

# Calculating evapotranspiration

## Objectives

In this data lab, students will gain practice in applying functions, visualizing data, and interpreting the outputs while working with water fluxes at two spatial and temporal scales.

## Background

We will first use the Thornthwaite (1948) method to calculate potential evapotranspiration ( $E_p$ ), which relies on two key assumptions. First, this approach assumes that net radiation is equivalent to latent heat:

$$R_n \approx \lambda E$$

In Thornthwaite's time, net radiation was not commonly measured, so his empirical approach also assumes that mean monthly air temperature (MMAT) is a good proxy for  $R_n$ . Thus, we can only obtain Thornthwaite  $E_p$  at the monthly scale.

Chapter 12.3 of the Bonan text introduces the Budyko curve. We will calculate  $E_p$  with Thornthwaite and then use the fitted Budyko relationship to estimate  $E$ . Finally, we will plot the monthly values from two contrasting sites to see how water and energy limitation vary seasonally.

In Part II, we will use daily, plant-level sapflux data and invert a simplified form of Penman-Monteith to obtain surface conductance for nine species of riparian trees. Surface conductance of plant canopies vary widely between species and over time, with implications for plant carbon uptake.

## Part I: Budyko curves at NEON sites

NEON is a federally funded network of ecological observatories that endeavor to provide harmonized, continental-scale ecological data to improve understanding of the earth system. We will be calculating monthly  $E_p$  and  $E$  to plot a Budyko curve for two contrasting sites.

### Q1a (0 points)

First, use the NEON [site explorer tool](#) to select two sites with contrasting vegetation and climate, including one you suspect is water-limited and one you suspect is energy-limited. Note their 4 letter codes here.

- Moab NEON (MOAB), probably water-limited
- Lower Tombigbee River NEON (TOMB), probably energy-limited

### Q1b (0 points)

Then, read in the 'PRISM\_env\_NEON.csv' with the tidyverse package. These climate variables are 30-year normals from 1981-2010 that I extracted from a gridded [PRISM product](#). Please see the companion script 'get\_PRISM\_data.R' for further details.

```
library(tidyverse)

raw <- read_csv("PRISM_env_NEON.csv")
```

Many tidyverse functions exist for selecting, filtering, and otherwise manipulating data. These functions can be strung together using a pipe operator, or |>.

### Q1c (0 points)

Modify the code below to reflect the two sites you have chosen. Site names and other character strings need to be enclosed in double quotes.

```
env <- raw |>
  filter(field_site_id %in% c('MOAB', 'TOMB'))

#View(env)
```

### Q1d (0 points)

Next, examine the data to see what is present. How many types of **var** are included, and what units do you infer for the **value** column?

- 2 types of **var**, precipitation (ppt) and mean monthly air temperature (tmean)
- mm precipitation and degrees Celsius

## Q2 (0.25 points)

We need to use the `var` of 'tmean' to calculate Thornthwaite  $E_p$ . Examine equations 10.1-10.3 from the Bonan text. How many variables do we need for this calculation?

- 3 variables: L, N, T

The package 'SPEI' has a `thornthwaite()` function. Install this package (first time only; then re-comment out so the file will render), load the library, and type `?thornthwaite` into your console to see the documentation.

```
# install.packages("SPEI")  
library(SPEI)
```

```
# Package SPEI (1.8.1) loaded [try SPEInews()].
```

## Q3 (0.25 points)

What are the inputs needed by `thornthwaite()`? How do these inputs map onto the variables in 10.1-10.3?

- 'Tave' (MMAT) and 'lat' (latitude)
- Tave -> T, lat helps to calculate day length

## Q4 (0.25 points)

Now, let's isolate the variable `tmean` (equivalent to MMAT or Tave) to calculate Thornthwaite's  $E_p$ . Write some code to divide the dataframe into one for each variable.

```
temp <- env |>  
  filter(var == 'tmean')  
  
ppt <- env |>  
  filter(var == 'ppt')
```

Because we have two sites, we need to use `group_by()` in order to calculate  $E_p$  separately for each site.

```

Ep <- temp |>
  group_by(field_site_id) |>
  reframe(Ep_thorn = thornthwaite(Tave = value,
                                lat = unique(lat))) |>

  # Add the months back in
  mutate(month = rep(1:12, 2)) |>
  # Join ppt back in
  left_join(ppt, by = join_by(field_site_id, month)) |>
  # Tidy up
  rename(ppt_mm = value) |>
  select(-temp, -var) |>
  relocate(Ep_thorn, .after = ppt_mm)

```

```

[1] "Checking for missing values (`NA`): all the data must be complete. Input type is vector
[1] "Checking for missing values (`NA`): all the data must be complete. Input type is vector

```

Examine the output of this code chunk to ensure that it looks correct.

Next, we will program the Budyko curve function given by 12.6 in the Bonan text. First, re-write the 12.6 in terms of  $E_p$ .

### Q5 (0.25 points)

I have provided an empty R function below. Please populate it with your rewritten version. It can be easier to break up complicated functions into components, if only to reduce errors in parentheses.

```

budyko <- function(Ep, ppt) { # these are the two input variable names
  part1 <- Ep*ppt
  part2 <- tanh(ppt/Ep)
  part3 <- 1 - cosh(Ep/ppt) + sinh(Ep/ppt)

  E <- (part1*part2*part3)^(1/2)

  return(E)
}

```

Test this function in your console to assure yourself that it works. Choose reasonable values for  $E_p$  and  $ppt$ . The output  $E$  should always be less than  $ppt$ .

### Q6 (0.25 points)

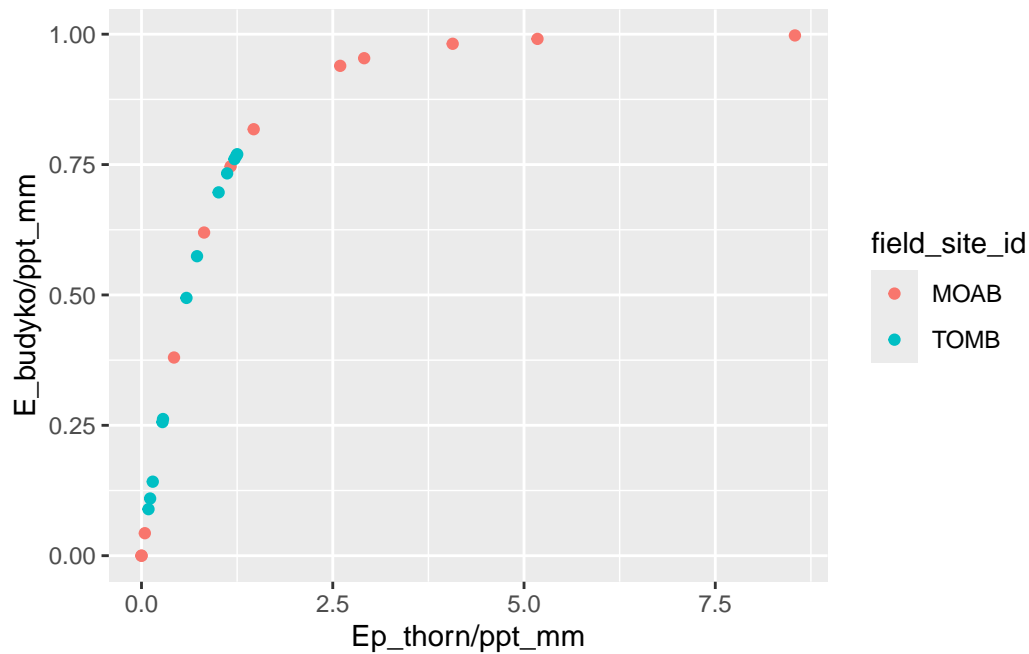
If you are reasonably sure that this function is working correctly, then the next step is to apply it across the dataframe. We will use `mutate()` to generate a new column.

```
Ep <- Ep |>
  mutate(E_budyko = budyko(Ep_thorn, ppt_mm))
```

### Q7 (0.25 points)

Let's check that our calculations are correct by plotting them. Use your knowledge of `ggplot()` to start a simple scatterplot of  $E/P$  vs.  $E_p/P$ , and color each site differently. Note that you can perform calculations within the x and y arguments of `aes()`.

```
Ep |>
  ggplot() +
  geom_point(aes(y = E_budyko/ppt_mm,
                 x = Ep_thorn/ppt_mm,
                 color = field_site_id))
```



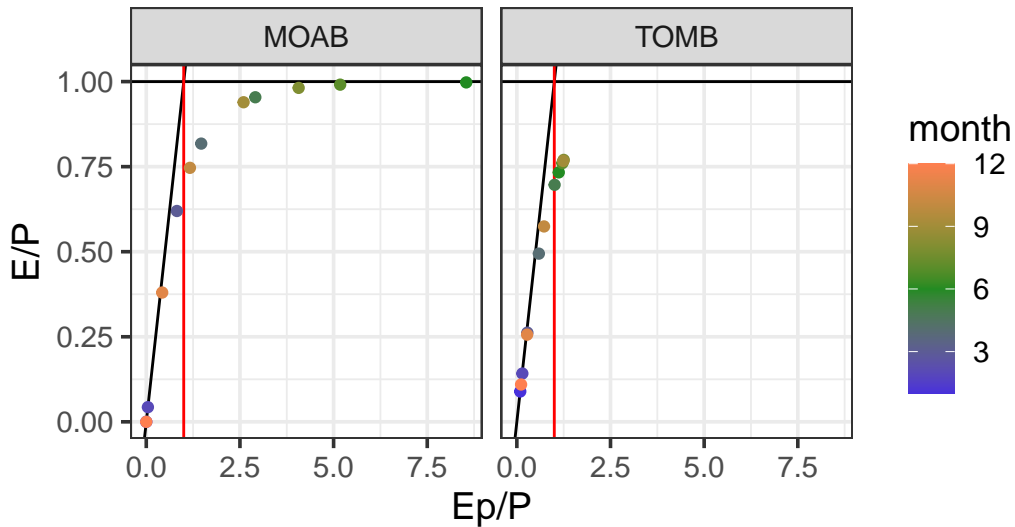
### Q8 (0.5 points)

To enhance your interpretation of the plots, use `facet_wrap()` to arrange one site per panel and two rows. Represent each month by color and use `scale_color_gradient2()` to select a 3-point color ramp. Add a vertical line at 1 to denote the energy- and water-limitations. Finally, add `theme_bw()`.

```
Ep |>
  ggplot() +
  geom_hline(yintercept = 1) +
  geom_abline(slope = 1) +
  geom_vline(xintercept = 1,
             color = 'red') +
  geom_point(aes(y = E_budyko/ppt_mm,
                 x = Ep_thorn/ppt_mm,
                 color = month)) +
  scale_color_gradient2(low = "blue", mid = "forestgreen", high = "coral",
                        midpoint = 6) +
  facet_wrap(~field_site_id) +
  labs(title = 'Budyko Curves',
       subtitle = 'Moab, UT & Tombigbee River, AL',
       x = 'Ep/P',
       y = 'E/P') +
  theme_bw(base_size = 14)
```

## Budyko Curves

Moab, UT & Tombigbee River, AL



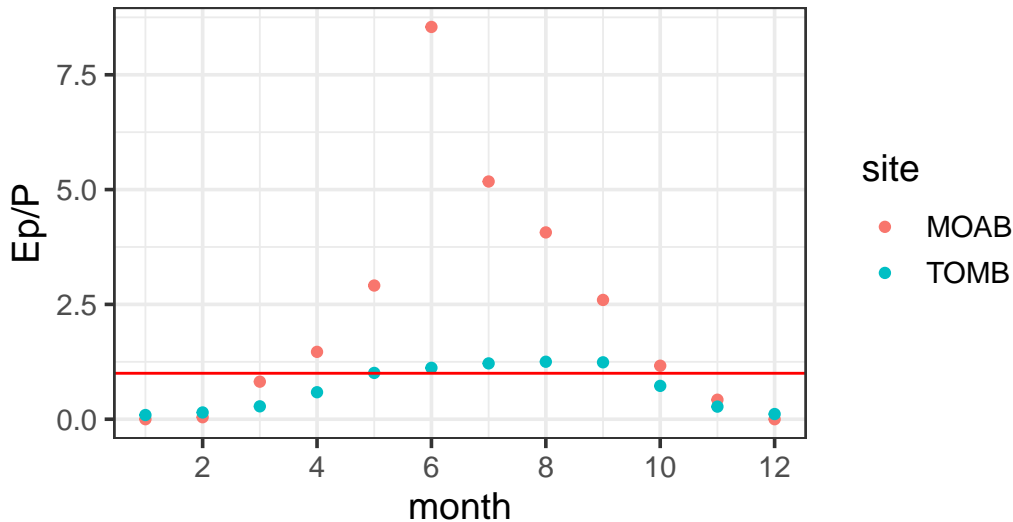
### Q9 (0.5 points)

Finally, let's visualize the same data as a time series of  $E_p/P$ . Create another figure wherein month is on the x-axis,  $E_p/P$  is on the axis, and the two sites are different colors. Add a line representing the division between energy- and water-limitation.

```
Ep |>
  ggplot() +
    geom_point(aes(y = Ep_thorn/ppt_mm,
                  x = month,
                  color = field_site_id)) +
    geom_hline(yintercept = 1,
              color = 'red') +
    labs(title = 'Timeseries of Ep/P',
         subtitle = 'With threshold between energy-limited and water-limited',
         y = 'Ep/P',
         color = 'site') +
    scale_x_continuous(breaks = (c(1:6)*2)) +
    theme_bw(base_size = 14)
```

## Timeseries of $E_p/P$

With threshold between energy-limited and water-limited



### Q10 (1.5 points)

Using both figures and any additional code necessary, address the following questions. Refer to the Budyko figure as Fig. 1 and the time series plot as Fig. 2. Read the site descriptions from [NEON](#) for additional insight, and cite as needed.

- For each site, what are the limitations to evapotranspiration and in which months do they occur?
  - Moab varies greatly, because it's very dry, and so  $E_p/P$  depends heavily on temperature, which is very seasonal (Fig. 2). energy-limited in the winter (November through March), and water-limited for the rest of the year when the temperature is high and outpaces precipitation in arid Utah.
  - Tombigbee experiences a similar pattern, with variable temperatures throughout the year, however precipitation is instead consistently high year-round (NEON). It therefore requires very high temperatures for  $E_p$  to exceed  $P$ . Most of the year, due to this precipitation, it is energy-limited, but when temperatures are highest in the summer, May through September, it becomes very slightly water-limited (Fig. 2).
- On average, were your sites energy- and water- limited as you expected based on site climatology and vegetation?



- For the most part. Utah is dry and can get quite hot so it experiences the full length of the Budyko curve (Fig. 1). I was expecting the site in the humid South to be consistently energy-limited thanks to the very high amount of moisture and precipitation available there, but it turns out it also gets hot enough that water can be the limiting factor instead for part of the year.
- How might the climate change factor of drought differentially impact evapotranspiration and therefore plant carbon uptake at your sites?
  - The term  $E_p/P$  is more sensitive when precipitation is low (Fig. 2). A drought resulting in a decrease in evapotranspiration would be more impactful on a site with less water available in the system to begin with, as such a system is more susceptible to evapotranspiration becoming water-limited. Transpiration, alongside carbon output, is dependent on stomatal conductance, so if plants undergo less transpiration, they will expel and therefore uptake less carbon. The humid, high-moisture ecosystem would be more resistant to these changes (Smith et al., 2024).
  - Since the limiting factor in these environments fluctuates seasonally, droughts would be more impactful during the summer, when they are water-limited. They may also contribute to the water-limited part of the year lengthening.

## Part II: Inverting a simplified Penman model

From class, recall that plant representation in the Penman-Monteith model of evapotranspiration was limited to the  $g_w$  term, or surface conductance to water. At the whole-plant scale, we can estimate transpiration via sap flow methods, and then use a simplified version of the Monteith model (Monteith & Unsworth 1990 Chapter 13) that assumes a canopy well-coupled with the bulk air.

### Q11 (0.25 points)

First, rearrange equation 13.42 (reproduced below) to isolate  $g_s$ . Recall that resistance is the inverse of conductance, so  $r_s = 1/g_s$ , and rewrite until the constants (Greek letters and  $c_p$ ) are separated from the state variables of  $E$  and  $D$ , or vapor pressure deficit.

$$\lambda E_i \approx \rho c_p D / (\gamma r_s)$$

Let's use values of  $\gamma = 0.0559 \text{ kPa } K^{-1}$ ,  $\lambda = 2300 \text{ J } g^{-1}$ ,  $\rho = 1020 \text{ g } m^{-3}$ , and  $c_p = 0.00119 \text{ J } g^{-1} K^{-1}$ .

- $y_l E / g = p_c D$
- $g = y_l E / p_c D$
- $g = E / D * 0.0559 * 2300 / 1020 * 0.00119$
- $g = 105.9235 * E / D$

### Q12 (0.5 points)

Create an R function to estimate canopy surface conductance  $g_s$  from  $E$  and  $D$ .

```
gs_monteith <- function(E, D){ # only inputs needed are the 2 state variables
  # declare constants
  gamma = 0.0559
  lambda = 2300
  rho = 1020
  cp = 0.00119
  # write rearranged Monteith equation
  gs <- (gamma * lambda * E) / (rho * cp * D)
  return(gs)
}
```

We will be working with a [dataset](#) of sapflow from riparian trees near Salt Lake City, Utah. Sapflow techniques involve inserting a pair of heating and sensing probes into the active sapwood (where the xylem is) of a tree, then using the principles of heat conduction and approximations of tree geometry to estimate the flow rate of water within the sapwood. This is one way to obtain tree-level, hourly to daily values of transpiration. In this case, we will use tree-level sapflux density as  $E$ ,  $g\ m^{-2}\ s^{-1}$  and bulk air vapor pressure deficit  $D$ ,  $kPa$ .

We are going to load in the “riparian\_sapflow.csv”. The customary abbreviation for sapflux density is  $J_s$ , and this dataset includes the mean daily sum of  $J_s$  for 9 species at 4 sites as well as the associated mean  $D$ .

### Q13 (0.25 points)

Use the function you created above to create a new column called **Gs**.

```
Js <- read_csv("riparian_sapflow.csv") |>
  mutate(species = factor(species, levels = c("P. fremontii", "T. ramosissima", "E. angustifolius",
                                              "P. hybrid", "S. hybrid",
                                              "A. negundo",
```

```

                                "A. grandidentatum", "B. occidentalis", "P. ang
site = factor(site, levels = c("Jordan", "Reservoir", "Todds", "Upper"))

```

Rows: 365 Columns: 8

-- Column specification -----

Delimiter: ","

chr (2): site, species

dbl (5): doy, Js\_mean, Js\_sd, n, Dmean

dtm (1): date

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

```

Js <- Js |>
  mutate(Gs = gs_monteith(Js_mean, Dmean))

```

Because I did not include the scaling factors from sapwood area to leaf area, the final units here are non-standard, but the trends over time and inter-species comparison can still be analyzed.

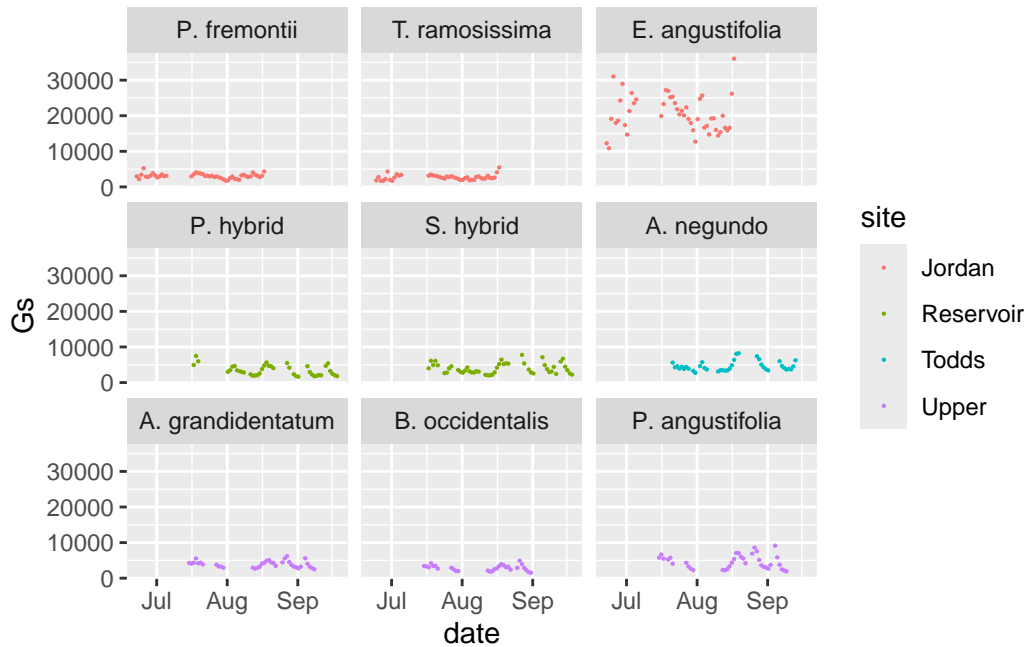
#### Q14 (0.25 points)

Plot the trend in **Gs** over time, faceting over the species and coloring the points by site. The sites are listed in elevation order from Jordan to Upper.

```

Js |>
  ggplot() +
    geom_point(aes(x = date,
                  y = Gs,
                  color = site),
              size = 0.1) +
    facet_wrap(~species,
              #scales = 'free'
              )

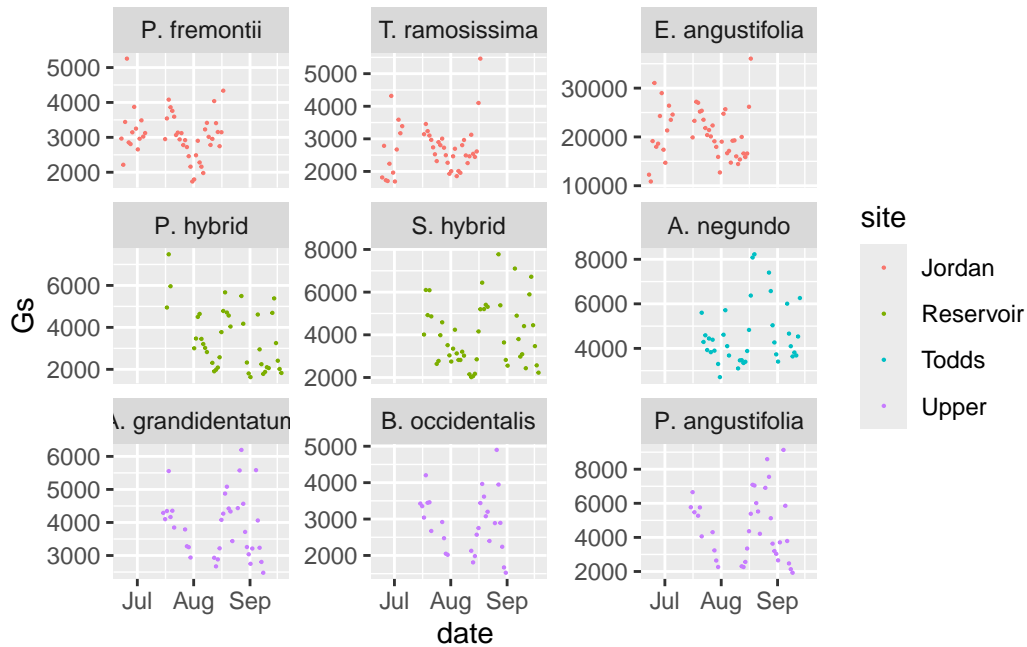
```



### Q15 (0.25 points)

One of the species has much higher  $G_s$  than the others, owing to much higher  $J_s$ . Try remaking the plot with the argument 'scale = "free\_y"' within `facet_wrap()`. What does this do?

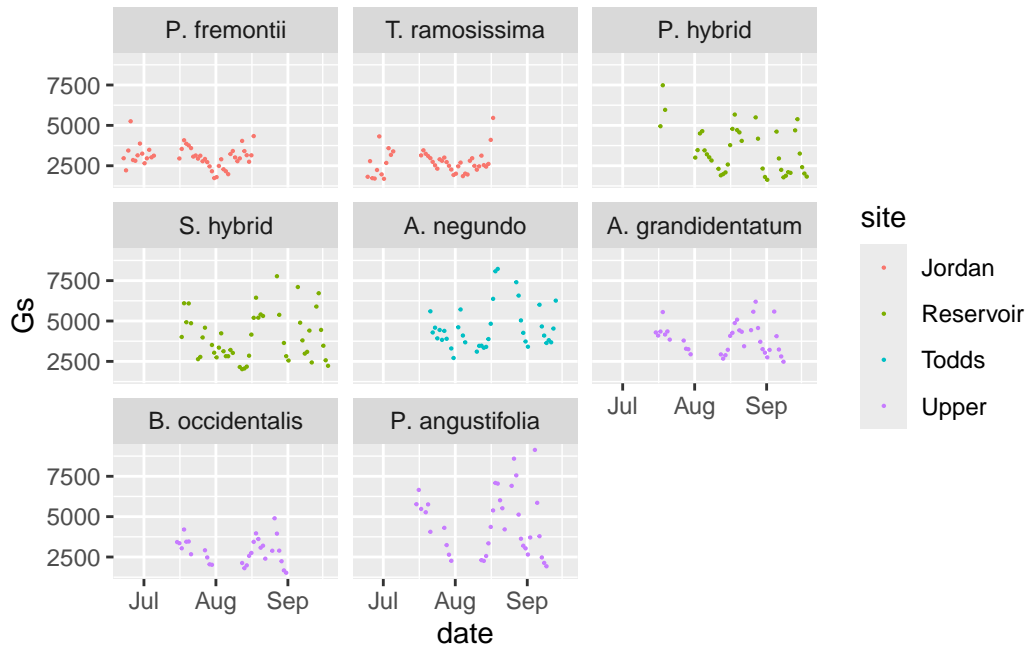
```
Js |>
ggplot() +
  geom_point(aes(x = date,
                 y = Gs,
                 color = site),
             size = 0.1) +
  facet_wrap(~species,
            scales = 'free_y'
            )
```



- This adds a y scale to each individual plot, which makes each one take up less space which makes it overall less readable, but you can see the similar fluctuating trends in each species.

You can also use the `filter()` function within `ggplot()` to remove *E. angustifolia*, which allows for easier comparison among the remaining species.

```
Js |>
  filter(species != 'E. angustifolia') |>
  ggplot() +
  geom_point(aes(x = date,
                  y = Gs,
                  color = site),
             size = 0.1) +
  facet_wrap(~species,
             #scales = 'free'
             )
```



### Q16 (2.5 points)

In this study, all trees could access groundwater such that their transpiration rate  $J_s$  is largely controlled by atmospheric demand  $D$  and species differences. The  $D$  also differs as the sites are arrayed along an elevation gradient from Jordan (1314 m) to Upper (1900 m).

- (i) Generally,  $D$  decreases with elevation due to the adiabatic rate of cooling. What might the elevation pattern of  $D$  look like under climate change? Explain your reasoning.
  - (i) Saturation vapor pressure is more sensitive to temperature at higher temperatures. VPD, therefore, is not only higher at warmer temperatures, but is more susceptible to changes at warmer temperatures. At lower altitudes, VPD will respond to global warming to a greater extent, while it won't be influenced as much at higher elevations. The disparity between VPD at different elevations will be widened.
- (ii) Excluding *E. angustifolia*, is there an elevational pattern in  $G_s$  across species? Why might this be so? Refer to your rearranged Monteith equation. What might happen to  $G_s$  across elevations under climate change?
  - (i)  $G_s$  is directly proportional to  $E$  and inversely proportional to  $D$  according to the equation. If the vapor pressure deficit goes down at high elevations due to water storage and temperature, that means that  $G_s$  would increase. Assuming that VPD changes disproportionately at different elevations with climate change,  $G_s$  would

become significantly lower at lower altitudes, and only slightly lower at higher altitudes.

- (iii) Both *T. ramosissima* and *E. angustifolia* are non-native species (more commonly known as tamarix and Russian olive). What is their  $G_s$  relative to the other native riparian trees? How might water use by non-native species play out ecologically under climate change, given the tradeoff between water use and  $CO_2$  uptake?
- (i) While tamarix'  $G_s$  is comparable to native species, especially the one at the same site, Russian olive has an extremely high  $G_s$ , despite similar conditions to other plants. It transpires at a rate nearly an order of magnitude greater than the other riparians, and it would have to take up a much higher amount of water to enable this.
- (ii) As atmospheric  $CO_2$  increases, stomatal density decreases, leading to lower water transport and transpiration. Less water would be available overall for native plants to utilize, and while stomatal conductance would facilitate greater water storage and conservation, it would also hinder those plants' ability to uptake necessary  $CO_2$ , overall contributing to native plants having a lower capability of acquiring necessary resources for respiration.