Appendix 2: Analysis for Partitioning inorganic carbon fluxes using paired O_2-CO_2 sensors in a headwater stream, Costa Rica, Submitted to Biogeochemistry

Nicholas S. Marzolf

Last updated: June 22, 2022

Contents

Publication information

Full authorship and affiliations:

Nicholas S. Marzolf[1 *], Gaston E. Small[2], Diana Oviedo-Vargas[3], Carissa N. Ganong[4], John H. Duff[5], Alonso Ramírez[6], Catherine M. Pringle[7], David P. Genereux[8], Marcelo Ardón[1,9]

1 Department of Forestry and Environmental Resources, North Carolina State University, NC, USA (ORCiD: 0000-0001-9146-1643)
2 College of Arts and Sciences, University of St. Thomas, MN, USA (ORDiD: 0000-0002-9018-7555)
3 Stroud Water Research Center, Avondale, PA, USA (ORCiD: 0000-0001-9333-0962)
4 Department of Biology, Missouri Western State University, MO, USA
5 US Geological Survey, Menlo Park, CA, USA
6 Department of Applied Ecology, North Carolina State University, NC, USA (ORCiD: 0000-0001-9985-5719)
7 Odum School of Ecology, University of Georgia, GA, USA (ORCiD: 0000-0002-3522-7315)

8 Marine, Earth, and Atmospheric Sciences, North Carolina State University, NC, USA (ORCiD: 0000-0001-5601-7088)
9 ORCiD: 0000-0001-7275-2672
* Corresponding author: nmarzol@alumni.ncsu.edu

Load packages and explore the data

```
# data manipulation
library(dplyr)
library(tidyr)
library(lubridate)
library(tidyquant)
# plotting
library(ggplot2)
library(cowplot)
library(ggrepel)
library(scales)
library(ggsci)
# stats
library(lmodel2)
library(ellipse)
library(purrr)
library(lsmeans)
library(emmeans)
library(ARTool)
# hydrology
library(hydrostats)
# reproducibility
library(pander)

# Read in time-series data
Tac_all <- readRDS("Tac_all.rds")
```

This RDS file contains hourly data from April 1 - September 27, including: stream and well Vaisala pCO2 sensors (CO2_ppm, wellCO2), YSI sonde (temp.water, pH, DO.obs, DO.sat, cond), discharge (sumQ_m3_hr, meanQ_m3_s), and meterological data (relative humidity (RH), Rainfall, barometric pressure (bp_mmHg)) from the weather station located at La Selva Biological Station. Rainfall is collected at 15-minute intervals, and there data here are aggregated to hourly sums. Discharge is collected at 15-minute intervals, and is shown as both mean discharge for the hour (m3/s) and sum of the 4 contributing measurements (m3/hr). Groundwater discharge (GW_Q) is described in the text. Mean depth is taken from the weir.

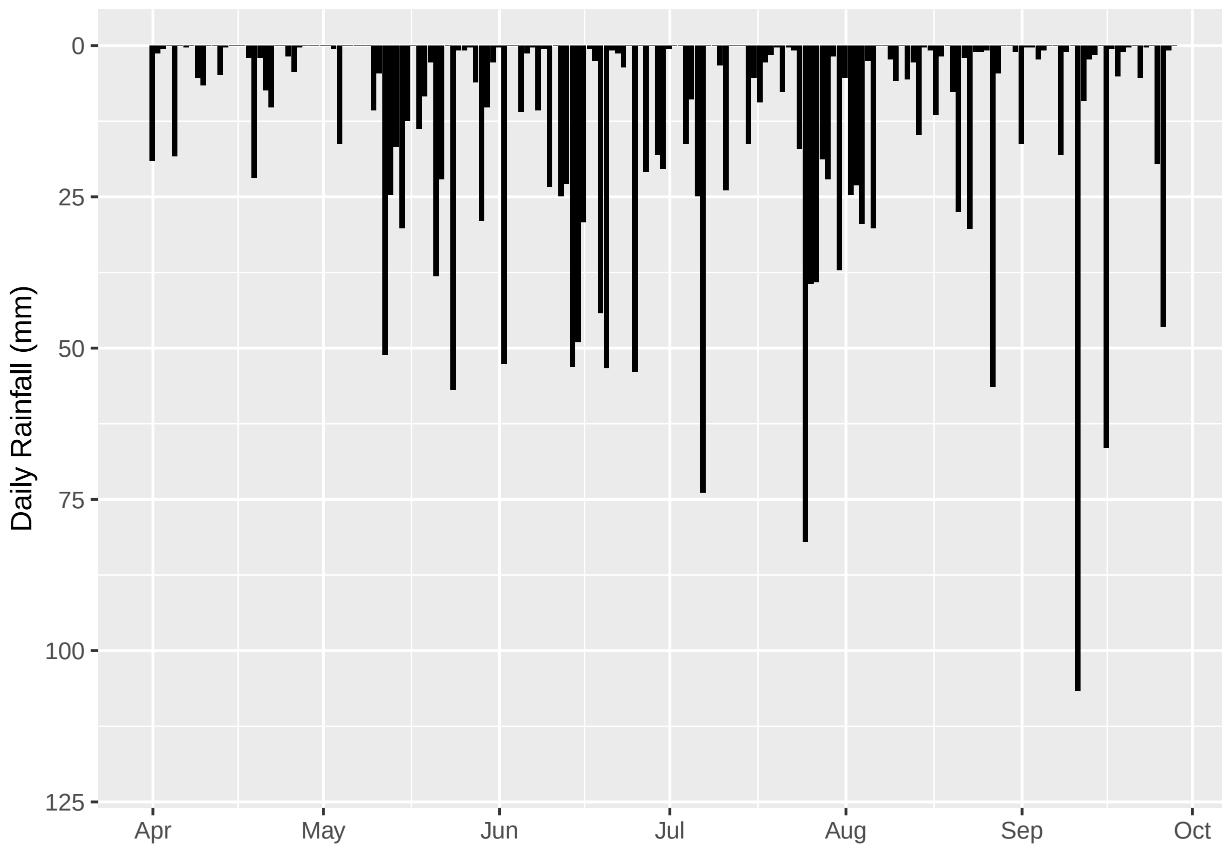
We will use this dataset build on with the following calculations at the hourly resolution and aggregate to daily scale at the end of the document.

Figure 2- Time Series

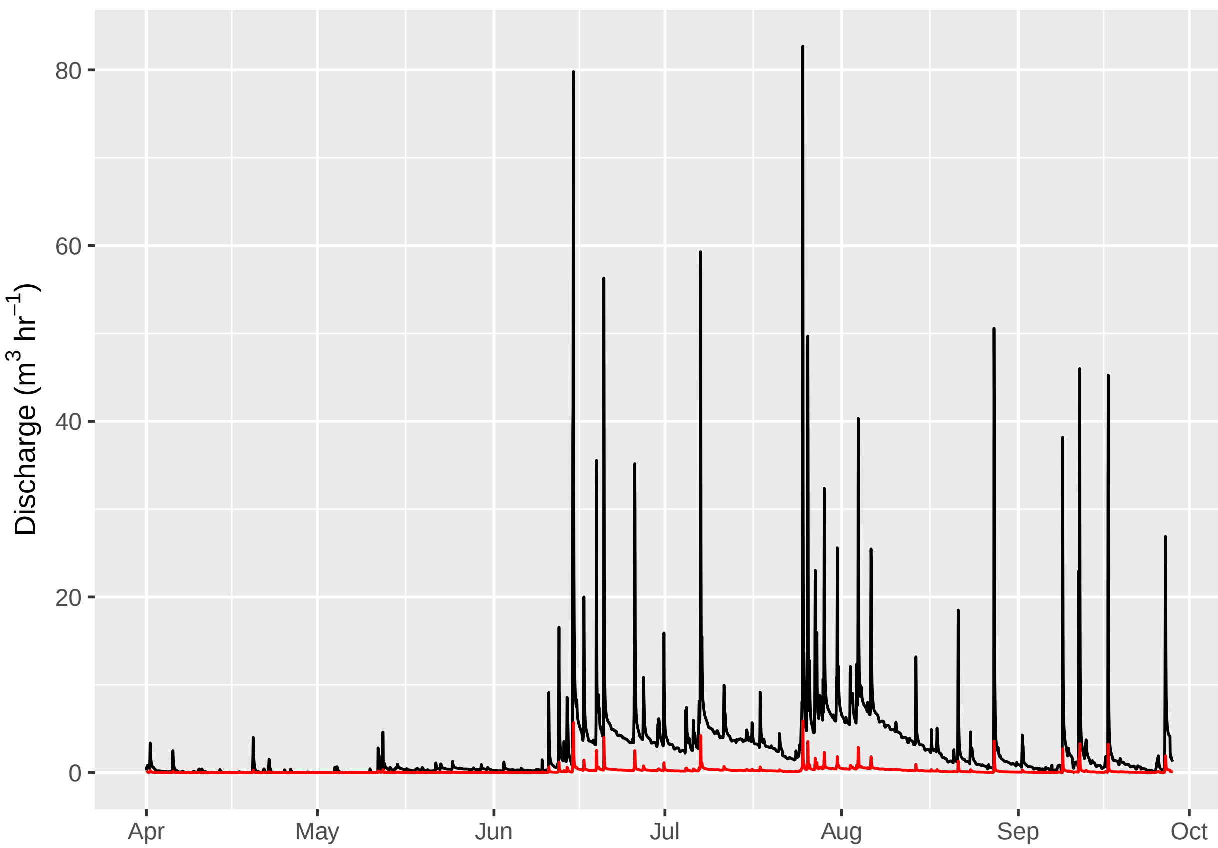
Figure 2 details the stream measurements. We plot rainfall, discharge, groundwater discharge, dissolved oxygen, CO2 in the stream and in the riparian well.

```
plot_rain <- ggplot(Tac_all %>%
                      select(Timestamp, Rainfall) %>%
```

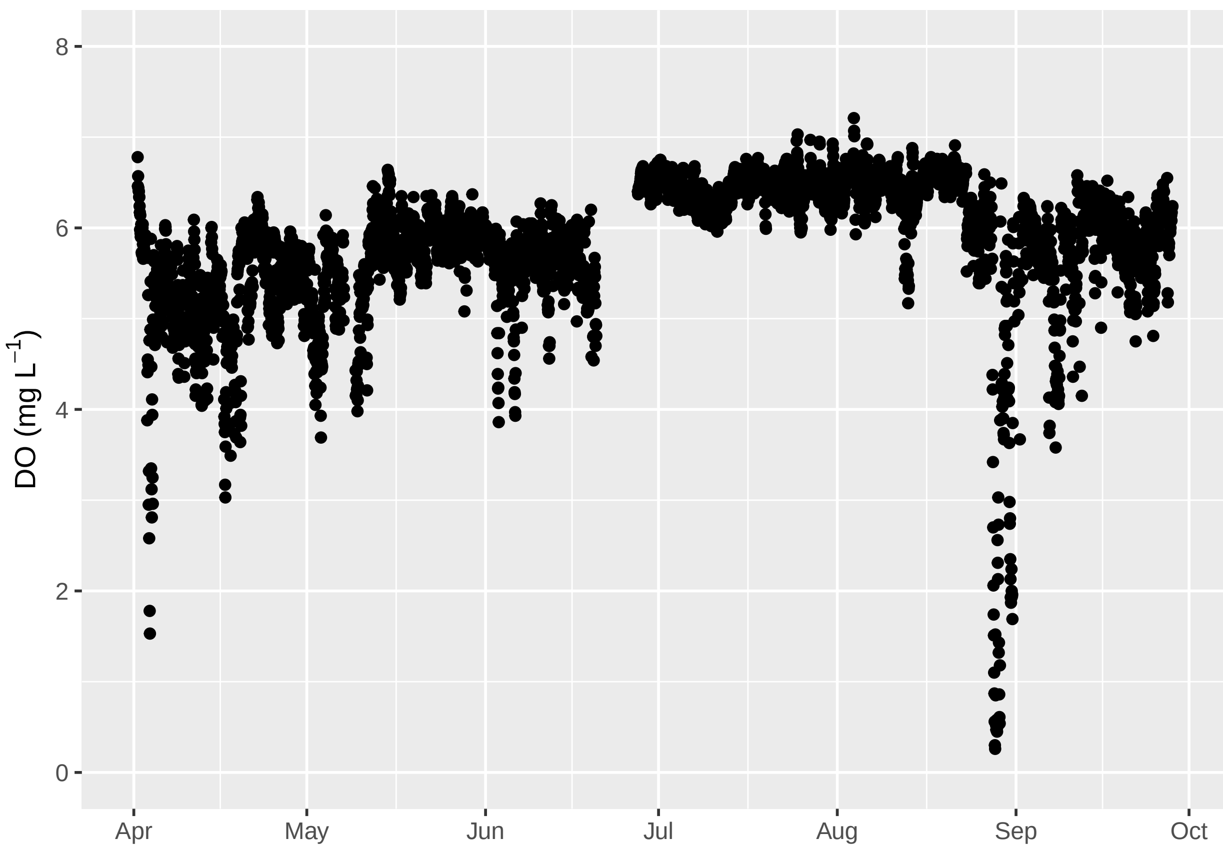
```
                      group_by(Date = date(Timestamp)) %>%
                      summarise(dayRain = sum(Rainfall, na.rm = TRUE))+
  geom_bar(aes(x = as.POSIXct(Date),
               y = dayRain),
           stat = 'identity', fill = 'black')+
  ylim(120, 0)+
  ylab('Daily Rainfall (mm)')+
  scale_x_datetime(date_breaks = "1 month", date_labels =  "%b")+
  theme(axis.title.x = element_blank())
plot_rain
```



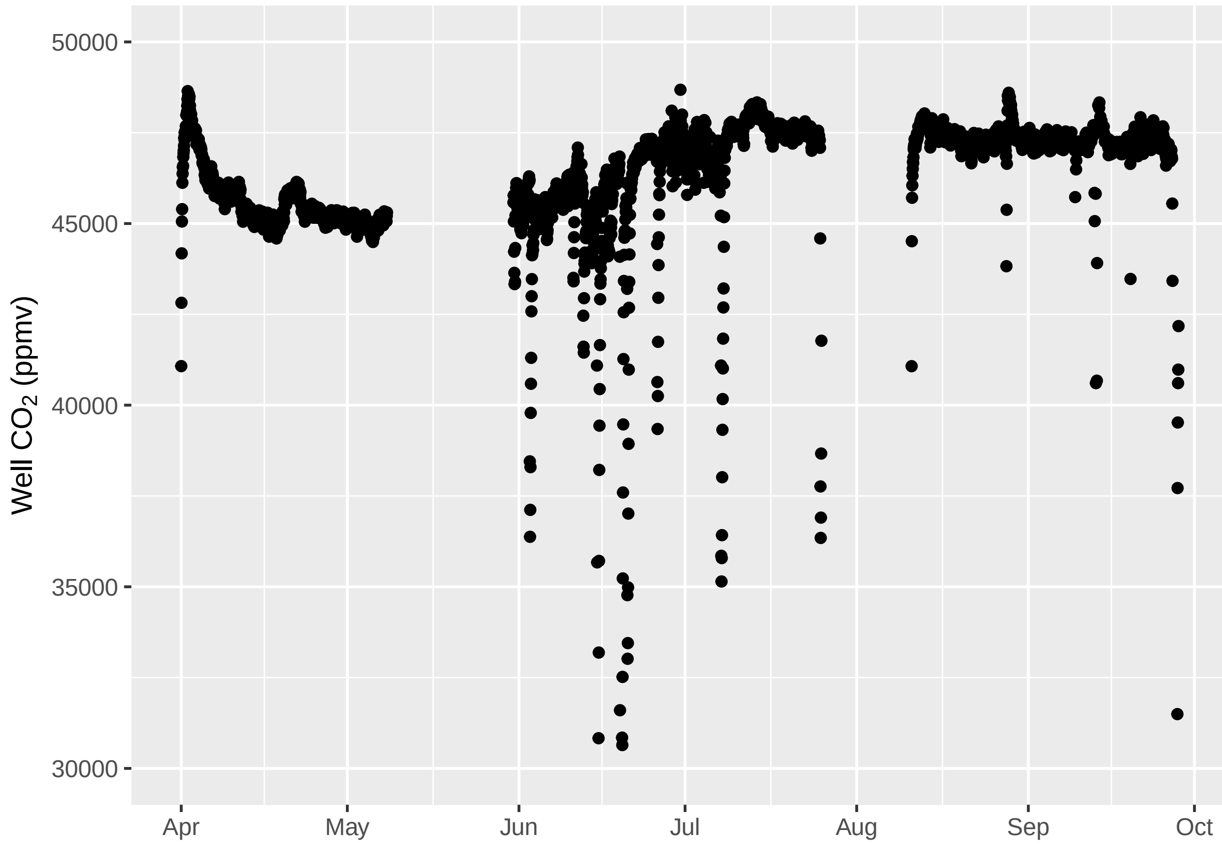

```
plot_Q <- ggplot(Tac_all,
                 aes(x = Timestamp))+
  geom_line(aes(y = sumQ_m3_hr),
            color = 'black')+
  geom_line(aes(y = GW_Q),
            color = 'red')+
  ylab(expression(paste("Discharge (", m^3," ", hr^-1,")")))+
  scale_color_manual(values = c('black',
                                'red'))+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  theme(axis.title.x = element_blank())
plot_Q
```



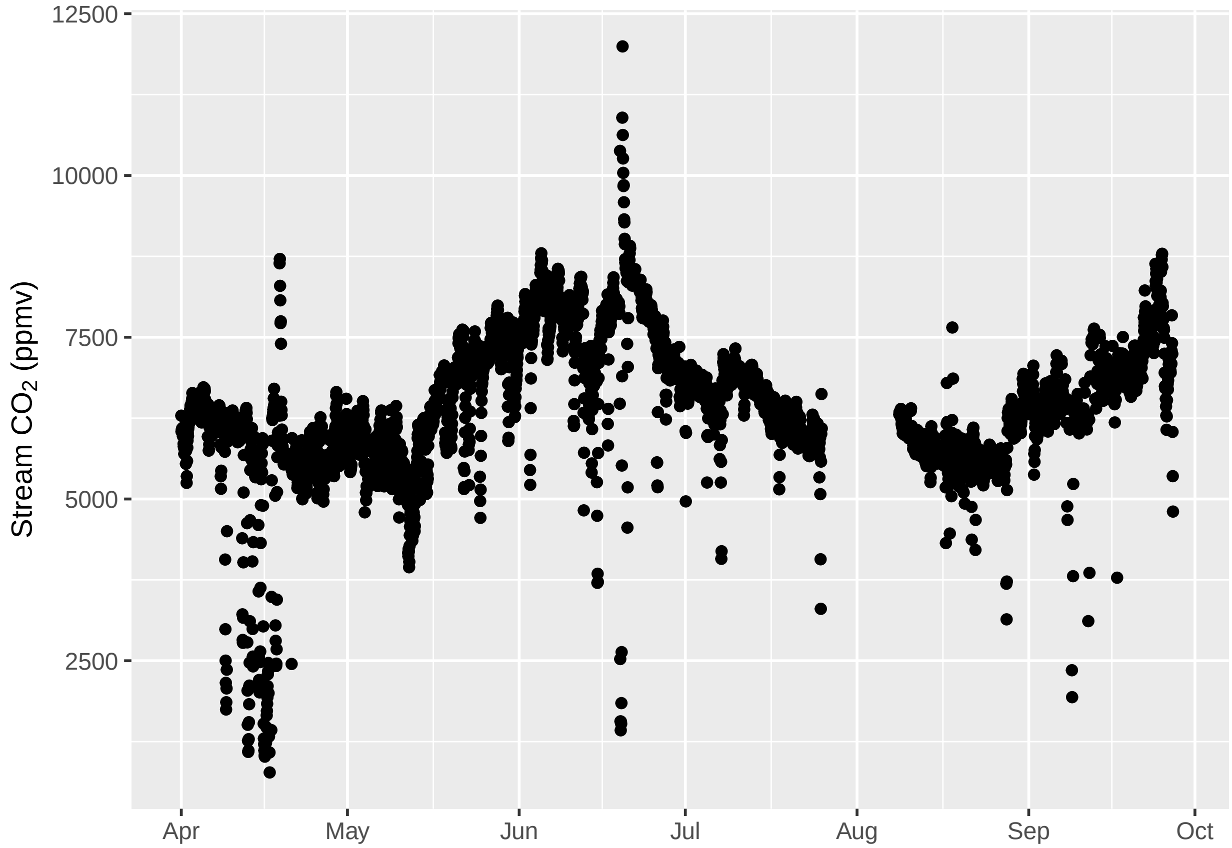

```
plot_do <- ggplot(data = Tac_all, aes(x = Timestamp))+
  geom_point(aes(y = DO.obs))+
  labs(x = "Time",
       y = expression(paste("DO (mg " ,L^{-1},")")))+
  ylim(0,8)+
  theme(axis.title.x = element_blank())+
  scale_x_datetime(date_breaks = "1 month",
                   labels = date_format("%b"))
plot_do
```




```
plot_wellCO2 <- ggplot(Tac_all,
                       aes(x = Timestamp))+
  geom_point(aes(y = wellCO2),
             color = 'black')+
  ylab(expression(paste("Well ", CO[2]," ", "(ppmv)")))+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  theme(axis.title.x = element_blank())+
  ylim(30000, 50000)
plot_wellCO2
```




```
plot_streamCO2 <- ggplot(Tac_all,
                         aes(x = Timestamp))+
  geom_point(aes(y = CO2_ppm),
             color = 'black')+
  ylab(expression(paste("Stream ", CO[2]," ", "(ppmv)")))+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  theme(axis.title.x = element_blank())
plot_streamCO2
```




```
Fig2 <- plot_grid(plot_rain, plot_Q, plot_do, plot_streamCO2, plot_wellCO2,
                  nrow = 5,
                  align = 'hv',
                  labels = 'auto', label_fontface = 'plain',
                  hjust = c(-9, -10, -10.5,-10,-9), vjust = 2)
Fig2
```

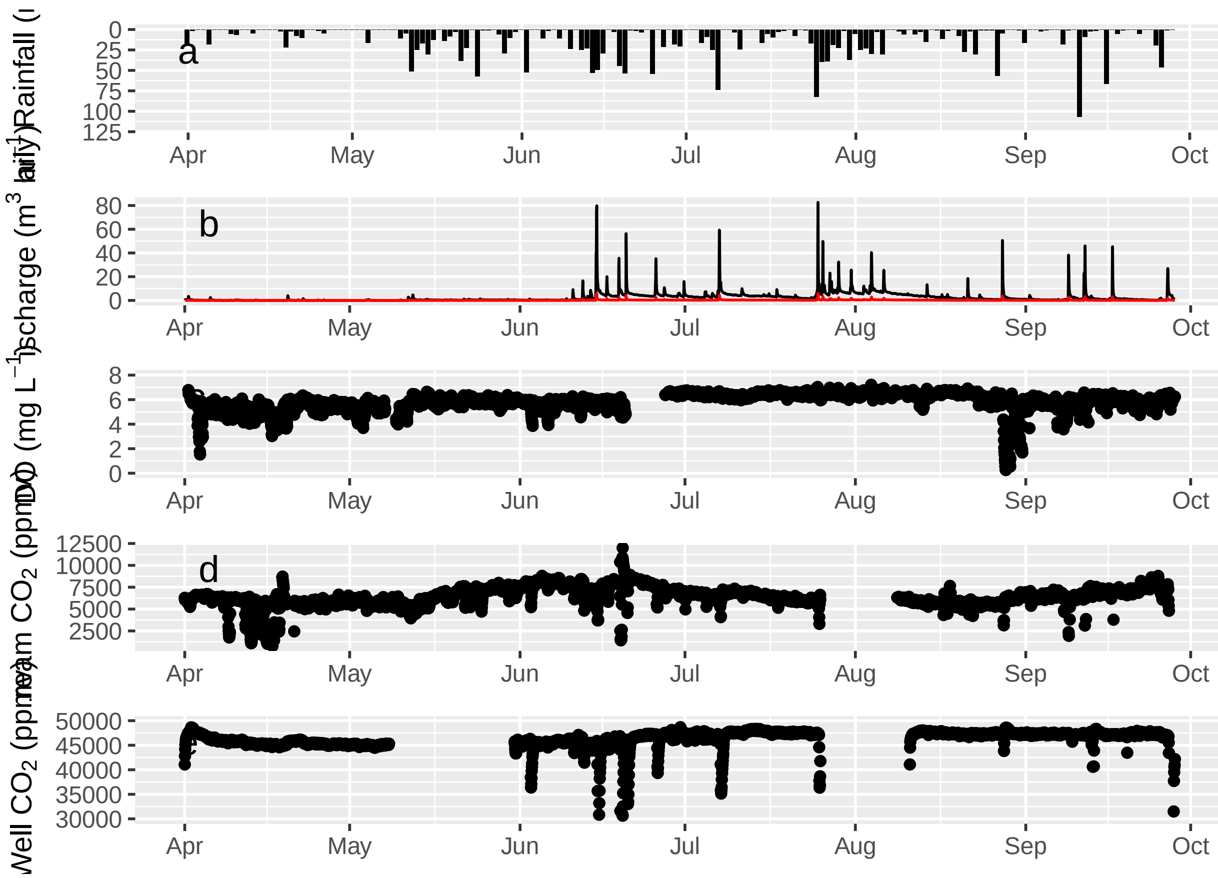


Figure 3- Stream and Well CO2

```
Fig3 <- ggplot(Tac_all,
               aes(x = wellCO2,
                   y = CO2_ppm,
                   color = month(Timestamp, abbr = TRUE, label = TRUE)))+
  geom_point(alpha = 0.7)+
  scale_color_viridis_d(name = element_blank())+
  labs(y = expression(paste('Stream ',italic(p),CO[2], ' (ppmv)')),
       x = expression(paste('Well ',italic(p),CO[2], ' (ppmv)')))
Fig3
```

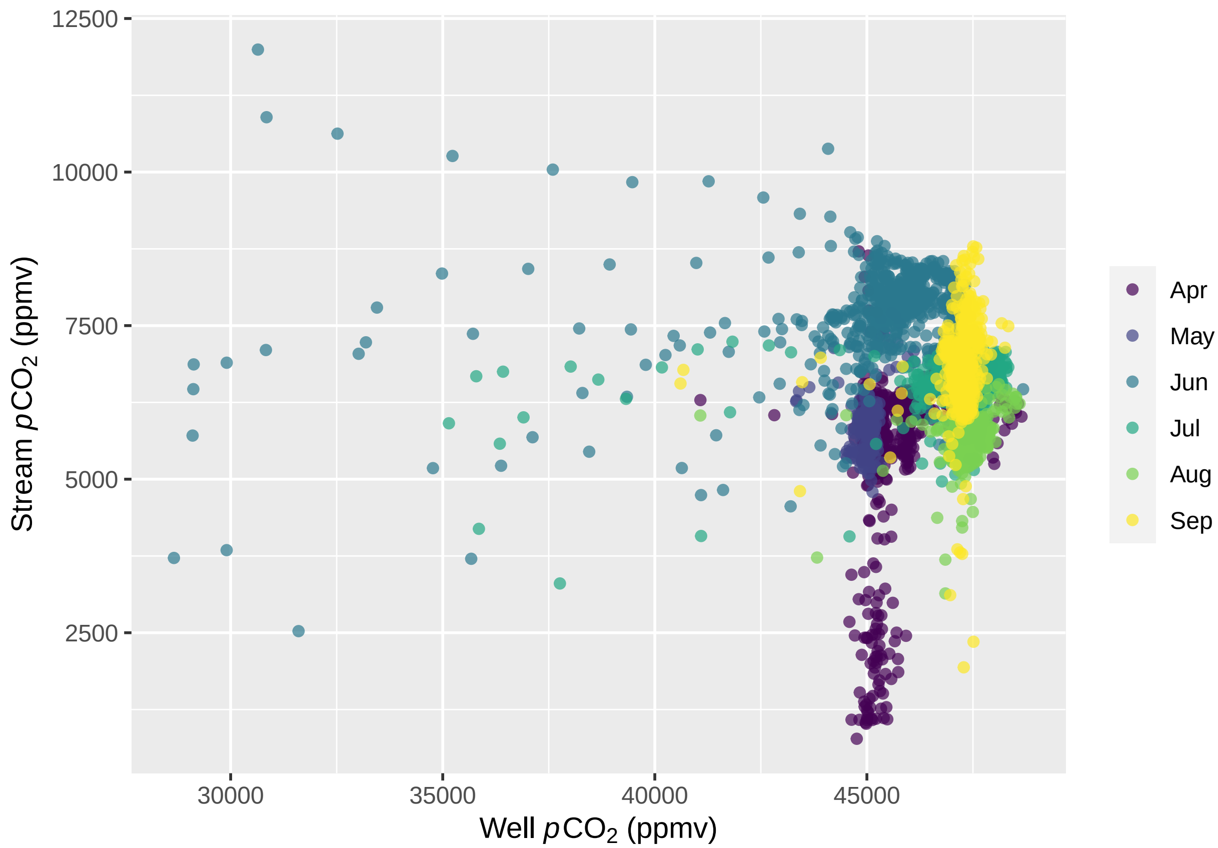


Figure 4- Departure

Here, we plot CO2 and O2 in respect to their atmospheric saturation concentrations. We calculate saturation concentrations for O2 using a barometric pressure model (Hall and Hotchkiss 2017) and for CO2 assuming atmopsheric CO2 of 400 ppm. We then explore the mean departure coordinates for the entire data collection period and for each month. The coordinates and slope inform processes that simultaneouly control O[2] and CO[2] in the stream.

```
Tac_departure <- Tac_all %>%
  mutate(
    # temperature-corrected Henry's Law constant (Plummer and Busenberg 1982)
    kH = 29.41*exp(-2400*(1/(273 + temp.water) - (1/298))),

    # in-stream CO2-aq calculations
    CO2_aq = ((CO2_ppm/1e6)/kH)*1000,       # aqueous CO2: mol CO2 m-3
    CO2_sat = ((400/1e6)/kH)*1000,

    ts = log((298.15 - temp.water)/(298.15 + temp.water)),    # function as part of O2 saturation, Hall
    u = 10^(8.10765 - (1750.3/(235 + temp.water))),           # function as part of O2 saturation

    Osat = exp(2.00907 +
                 (3.22014*ts) +
                 (4.0501*ts^2) +
                 (4.94457*ts^3) -
                 (0.256847*ts^4) +
```

```
                 (3.88767*ts^5)) *
      ((bp_mmHg - u)/(760 - u))*1.42905,

    CO2_dep = (CO2_aq - CO2_sat)*1000,   # umol/L
    O2_dep = ((DO.obs - Osat)/32)*1000)  # umol/L


Tac_dep_clean <- na.omit(Tac_departure)

Tac_dep_clean$month = lubridate::month(Tac_dep_clean$Timestamp,
                                       abbr = TRUE, label = TRUE)

# calculate mean and sd
Tac_mu <- c(mean(Tac_dep_clean$CO2_dep, na.rm = TRUE),
            mean(Tac_dep_clean$O2_dep, na.rm = TRUE))
Tac_sd <- c(sd(Tac_dep_clean$CO2_dep, na.rm = TRUE),
            sd(Tac_dep_clean$O2_dep, na.rm = TRUE))
# correlation and covariance matrix
Tac_corMat <- cor(cbind(Tac_dep_clean$CO2_dep,
                        Tac_dep_clean$O2_dep))
Tac_covMat <- cov(cbind(Tac_dep_clean$CO2_dep,
                        Tac_dep_clean$O2_dep))
# eigen values
Tac_evals <- eigen(Tac_covMat)$values
Tac_ell_length <- 2*sqrt(5.991*Tac_evals)

# run linear model
Tac_reg <- lmodel2(data = Tac_dep_clean,
                   O2_dep ~ CO2_dep,
                   nperm = 99)
```

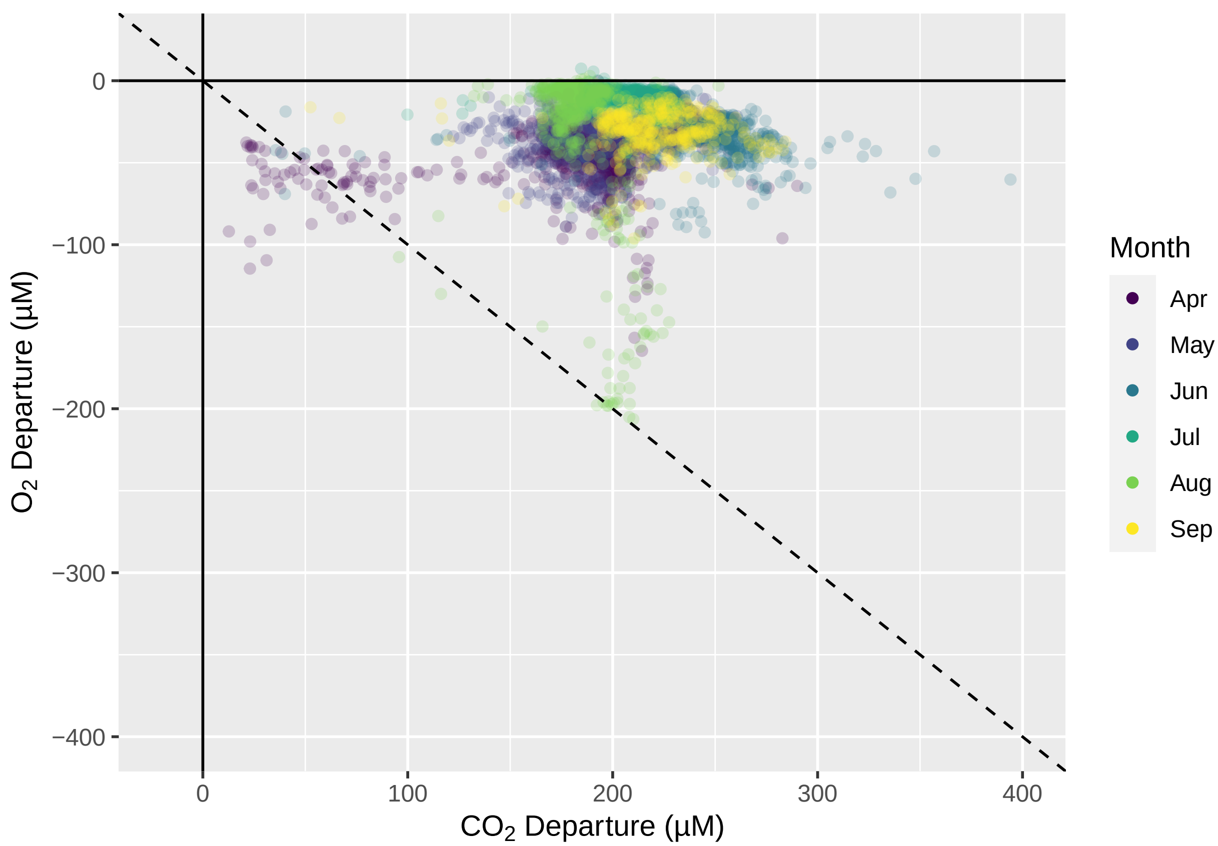
```
## RMA was not requested: it will not be computed.
```

```
# extract values from regression
Tac_inter = Tac_reg$regression.results[2,2]
Tac_slope = Tac_reg$regression.results[2,3]
Tac_correlation = Tac_reg$r

# create new data frame
Tac_metrics <- data.frame(meanCO2dep = Tac_mu[1],
                          meanO2dep = Tac_mu[2],
                          offset = Tac_mu[1] + Tac_mu[2],
                          EQ = 1/abs(Tac_slope),
                          width = Tac_ell_length[2],
                          stretch = Tac_ell_length[1])
Tac_metrics
```

```
##   meanCO2dep meanO2dep   offset       EQ    width  stretch
## 1   203.8241 -32.04432 171.7798 9.461728 128.9109 188.8561
```

```
# plot- all data points with 95% CI ellipsis
Fig4 <- ggplot(data = Tac_departure,
```

```
              aes(x = CO2_dep,
                  y = O2_dep,
                  color = factor(month(Timestamp))))+
  geom_point(alpha = 0.2)+
  geom_hline(yintercept = 0)+
  geom_vline(xintercept = 0)+
  geom_abline(intercept = 0, slope = -1, linetype = 2)+
  xlim(-20, 400)+
  ylim(-400, 20)+
  xlab(expression(paste(CO[2], " Departure (µM)")))+
  ylab(expression(paste(O[2], " Departure (µM)")))+
  guides(colour = guide_legend(override.aes = list(alpha = 1)))+
  scale_color_viridis_d(name = "Month",
                       labels = c("Apr", "May", "Jun", "Jul", "Aug", "Sep"))
Fig4
```




```
# summary for each month of the data
Tac_summary_mon <- Tac_dep_clean %>%
  group_by(month(Timestamp, label = TRUE, abbr = TRUE)) %>%
  dplyr::summarise(meanCO2_dep = mean(CO2_dep, na.rm = TRUE),
                   meanO2_dep = mean(O2_dep, na.rm = TRUE),
                   sdCO2_dep = sd(CO2_dep, na.rm = TRUE),
                   sdO2_dep = sd(O2_dep, na.rm = TRUE),
                   corMat = cor(cbind(CO2_dep, O2_dep)),
                   covMat = cor(cbind(CO2_dep, O2_dep))) %>%
```

```
  dplyr::mutate(offset = meanCO2_dep + meanO2_dep,
                evals_1 = eigen(covMat)$values[1],
                evals_2 = eigen(covMat)$values[2],
                ell_length_1 = 2*sqrt(5.991*evals_1),
                ell_length_2 = 2*sqrt(5.991*evals_2)) %>%
  dplyr::rename(month = `month(Timestamp, label = TRUE, abbr = TRUE)`) %>%
  group_by(month) %>%
  filter(row_number() %% 2==0)
```

```
## `summarise()` has grouped output by 'month(Timestamp, label = TRUE, abbr =
## TRUE)'. You can override using the `.groups` argument.
```

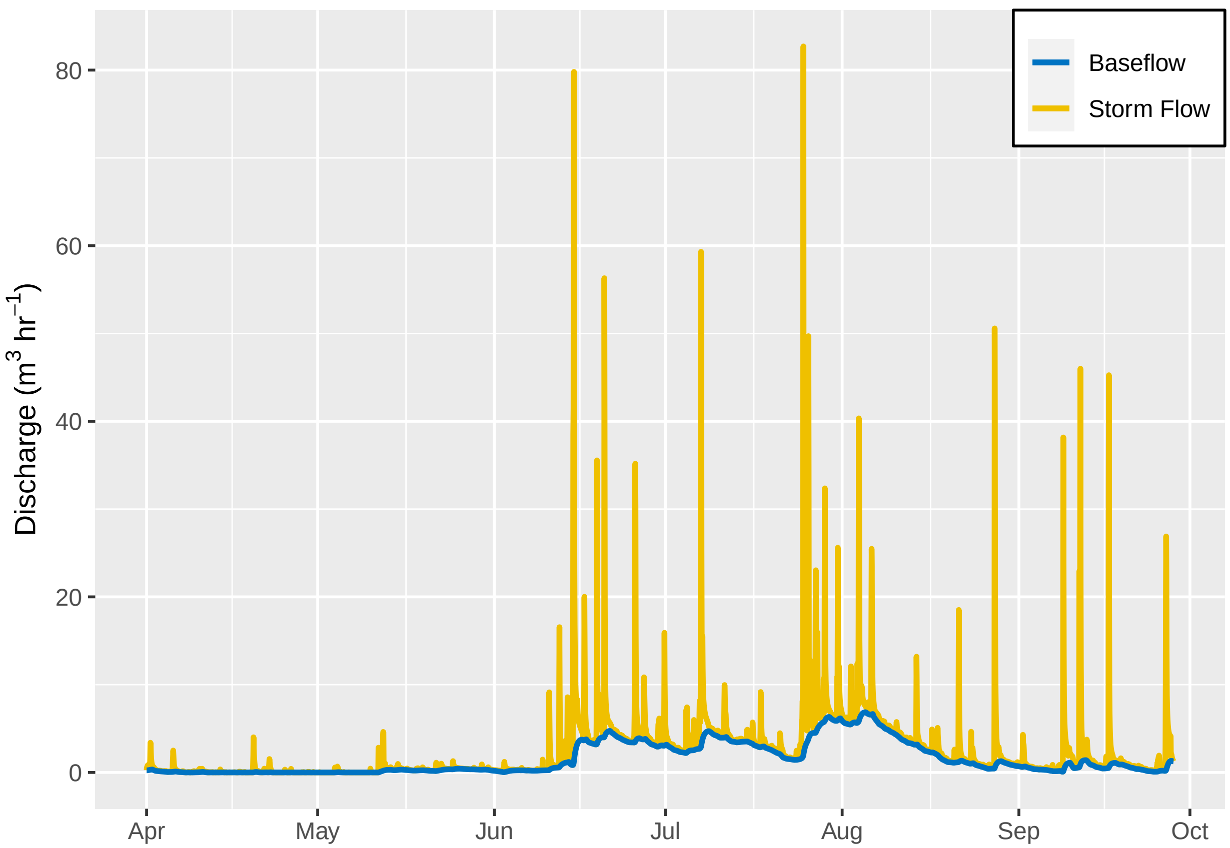
```
Tac_model_metrics <- Tac_dep_clean %>%
  select(Timestamp, CO2_dep, O2_dep) %>%
  dplyr::mutate(month = factor(month(Timestamp, abbr= TRUE, label = TRUE))) %>%
  nest(-month) %>%
  mutate(model = map(data,
                   ~lmodel2(O2_dep ~ CO2_dep, nperm = 99,
                            data = .)),
       tidy = map(model,
                  broom::tidy),
       glance = map(model,
                   broom::glance)) %>%
  unnest(tidy, glance) %>%
  select(month, method, term, estimate, r.squared) %>%
  mutate(r = -sqrt(r.squared)) %>%
  filter(method == 'OLS') %>%
  spread(term,
       estimate)
```

```
## RMA was not requested: it will not be computed.
##
## RMA was not requested: it will not be computed.
##
## RMA was not requested: it will not be computed.
##
## RMA was not requested: it will not be computed.
##
## RMA was not requested: it will not be computed.
##
## RMA was not requested: it will not be computed.
```

```
Tac_metrics_mon <- data.frame(month = Tac_summary_mon$month,
                              meanCO2_dep = Tac_summary_mon$meanCO2_dep,
                              meanO2_dep = Tac_summary_mon$meanO2_dep,
                              sdCO2_dep = Tac_summary_mon$sdCO2_dep,
                              sdO2_dep = Tac_summary_mon$sdO2_dep,
                              cor = Tac_summary_mon$corMat[,1],
                              offset = Tac_summary_mon$offset,
                              slope = 1/abs(Tac_model_metrics$Slope),
                              width = Tac_summary_mon$ell_length_2,
                              stretch = Tac_summary_mon$ell_length_1)
```

Figure 5- Hydrograph Separation

```
# baseflow separation approach
# see Fig S3 for the determination of alpha

# use alpha = 0.96
Tac_baseflow <- baseflows(Tac_all %>%
                            select(Date = Timestamp,
                                   Q = sumQ_m3_hr),
                          a = 0.96,
                          n.reflected = 30,
                          ts = 'daily') %>%
  mutate(bfi = bf/Q,
         stormflow = Q - bf,
         source = ifelse(bfi > 0.5,
                         'Baseflow',
                         'Storm Flow'),
         month = month(Date, label = TRUE, abbr = TRUE)) %>%
  rename(Timestamp = Date)
# bfi = fraction of baseflow calculated; bfi = 1 means all baseflow, 0 = all stormflow


# plot- baseflow on top of stream discharge
Fig5a <- ggplot(Tac_baseflow)+
  geom_line(aes(x = Timestamp,
                y = Q,
                color = 'Storm Flow'),
            size = 1)+
  geom_line(aes(x = Timestamp,
                y = bf,
                color = 'Baseflow'),
            size = 1)+
  scale_color_jco()+
  ylab(expression(paste('Discharge (', m^3, ' ',hr^-1,')')))+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  theme(axis.title.x = element_blank(),
        legend.position = c(1, 1),
        legend.justification = c(1, 1),
        legend.background = element_rect(color = 'black'),
        legend.title = element_blank())
Fig5a
```




```
# join hydrograph separation with pCO2 data
Tac_baseflow_hyst <- left_join(
  Tac_baseflow,
  Tac_all %>%
    select(Timestamp, CO2_ppm, sumQ_m3_hr, GW_Q))
```

```
## Joining, by = "Timestamp"
```

```
shapiro.test(Tac_baseflow_hyst$CO2_ppm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Tac_baseflow_hyst$CO2_ppm
## W = 0.91907, p-value < 2.2e-16
```

```
pco2_source_art <- art(data = Tac_baseflow_hyst %>%
                         dplyr::mutate(source_f = factor(source)) %>%
                         dplyr::select(month, source_f, CO2_ppm) %>%
                         tidyr::drop_na(),
                       CO2_ppm ~ month * source_f)

anova(pco2_source_art)
```

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Anova Table (Type III tests)
## Model: No Repeated Measures (lm)
## Response: art(CO2_ppm)
##
##                  Df Df.res F value     Pr(>F)
## 1 month           5   3717 478.189 < 2.22e-16 ***
## 2 source_f        1   3717 660.905 < 2.22e-16 ***
## 3 month:source_f  5   3717  41.348 < 2.22e-16 ***
## ---
## Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
art.con(pco2_source_art,
        'source_f')
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  contrast              estimate   SE   df t.ratio p.value
##  Baseflow - Storm Flow     1146 44.6 3717  25.708  <.0001
##
## Results are averaged over the levels of: month
```

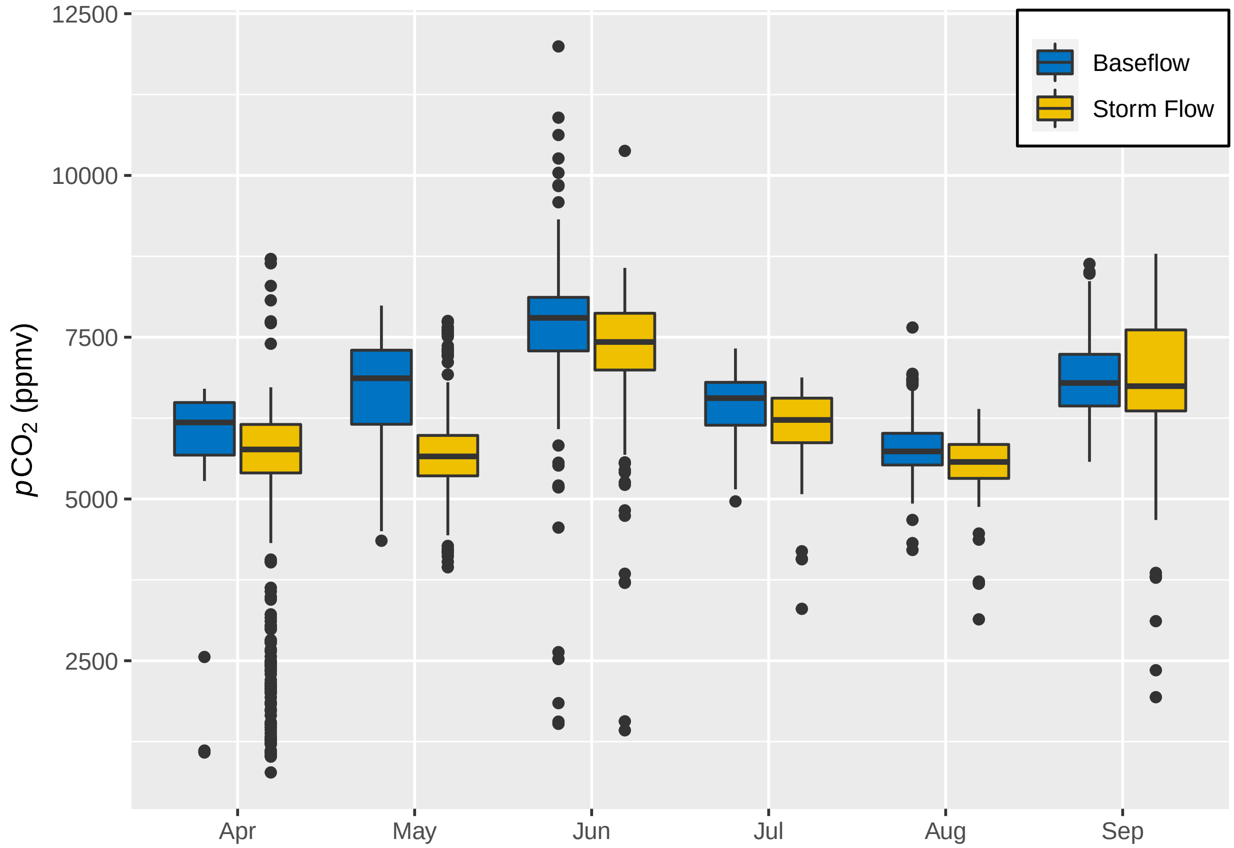
```
art.con(pco2_source_art,
        'month',
        method = "pairwise")
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  contrast  estimate   SE   df t.ratio p.value
##  Apr - May   -816.4 46.0 3717 -17.767  <.0001
##  Apr - Jun  -2178.7 50.5 3717 -43.125  <.0001
##  Apr - Jul  -1086.8 63.2 3717 -17.196  <.0001
##  Apr - Aug    -71.4 68.1 3717  -1.048  0.9017
##  Apr - Sep  -1536.3 53.8 3717 -28.567  <.0001
##  May - Jun  -1362.3 44.3 3717 -30.774  <.0001
##  May - Jul   -270.4 58.3 3717  -4.636  0.0001
##  May - Aug    745.1 63.6 3717  11.709  <.0001
##  May - Sep   -719.9 48.0 3717 -15.012  <.0001
##  Jun - Jul   1091.9 62.0 3717  17.616  <.0001
##  Jun - Aug   2107.4 67.0 3717  31.452  <.0001
##  Jun - Sep    642.5 52.3 3717  12.273  <.0001
##  Jul - Aug   1015.4 77.0 3717  13.185  <.0001
##  Jul - Sep   -449.5 64.7 3717  -6.950  <.0001
##  Aug - Sep  -1464.9 69.5 3717 -21.080  <.0001
##
## Results are averaged over the levels of: source_f
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
contrast(emmeans(artlm(pco2_source_art, 'month:source_f'),
                 ~ month:source_f),
         method = 'pairwise', interaction = TRUE)
```

```
##  month_pairwise source_f_pairwise     estimate  SE   df t.ratio p.value
##  Apr - May      Baseflow - Storm Flow     -784 119 3717  -6.611  <.0001
##  Apr - Jun      Baseflow - Storm Flow      240 130 3717   1.838  0.0661
##  Apr - Jul      Baseflow - Storm Flow      396 163 3717   2.430  0.0151
##  Apr - Aug      Baseflow - Storm Flow      576 176 3717   3.277  0.0011
##  Apr - Sep      Baseflow - Storm Flow      744 139 3717   5.361  <.0001
##  May - Jun      Baseflow - Storm Flow     1024 114 3717   8.960  <.0001
##  May - Jul      Baseflow - Storm Flow     1181 151 3717   7.842  <.0001
##  May - Aug      Baseflow - Storm Flow     1360 164 3717   8.283  <.0001
##  May - Sep      Baseflow - Storm Flow     1528 124 3717  12.348  <.0001
##  Jun - Jul      Baseflow - Storm Flow      157 160 3717   0.980  0.3273
##  Jun - Aug      Baseflow - Storm Flow      337 173 3717   1.946  0.0517
##  Jun - Sep      Baseflow - Storm Flow      505 135 3717   3.734  0.0002
##  Jul - Aug      Baseflow - Storm Flow      180 199 3717   0.904  0.3658
##  Jul - Sep      Baseflow - Storm Flow      348 167 3717   2.083  0.0373
##  Aug - Sep      Baseflow - Storm Flow      168 179 3717   0.936  0.3492
```

```r
# plot CO2 vs stormflow
Fig5b <- ggplot(Tac_baseflow_hyst,
                aes(x = month,
                    y = CO2_ppm,
                    fill = source))+
  geom_boxplot()+
  ylab(expression(paste(italic(p), CO[2], ' (ppmv)')))+
  scale_fill_jco()+
  theme(axis.title.x = element_blank(),
        legend.position = c(1, 1),
        legend.justification = c(1, 1),
        legend.background = element_rect(color = 'black'),
        legend.title = element_blank())
Fig5b
```




```
# combine plots
Fig5 <- plot_grid(Fig5a, Fig5b,
                  align = 'v', nrow = 2,
                  labels = 'auto', label_fontface = 'plain',
                  hjust = -10)
Fig5
```

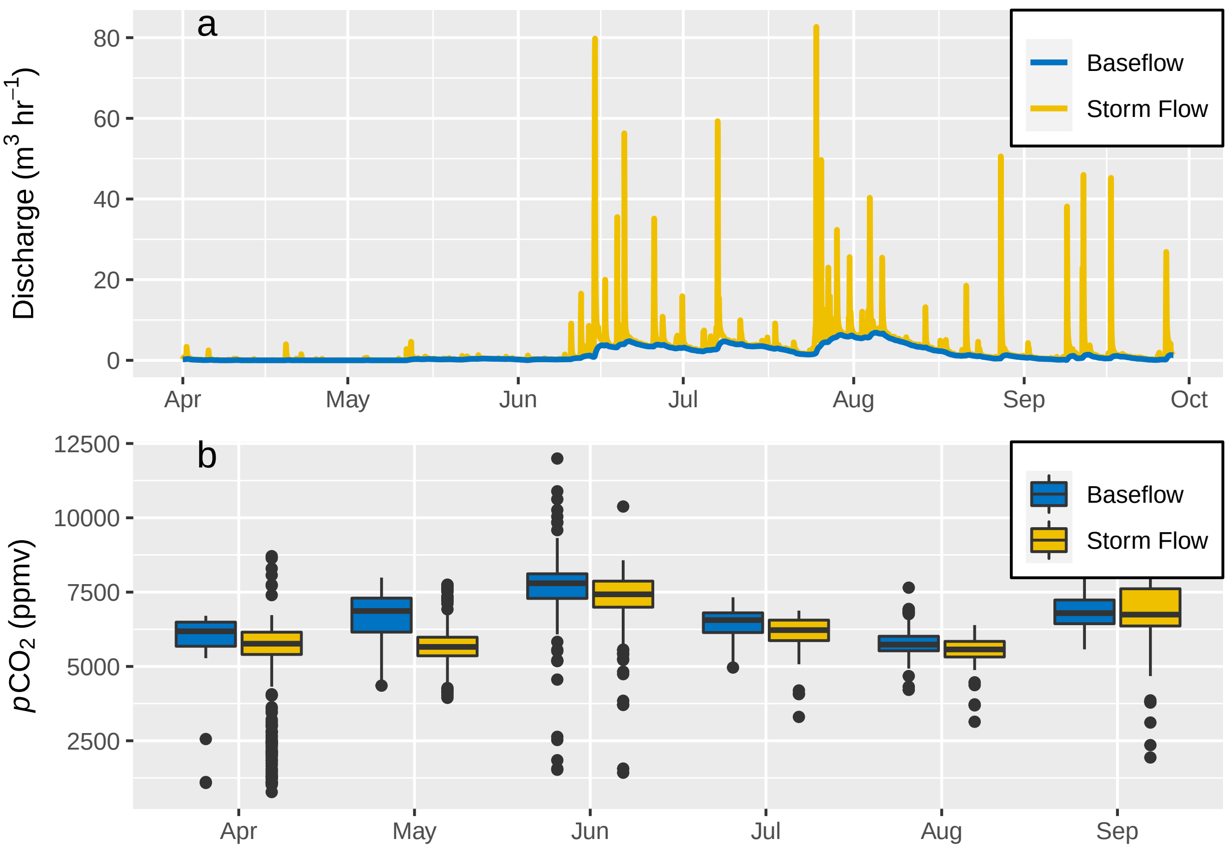


Figure 6- Areal Fluxes

```
Tac_length = 1587.4        # total length of the main-stem Taconazo
Tac_reach = 75             # m
Tac_width_upper_dry = 1.4 # m
Tac_width_upper_wet = 1.8 # m
Tac_width_lower_dry = 1.5 # m
Tac_width_lower_wet = 2.8 # m
Tac_slope = 0.0024         # m/m

# calculate mean temperature
temp_mean = mean(Tac_all$temp.water,
                 na.rm = TRUE)

# estimate Schmidt numbers
Sc_prop = 3545.60 - (203.41*temp_mean) + (4.78*(temp_mean^2)) - (0.0404*(temp_mean^3))
Sc_CO2 = 1686.08 - (89.66*temp_mean) + (2.07*(temp_mean^2)) - (0.018*(temp_mean^3))
Sc_O2 = 1568 - (86.04*temp_mean) + (2.142*(temp_mean^2)) - (0.0216*(temp_mean^3))

K_prop_wet = 10.2     # gas exchange coefficient for wet season propane (d^-1); 95% CI = 4.2
K_prop_dry = 27.2     # gas exchange coefficient for dry season propane (d^-1); 95% CI = 3.6

K_CO2_dry = K_prop_dry*(Sc_prop/Sc_CO2)^-0.5  # dry season gas exchange coefficient for CO2, d^-1
K_CO2_wet = K_prop_wet*(Sc_prop/Sc_CO2)^-0.5  # wet gas exchange coefficient for CO2, d^-1
```

```r
# Convert to units for O2
K_O2_dry = K_prop_dry*(Sc_prop/Sc_O2)^-0.5   # gas exchange coefficient for O2, d^-1
K_O2_wet = K_prop_wet*(Sc_prop/Sc_O2)^-0.5   # gas exchange coefficient for O2, d^-1
```

```r
Tac_gas_flux <- Tac_all %>%
  # get the necessary data
  select(Timestamp,
         temp.water,
         CO2_ppm,
         DO.obs, bp_mmHg,
         sumQ_m3_hr,
         mean_depth) %>%
  mutate(

    mean_depth = mean_depth*0.7,

    # Henry's Law constant based on temperature correction (Plummer and Busenberg 1982)
    kH = 29.41*exp(-2400*(1/(273 + temp.water) - (1/298))),
    # calculate aqueous CO2 and CO2-sat (mol CO2 m-3)
    CO2_aq = ((CO2_ppm/1e6)/kH)*1000,
    CO2_sat = ((400/1e6)/kH)*1000,
    # calculate CO2 efflux (mol CO2 m-2 h-1)
    CO2_efflux = ifelse(month(Timestamp) == '4',
                        (CO2_aq - CO2_sat)*(K_CO2_dry/24)*mean_depth,
                        (CO2_aq - CO2_sat)*(K_CO2_wet/24)*mean_depth
    ),

    # calculate NEP
    ts = log((298.15 - temp.water)/(298.15 + temp.water)),   # function as part of O2 saturation
    u = 10^(8.10765 - (1750.3/(235 + temp.water))),          # function as part of O2 saturation

    # calculate O2 saturation
    Osat = exp(2.00907 +
               (3.22014*ts) +
               (4.0501*ts^2) +
               (4.94457*ts^3) -
               (0.256847*ts^4) +
               (3.88767*ts^5)) *
      ((bp_mmHg - u)/(760 - u))*1.42905,

    # hourly NEP (mol C m-2 h-1)
    direct_met_mol = ifelse(month(Timestamp) == '4',
                            (diff(DO.obs) - ((K_O2_dry/24)*(Osat - DO.obs))*(1/32))*-1*mean_depth,
                            (diff(DO.obs) - ((K_O2_wet/24)*(Osat - DO.obs)) * (1/32))*-1*mean_depth
    )
  )
```

```r
Tac_GWCO2 <- Tac_all %>%
  select(Timestamp,
         pH,
         temp.water, wellCO2,
         sumQ_m3_hr) %>%
  left_join(Tac_baseflow_hyst) %>%
```

```
  mutate(

    # temperature-corrected Henry's Law constant (Plummer and Busenberg 1982)
    kH = 29.41*exp(-2400*(1/(273 + temp.water) - (1/298))),

    # CO2-well
    wellCO2_aq = ((wellCO2/1e6)/kH)*1000,  # well aqueous CO2: mol CO2 m-3

    # Groundwater discharge (m3 h-1)
    GW_Q = ifelse(month(Timestamp) == 'April',    # calculate total groundwater discharge (m3/h)
                  (0.1779*sumQ_m3_hr),            # 17.79% of stream Q in dry season (April); m3/h
                  (0.0715*sumQ_m3_hr)),           # 7.15% of stream Q in wet season; m3/h

    # groundwater velocity (m h-1); divide GW discharge by reach area (length * width)
    GW_v = ifelse(month(Timestamp) == 'April',
                  GW_Q/(Tac_reach * Tac_width_lower_dry),
                  GW_Q/(Tac_reach * Tac_width_lower_wet)
    ),

    # GWCO2 flux (mol CO2 m-2 h-1)
    GWCO2 = GW_v * wellCO2_aq
  )
```

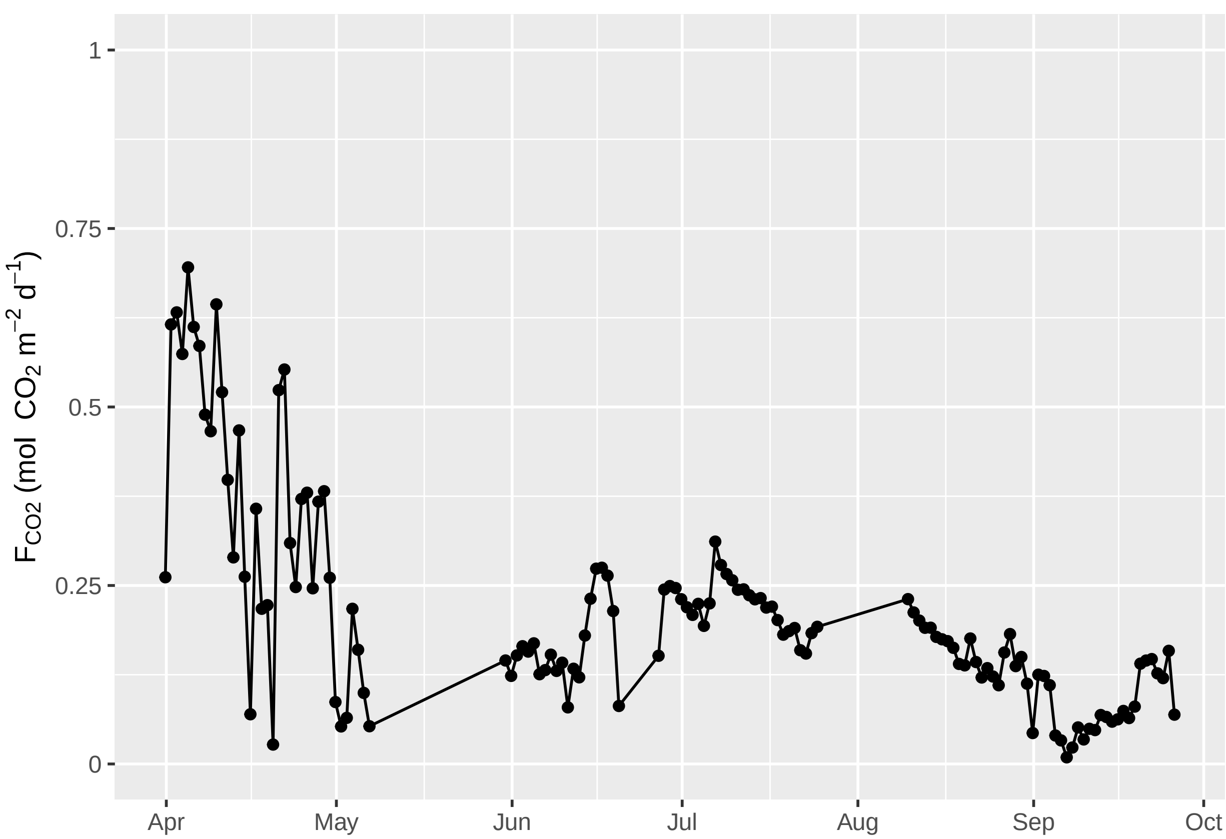
```
## Joining, by = c("Timestamp", "sumQ_m3_hr")
```

```
# aggregate to the daily time-step

Tac_fluxes <- left_join(Tac_gas_flux,
                        Tac_GWCO2,
                        by = 'Timestamp') %>%
  select(Timestamp, GWCO2, CO2_efflux, direct_met_mol) %>%
  group_by(day = date(Timestamp)) %>%
  summarise(dayGW = sum(GWCO2, na.rm = TRUE),
            dayF = sum(CO2_efflux, na.rm = TRUE),
            dayNEP = sum(direct_met_mol, na.rm = TRUE)) %>%
  filter(dayNEP < 1.5,
         dayGW > 0,
         dayF > 0)

Tac_fluxes %>%
  group_by(month(day)) %>%
  summarise(F_med = median(dayF),
            F_sd = sd(dayF),
            GW_med = median(dayGW),
            GW_sd = sd(dayGW),
            NEP_med = median(dayNEP),
            NEP_sd = sd(dayNEP))
```

```
## # A tibble: 6 x 7
##   `month(day)`  F_med  F_sd    GW_med   GW_sd NEP_med NEP_sd
##          <dbl>  <dbl> <dbl>     <dbl>   <dbl>   <dbl>  <dbl>
## 1            4 0.381  0.173  0.000876 0.00251  0.128  0.114
## 2            5 0.0931 0.0592 0.0000701 0.00189  0.0311 0.0329
```

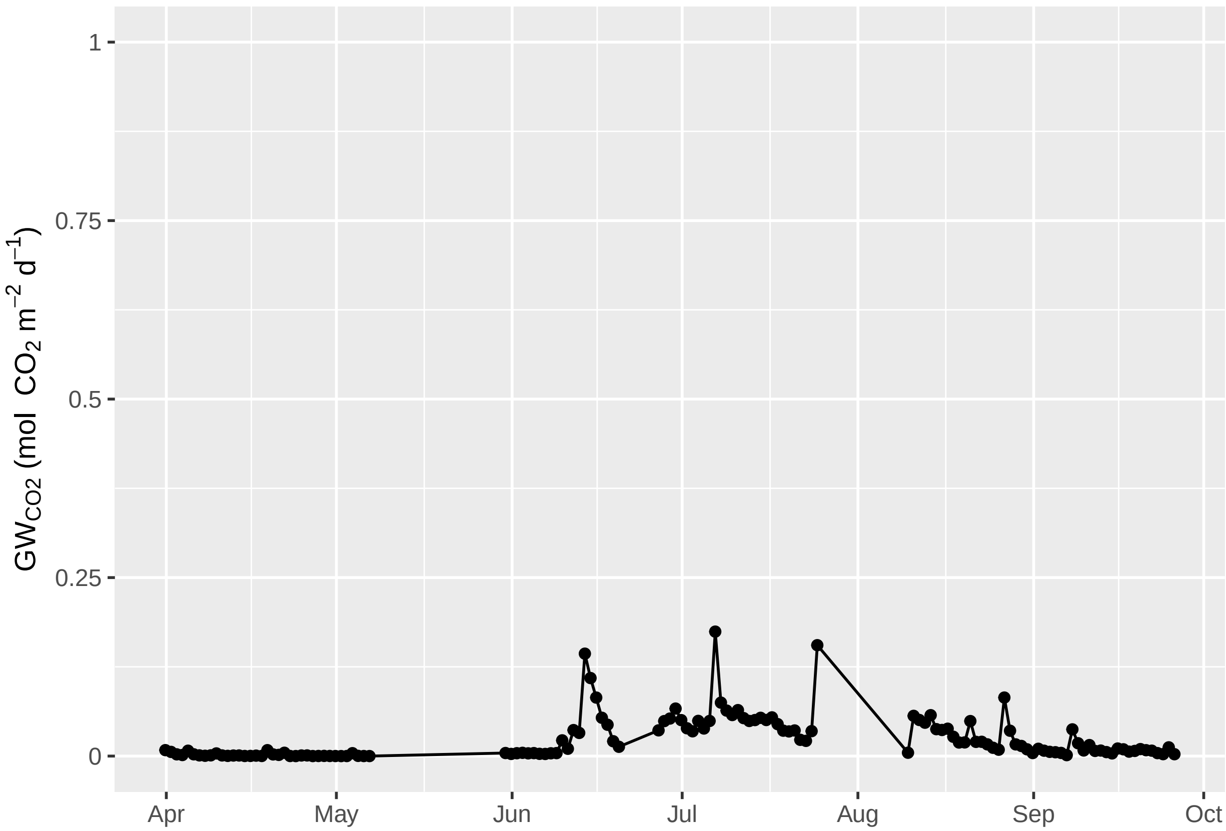
```
## 3              6 0.155  0.0598 0.0214    0.0371   0.0281 0.277
## 4              7 0.220  0.0365 0.0493    0.0352   0.0150 0.0498
## 5              8 0.159  0.0330 0.0235    0.0199   0.0121 0.196
## 6              9 0.0672 0.0433 0.00736   0.00701  0.0241 0.0759
```

```
Fig6_F <- ggplot(Tac_fluxes %>%
                   filter(dayF > 0),
                 aes(x = as.POSIXct(day), y = dayF))+
  geom_point()+
  geom_line()+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1),
                     labels = c(0, 0.25, 0.5, 0.75, 1),
                     limits = c(0, 1))+
  labs(x = element_blank(),
       y = expression(paste(F[CO2], ' (mol  ', CO[2],' ', m^-2,' ', d^-1, ')')))
Fig6_F
```

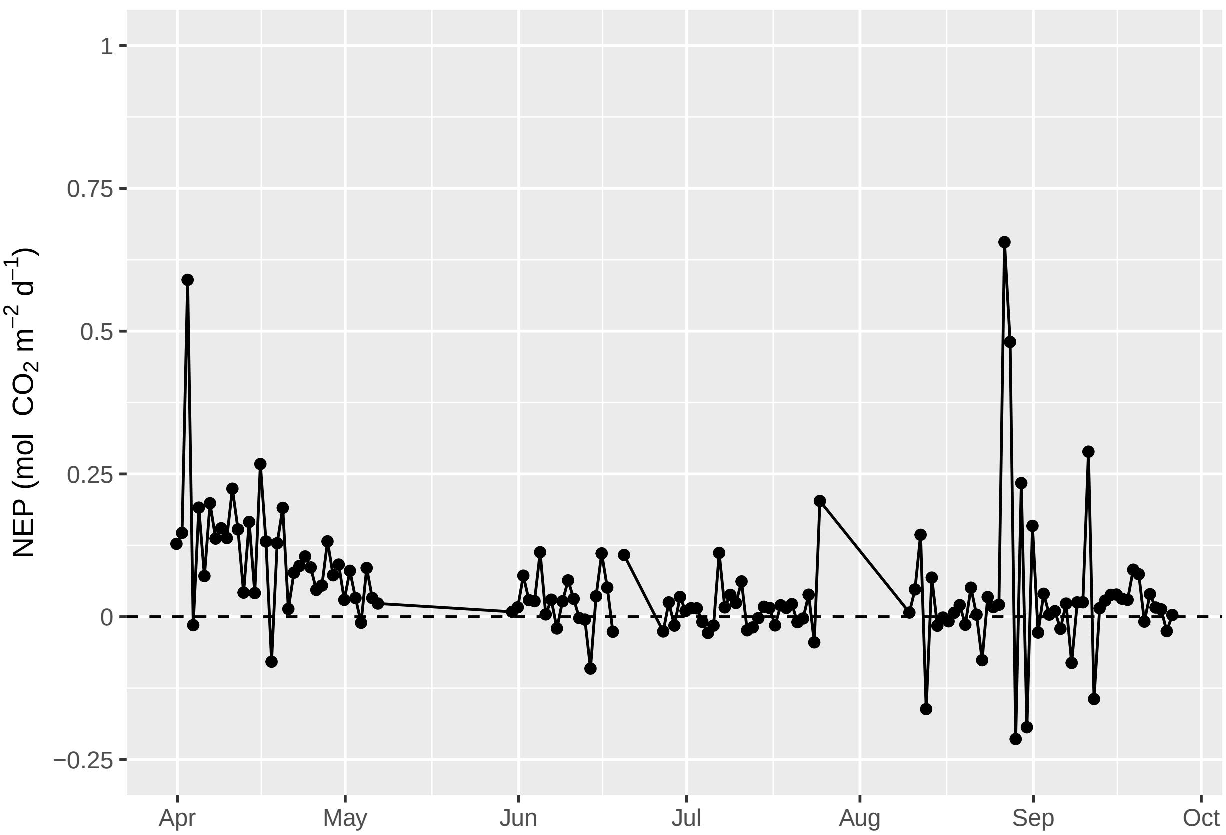



```
Fig6_GW <- ggplot(Tac_fluxes %>%
                    mutate(GWCO2_mean = rowMeans(dplyr::select(.,starts_with('dayGW')))),
                  aes(x = as.POSIXct(day)))+
  geom_point(aes(y = GWCO2_mean
  ))+
  geom_line(aes(y = GWCO2_mean
```

```
  ))+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  scale_y_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1),
                     labels = c(0, 0.25, 0.5, 0.75, 1),
                     limits = c(0, 1))+
  labs(x = element_blank(),
       y = expression(paste(GW[CO2], ' (mol  ', CO[2],' ', m^-2,' ', d^-1, ')')))
Fig6_GW
```

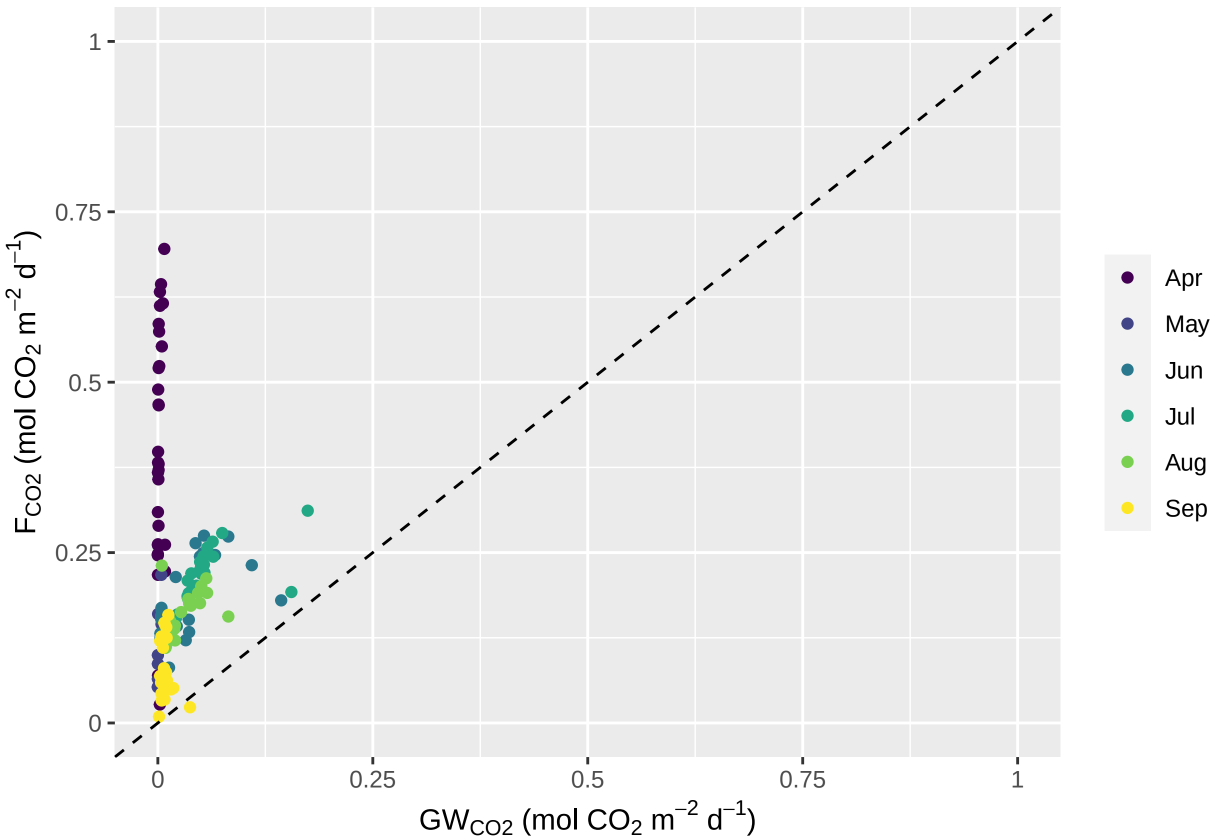



```
Fig6_NEP <- ggplot(Tac_fluxes %>%
                     filter(dayNEP < 1.5),
                   aes(x = as.POSIXct(day), y = dayNEP))+
  geom_point()+
  geom_line()+
  geom_hline(yintercept = 0, linetype = 'dashed')+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  scale_y_continuous(breaks = c(-0.25, 0, 0.25, 0.5, 0.75, 1),
                     labels = c(-0.25, 0, 0.25, 0.5, 0.75, 1),
                     limits = c(-0.25, 1))+
  labs(x = element_blank(),
       y = expression(paste('NEP (mol  ', CO[2],' ', m^-2,' ', d^-1, ')')))
Fig6_NEP
```

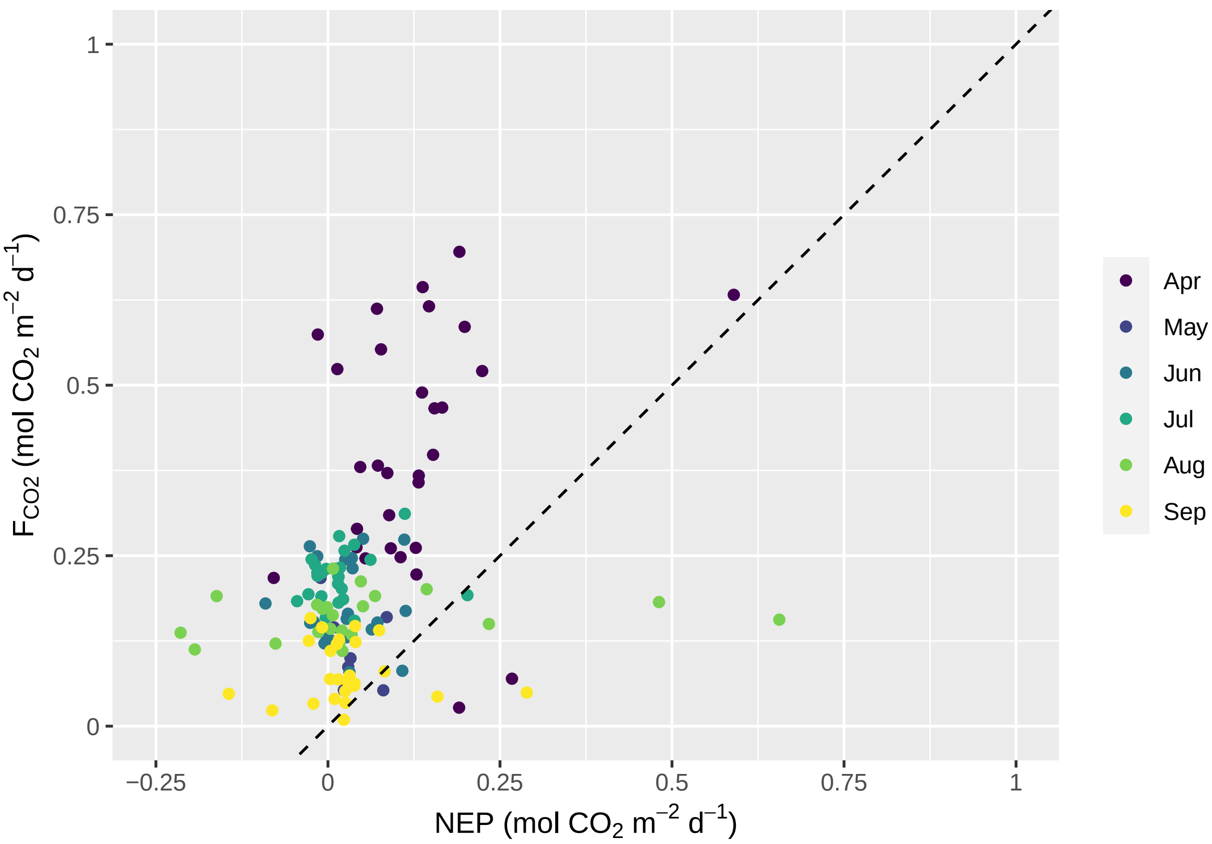



```
Fig6a <- plot_grid(Fig6_GW,
                   Fig6_NEP,
                   Fig6_F,
                   align = 'v', nrow = 3,
                   labels = 'auto', label_fontface = 'plain',
                   hjust = -9)


Fig6d <- ggplot(Tac_fluxes %>%
                  filter(dayGW > 0,
                         dayF > 0),
                aes(x = dayGW,
                    y = dayF,
                    color = month(day, abbr = TRUE, label = TRUE)))+
  geom_point()+
  geom_abline(intercept = 0, slope = 1,
              linetype = 'dashed')+
  labs(x = expression(paste(GW[CO2], ' (mol ', CO[2], ' ', m^-2,' ', d^-1, ')')),
       y = expression(paste(F[CO2], ' (mol ', CO[2], ' ',m^-2,' ', d^-1, ')')))+
  scale_x_continuous(limits = c(0, 1),
                     breaks = c(0,0.25,0.5,0.75,1),
                     labels = c(0,0.25,0.5,0.75,1))+
  scale_y_continuous(limits = c(0, 1),
                     breaks = c(0,0.25,0.5,0.75,1),
                     labels = c(0,0.25,0.5,0.75,1))+
  scale_color_viridis_d(name = element_blank())
```

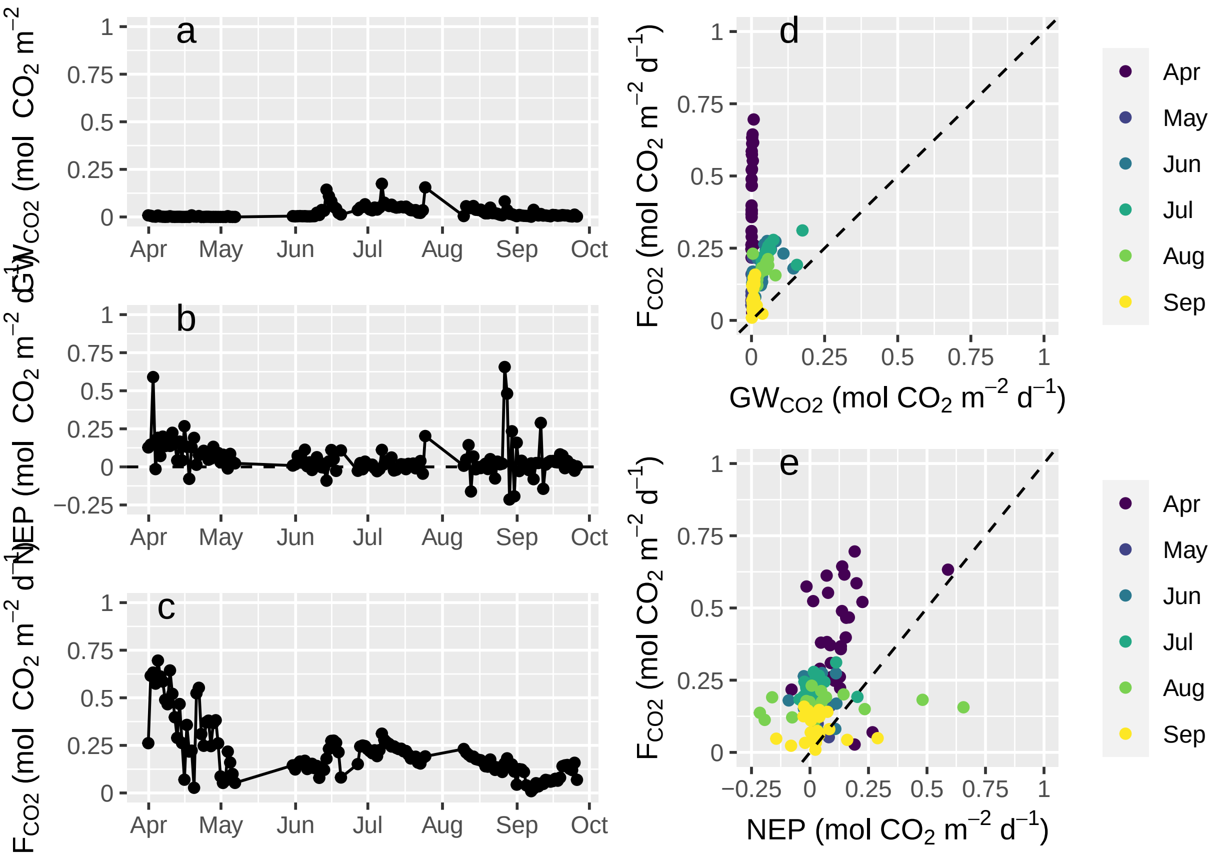
```
Fig6d
```




```
Fig6e <- ggplot(Tac_fluxes %>%
                  filter(dayF > 0,
                         dayNEP < 1.5),
                aes(x = dayNEP,
                    y = dayF,
                    color = month(day, abbr = TRUE, label = TRUE)))+
  geom_point()+
  geom_abline(intercept = 0, slope = 1,
              linetype = 'dashed')+
  labs(x = expression(paste('NEP (mol ', CO[2],' ', m^-2,' ', d^-1, ')')),
       y = expression(paste(F[CO2], ' (mol ', CO[2], ' ',m^-2,' ', d^-1, ')')))+
  scale_color_viridis_d(name = element_blank())+
  scale_x_continuous(limits = c(-0.25, 1),
                     breaks = c(-0.25, 0,0.25,0.5,0.75,1),
                     labels = c(-0.25, 0,0.25,0.5,0.75,1))+
  scale_y_continuous(limits = c(0, 1),
                     breaks = c(0,0.25,0.5,0.75,1),
                     labels = c(0,0.25,0.5,0.75,1))
Fig6e
```




```
Fig6b <- plot_grid(Fig6d, Fig6e,
                   nrow = 2, align = 'v',
                   labels = c('d', 'e'), label_fontface = 'plain',
                   hjust = -8)


Fig6 <- plot_grid(Fig6a, Fig6b,
                  align = 'hv')
Fig6
```




```
summary(lm(
  data = Tac_fluxes,
  dayF ~ dayNEP)
)
```

```
##
## Call:
## lm(formula = dayF ~ dayNEP, data = Tac_fluxes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26106 -0.07823 -0.02505  0.04274  0.45999
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.19652    0.01301  15.103  < 2e-16 ***
## dayNEP       0.20449    0.07669   2.666  0.00862 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.141 on 133 degrees of freedom
## Multiple R-squared:  0.05074,    Adjusted R-squared:  0.04361
## F-statistic:  7.11 on 1 and 133 DF,  p-value: 0.008619
```

```r
summary(lm(
  data = Tac_fluxes,
  dayF ~ dayGW)
)
```

```
##
## Call:
## lm(formula = dayF ~ dayGW, data = Tac_fluxes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20111 -0.08685 -0.03017  0.03748  0.48557
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21048    0.01574  13.376   <2e-16 ***
## dayGW       -0.06226    0.41032  -0.152     0.88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1447 on 133 degrees of freedom
## Multiple R-squared:  0.0001731,  Adjusted R-squared:  -0.007344
## F-statistic: 0.02303 on 1 and 133 DF,  p-value: 0.8796
```

```r
Tac_fluxes %>%
  mutate(day_F_all = ifelse(month(day) == 'April',
                            dayF * 3008,           # dry season mol CO2 per day
                            dayF * 5022)) %>%      # wet season mol CO2 per day
  summarise(mean_F_all = mean(day_F_all),
            sd_F_all = sd(day_F_all))          # mean daily CO2 loss (mol CO2 per day)
```

```
## # A tibble: 1 x 2
##   mean_F_all sd_F_all
##        <dbl>    <dbl>
## 1      1050.     724.
```

```r
Tac_gas_flux %>%
  mutate(C_export = sumQ_m3_hr * CO2_aq) %>%              # mol CO2 per hour
  group_by(day = date(Timestamp)) %>%                     # mol CO2 per day
  summarise(daily_export = sum(C_export)) %>%             # daily C export (mol C per day)
  summarise(mean_export = mean(daily_export, na.rm = TRUE),
            sd_export = sd(daily_export, na.rm = TRUE)) # mean daily export mol CO2 per day
```

```
## # A tibble: 1 x 2
##   mean_export sd_export
##         <dbl>     <dbl>
## 1        8.23      9.97
```

Figure 7- GW on pH

Finally, we determine the effect of groundwater DIC inputs to the reach as drivers of pH decreases, both at seasonal and episodic scales.

```r
Tac_GWCO2_day <- Tac_GWCO2 %>%
  group_by(day = date(Timestamp)) %>%
  summarise(meanpH = mean(pH, na.rm = TRUE),
            sdpH = sd(pH, na.rm = TRUE),
            dayGW = sum(GWCO2, na.rm = TRUE))

# remove low pH outliers
outliers <- boxplot.stats(Tac_GWCO2_day$meanpH)$out
# below 4.8 and above 5.91

Tac_GWCO2_day <- Tac_GWCO2_day[-which(Tac_GWCO2_day$meanpH %in% outliers),]

summary(lm(data = Tac_GWCO2_day %>%
             filter(dayGW > 0,
                    meanpH > 4.75),
           meanpH ~ log10(dayGW))
)
```

```
##
## Call:
## lm(formula = meanpH ~ log10(dayGW), data = Tac_GWCO2_day %>%
##     filter(dayGW > 0, meanpH > 4.75))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55538 -0.07368  0.03558  0.08386  0.25283
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.18175    0.02831 183.044  < 2e-16 ***
## log10(dayGW) -0.07154    0.01231  -5.811 4.47e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1242 on 131 degrees of freedom
## Multiple R-squared:  0.205,  Adjusted R-squared:  0.1989
## F-statistic: 33.77 on 1 and 131 DF,  p-value: 4.47e-08
```

```r
Fig7 <- ggplot(Tac_GWCO2_day %>%
                 filter(dayGW > 0),
               aes(x = dayGW,
                   y = meanpH))+
  geom_point(aes(color = month(day, label = TRUE, abbr = TRUE)))+
  geom_errorbar(aes(ymin = meanpH - sdpH,
                    ymax = meanpH + sdpH,
                    color = month(day, label = TRUE, abbr = TRUE)))+
  geom_abline(color = 'black',
              intercept = 5.20, slope = -0.06)+
  #ylim(3.8, 6)+
  scale_x_log10(breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x)),
                #limits = c(1e-5, 10^(-0.5))
  )+
```

```
  scale_color_viridis_d(name = 'Month')+
  xlab(expression(paste('Groundwater Flux (mol ', CO[2],' ',m^-2,' ', d^-1, ")")))+
  ylab('Mean Daily pH')+
  annotate(x = 0.1, y = 4, 'text', label = expression(italic("p < 0.01")))+
  annotate(x = 0.1, y = 4.2, 'text', label = expression(italic(paste(R^2,"= 0.20"))))
Fig7
```

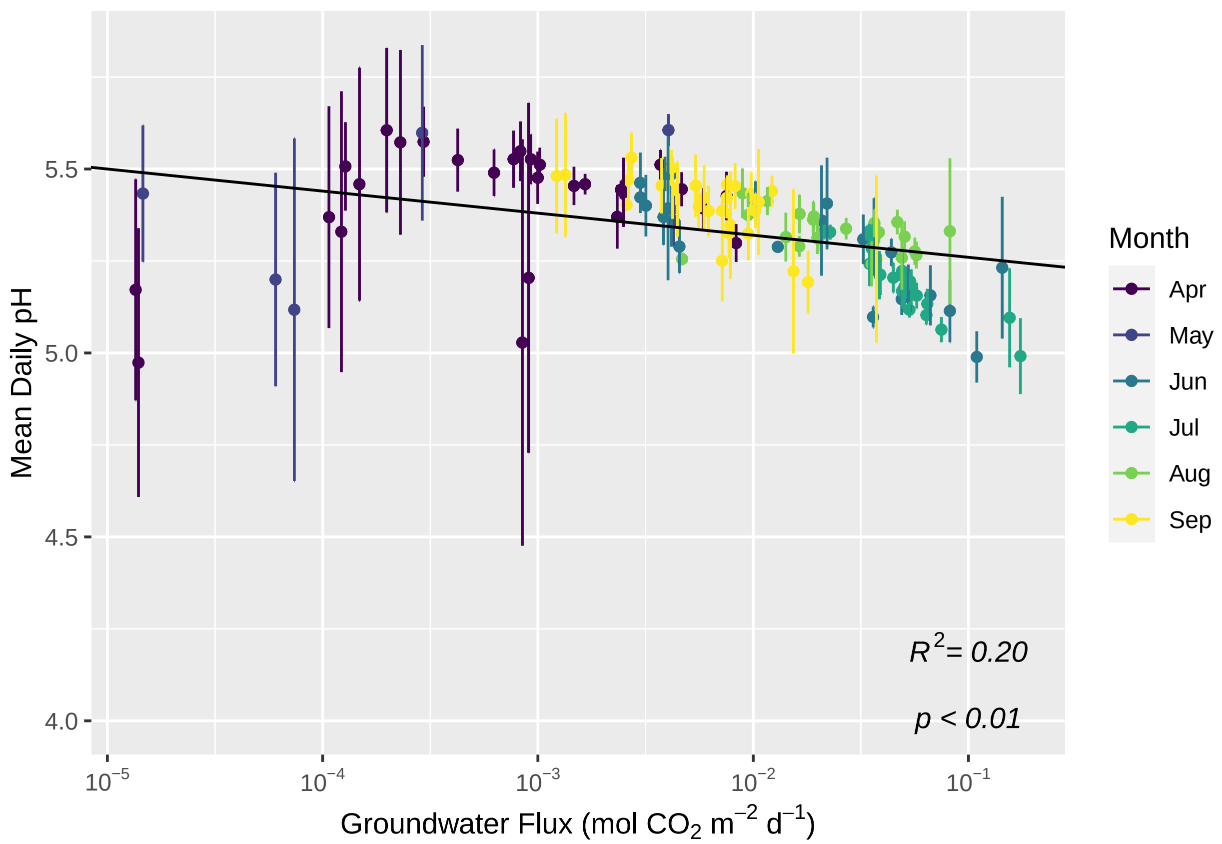


Fig S2- Hydrograph Separation

Here, we explore the variation in pCO2 during high and low flow events. We use a baseflow separation approach to distinguish flow and storm conditions by means of a filter parameter, alpha. To assess which alpha fits our reach, we vary alpha and visually evaluate the resulting separation time series.

```
# baseflow separation approach
# iterate alphas to determine the 'best' to use
alphas <- seq(0.9, 0.99, by = 0.01)

hydro_output_all <- tibble()

for(p in 1:length(alphas)) {
  hydro_output <- baseflows(flow.ts = Tac_all %>%
                              select(Date = Timestamp,
                                     Q = sumQ_m3_hr),
                            a = alphas[p],
```

```
                                n.reflected = 30,
                                ts = 'daily') %>%
    mutate(bfi = bf/Q,
           alpha = !!alphas[p])

  hydro_output_all <- rbind(hydro_output, hydro_output_all)
}

# plot Q and bf across the different alphas
FigS2 <- ggplot(hydro_output_all %>%
                  filter(alpha == 0.96),
                aes(x = Date))+
  geom_line(aes(y = Q, color = 'Storm Flow'))+
  geom_line(aes(y = bf, color = 'Baseflow'))+
  facet_wrap(. ~ alpha,
             ncol = 3)+
  ylab('Discharge (m3/h)')+
  theme(axis.title.x = element_blank(),
        legend.title = element_blank())
FigS2
```

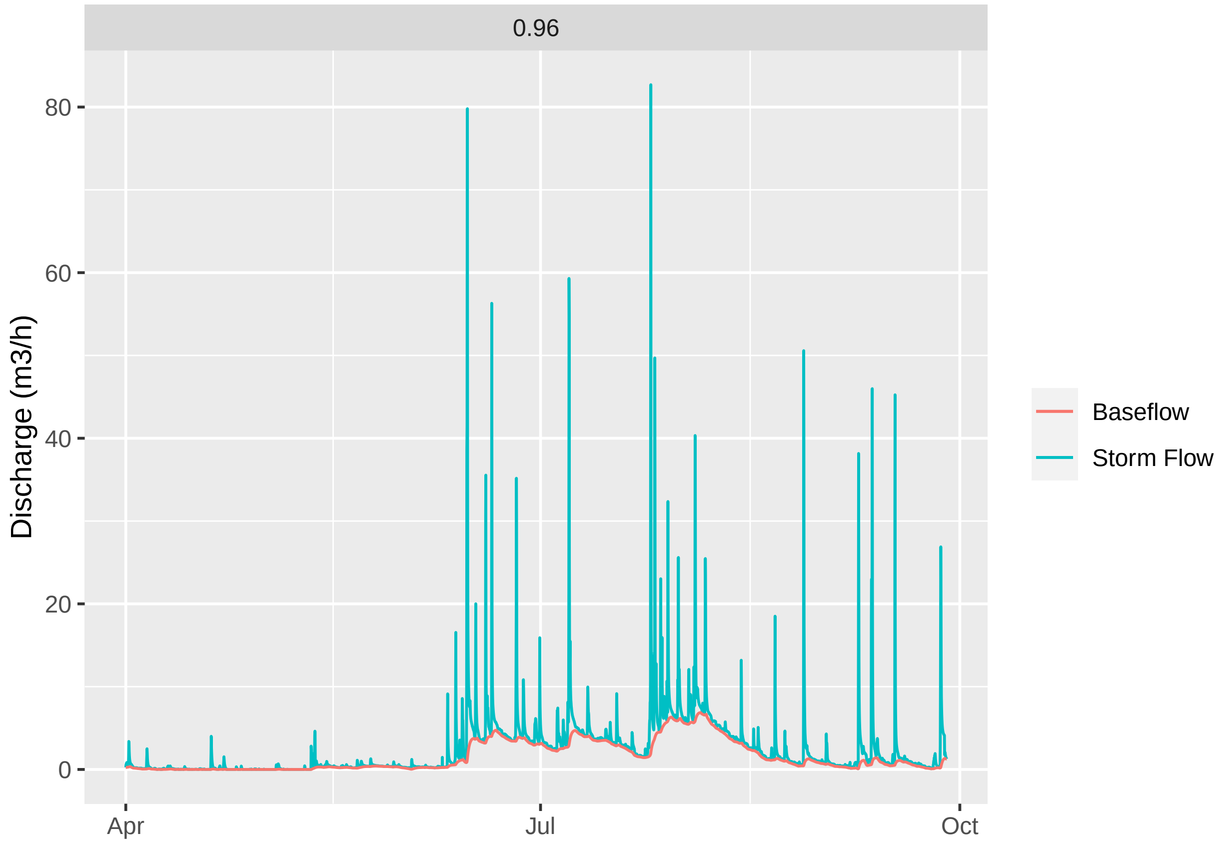
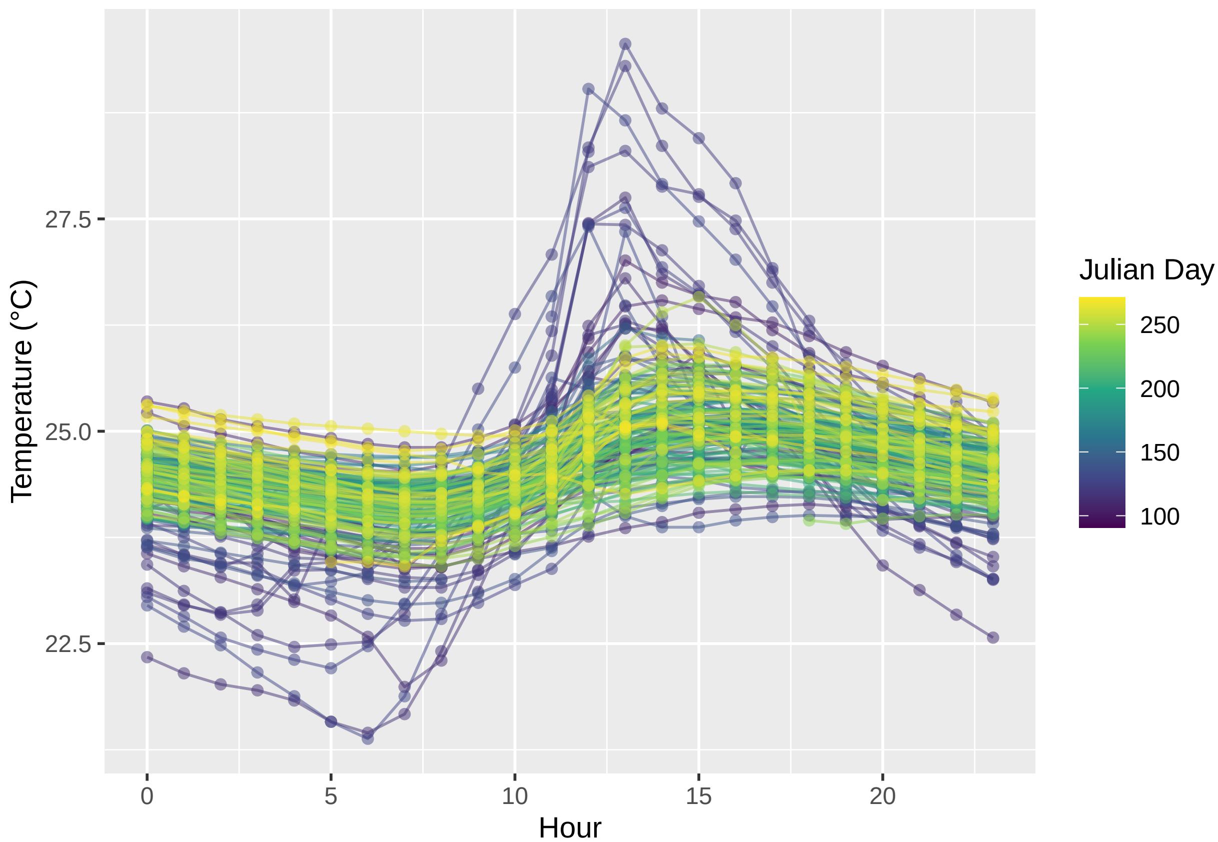


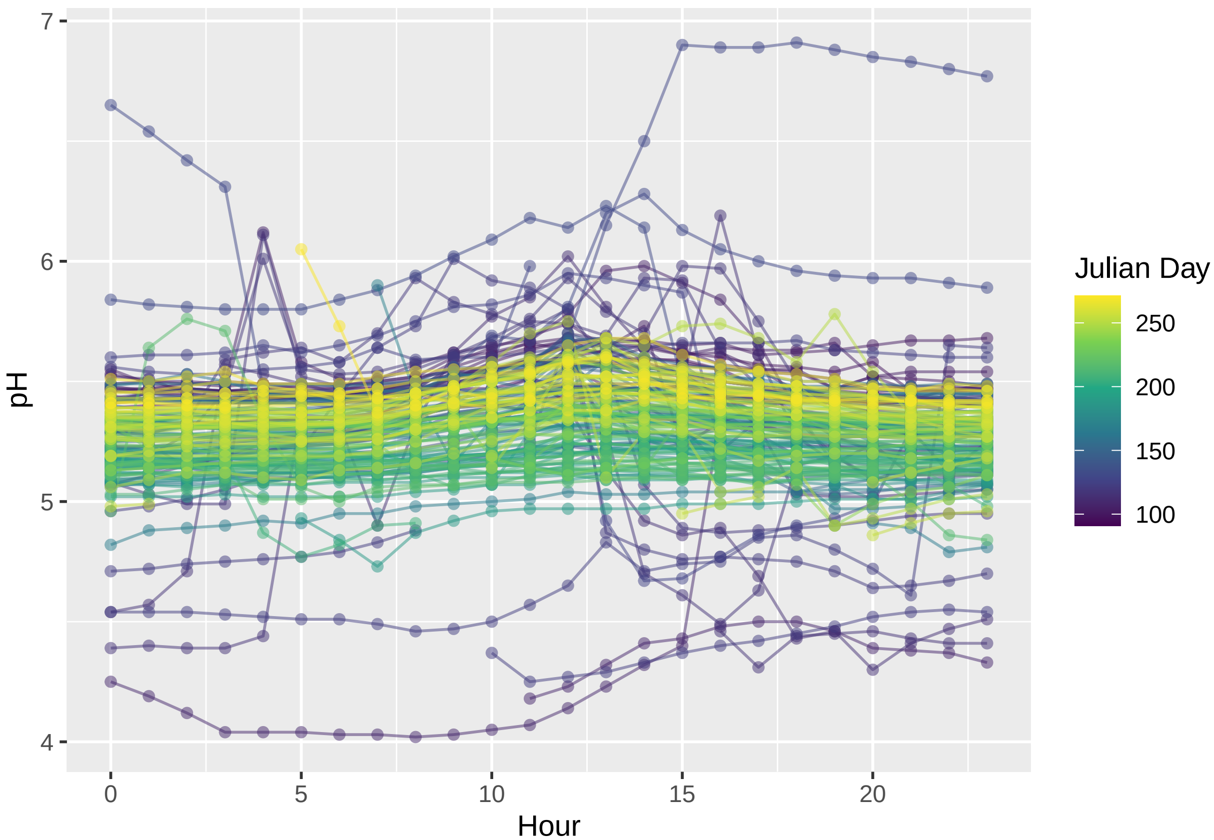
Fig S3- Diel Variation

```
tempHour <- ggplot(Tac_all,
                   aes(x = hour(Timestamp), y = temp.water))+
  geom_line(alpha = 0.5,
            aes(group = yday(Timestamp),
                color = yday(Timestamp)))+
  geom_point(alpha = 0.5,
             aes(group = yday(Timestamp),
                 color = yday(Timestamp)))+
  xlab('Hour')+
  scale_colour_viridis_c(name = "Julian Day")+
  theme(legend.title = element_text(face = 'plain'))+
  ylab("Temperature (°C)")
tempHour
```

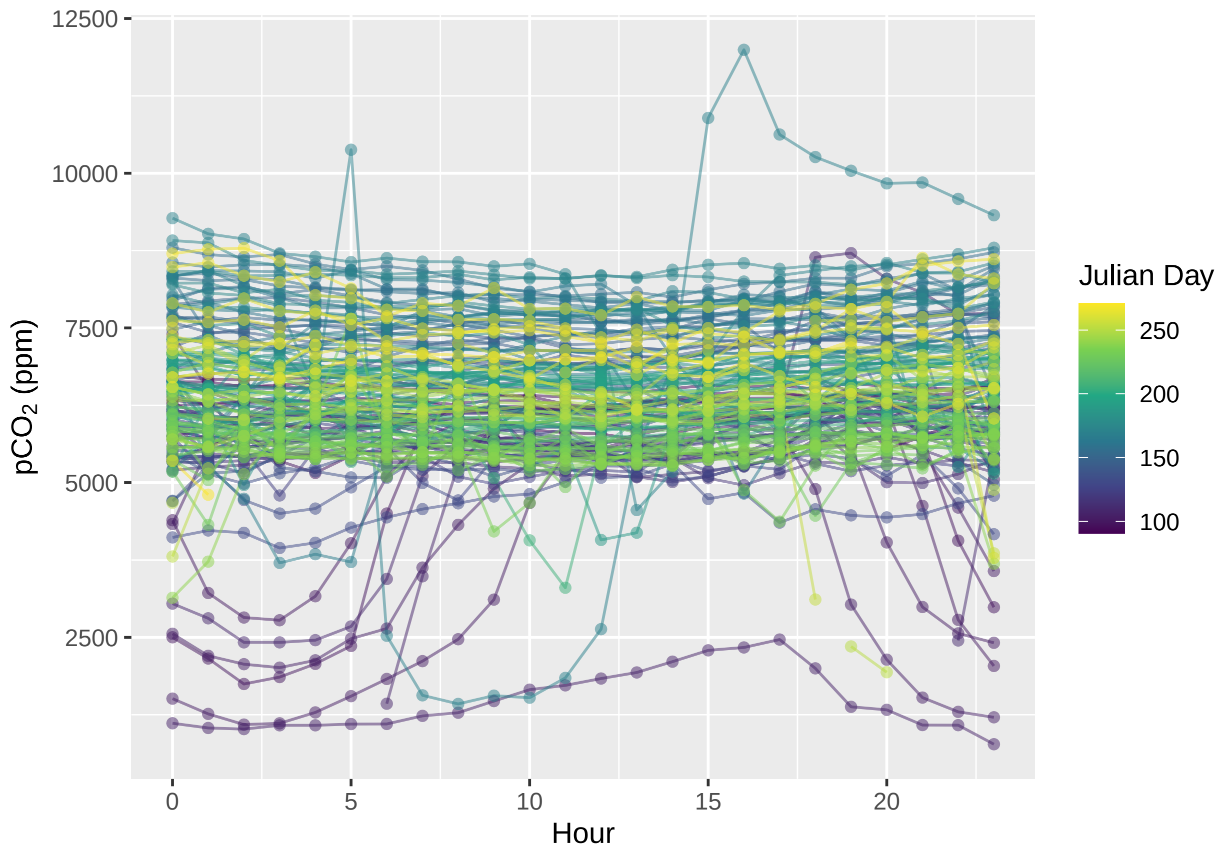



```
# pH
pHHour <- ggplot(Tac_all,
                 aes(x = hour(Timestamp), y = pH))+
  geom_line(alpha = 0.5,
            aes(group = yday(Timestamp),
                color = yday(Timestamp)))+
  geom_point(alpha = 0.5,
             aes(group = yday(Timestamp),
```

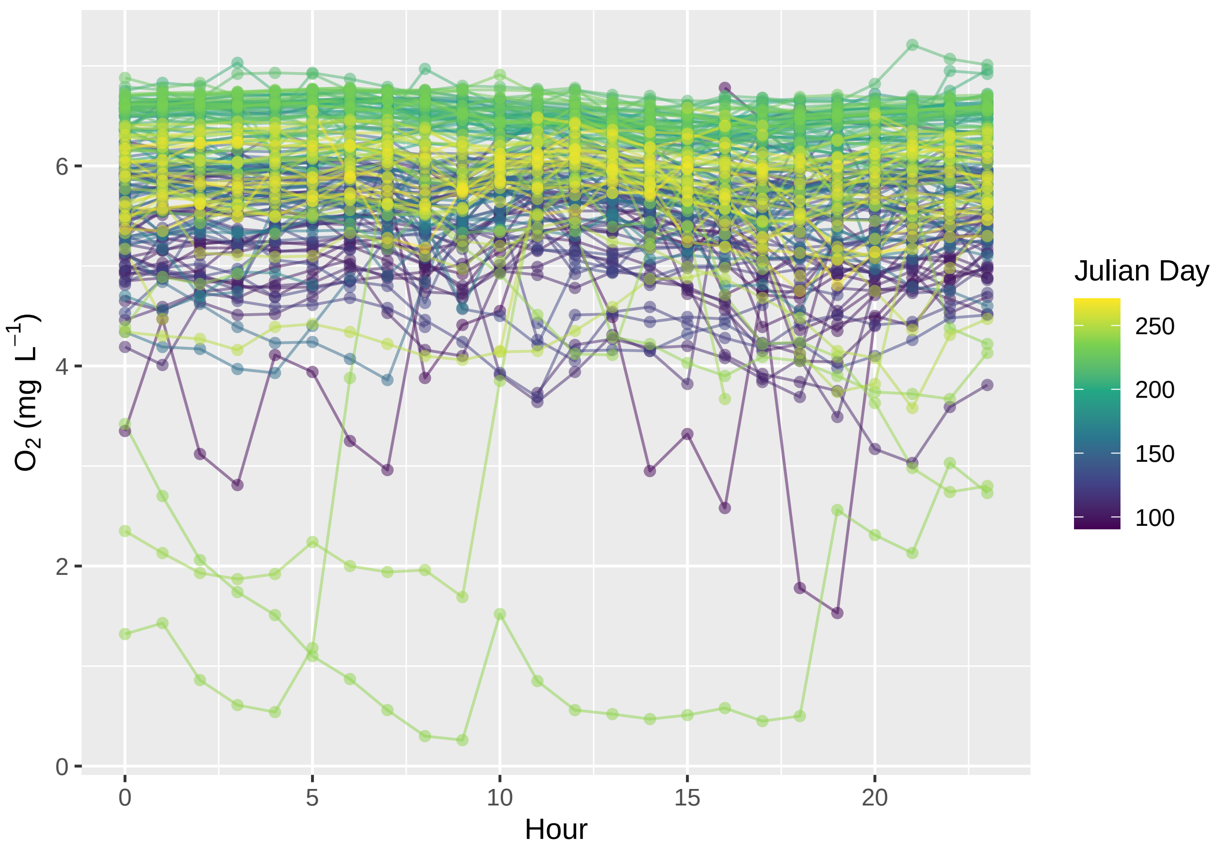
```r
                  color = yday(Timestamp)))+
  labs(x = 'Hour', y = "pH")+
  scale_colour_viridis_c(name = "Julian Day")+
  theme(legend.title = element_text(face = 'plain'))
pHHour
```




```r
# stream CO2, ppm
CO2Hour = ggplot(Tac_all,
                 aes(x = hour(Timestamp), y = CO2_ppm))+
  geom_line(alpha = 0.5,
            aes(group = yday(Timestamp), color = yday(Timestamp)))+
  geom_point(alpha = 0.5,
             aes(group = yday(Timestamp), color = yday(Timestamp)))+
  xlab('Hour')+
  ylab(expression(paste(pCO[2], ' (ppm)')))+
  scale_colour_viridis_c(name = "Julian Day")+
  theme(legend.title = element_text(face = 'plain'))
CO2Hour
```




```
# dissolved oxygen
DOHour = ggplot(Tac_all,
                aes(x = hour(Timestamp), y = DO.obs))+
  geom_line(alpha = 0.5,
            aes(group = yday(Timestamp), color = yday(Timestamp)))+
  geom_point(alpha = 0.5,
             aes(group = yday(Timestamp), color = yday(Timestamp)))+
  xlab('Hour')+
  ylab(expression(paste(O[2], " (mg "," ",L^{-1},")")))+
  scale_colour_viridis_c(name = 'Julian Day')+
  theme(legend.title = element_text(face = 'plain'))
DOHour
```




```r
FigS3 <- plot_grid(tempHour, pHHour, CO2Hour, DOHour,
                   ncol = 2, nrow = 2, align = 'hv',
                   labels = 'auto')
FigS3
```

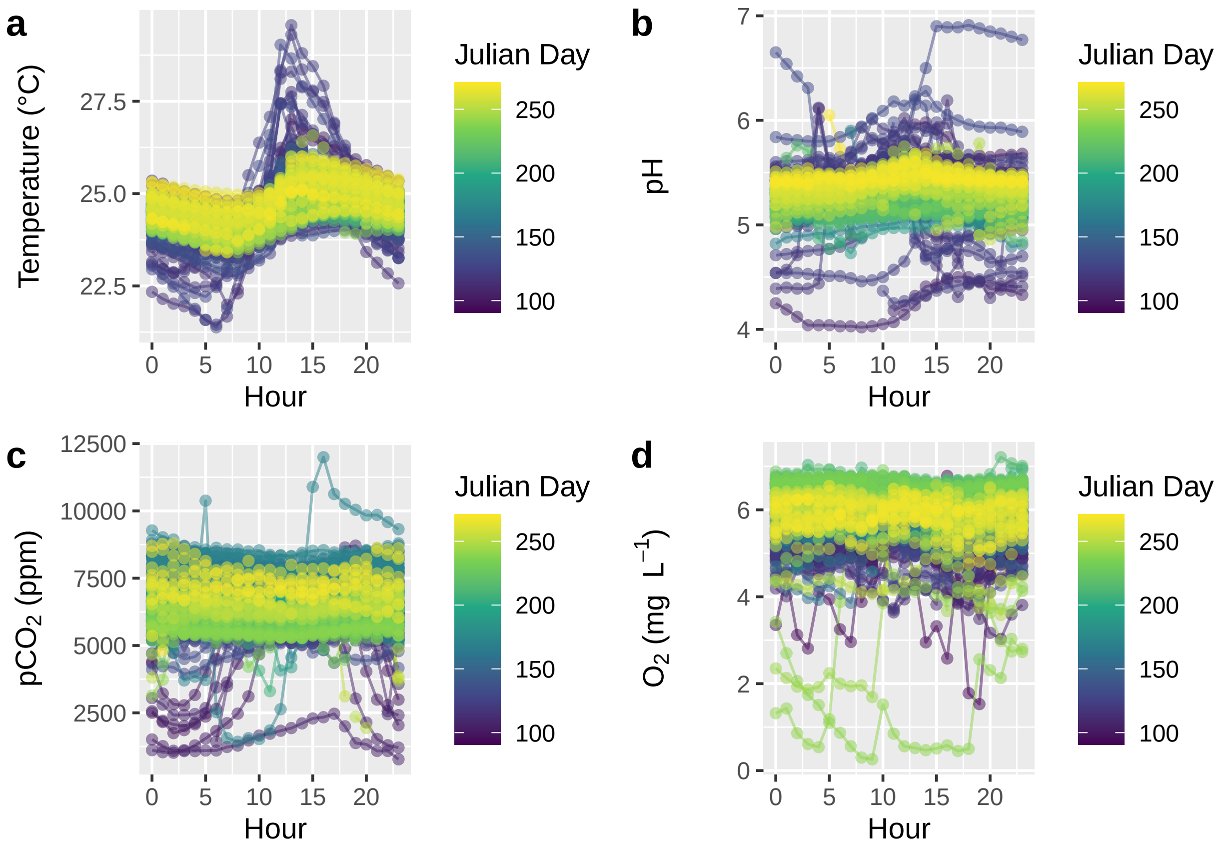
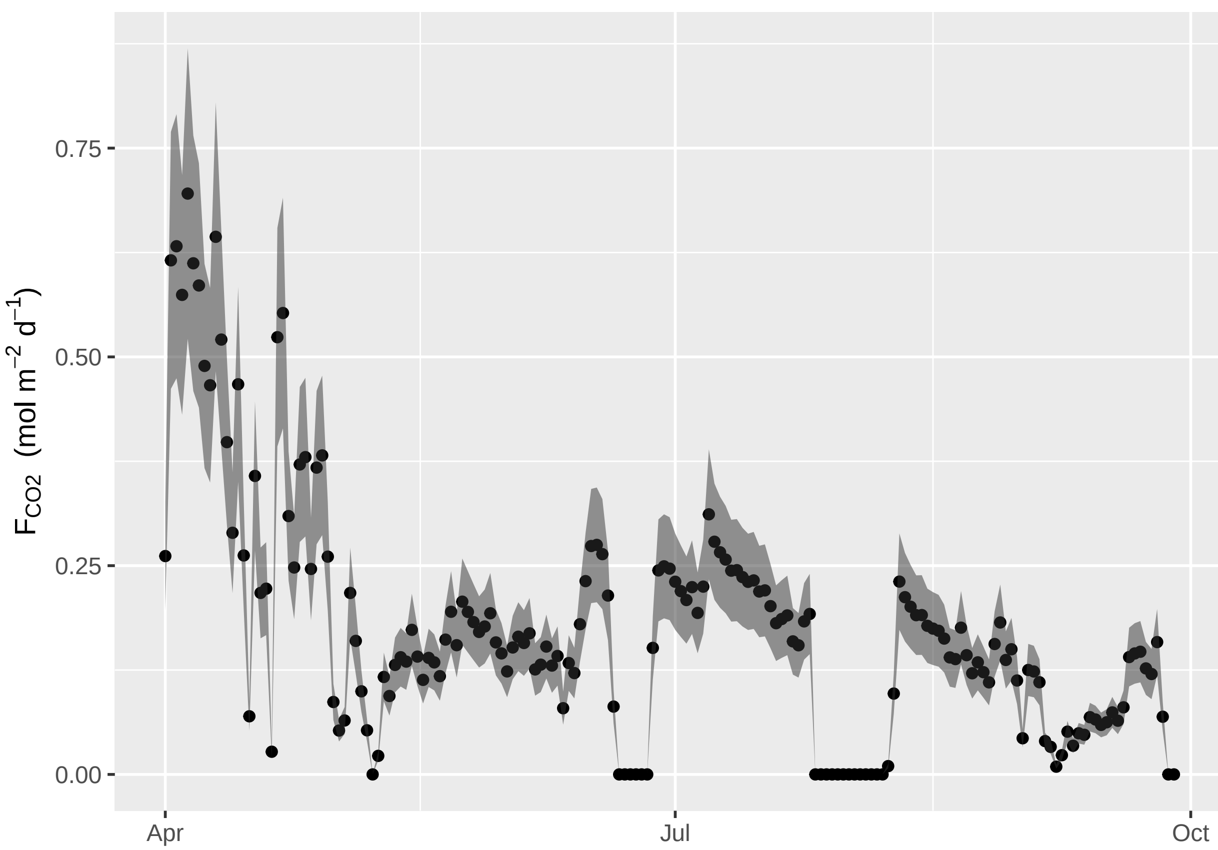


Fig S4- Sensitivity Analysis

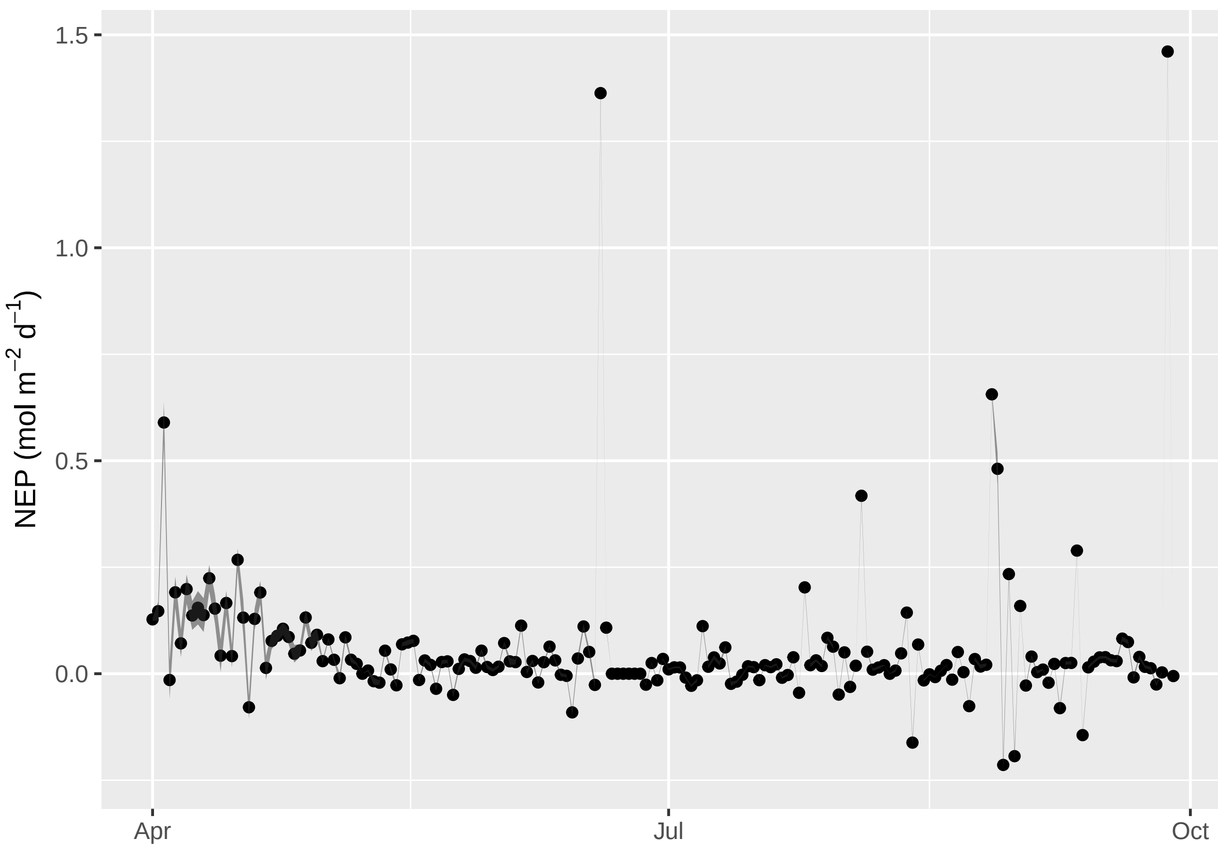
Here, we perform a sensitivity analysis on areal fluxes. We determine to what extent variation in 1) gas exchange influences changes in F[CO2] and NEP and 2) groundwater discharge influences groudwater CO2 inputs. We explore variation at +-5% and +-25% of parameter values.

```
Tac_sens_K <- Tac_gas_flux %>%
  select(Timestamp, mean_depth,
         CO2_aq, CO2_sat, CO2_efflux,
         DO.obs, Osat, direct_met_mol) %>%
  mutate(
    FCO2_high = ifelse(month(Timestamp) == '4',
                       (CO2_aq - CO2_sat)*((K_CO2_dry*1.25)/24)*mean_depth,
                       (CO2_aq - CO2_sat)*((K_CO2_wet*1.25)/24)*mean_depth),
    FCO2_low = ifelse(month(Timestamp) == '4',
                      (CO2_aq - CO2_sat)*((K_CO2_dry*0.75)/24)*mean_depth,
                      (CO2_aq - CO2_sat)*((K_CO2_wet*0.75)/24)*mean_depth),
    NEP_high = ifelse(month(Timestamp) == '4',
                      (diff(DO.obs) - (((K_O2_dry*1.25)/24)*(Osat - DO.obs))*(1/32))*-1*mean_depth,
                      (diff(DO.obs) - (((K_O2_wet*1.25)/24)*(Osat - DO.obs))*(1/32))*-1* mean_depth),
    NEP_low = ifelse(month(Timestamp) == '4',
                     (diff(DO.obs) - (((K_O2_dry*0.75)/24)*(Osat - DO.obs))*(1/32))* -1 * mean_depth,
                     (diff(DO.obs) - (((K_O2_wet*0.75)/24)*(Osat - DO.obs)) * (1/32)) * -1 * mean_depth)
  ) %>%
  group_by(day = date(Timestamp)) %>%
```

```
  summarise(sumF = sum(CO2_efflux, na.rm = TRUE),
            sumF_hi = sum(FCO2_high, na.rm = TRUE),
            sumF_low = sum(FCO2_low, na.rm = TRUE),
            sumNEP = sum(direct_met_mol, na.rm = TRUE),
            sumNEP_hi = sum(NEP_high, na.rm = TRUE),
            sumNEP_low = sum(NEP_low, na.rm = TRUE))

plot_sens_FCO2 <- ggplot(Tac_sens_K,
                         aes(x = day))+
  geom_point(aes(y = sumF))+
  geom_ribbon(aes(ymin = sumF_low, ymax = sumF_hi),
              alpha = 0.5)+
  ylab(expression(paste(F[CO2], '  (mol ', m^-2, ' ',d^-1,')')))+
  theme(axis.title.x = element_blank())
plot_sens_FCO2
```

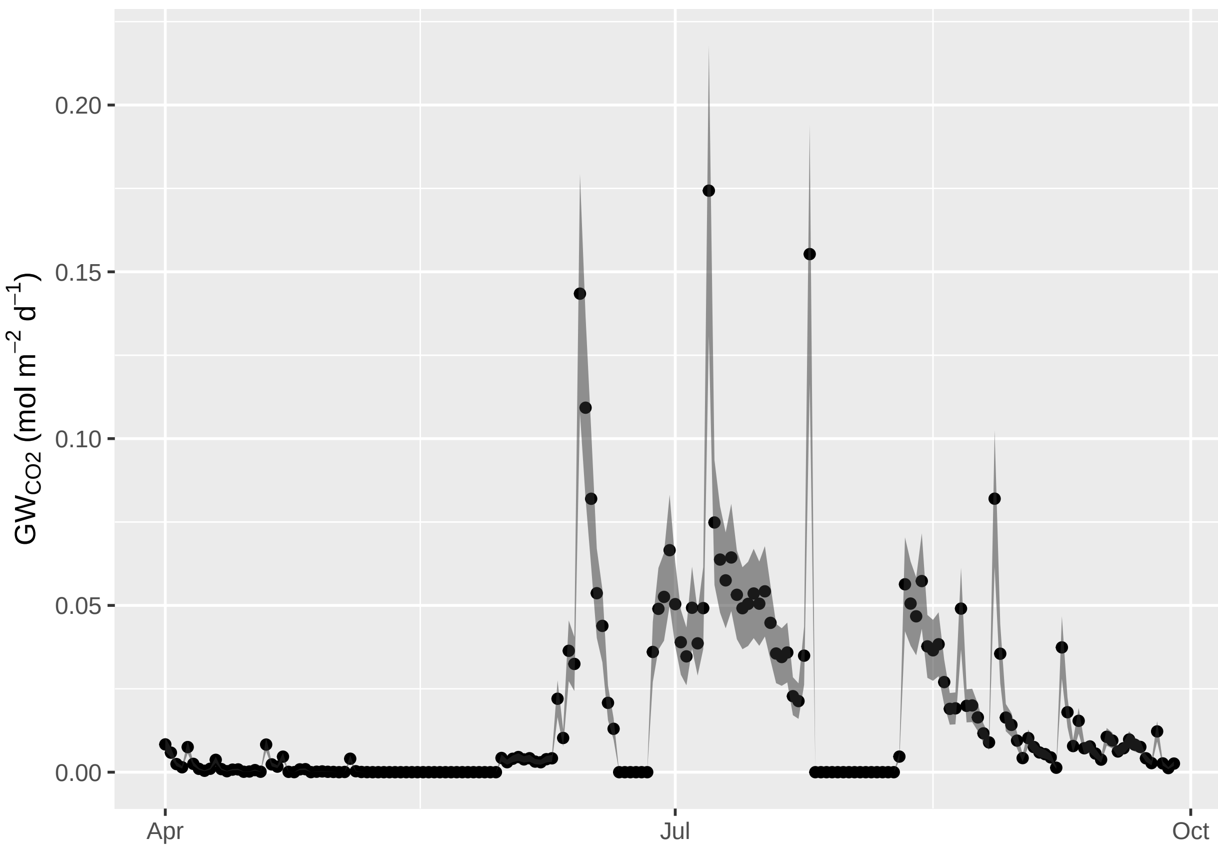



```
plot_sens_NEP <- ggplot(Tac_sens_K %>%
                          filter(sumNEP < 1.5),
                        aes(x = day))+
  geom_point(aes(y = sumNEP))+
  geom_ribbon(aes(ymin = sumNEP_low, ymax = sumNEP_hi),
              alpha = 0.5)+
  ylab(expression(paste('NEP (mol ', m^-2, ' ',d^-1,')')))+
  theme(axis.title.x = element_blank())
plot_sens_NEP
```

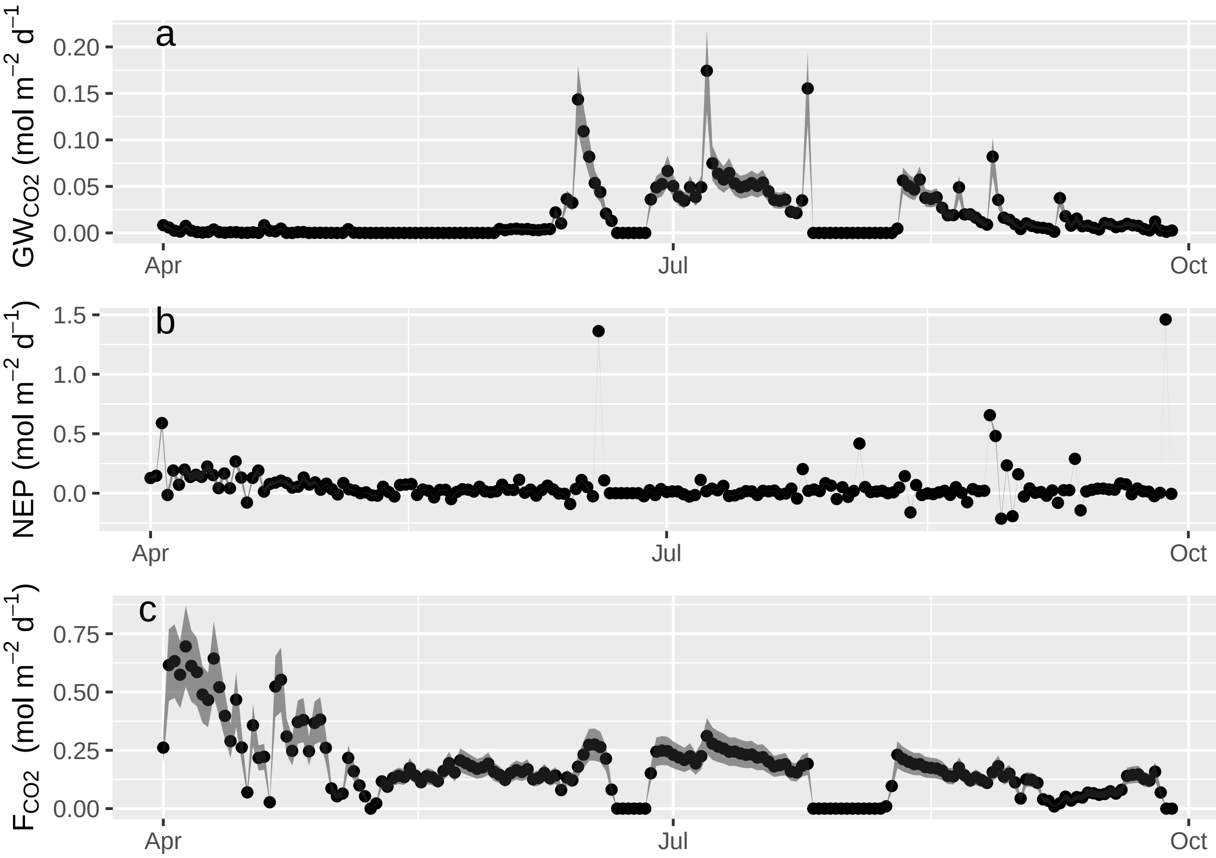



```
Tac_sens_gw <- Tac_GWCO2 %>%
  select(Timestamp,
         wellCO2_aq,
         sumQ_m3_hr,
         GWCO2) %>%
  mutate(

    GW_Q_hi = ifelse(month(Timestamp) == 'April',    # calculate total groundwater discharge (m3/h)
                     ((0.1779*1.25)*sumQ_m3_hr),           # 17.79% of stream Q in dry season (April);
                     ((0.0715*1.25)*sumQ_m3_hr)),
    GW_Q_low = ifelse(month(Timestamp) == 'April',    # calculate total groundwater discharge (m3/h)
                      ((0.1779*0.75)*sumQ_m3_hr),           # 17.79% of stream Q in dry season (April)
                      ((0.0715*0.75)*sumQ_m3_hr)),

    # groundwater velocity (m h-1); divide GW discharge by reach area (length * width)
    GW_v_hi = ifelse(month(Timestamp) == 'April',
                     GW_Q_hi/(Tac_reach * Tac_width_lower_dry),
                     GW_Q_hi/(Tac_reach * Tac_width_lower_wet)),
    GW_v_low = ifelse(month(Timestamp) == 'April',
                      GW_Q_low/(Tac_reach * Tac_width_lower_dry),
                      GW_Q_low/(Tac_reach * Tac_width_lower_wet)),

    # GWCO2 flux (mol CO2 m-2 h-1)
    GWCO2_hi = GW_v_hi * wellCO2_aq,
    GWCO2_low = GW_v_low * wellCO2_aq
  ) %>%
```

```
  group_by(day = date(Timestamp)) %>%
  summarise(sumGW = sum(GWCO2, na.rm = TRUE),
            sumGW_hi = sum(GWCO2_hi, na.rm = TRUE),
            sumGW_low = sum(GWCO2_low, na.rm = TRUE))

plot_sens_GW <- ggplot(Tac_sens_gw,
                       aes(x = day))+
  geom_point(aes(y = sumGW))+
  geom_ribbon(aes(ymin = sumGW_low, ymax = sumGW_hi),
              alpha = 0.5)+
  ylab(expression(paste(GW[CO2], ' (mol ', m^-2, ' ', d^-1,')')))+
  theme(axis.title.x = element_blank())
plot_sens_GW
```




```
FigS4 <- plot_grid(plot_sens_GW,
                   plot_sens_NEP,
                   plot_sens_FCO2,
                   nrow = 3,
                   labels = 'auto', label_fontface = 'plain',
                   hjust = -8)
FigS4
```



Figs S5-11- Groundwater input method evaluation —-

Here, we evaluate approaches to estimate groundwater inputs to our study reach. In addition to the approach used in the manuscript, which uses a fixed seasonal percentage based on empirical field data, we compare baseflow separation techniques and a linear interpolation of groundwater inputs.

```
# compare 4 methods
## 1: fixed percent- used in the ms
## 2: interpolation- develop a rating curve between discharge and the fraction of GW inputs using publi
## 3: Baseflow separation
## 4: arealy corrected baseflow- using the baseflow calculated in #2, scale this estimate to our study r


# For method 2, create a data frame with discharge and the measured fGW
rc <- data.frame(mon = c('Feb','Aug'),
                 discharge = c(14.16, 109.8),
                 fGW = c(17.8, 7.2))

# develop a ratin curve between fGW and discharge
summary(lm(data = rc,
           fGW ~ discharge))
```

```
##
## Call:
## lm(formula = fGW ~ discharge, data = rc)
```

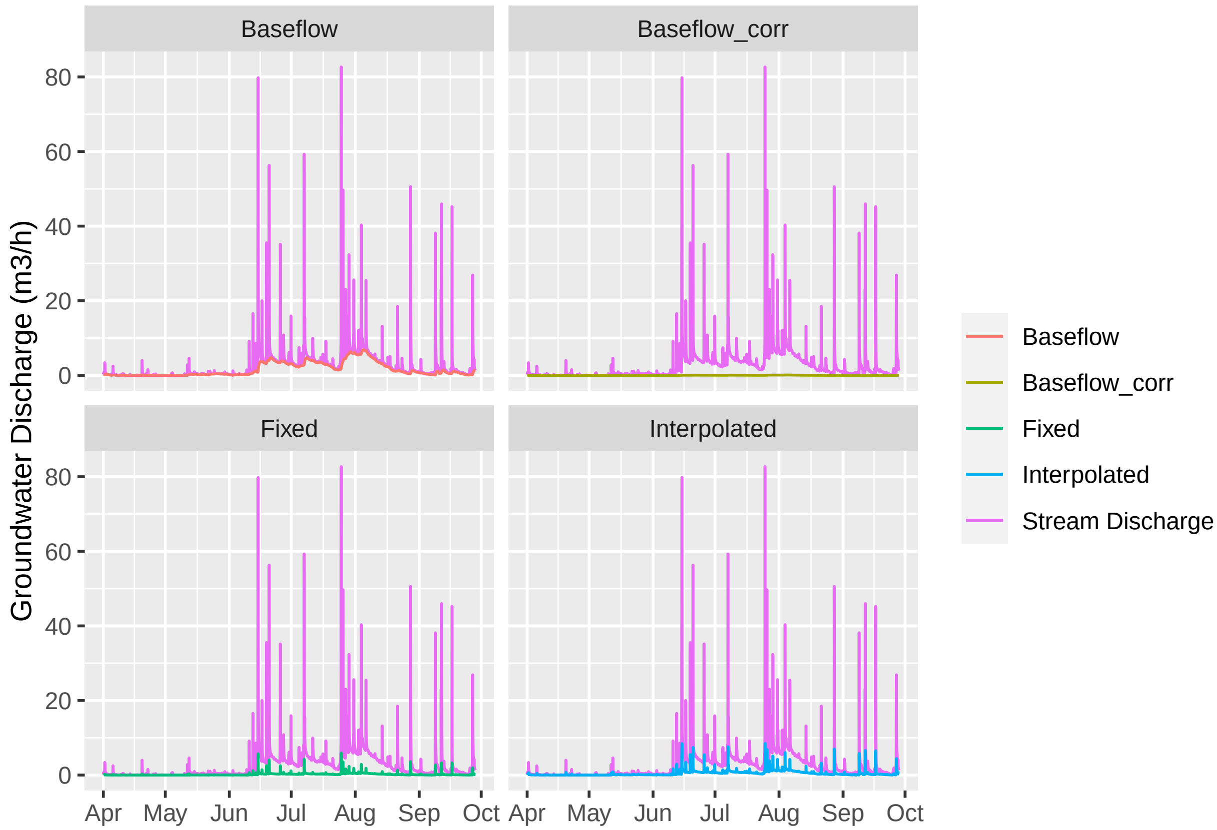
```
##
## Residuals:
## ALL 2 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.3694        NaN     NaN      NaN
## discharge    -0.1108        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 1 and 0 DF,  p-value: NA
```

```
# fGW = 19.37 - discharge*0.11

# estimate fGW as a linear function of discharge for all discharges
Tac_baseflow_hyst_pred <- Tac_baseflow_hyst %>%
  mutate(fGW_pred = (19.37 - Q*0.11)/100,
         GW_Q_pred = Q*fGW_pred)

# merge all fGW estimates into one data frame
Tac_GWCO2_eval <- Tac_all %>%
  select(Timestamp,
         pH,
         temp.water, wellCO2,
         sumQ_m3_hr) %>%
  left_join(Tac_baseflow_hyst_pred) %>%
  mutate(
    # temperature-corrected Henry's Law constant (Plummer and Busenberg 1982)
    kH = 29.41*exp(-2400*(1/(273 + temp.water) - (1/298))),

    # CO2-well
    wellCO2_aq = ((wellCO2/1e6)/kH)*1000,  # well aqueous CO2: mol CO2 m-3

    # Method 1: Fixed percentage (m3 h-1)
    m1_GW_Q = ifelse(month(Timestamp) == 'April',     # calculate total groundwater discharge (m3/h)
                      (0.1779*sumQ_m3_hr),             # 17.79% of stream Q in dry season (April); m3/h
                      (0.0715*sumQ_m3_hr)),            # 7.15% of stream Q in wet season; m3/h

    # groundwater velocity (m h-1); divide GW discharge by reach area (length * width)
    m1_GW_v = ifelse(month(Timestamp) == 'April',
                     GW_Q/(Tac_reach * Tac_width_lower_dry),
                     GW_Q/(Tac_reach * Tac_width_lower_wet)
    ),

    # GWCO2 flux (mol CO2 m-2 h-1)
    m1_GWCO2 = m1_GW_v * wellCO2_aq,

    # Method 2: interpolation
    m2_GW_v = ifelse(month(Timestamp) == 'April',
                     GW_Q_pred/(Tac_reach * Tac_width_lower_dry),
                     GW_Q_pred/(Tac_reach * Tac_width_lower_wet)
    ),
    m2_inter_GWCO2 = m2_GW_v * wellCO2_aq,
```

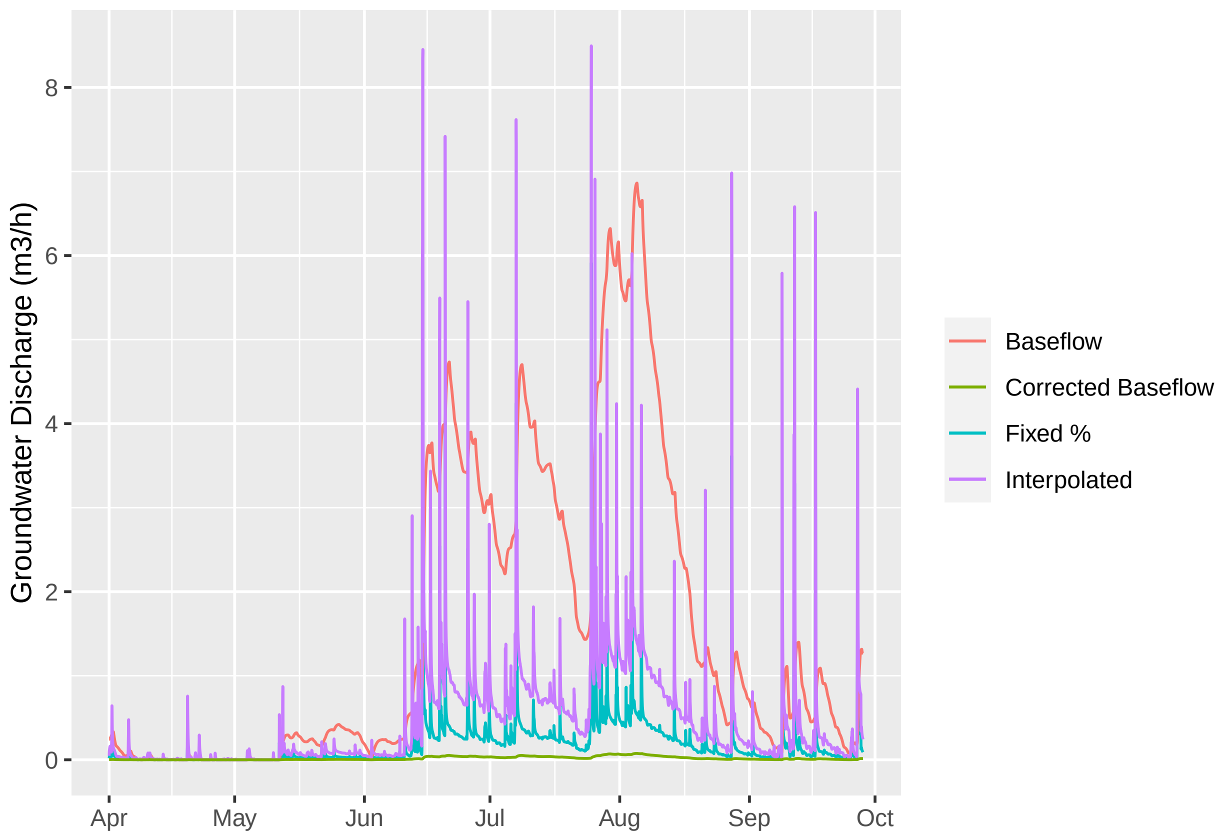
```
    # Method 3: baseflow separation
    m3_bf_v = ifelse(month(Timestamp) == 'April',
                     bf/(Tac_reach * Tac_width_lower_dry),
                     bf/(Tac_reach * Tac_width_lower_wet)),
    m3_bf_GWCO2 = m3_bf_v * wellCO2_aq,

    # Method 4: area-corrected baseflow
    m4_bf_corr = 0.011*bf,
    m4_bf_corr_v = ifelse(month(Timestamp) == 'April',
                     m4_bf_corr/(Tac_reach * Tac_width_lower_dry),
                     m4_bf_corr/(Tac_reach * Tac_width_lower_wet)),
    m4_bf_corr_GWCO2 = m4_bf_corr_v * wellCO2_aq
  )
```

```
## Joining, by = c("Timestamp", "sumQ_m3_hr")
```

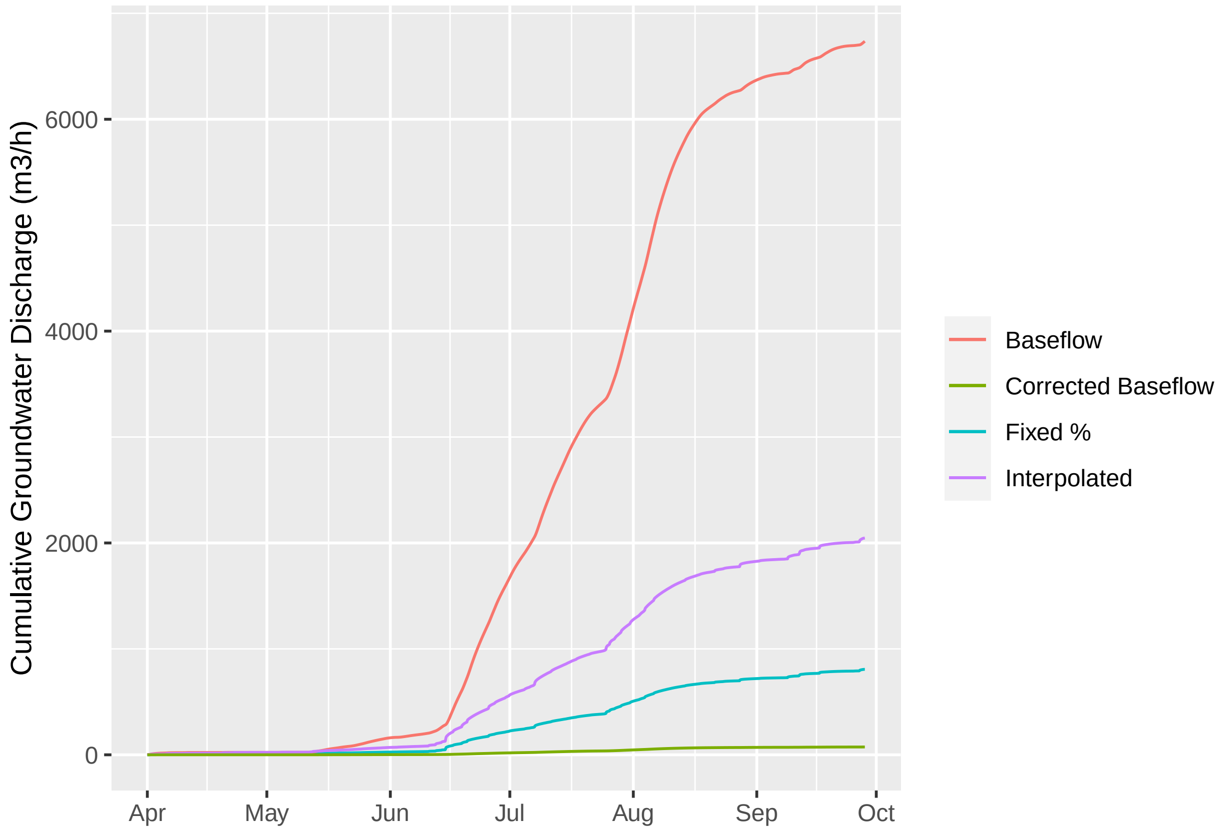
```
# plot: each estimate compared to stream discharge
Tac_GWCO2_eval %>%
  select(Timestamp, sumQ_m3_hr,
         Fixed = m1_GW_Q,
         Baseflow = bf,
         Interpolated = GW_Q_pred,
         Baseflow_corr = m4_bf_corr) %>%
  pivot_longer(cols = 3:6,
               names_to = 'vars',
               values_to = 'vals') %>%
  ggplot(., aes(x = Timestamp))+
  geom_line(aes(y = sumQ_m3_hr, color = 'Stream Discharge'))+
  geom_line(aes(y = vals, color = vars))+
  facet_wrap(. ~ vars,
             nrow = 2, ncol = 2)+
  labs(x = element_blank(),
       y = 'Groundwater Discharge (m3/h)')+
  theme(legend.title = element_blank())+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")
```



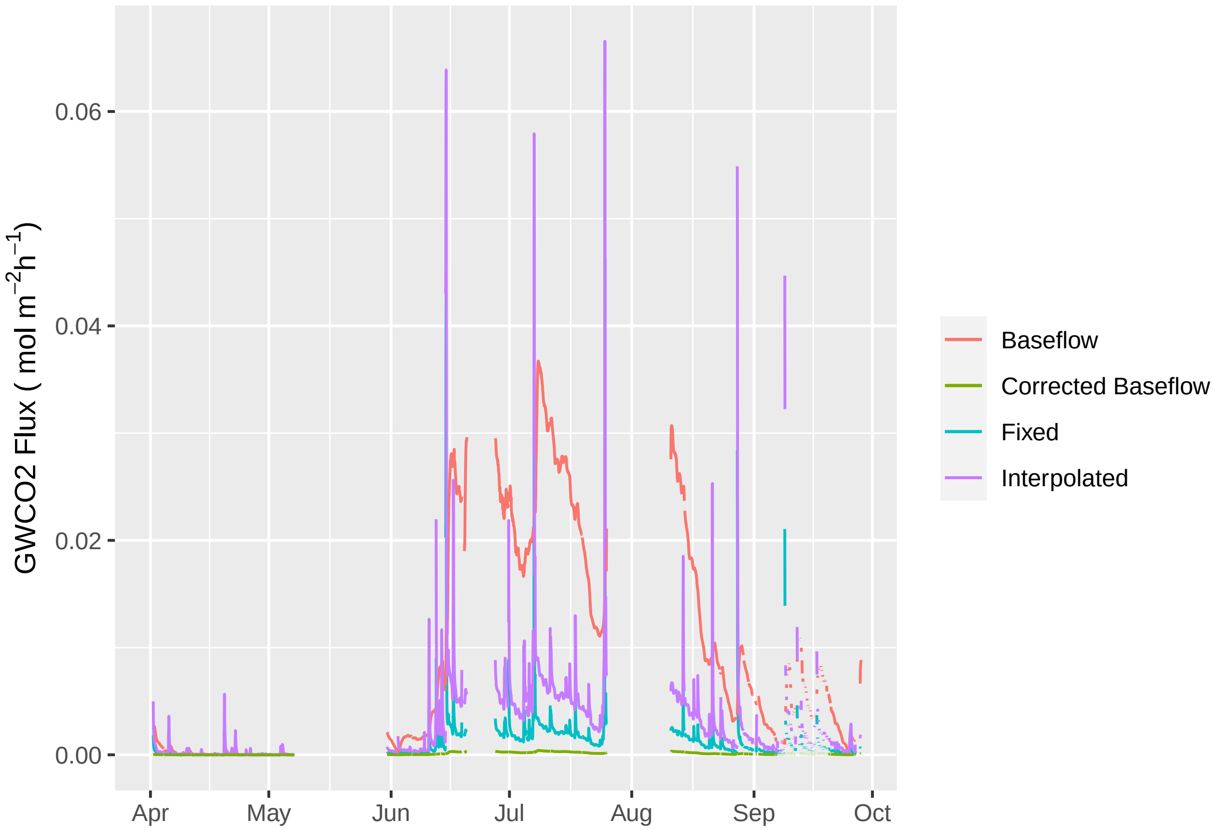

```r
# plot: compare estimates of groundwater inputs
Tac_GWCO2_eval %>%
  ggplot(., aes(x = Timestamp))+
  geom_line(aes(y = m1_GW_Q, color = 'Fixed %'))+
  geom_line(aes(y = bf, color = 'Baseflow'))+
  geom_line(aes(y = GW_Q_pred, color = 'Interpolated'))+
  geom_line(aes(y = m4_bf_corr, color = 'Corrected Baseflow'))+
  labs(x = element_blank(),
       y = 'Groundwater Discharge (m3/h)')+
  theme(legend.title = element_blank())+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")
```



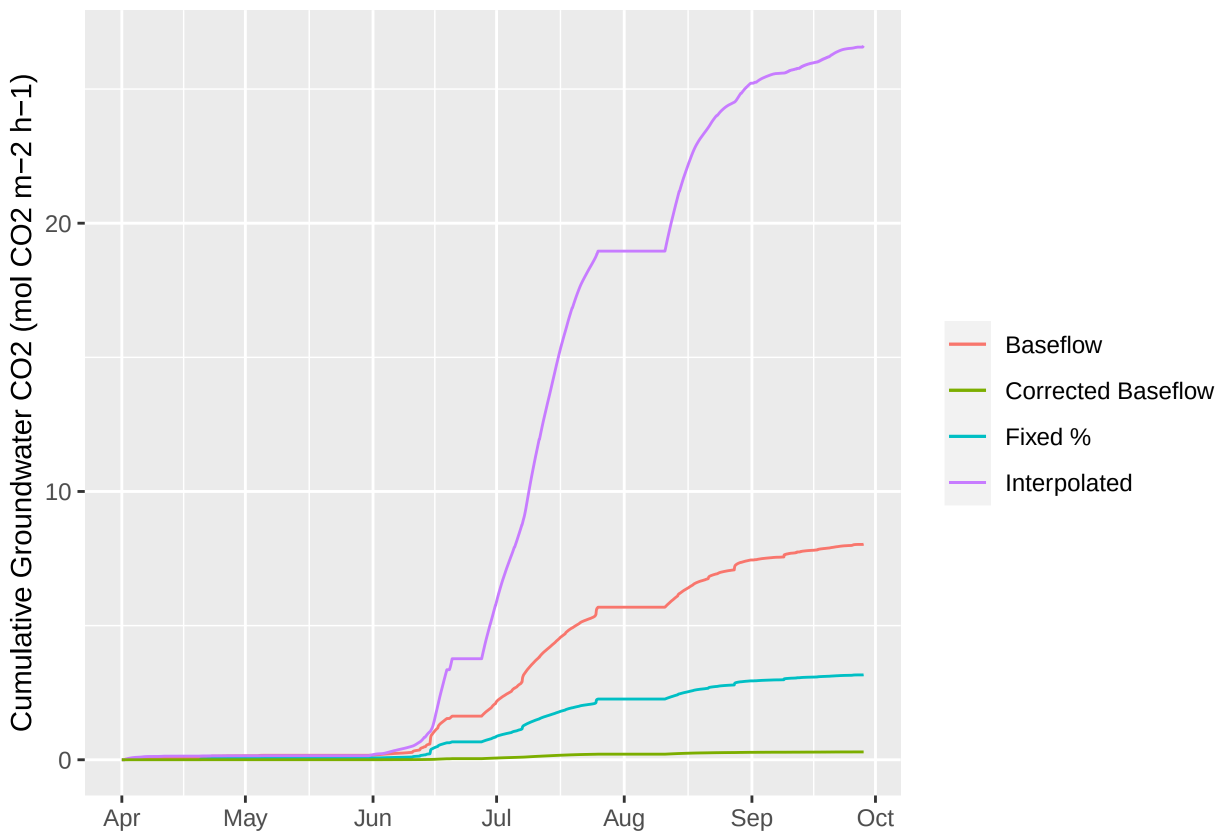

```r
# plot: cumulative GW inputs
Tac_GWCO2_eval %>%
  mutate(cum_m1 = cumsum(m1_GW_Q),
         cum_m2 = cumsum(bf),
         cum_m3 = cumsum(GW_Q_pred),
         cum_m4 = cumsum(m4_bf_corr)) %>%
  ggplot(., aes(x = Timestamp))+
  geom_line(aes(y = cum_m1, color = 'Fixed %'))+
  geom_line(aes(y = cum_m2, color = 'Baseflow'))+
  geom_line(aes(y = cum_m3, color = 'Interpolated'))+
  geom_line(aes(y = cum_m4, color = 'Corrected Baseflow'))+
  labs(x = element_blank(),
       y = 'Cumulative Groundwater Discharge (m3/h)')+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  scale_color_discrete(name = element_blank())
```



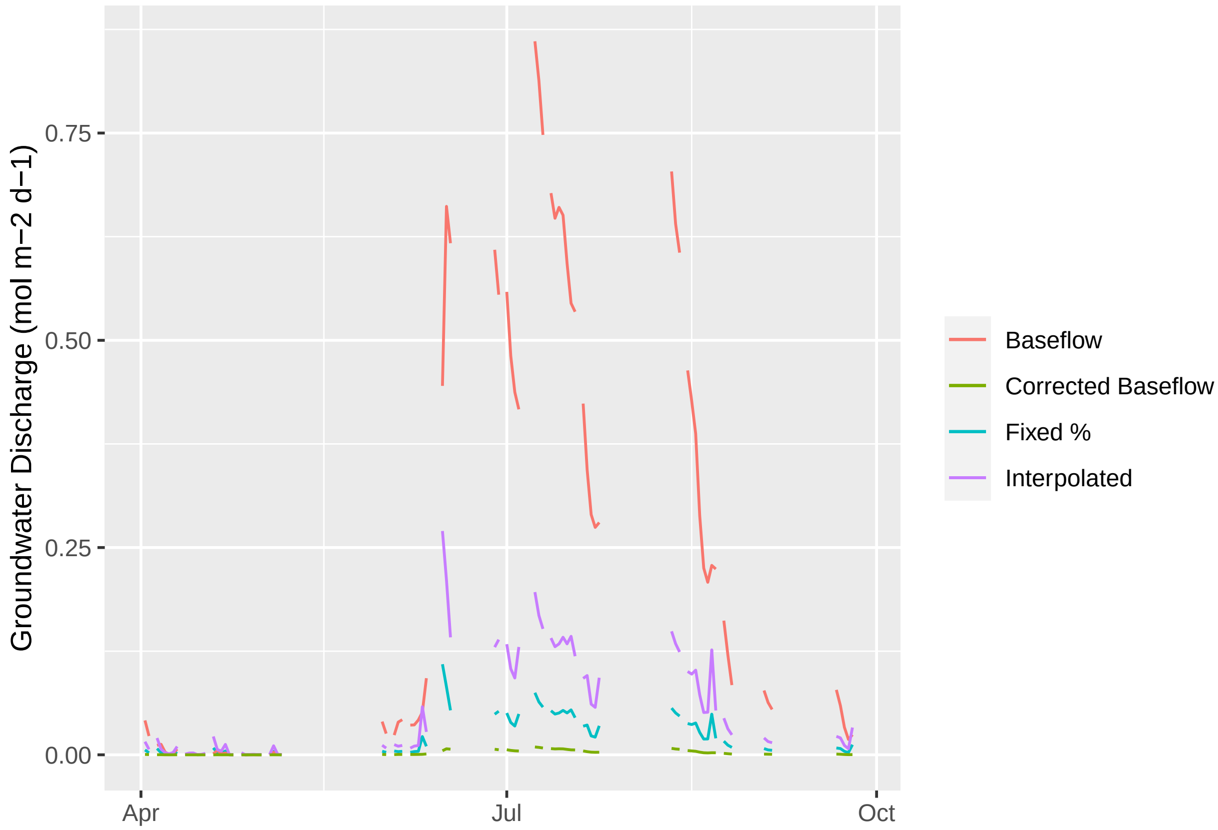

```r
# plot: compare estimates of GWCO2 flux
Tac_GWCO2_eval %>%
  ggplot(., aes(x = Timestamp))+
  geom_line(aes(y = m1_GWCO2, color = 'Fixed'))+
  geom_line(aes(y = m3_bf_GWCO2, color = 'Baseflow'))+
  geom_line(aes(y = m2_inter_GWCO2, color = 'Interpolated'))+
  geom_line(aes(y = m4_bf_corr_GWCO2, color = 'Corrected Baseflow'))+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  scale_color_discrete(name = element_blank())+
  labs(x = element_blank(),
       y = expression(paste('GWCO2 Flux ( mol ', m^-2, h^-1,')')))
```



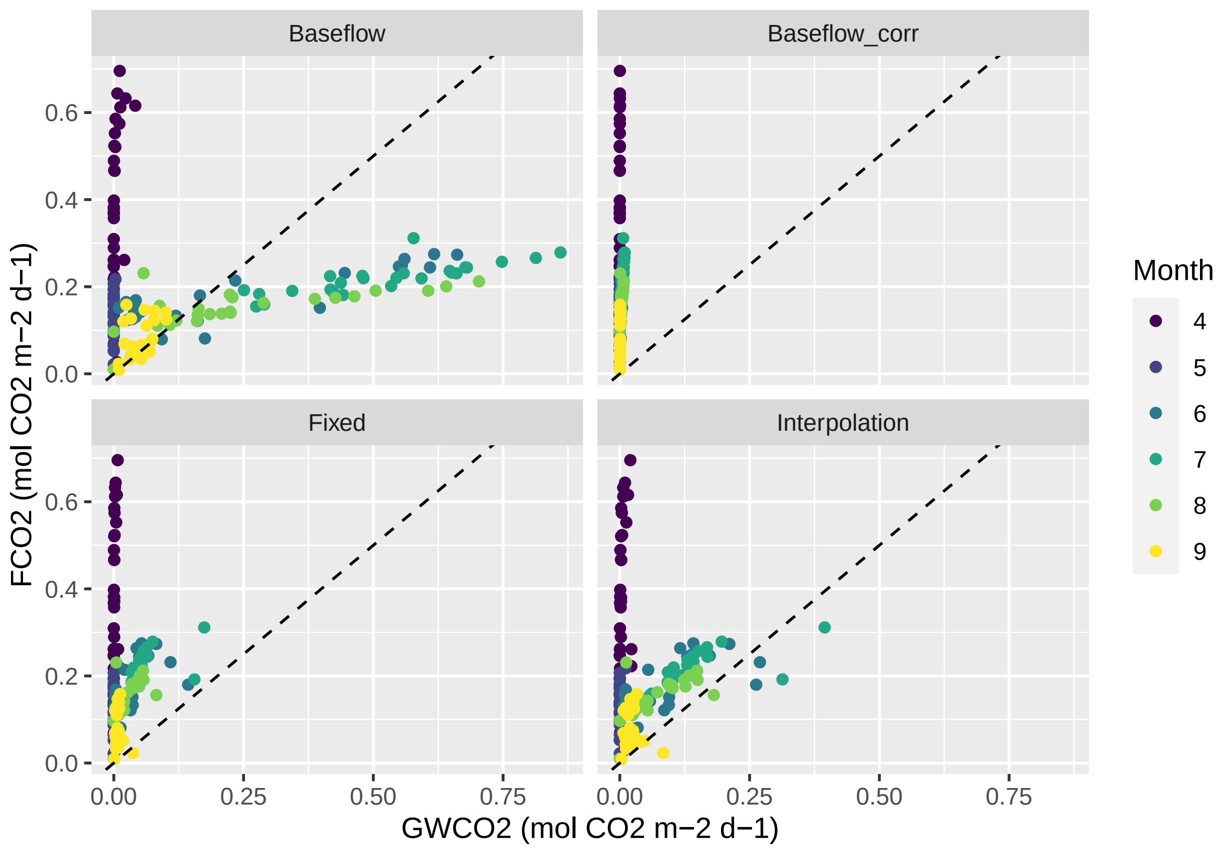

```
# plot: compare cumulative GWCO2 inputs
Tac_GWCO2_eval %>%
  mutate(cum_m1 = cumsum(replace_na(m1_GWCO2, 0)),
         cum_m3 = cumsum(replace_na(m3_bf_GWCO2, 0)),
         cum_m2 = cumsum(replace_na(m2_inter_GWCO2, 0)),
         cum_m4 = cumsum(replace_na(m4_bf_corr_GWCO2, 0))) %>%
  ggplot(., aes(x = Timestamp))+
  geom_line(aes(y = cum_m1, color = 'Fixed %'))+
  geom_line(aes(y = cum_m2, color = 'Baseflow'))+
  geom_line(aes(y = cum_m3, color = 'Interpolated'))+
  geom_line(aes(y = cum_m4, color = 'Corrected Baseflow'))+
  labs(x = element_blank(),
       y = 'Cumulative Groundwater CO2 (mol CO2 m-2 h-1)')+
  scale_x_datetime(date_breaks = "1 month",
                   date_labels =  "%b")+
  scale_color_discrete(name = element_blank())
```




```
# plot: compare daily GWCO2 inputs from the methods
Tac_GWCO2_eval %>%
  group_by(day = date(Timestamp)) %>%
  summarise(m1_day = sum(m1_GWCO2),
            m3_day = sum(m3_bf_GWCO2),
            m2_day = sum(m2_inter_GWCO2),
            m4_day = sum(m4_bf_corr_GWCO2)) %>%
  ggplot(., aes(x = day))+
  geom_line(aes(y = m1_day, color = 'Fixed %'))+
  geom_line(aes(y = m3_day, color = 'Baseflow'))+
  geom_line(aes(y = m2_day, color = 'Interpolated'))+
  geom_line(aes(y = m4_day, color = 'Corrected Baseflow'))+
  labs(x = element_blank(),
       y = 'Groundwater Discharge (mol m-2 d-1)')+
  # scale_x_datetime(date_breaks = "1 month",
  #                  date_labels =  "%b")+
  scale_color_discrete(name = element_blank())
```




```r
# plot: re-make Fig 6, using different estimates of GWCO2 as drivers of FCO2
left_join(Tac_gas_flux,
                         Tac_GWCO2_eval,
                         by = 'Timestamp') %>%
  select(Timestamp, CO2_efflux, direct_met_mol,
         m1_GWCO2, m3_bf_GWCO2, m2_inter_GWCO2, m4_bf_corr_GWCO2) %>%
  group_by(day = date(Timestamp)) %>%
  summarise(dayF = sum(CO2_efflux, na.rm = TRUE),
            dayNEP = sum(direct_met_mol, na.rm = TRUE),
            Fixed = sum(m1_GWCO2, na.rm = TRUE),
            Baseflow = sum(m3_bf_GWCO2, na.rm = TRUE),
            Interpolation = sum(m2_inter_GWCO2, na.rm = TRUE),
            Baseflow_corr = sum(m4_bf_corr_GWCO2, na.rm = TRUE)) %>%
  filter(dayNEP < 1.5,
         dayF > 0) %>%
  pivot_longer(cols = 4:7,
               names_to = 'vars',
               values_to = 'vals') %>%
  ggplot(., aes(x = vals, y =  dayF))+
  geom_point(aes(color = factor(month(day, abbr = TRUE))))+
  geom_abline(intercept = 0, slope = 1, linetype = 'dashed')+
  facet_wrap(. ~ vars, ncol = 2)+
  scale_color_viridis_d(name = 'Month')+
  labs(x = 'GWCO2 (mol CO2 m-2 d-1)',
      y = 'FCO2 (mol CO2 m-2 d-1)')
```



Session info

```
pander(sessionInfo())
```

R version 4.1.3 (2022-03-10)

Platform: x86_64-w64-mingw32/x64 (64-bit)

locale: *LC_COLLATE=English_United States.1252*, *LC_CTYPE=English_United States.1252*, *LC_MONETARY=English_United States.1252*, *LC_NUMERIC=C* and *LC_TIME=English_United States.1252*

attached base packages: *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

other attached packages: *pander(v.0.6.5)*, *hydrostats(v.0.2.9)*, *ARTool(v.0.11.1)*, *lsmeans(v.2.30-0)*, *emmeans(v.1.7.5)*, *purrr(v.0.3.4)*, *ellipse(v.0.4.3)*, *lmodel2(v.1.7-3)*, *ggsci(v.2.9)*, *scales(v.1.2.0)*, *ggrepel(v.0.9.1)*, *cowplot(v.1.1.1)*, *ggplot2(v.3.3.6)*, *tidyquant(v.1.0.4)*, *quantmod(v.0.4.20)*, *TTR(v.0.24.3)*, *PerformanceAnalytics(v.2.0.4)*, *xts(v.0.12.1)*, *zoo(v.1.8-10)*, *lubridate(v.1.8.0)*, *tidyr(v.1.2.0)* and *dplyr(v.1.0.8)*

loaded via a namespace (and not attached): *httr(v.1.4.3)*, *viridisLite(v.0.4.0)*, *jsonlite(v.1.8.0)*, *splines(v.4.1.3)*, *carData(v.3.0-5)*, *assertthat(v.0.2.1)*, *highr(v.0.9)*, *yaml(v.2.3.5)*, *backports(v.1.4.1)*, *pillar(v.1.7.0)*, *lattice(v.0.20-45)*, *glue(v.1.6.2)*, *quadprog(v.1.5-8)*, *digest(v.0.6.29)*, *minqa(v.1.2.4)*, *colorspace(v.2.0-3)*, *sandwich(v.3.0-2)*, *htmltools(v.0.5.2)*, *Matrix(v.1.4-1)*, *plyr(v.1.8.7)*, *pkgconfig(v.2.0.3)*, *broom(v.0.8.0)*, *xtable(v.1.8-4)*, *mvtnorm(v.1.1-3)*, *lme4(v.1.1-29)*, *tibble(v.3.1.6)*, *farver(v.2.1.0)*, *generics(v.0.1.2)*, *car(v.3.1-0)*, *ellipsis(v.0.3.2)*, *TH.data(v.1.1-1)*, *withr(v.2.5.0)*, *cli(v.3.2.0)*, *survival(v.3.3-1)*, *magrittr(v.2.0.2)*, *crayon(v.1.5.1)*, *estimability(v.1.3)*, *evaluate(v.0.15)*, *fansi(v.1.0.2)*, *nlme(v.3.1-158)*, *MASS(v.7.3-57)*, *tools(v.4.1.3)*, *lifecycle(v.1.0.1)*, *multcomp(v.1.4-19)*, *stringr(v.1.4.0)*, *munsell(v.0.5.0)*, *compiler(v.4.1.3)*, *rlang(v.1.0.2)*, *grid(v.4.1.3)*, *nloptr(v.2.0.3)*, *rstudioapi(v.0.13)*, *labeling(v.0.4.2)*, *rmarkdown(v.2.14)*, *boot(v.1.3-28)*, *gtable(v.0.3.0)*, *codetools(v.0.2-18)*, *abind(v.1.4-5)*, *DBI(v.1.1.3)*, *curl(v.4.3.2)*, *R6(v.2.5.1)*, *knitr(v.1.39)*, *fastmap(v.1.1.0)*, *utf8(v.1.2.2)*, *Quandl(v.2.11.0)*, *stringi(v.1.7.6)*, *Rcpp(v.1.0.8.3)*, *vctrs(v.0.3.8)*, *tidyselect(v.1.1.2)*, *xfun(v.0.30)* and *coda(v.0.19-4)*