* 1. **Create** “**layers**” of CTD data
     + Input file: CTD\_FCR.csv
     + Take all the ctd data and select data for specific depths (based on the nutrient chemistry sampling depths)
     + Replace CTD depth with the sampling depth to create “layers” (e.g., 0.1m, 0.8m, etc.)
     + R File: 1\_FCR\_CTDlayers\_Site50\_ONLY\_replacedepth
     + Files created: FCR\_CTD\_50surf\_binned.csv, FCR\_CTD\_50\_binned.csv
  2. **Summarize** meteorological data on a daily basis
     + Calculate summary statistics (mean, median, sum, etc.) for meteorological data to go from hourly met data to daily statistics
     + R File: 2\_summarize\_daily\_met\_TIDY
  3. **Calculate** specific conductance for all CTD data at site 50 and remove CTD-generated specific conductance
     + Input file: FCR\_CTD\_50\_binned.csv
     + R File: 3\_calculate\_spcond\_CTD
     + File created: FCR\_CTD\_50\_binned.csv, same file but new specific conductance column is added
  4. **Format** fluoroprobe data
     + Pull out only relevant columns from 2017 data
     + Put 2014-2017 data together
     + Create “layers” and replace with proper depth labels
     + File created: Fluoro\_FCR50\_2014\_2017.csv
     + R File: 4\_format\_fluora
  5. **Calculate** light extinction coefficient
     + File created: FCR\_kd.csv
     + R File: 5\_calculate\_Kd.R
  6. **Merge** together data that is on a weekly timestep
     + Input files: CTD, chemistry, kd, inflow
     + Pull together all important data sets with some small formatting before merging to make sure everything will align properly
       - TP/TN and NH4/NO3NO2 ratio columns created
       - New columns for inflow chem data created
       - 0.8m chemistry data is now called 1.0m chemistry data
     + File created: FCR\_VT\_data\_2013\_2017.csv
     + R File: 6\_merge\_FCR\_data.R
  7. **Plot** at the data!
     + R File: 7\_plot\_fun.R
  8. Determine the **timestep** of the chlorophyll data we have
     + Create weekly divisions
     + Determine which weeks need to be interpolated (18 dates)
     + R file:
  9. **Interpolate** all data collected on a ~ weekly basis for weeks that were not sampled from the May 01-Oct 31 time frame
     + Input file: FCR\_VT\_data\_2013\_2017.csv
     + TN:TP and NH4NO3NO2:SRP ratio columns created with newly interpolated nutrient data
     + Dataset trimmed down to the May-Oct timeframe for 2013-2016 after interpolation
     + Column names changed (‘interp’ removed)
     + R file: 9\_interpolate\_allcollecteddata.R
     + File created: data\_interpolated\_MayOct13\_16.csv
  10. Using **inflow** data and **interpolated chemistry** data, calculate various summary statistics
      + Input files
        - Chemistry: data\_interpolated\_MayOct13\_16.csv
        - Inflow: inflow\_interpolation\_2013\_2017.csv
      + mean daily nutrients loads
      + mean daily residence time
      + mean daily water temperature at inflow (should this also include max and min??)
      + mean, min, max, and median flow (m3/s)
      + R file: 10\_inflow.R
      + File created: data\_interpolated\_plusinflowcalcs\_MayOct13\_16.csv
  11. **Weeks: Select** one datapoint per week and **calculate** some weekly summary identifiers
      + Input file: data\_interpolated\_plusinflowcalcs\_MayOct13\_16.csv
      + Randomly select one datapoint per week for weeks where there is more than one so that dataset has **one datapoint per week**
        - 2013-2016 only
        - 2017 CTD has missing data for chlorophyll so this is not currently included
      + Add week numbers
        - Week\_jul
          * “Julian” week (starts over each year)
          * 17-43
        - Week\_series
          * Week number within the series, but starts over each year
          * 1-27
        - Week\_cum
          * Cumulative week number
          * i.e., Oct 2013 (#27), May 2014 (#28)
          * 2016 does not have a week 1
          * 1-108
      + R file: 11\_weeks.R
      + File created: interpolated\_weeks\_2013\_2016.csv
  12. Merge together interpolated, weekly dataset and meteorological data
      + Input file: interpolated\_weeks\_2013\_2016.csv
      + R file: 12\_merge\_weekly\_met.R
      + File created: variables\_all\_2013\_2016.csv
  13. AR Lag
      + Input file: variables\_all\_2013\_2016.csv
      + Use package astsa to determine what time lag is necessary to include in auto-regressive chl model
        - AR lag = 1
      + Then add a new column to dataframe of chl at the previous timestep
      + **Keep dataframe for surface (0.1m) data only**
      + Input file: model\_2013\_2016.csv
      + R file: 13\_AR\_lag.R
      + File created: variables\_all\_pluslag\_2013\_2016.csv
  14. Transformations
      + Assess histograms of each variable to determine if and what transformation is needed to satisfy normality
      + Detailed notes on which variables are transformed can be found within the R script
        - Variables that included zeroes that required log transformation first had the lowest non-zero positive value added, then logged
      + Input file: model\_lag\_2013\_2016.csv
      + R file: 14\_transformations.R
      + File created: model\_transformed\_chlalog\_2013\_2016.csv, model\_transformed\_chlasqrt\_2013\_2016.csv
  15. Correlation coefficients
      + TWO R FILES: one for log transformed chla and one for sqrt transformed chla
        - R file: 15\_correlation\_coefficients\_chlalog.R
        - R file: 15\_correlation\_coefficients\_chlasqrt.R
      + Input file: TWO DIFFERENT INPUT FILES, one for each R script
        - model\_transformed\_chlalog\_2013\_2016.csv
        - model\_transformed\_chlasqrt\_2013\_2016.csv
      + Using package Hmisc, create correlation matrices for the dataframe
      + Correlation matrices were created for the entire dataset and for each year, 2013-2016
      + For each matrix, within groups of correlated variables, one variable was chosen for inclusion in the model based on
        - Start with variable that has the highest spearman’s r with chl and select from there based on:
        - 1) a visual assessment of which variable has the strongest relationship with chlorophyll, IF no clear relationship:
        - 2) the largest spearman’s r value between the variable and chlorophyll, IF no meaningful difference in spearman’s r:
        - 3) the variable with the most meaningful biological importance was chosen (e.g., water temp is chosen over air temp if 1 and 2 above cannot be determined)
      + File created: correlation\_matrix\_YEAR.csv
  16. Linear model iterations
      + Input file: model\_transformed\_2013\_2016.csv
        - Also selected correlation files from each year to subset to the needed variables for model analysis
      + Using packing MuMIn, determine which iterations of the global model have AICc within 2 units of difference
        - From this subset of models, create individual linear models, create visual and summary statistics and put into a table
      + Was forced to subset the 2013 dataset down to late June because there are NAs in some of the candidate predictor variables at the beginning of the dataset
      + R file: 16\_lm\_iterations
        - First run at lm iterations
        - Used 0.1m chl, square root transformed
      + R File: 16\_lm\_iterations\_chlasqrt\_1.0m\_predictable\_vs\_not\_2013
        - Chl data at 1.0m, square root transformed, 2013 only
        - Are predictable drivers (e.g., water temp, inflow, met variables) as good as non-predictable drivers (i.e., those selected without regard to predictability) at predicting chlorophyll?
          * R^2, RMSE, AICc if applicable
      + R File: 16\_lm\_iterations\_chlasqart\_1.0m\_predictabledrivers
        - Chla data at 1.0m, square root transformed, data from 2013 to 2016
        - Are the same predictable drivers important from year to year?
        - Look at linear models of chl from 2013, 2014, 2015, 2016, and 2013-2016
      + File created:
  17. Linear Model Diagnostic tables created
      + Used R Markdown to make nice exportable tables for model diagnostics
      + Input files: model\_transformed\_2013\_2016.csv, and the correlation matrices for selected variables
      + R File: Meeting\_11282018.Rmd
  18. Variability of drivers
      + Looking at the distribution of drivers throughout the time series
        - i.e., is there an extreme weather event that is driving phyto trends in one year? Was one year super high inflow? Etc.
      + R file: 18\_DriverVariability.R
      + Input files: model\_transformed\_chlasqrt\_2013\_2016.csv, Met\_FCR\_daily.csv
  19. Testing model predictions with hard-coded equations
      + Using selected driver variables and linear model coefficients, calculate predictions
      + R file: 19\_HardCodingModelEquations.R
  20. Validate model predictions with 2018 out of sample data
      + R file: 20\_2018\_Validation.R
  21. Calculate model assessment metrics
      + R file: 21\_model\_assessment\_metrics.R
  22. Add 2019 CTD data to dataset
      + R file: 22\_create2018\_2019\_CTD\_input.R
  23. Other scripts of importance
      + Compare\_CTD\_EXO\_overtime.R
        - Script which compares chla data (ug/L) from both the CTD and EXO sensors converted into ‘CTD units’ using a linear model
      + Compare\_interpolated\_and\_no\_interpolated\_training\_data.R
        - This file calculates model assessment metrics between training datasets which interpolate missing data and which do not interpolate any missing data
      + CTD\_layers\_2018.R
        - Calculate discrete layers from high spatial-resolution CTD data for 2018
      + High\_frequency\_AR\_analysis\_fullmodelselection.R
        - Gather predictable driver data and run linear model selection on a daily timestep to determine which drivers are significant
      + Lm\_iterations\_fortnightly\_data.R
        - Run model selection on data on a fortnightly timestep to determine which drivers are significant
      + Lm\_iterations\_with\_dischargelag.R
        - Run model selection with a lagged discharge term to determine if this is a better driver than discharge on the predicted timestep
      + Merge\_NLDAS\_years.R
        - Merge different years of NLDAS data
      + TrainingDataSummaryStats.R
        - Calculate summary stats (min, max, median, etc.) on training data