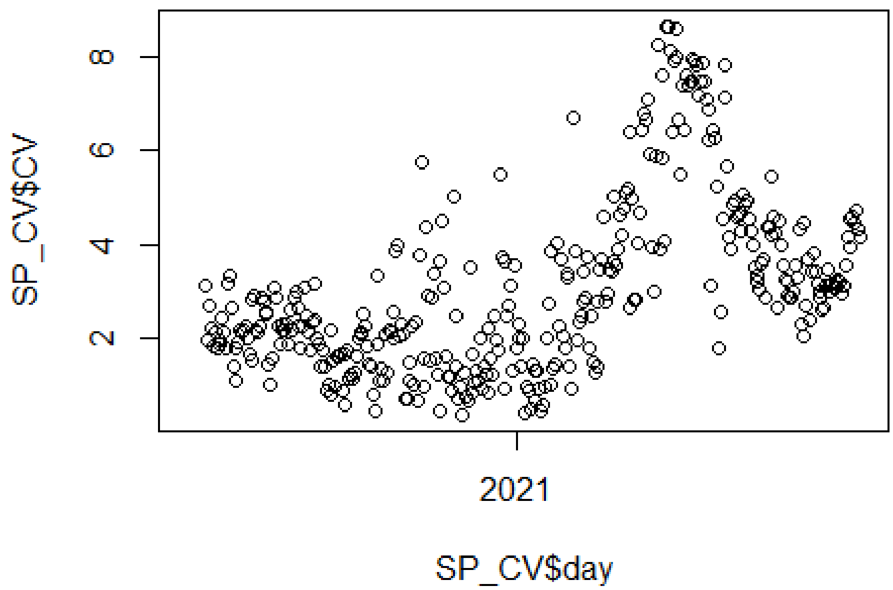
```
quarterHour_condensed$day <-
lubridate::floor_date(quarterHour_condensed$quarterHour, unit="day") #Bin by
day

#Create a function for coefficient of variance (CV)
CV <- function(x){
        (sd(x)/mean(x))*100
}
```

##Individual CVs

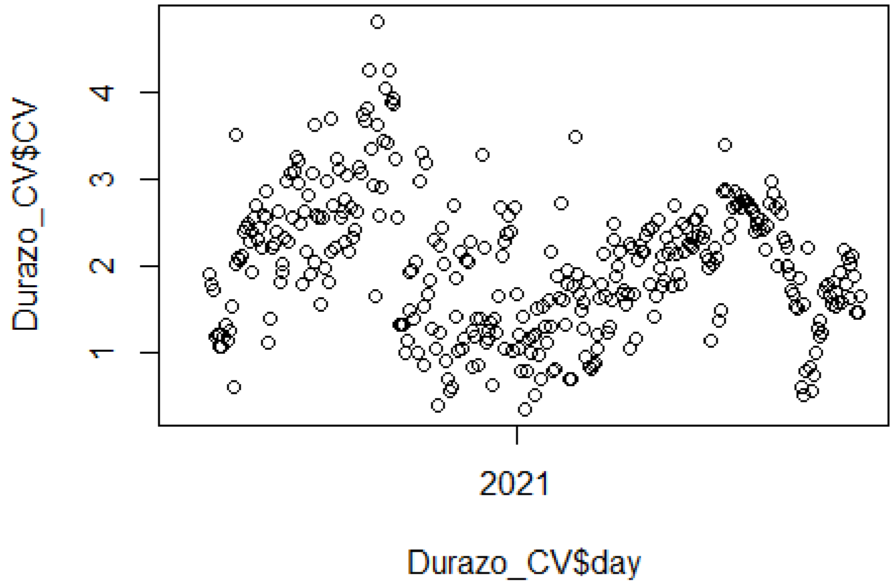
```
#Alexander Pond
AP_CV <- quarterHour_condensed %>%       #15 min data
  group_by(day) %>%                      #Group by day
  dplyr::select(AP) %>%                  #Select site
  summarize(CV = CV(AP))                 #Calculate CV per day for one site

## Adding missing grouping variables: `day`

AP_CV_mean <- mean(AP_CV$CV,na.rm=T)     #Take all days as mean over entire
year
plot(AP_CV$day,AP_CV$CV)                 #Plot a time series of CV
```



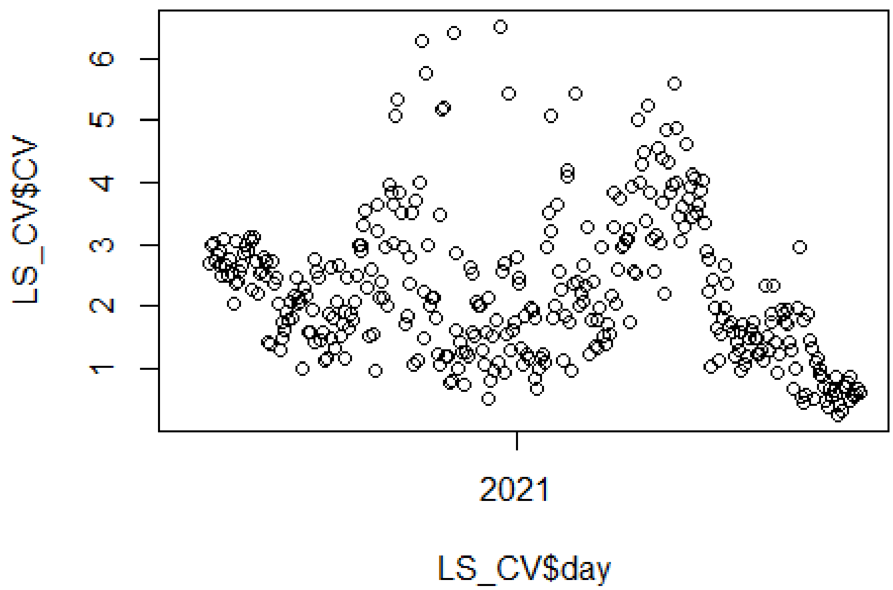

```
#Stender Pond
SP_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(SP) %>%
  summarize(CV = CV(SP))

## Adding missing grouping variables: `day`

SP_CV_mean <- mean(SP_CV$CV,na.rm=T)
plot(SP_CV$day,SP_CV$CV)
```



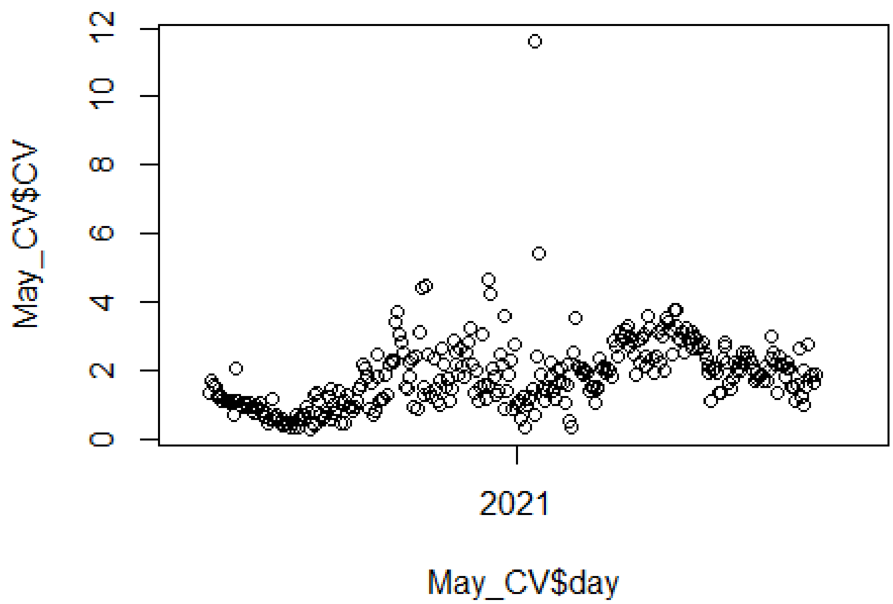

```r
#Durazo Pond
Durazo_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(Durazo) %>%
  summarize(CV = CV(Durazo))

## Adding missing grouping variables: `day`

Durazo_CV_mean <- mean(Durazo_CV$CV,na.rm=T)
plot(Durazo_CV$day,Durazo_CV$CV)
```



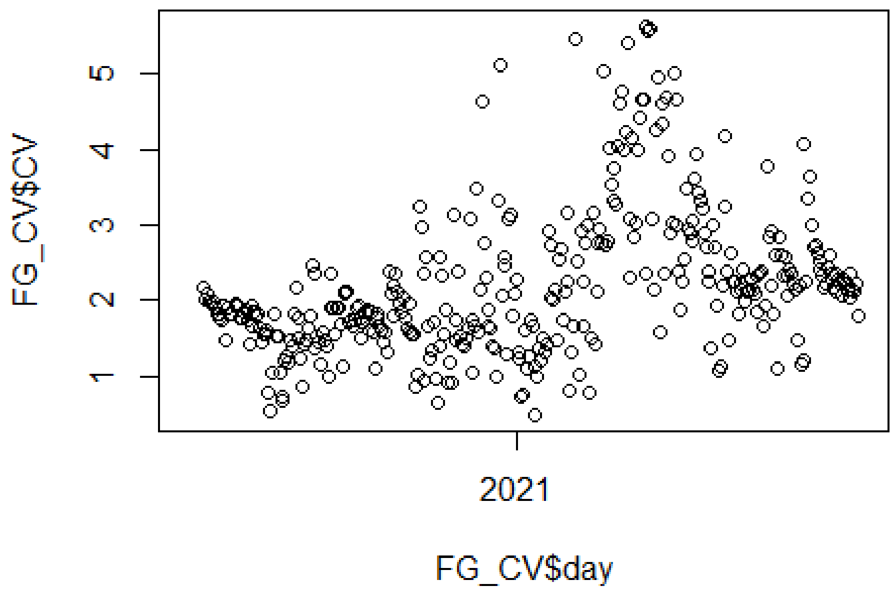

```
#Lower Seiad
LS_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(LS) %>%
  summarize(CV = CV(LS))

## Adding missing grouping variables: `day`

LS_CV_mean <- mean(LS_CV$CV,na.rm=T)
plot(LS_CV$day,LS_CV$CV)
```



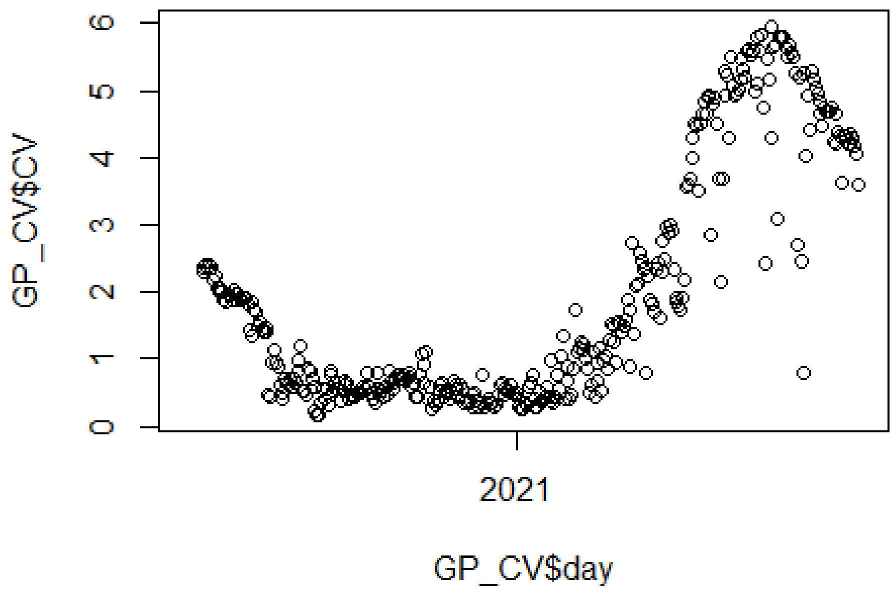

```
#May Pond
May_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(May) %>%
  summarize(CV = CV(May))

## Adding missing grouping variables: `day`

May_CV_mean <- mean(May_CV$CV,na.rm=T)
plot(May_CV$day,May_CV$CV)
```



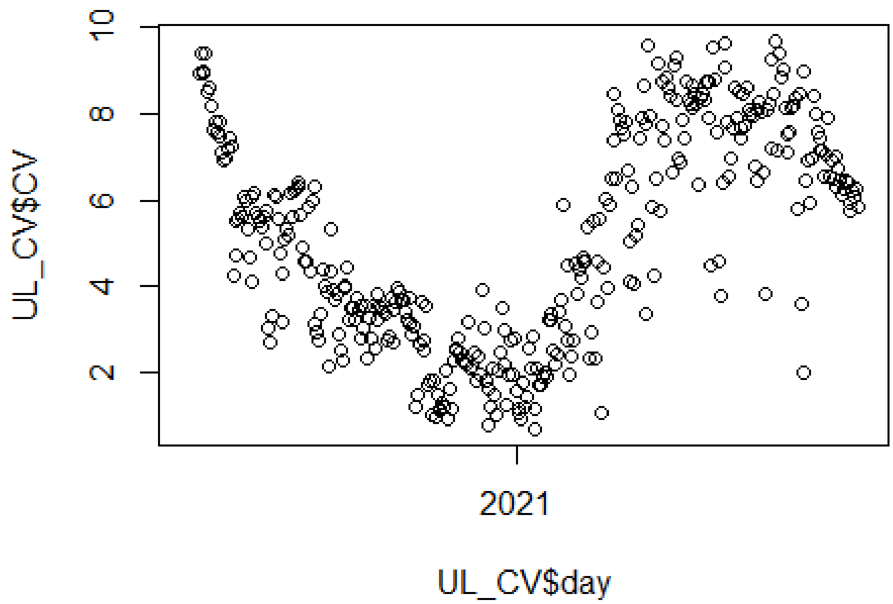

```
#Fish Gulch
FG_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(FG) %>%
  summarize(CV = CV(FG))

## Adding missing grouping variables: `day`

FG_CV_mean <- mean(FG_CV$CV,na.rm=T)
plot(FG_CV$day,FG_CV$CV)
```



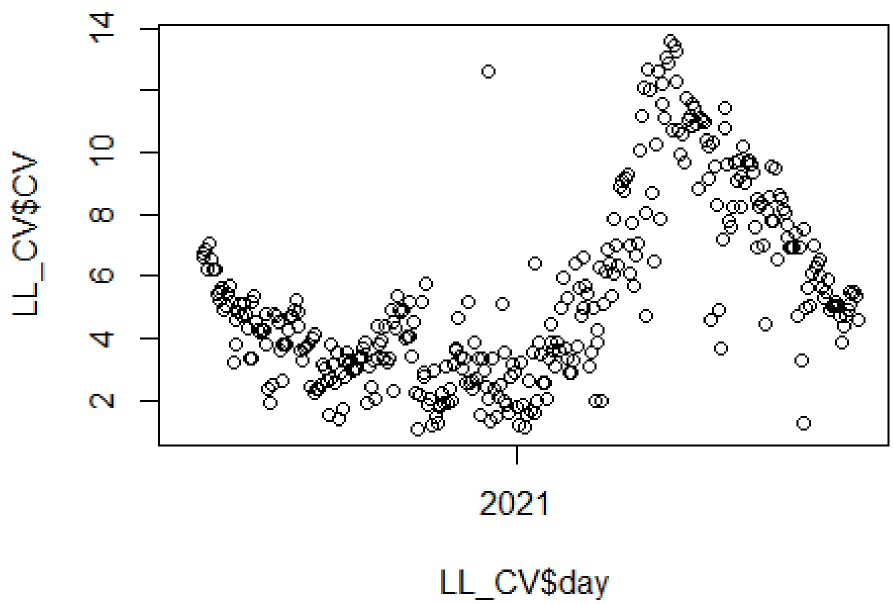

```
#Goodman Pond
GP_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(GP) %>%
  summarize(CV = CV(GP))

## Adding missing grouping variables: `day`

GP_CV_mean <- mean(GP_CV$CV,na.rm=T)
plot(GP_CV$day,GP_CV$CV)
```



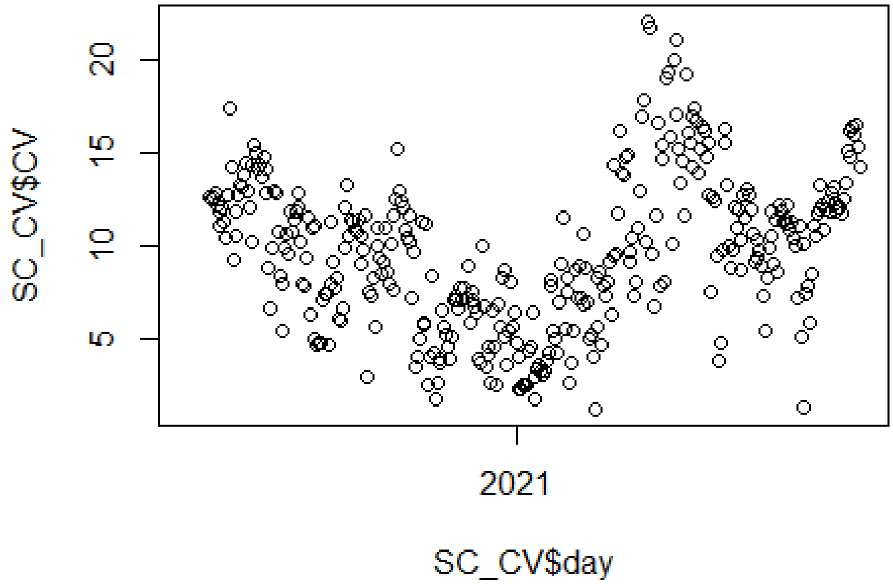

```
#Upper Lawrence
UL_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(UL) %>%
  summarize(CV = CV(UL))

## Adding missing grouping variables: `day`

UL_CV_mean <- mean(UL_CV$CV,na.rm=T)
plot(UL_CV$day,UL_CV$CV)
```



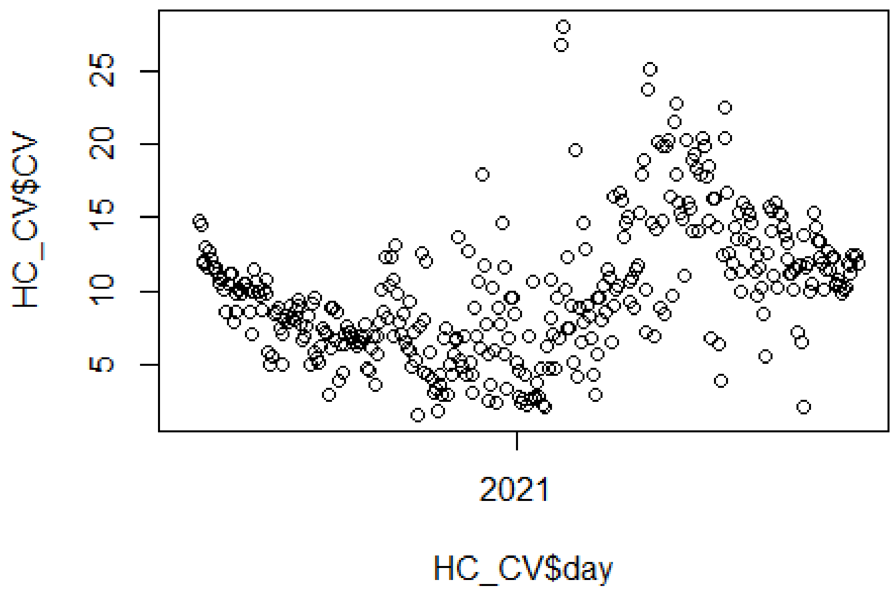

```
#Lower Lawrence
LL_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(LL) %>%
  summarize(CV = CV(LL))

## Adding missing grouping variables: `day`

LL_CV_mean <- mean(LL_CV$CV,na.rm=T)
plot(LL_CV$day,LL_CV$CV)
```




```
#Seiad Creek
SC_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(SC) %>%
  summarize(CV = CV(SC))

## Adding missing grouping variables: `day`

SC_CV_mean <- mean(SC_CV$CV,na.rm=T)
plot(SC_CV$day,SC_CV$CV)
```




```
#Horse Creek
HC_CV <- quarterHour_condensed %>%
  group_by(day) %>%
  dplyr::select(HC) %>%
  summarize(CV = CV(HC))

## Adding missing grouping variables: `day`

HC_CV_mean <- mean(HC_CV$CV,na.rm=T)
plot(HC_CV$day,HC_CV$CV)
```



##CV Ratios

```
#CV Ratios are Creeks/Ponds
#No Buffering = <1
#Buffering = >1

#Alexander Pond
SC_AP <- SC_CV_mean/AP_CV_mean

#Stender Pond
SC_SP <- SC_CV_mean/SP_CV_mean

#Durazo Pond
SC_Durazo <- SC_CV_mean/Durazo_CV_mean

#Lower Seiad
SC_LS <- SC_CV_mean/LS_CV_mean

#May Pond
SC_May <- SC_CV_mean/May_CV_mean

#Fish Gulch
HC_FG <- HC_CV_mean/FG_CV_mean

#Goodman Pond
HC_GP <- HC_CV_mean/GP_CV_mean
```

```r
#Upper Lawrence
HC_UL <- HC_CV_mean/UL_CV_mean

#Lower Lawrence
HC_LL <- HC_CV_mean/LL_CV_mean
```

##CV Data Table

```r
data.frame(Creek=c("Seiad Creek","Seiad Creek","Seiad Creek","Seiad
Creek","Seiad Creek","Horse Creek","Horse Creek","Horse Creek","Horse
Creek"),

CreekCV=round(c(SC_CV_mean,SC_CV_mean,SC_CV_mean,SC_CV_mean,SC_CV_mean,HC_CV_
mean,HC_CV_mean,HC_CV_mean,HC_CV_mean),2),
           Pond=c("Alexander","Stender","Durazo","Lower Seiad","May","Fish
Gulch","Goodman","Upper Lawrence","Lower Lawrence"),

PondCV=round(c(AP_CV_mean,SP_CV_mean,Durazo_CV_mean,LS_CV_mean,May_CV_mean,FG
_CV_mean,GP_CV_mean,UL_CV_mean,LL_CV_mean),2),
           CV_Ratio=(round(c(SC_AP,
                        SC_SP,
                        SC_Durazo,
                        SC_LS,
                        SC_May,
                        HC_FG,
                        HC_GP,
                        HC_UL,
                        HC_LL),2)))

##         Creek CreekCV           Pond PondCV CV_Ratio
## 1 Seiad Creek    9.68      Alexander   6.02     1.61
## 2 Seiad Creek    9.68        Stender   3.08     3.14
## 3 Seiad Creek    9.68         Durazo   2.02     4.80
## 4 Seiad Creek    9.68    Lower Seiad   2.25     4.31
## 5 Seiad Creek    9.68            May   1.83     5.30
## 6 Horse Creek    9.94     Fish Gulch   2.21     4.49
## 7 Horse Creek    9.94        Goodman   1.97     5.06
## 8 Horse Creek    9.94 Upper Lawrence   5.09     1.95
## 9 Horse Creek    9.94 Lower Lawrence   5.35     1.86
```

#Figure 3: Time Series Examples Figure

```r
daily_means_condensed <- readRDS('daily_means_condensed.RDS')
daily_means_condensed <- as.data.frame(t(daily_means_condensed))
daily_means_condensed <- daily_means_condensed %>%
mutate_all(~ifelse(is.nan(.), NA, .))
day <- seq(mdy('7/1/2020'),mdy('7/13/2021'),by = "day")
day <- as.data.frame(day) #convert "day" to data frame

#Attach day to data frame
```

```r
daily_means_condensed <- cbind(daily_means_condensed,day)
daily_means_condensed$day <- as.POSIXct(daily_means_condensed$day)

mycolors <- met.brewer(name="Archambault",n=4,type="discrete")

p1 <- ggplot()+
  geom_line(data = quarterHour_condensed, aes(x = quarterHour, y = AP), color
="#D3D3D3", size = 1) +
  geom_line(data = daily_means_condensed, aes(x = day, y = AP), color
="#88a0dc", size = 1) +
  labs(title = "(a) Alexander Pond", x = "Date", y = "Temperature (C)")+
  scale_y_continuous(name="Daily Mean Temp (C)",limits=c(0, 25))+
  theme_classic()+
  theme(text=element_text(size=16),legend.position =
"none",axis.title.x=element_blank())
p2 <- ggplot()+
  geom_line(data = quarterHour_condensed, aes(x = quarterHour, y = UL), color
="#D3D3D3", size = 1) +
  geom_line(data = daily_means_condensed, aes(x = day, y = UL), color =
"#ed968c", size = 1) +
  labs(title = "(c) Upper Lawrence Pond", x = "Date", y = "Temperature (C)")+
  scale_y_continuous(name="Daily Mean Temp (C)",limits=c(0, 25))+
  theme_classic()+
  theme(text=element_text(size=16),legend.position = "none")
p3 <- ggplot()+
  geom_line(data = quarterHour_condensed, aes(x = quarterHour, y = SC), color
="#D3D3D3", size = 1) +
  geom_line(data = daily_means_condensed, aes(x = day, y = SC), color =
"#e78429", size = 1) +
  labs(title = "(b) Seiad Creek", x = "Date", y = "Temperature (C)")+
  scale_y_continuous(name="Daily Mean Temp (C)",limits=c(0, 25))+
  theme_classic()+
  theme(text=element_text(size=16),legend.position = "none",
axis.title.x=element_blank())
p4 <- ggplot()+
  geom_line(data = quarterHour_condensed, aes(x = quarterHour, y = HC), color
="#D3D3D3", size = 1) +
  geom_line(data = daily_means_condensed, aes(x = day, y = HC), color =
"#f9d14a", size = 1) +
  labs(title = "(d) Horse Creek", x = "Date", y = "Temperature (C)")+
  scale_y_continuous(name="Daily Mean Temp (C)",limits=c(0, 25))+
  theme_classic()+
  theme(text=element_text(size=16),legend.position = "none")

png("Fig_TempTimeSeries.png",width = 1000, height = 600)
(p1 + p3) / (p2 + p4)

## Warning: Removed 931 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 1202 row(s) containing missing values (geom_path).

## Warning: Removed 13 row(s) containing missing values (geom_path).

## Warning: Removed 694 row(s) containing missing values (geom_path).

## Warning: Removed 9 row(s) containing missing values (geom_path).

## Warning: Removed 688 row(s) containing missing values (geom_path).

## Warning: Removed 9 row(s) containing missing values (geom_path).

dev.off()

## png
##   2
```

#May Pond Example

```
May <- readRDS("May.rds")
MayCk <- readRDS("MayCk.rds")

May_August <-May %>%filter(day>'2020-08-01'& day < '2020-08-31')
May_August <- May_August[,c("Temp","date")]
MayCk_August <-MayCk %>%filter(day>'2020-08-01'& day < '2020-08-31')
MayCk_August <- MayCk_August[,c("Temp","date")]

png("Fig_MayPond.png", width = 700, height = 400)
ggplot()+
  geom_line(data = May_August, aes(x = date, y = Temp, color = "May Pond"))+
  geom_line(data = MayCk_August, aes(x = date, y = Temp, color = "Seiad
Creek"))+
  labs(x = "Date",
       y = "Hourly Temperature (C)",
       title = "May Pond and Seiad Creek, August 2020")+
  theme_classic()+
  theme(text=element_text(size=16), legend.position = "bottom")+
  scale_colour_manual("", values = c("May Pond"="#88a0dc", "Seiad
Creek"="#ed968c")) +
  scale_y_continuous("Hourly Temperature (C)")
dev.off()

## png
##   2
```