SWAT Output Post-Processing (Part 2)

1. Overview

(5 min)

In post-processing part 1, we were introduced to the different SWAT output files, learned about tidy data concepts, and started importing SWAT outputs into R/RStudio. Here we will learn how to plot these data in ways that are helpful to our research.

By the end of this workshop session you will be able to:

1. *describe* common figures for exploring and publishing SWAT results
2. *modify* R code to import and plot SWAT data

2. Importing Data

(10 min)

First set your working directory as we did in post-processing part 1.

```
setwd('/Users/ssaia/Documents/GitHub/ecohydro-modeling-workshop-mar2018/data/swat_
output_data')
# copy/paste the path to the data directory inside the function setwd()
# windows users may have to change /'s to \\'s when copy/pasting
```

Next let's load our package.

```
library(tidyverse) # do this each time you open a new script and need it
```

Now let's import the raw `output.rch` file.

```
# You will need to change what's between the 's here. It should be output.rch.
raw_rch_data <- read_table2('/Users/ssaia/Documents/GitHub/ecohydro-modeling-works
hop-mar2018/data/swat_output_data/output.rch', col_names=FALSE, skip=9)
```

Let's look at the data? What's different about it from the first time we tried importing it? Hint: Look closely at the modifications made to the `read_table()` function. You can read more about `read_table2()` by searching for the function in your plot/help window.

But there are still some problems. For example, there is no simulation year column.

```
head(raw_rch_data, n = 10)
```

```
## # A tibble: 10 x 51
##       X1    X2    X3    X4      X5     X6     X7         X8    X9    X10
##    <chr> <int> <int> <int>   <dbl>  <dbl>  <dbl>      <dbl> <dbl>  <dbl>
##  1 REACH     1     0     1  157.50  3.712  3.706 0.0061770     0  787.4
##  2 REACH     2     0     1  191.70  4.410  4.404 0.0063450     0 1442.0
##  3 REACH     3     0     1   62.00  1.620  1.620 0.0005549     0  158.6
##  4 REACH     4     0     1  527.10 12.630 12.610 0.0150300     0 4963.0
##  5 REACH     5     0     1  180.40  4.689  4.680 0.0095240     0  548.1
##  6 REACH     6     0     1   79.87  1.936  1.936 0.0007220     0  187.2
##  7 REACH     7     0     1  103.10  2.758  2.756 0.0021910     0  880.6
##  8 REACH     8     0     1  699.60 16.830 16.820 0.0138400     0 5326.0
##  9 REACH     9     0     1  261.40  6.217  6.214 0.0032150     0 2177.0
## 10 REACH    10     0     1 1006.00 23.820 23.810 0.0142700     0 6058.0
## # ... with 41 more variables: X11 <dbl>, X12 <dbl>, X13 <dbl>, X14 <dbl>,
## #   X15 <dbl>, X16 <dbl>, X17 <dbl>, X18 <dbl>, X19 <dbl>, X20 <dbl>,
## #   X21 <dbl>, X22 <dbl>, X23 <dbl>, X24 <dbl>, X25 <dbl>, X26 <dbl>,
## #   X27 <dbl>, X28 <dbl>, X29 <dbl>, X30 <dbl>, X31 <dbl>, X32 <dbl>,
## #   X33 <dbl>, X34 <dbl>, X35 <dbl>, X36 <dbl>, X37 <dbl>, X38 <dbl>,
## #   X39 <dbl>, X40 <dbl>, X41 <dbl>, X42 <dbl>, X43 <dbl>, X44 <dbl>,
## #   X45 <dbl>, X46 <dbl>, X47 <dbl>, X48 <dbl>, X49 <dbl>, X50 <dbl>,
## #   X51 <dbl>
```

Ok, let’s reformat these raw data a little more using the `reformat_monthly_rch_file()` function.

```
source('reformat_monthly_rch_file.R') # add the reformat_rch_file() function to ou
r working environment

rch_data <- reformat_monthly_rch_file(raw_rch_data)
```

Let’s look at the result now. This should be more usable/tidy! :)

```
head(rch_data, n = 18)
```

```
## # A tibble: 18 x 50
##      rch month  year area_km2 flow_in_cms flow_out_cms  evap_cms tloss_cms
##    <int> <int> <int>    <dbl>       <dbl>        <dbl>     <dbl>     <dbl>
##  1     1     1  1997   157.50       3.712        3.706 0.0061770         0
##  2     2     1  1997   191.70       4.410        4.404 0.0063450         0
##  3     3     1  1997    62.00       1.620        1.620 0.0005549         0
##  4     4     1  1997   527.10      12.630       12.610 0.0150300         0
##  5     5     1  1997   180.40       4.689        4.680 0.0095240         0
##  6     6     1  1997    79.87       1.936        1.936 0.0007220         0
##  7     7     1  1997   103.10       2.758        2.756 0.0021910         0
##  8     8     1  1997   699.60      16.830       16.820 0.0138400         0
##  9     9     1  1997   261.40       6.217        6.214 0.0032150         0
## 10    10     1  1997  1006.00      23.820       23.810 0.0142700         0
## 11    11     1  1997   324.90       8.206        8.201 0.0045340         0
## 12    12     1  1997   102.60       2.457        2.455 0.0019000         0
## 13    13     1  1997  1426.00      33.710       33.690 0.0191500         0
## 14    14     1  1997   109.00       2.300        2.296 0.0036350         0
## 15    15     1  1997   256.40       5.048        5.042 0.0052540         0
## 16    16     1  1997    59.91       1.090        1.090 0.0003989         0
## 17    17     1  1997  1901.00      42.880       42.830 0.0412000         0
## 18     1     2  1997   157.50       3.947        3.940 0.0074620         0
## # ... with 42 more variables: sed_in_tons <dbl>, sed_out_tons <dbl>,
## #   sed_conc_mgperl <dbl>, orgn_in_kg <dbl>, orgn_out_kg <dbl>,
## #   orgp_in_kg <dbl>, orgp_out_kg <dbl>, no3_in_kg <dbl>,
## #   no3_out_kg <dbl>, nh4_in_kg <dbl>, nh4_out_kg <dbl>, no2_in_kg <dbl>,
## #   no2_out_kg <dbl>, minp_in_kg <dbl>, minp_out_kg <dbl>,
## #   chla_in_kg <dbl>, chla_out_kg <dbl>, cbod_in_kg <dbl>,
## #   cbod_out_kg <dbl>, disox_in_kg <dbl>, disox_out_kg <dbl>,
## #   solpst_in_mg <dbl>, solpst_out_mg <dbl>, sorpst_in_mg <dbl>,
## #   sorpst_out_mg <dbl>, reactpst_mg <dbl>, volpst_mg <dbl>,
## #   settlepst_mg <dbl>, resusppst_mg <dbl>, diffusepst_mg <dbl>,
## #   reactbedpst_mg <dbl>, burypst_mg <dbl>, bed_pst_mg <dbl>,
## #   bactp_out_ct <dbl>, bactlp_out_ct <dbl>, cmetal_num1_kg <dbl>,
## #   cmetal_num2_kg <dbl>, cmetal_num3_kg <dbl>, totn_kg <dbl>,
## #   totp_kg <dbl>, no3_conc_megagperl <dbl>, wtmp_deg_c <dbl>
```

Brainstorming Plots

(15 min)

Activity 1:

1. *(5 min on your own)* Think of some SWAT (or other model) papers you've read. What types of figure do you always see with regards to streamflow? Hint: If you were a hydrologist, what would you absolutely want to see? How might this change depending on the time step (i.e., daily, monthly, yearly) of your data and SWAT outputs?

2. *(5 min with a partner)* Share your figure ideas with a partner.
3. *(5 min class discussion)* Share what you and your partner discussed with the class.

Don't scroll down yet!

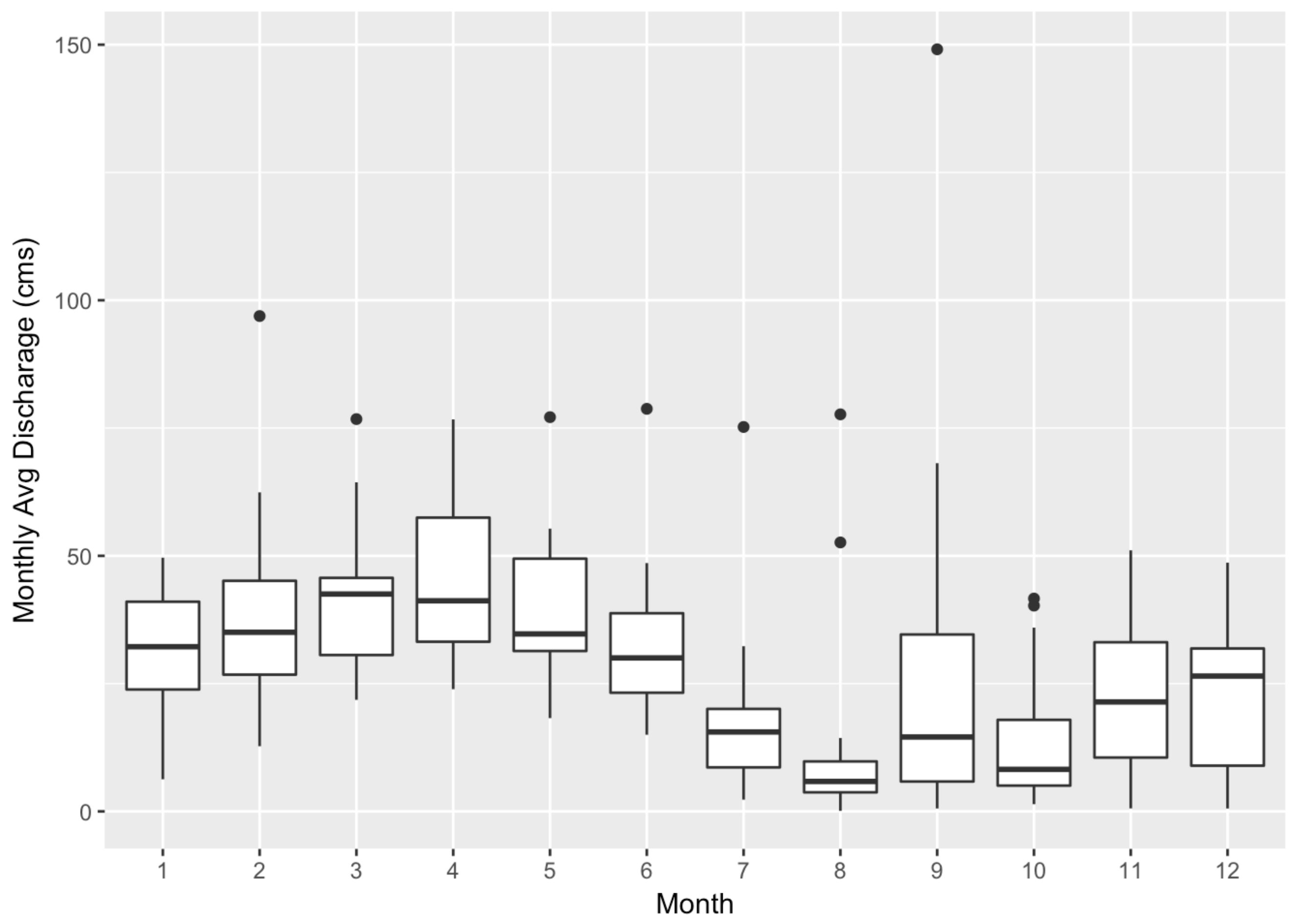
Plotting SWAT Outputs

(5 min)

Example 1 - Plot outlet simulation discharge data by month.

```
# select only outlet discharge data
sim_outlet_rch_data <- rch_data %>%
  filter(rch == 17) %>%
  mutate(date = paste0(month, "_", year))

# range of simulated outputs by month
ggplot(data = sim_outlet_rch_data, (aes(x = as.factor(month), y = flow_out_cms)))
+
  geom_boxplot() +
  xlab("Month") +
  ylab("Monthly Avg Discharage (cms)")
```

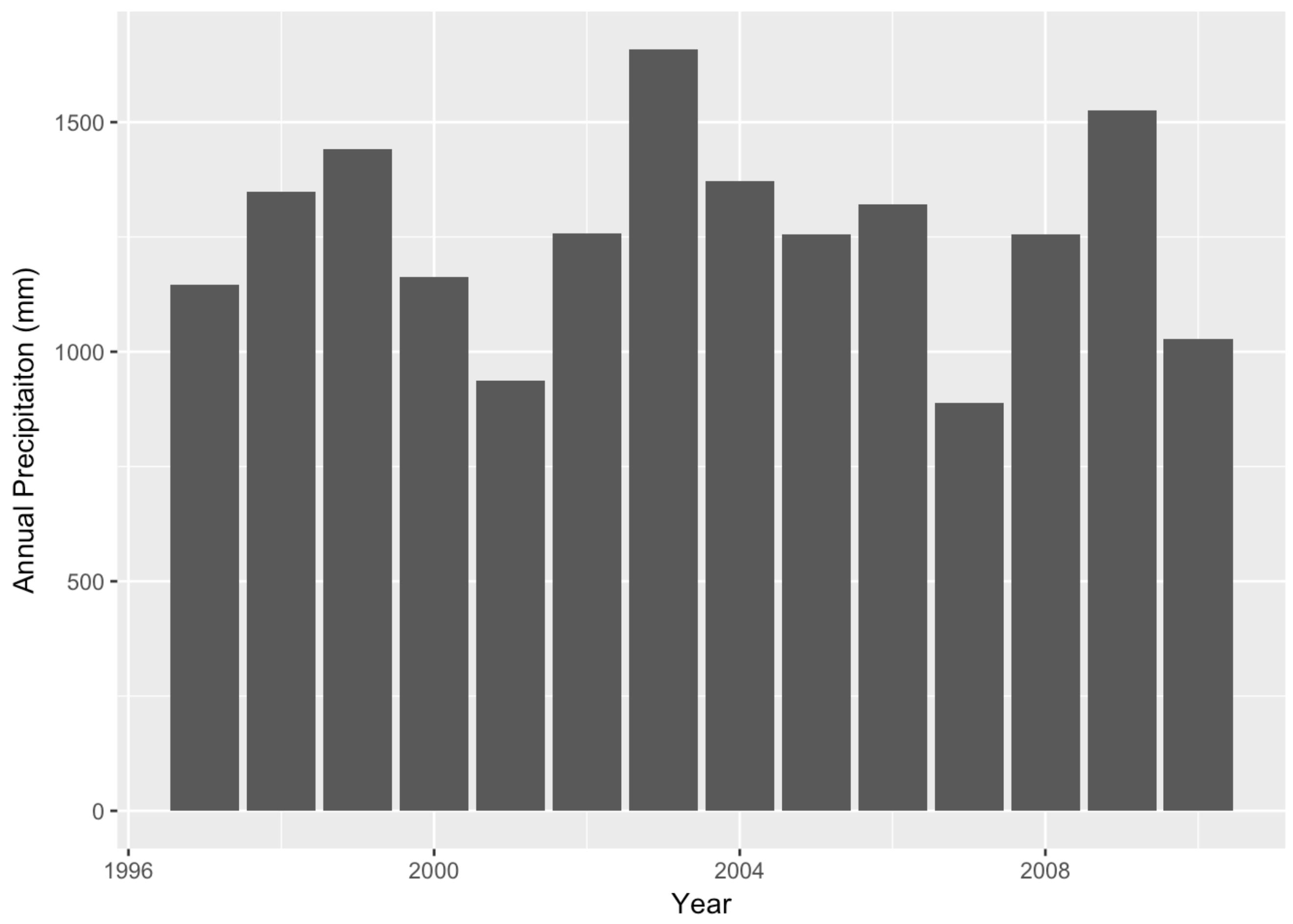


Example 2 - Plot annual precipitation total over time.

```
# import precip data
monthly_precip_data <- read_csv('/Users/ssaia/Documents/GitHub/ecohydro-modeling-w
orkshop-mar2018/data/swat_input_data/precip_data/p354-775_reformatted.csv', col_na
mes=TRUE)

# find annual sum
annual_precip_data <- monthly_precip_data %>%
  group_by(year) %>%
  summarize(annual_precip_mm = sum(monthly_precip_mm))

# bar chart
ggplot(data = annual_precip_data) +
  geom_bar(aes(x = year, y = annual_precip_mm), stat = "identity") +
  xlab("Year") +
  ylab("Annual Precipitaiton (mm)")
```



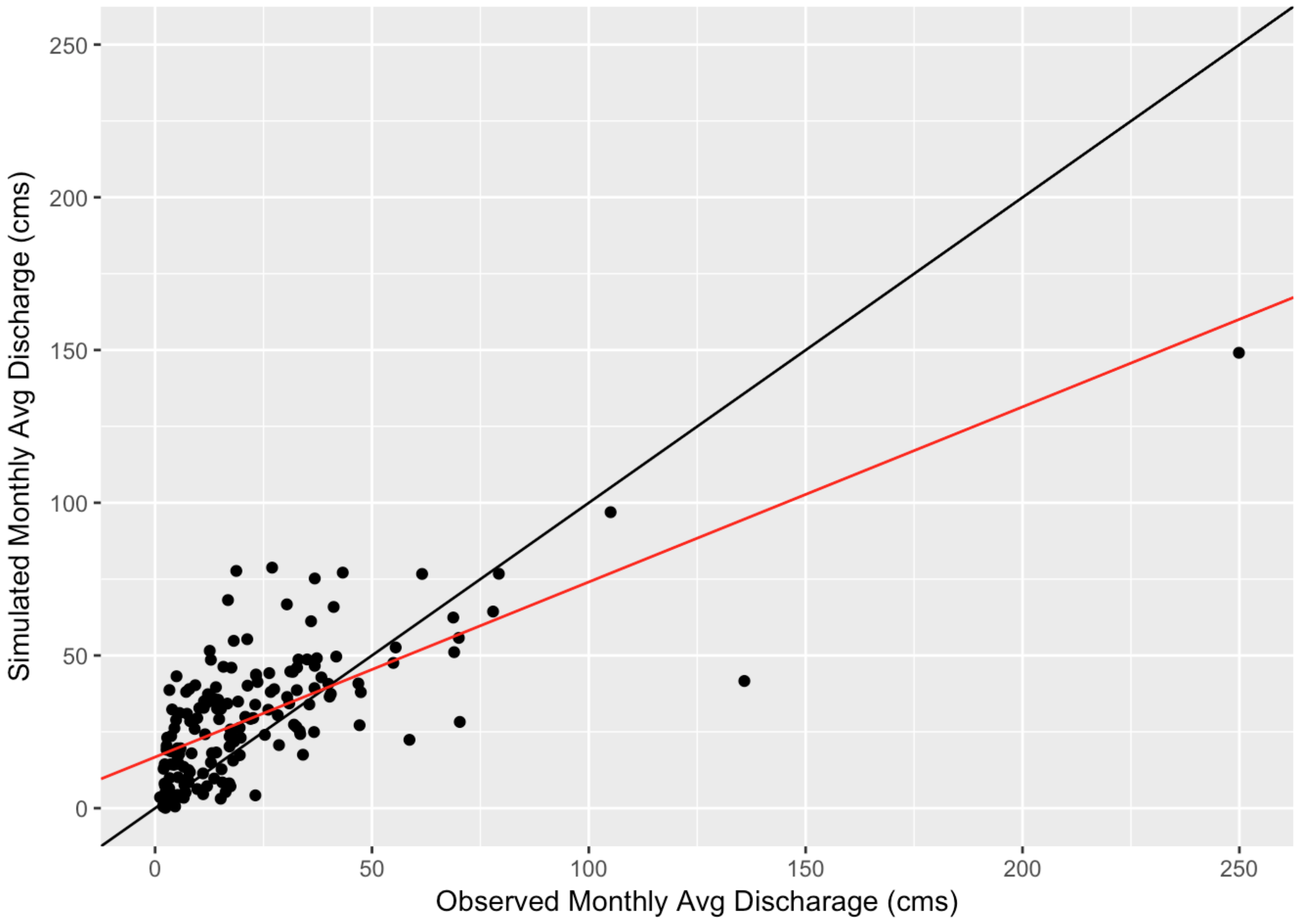
Example 3 - Plot SWAT simulated vs observed discharge data at the outlet.

```
# get observed discharge data
obs_outlet_rch_data <- read_csv('/Users/ssaia/Documents/GitHub/ecohydro-modeling-w
orkshop-mar2018/data/swat_input_data/streamflow_data/observed_monthly_outlet_flow.
csv', col_names=TRUE) %>%
  mutate(date = paste0(month, "_", year)) %>%
  select(-month, -year)

# join simulated and observed data into one table
sim_obs_outlet_rch_data <- left_join(sim_outlet_rch_data, obs_outlet_rch_data, by
= "date")

# fit a line to the simulated vs observed data and save linear model parameters
my_lm <- lm(flow_out_cms ~ obs_flow_out_cms, data = sim_obs_outlet_rch_data)
#summary(my_lm) # to see this uncomment it by removing the # before summary()
my_slope <- summary(my_lm)$coef[2]
my_intercept <- summary(my_lm)$coef[1]

# plot simulated vs observed monthly discharage with 1:1 line (black) and linear m
odel (red)
ggplot(data = sim_obs_outlet_rch_data) +
  geom_point(aes(x = obs_flow_out_cms, y = flow_out_cms)) +
  geom_abline(slope = 1, intercept = 0, color = "black") +
  geom_abline(slope = my_slope, intercept = my_intercept, color = "red") +
  xlim(0, 250) +
  ylim(0, 250) +
  xlab("Observed Monthly Avg Discharage (cms)") +
  ylab("Simulated Monthly Avg Discharge (cms)")
```



Your Turn - Modifying Code to Plot SWAT Inputs & Outputs

Activity 2:

(10 min)

1. *(5 min on your own)* Modify the code below (from example 1 above) to plot the range in monthly precipitation for each month. Try the bonus question below if you finish this.
2. *(5 min class discussion)* Share your modification with the class.

```
# code from example 1 above
ggplot(data = sim_outlet_rch_data, (aes(x = as.factor(month), y = flow_out_cms)))
+
  geom_boxplot() +
  xlab("Month") +
  ylab("Monthly Avg Discharage (cms)")

# don't forget to change the axes labels too!
```

Bonus - If you had daily data (instead of monthly data as we have here), how would you plot streamflow and precipitation together? Hint: You can use ggplot, or to make your life even easier, you can use a function in the `EcoHydRology` package (Go to Tools > Install Packages… > type in EcoHydrRology and look at the package functions in your help window).

Wrap up

I hope you can now:

1. *describe* common figures for exploring and publishing SWAT results
2. *modify* R code to import and plot SWAT data

Please fill out a paper teaching evaluation survey for me!