Assignment 6: Model Intercomparison for Potential Evapotranspiration

In homework #1 you wrote six functions to compute PET using six different methods. Now we will use those functions to explore the importance of PET and precipitation changes for water availability (measured as P - PET). By using multiple models and locations, you will be able to assess the importance of model uncertainty and spatial variability in making this assessment.

1 Retrieving the data (2 pts)

Choose five locations around the world that you wish to evaluate, e.g., research sites, hometowns, cities of interest, etc. Download daily weather data for any ten-year period you like from the NASA POWER dataset, available at http://earth-www.larc.nasa.gov/cgi-bin/cgiwrap/solar/agro.cgi. The GUI allows you to pick the lat, lon, dates, and variables that you want. After clicking the "Yes" button (so that it gives you the ICASA ASCII format that it mentions), a text file will appear in your browser window (it might take a couple minutes for processing). You can right-click and save as a .txt to your local directory, which you can then call `read.table()` on.

Now, with only five sites, it's bearable to sit and wait for each download to finish, save the results, go back to the web form, and rinse and repeat for each site. For hundreds or even just dozens of sites, you'd be looking for a minion. As part of the assignment, we're going to write code that automates this process. You'll notice that when you submit the web form, the URL in your browser's address bar ends up containing a long string of parameters, which are exactly the lat, lon, etc. that define the file that the server is retrieving for you. We can build such a string programmatically, and have R download the file at that location for us (this sort of task really does come up in research sometimes).

You'll use the `paste()` and `download.file()` functions to stitch together the url and download the file to a specified location on your hard drive. Look at the `downloadFromURL.R` file for more help.

You'll notice that each site's elevation is in the file's metadata, just a few rows above the beginning of the table. The `read.table()` function will only read tabular values, so it won't be able to retrieve the elevation for you, and you'll also have to tell `read.table()` how many rows it needs to skip before the tabular values begin. It's fine for this assignment to just count the number of rows and read the elevation with your eyeballs. If you had hundreds of files, though, you'd want a way to pick off the elevation value programmatically. **Extra credit:** Can you figure out how? You'll probably need the `readLines()` and `strsplit()` functions. We don't recommend attempting this until you've done the rest of the assignment.

2 World Map Plotting (2 pts)

Make a map of these locations by showing a map of the world with circles for each location. An easy way to do this is to use the function `map()` in the library `maps`.

Annotate each circle with the average summertime temperature (T) and precipitation (P). Scale the size of the circle according to one of these values, using the function `symbols`, or the cex argument in `plot` or `points`.

3 (5 pts)

Compute total seasonal PET for each year for a set growing season (e.g,. June-August in N Hemisphere, but you are free to choose your season). Make a barplot that shows the mean and standard deviation across years computed for all six methods at each site (remember the functions `barplot2` or `errbar` or `plotCI` if you don't want to code error bars yourself). How do differences between methods compare to differences between sites or between years?

There's no one right way to do this, but your solution must use dplyr's `group_by()` capability in some fashion. Also, to get full credit (and to keep yourselves from going insane by turning 100 lines of code into 1000) you must keep all of your site-specific data in one data structure. One big dataframe is a possibility given dplyr's tools, but you might find a list more straightforward.

Note: Be careful to check if your site has missing values. They'll probably be coded as something like -9999. If so, replace all of those values with NA's (this is a one-liner). There's a function called `na.approx()` in the `zoo` package that will then fill in all the NA's in a dataframe with interpolations from neighboring column elements.

4 (2 pts)

Make a similar barplot for total seasonal P-PET and answer the following questions: How does the standard deviation in P-PET compare to the standard deviation in PET? What does this tell you about sources of variation in year-to-year soil moisture? Is the answer different based on the method?

5 (4 pts)

Decide on your own metric to reflect the relative importance of P vs. PET in driving year-to-year variability in P-PET. Make a plot of this metric by method and site.

6 (5 pts)

Now imagine you have some climate projections for your site, that have a range of warming from 1 to 3^oC with a mean of 2^o C, and a range of precipitation changes from -10% to 10% with a mean of 0%. (These are typical of mid-century projections by GCMs). Recompute P, PET, and P-PET for your sites for high and low T change (with mean P change), and high and low P change (with mean T change). Choose one of the PET methods to do this, no need to run for multiple methods (although you are welcome to).

If we assume that differences between climate models are a good measure of climate uncertainty, what do you conclude about sources of uncertainty in projecting changes in water availability by mid-century? How does this compare to the conclusions you made regarding sources of interannual variability in P-PET? Make a single plot that summarizes your results (it may contain multiple panels).