

# Assignment 7: High Frequency Data

*Rachel Bash*

## OVERVIEW

This exercise accompanies the lessons in Hydrologic Data Analysis on high frequency data

## Directions

1. Change “Student Name” on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single pdf file.
5. After Knitting, submit the completed exercise (pdf file) to the dropbox in Sakai. Add your last name into the file name (e.g., “A07\_Chamberlin.pdf”) prior to submission.

The completed exercise is due on 16 October 2019 at 9:00 am.

## Setup

1. Verify your working directory is set to the R project file,
2. Load the StreamPULSE, streamMetabolizer and tidyverse packages.
3. Set your ggplot theme (can be theme\_classic or something else)

```
getwd()
```

```
## [1] "C:/Users/19524/Documents/DUKE/Hydrologic Data Analytics/Hydrologic_Data_Analysis/Assignments"
```

```
packages <- c(  
  "tidyverse",  
  "StreamPULSE",  
  "streamMetabolizer"  
)
```

```
invisible(  
  suppressPackageStartupMessages(  
    lapply(packages, library, character.only = TRUE)  
  )  
)
```

```
theme_set(theme_classic())
```

4. Download data from the Stream Pulse portal using `request_data()` for the Kansas River, (“KS\_KANSASR”). Download the discharge (`Discharge_m3s`), dissolved oxygen (`DO_mgL`) and nitrate data (`Nitrate_mgL`) for the entire period of record
5. Reformat the data into one dataframe with columns `DateTime_UTC`, `DateTime_Solar` (using `convert_UTC_to_solartime()`), `SiteName`, `DO_mgL`, `Discharge_m3s`, and `Nitrate_mgL`.

```
#query_available_data(region="KS", site="KANSASR")
```

```
#in order to find start and end date of period of record
```

```
Kansasdat <- StreamPULSE::request_data(  
  sitecode = "KS_KANSASR",  
  variables = c('Discharge_m3s', 'DO_mgL', 'Nitrate_mgL'),
```

```

startdate = "2018-02-01",
enddate = "2018-05-31"
)

```

```

## You may omit the "variables" parameter to automatically retrieve
## all variables necessary for metabolism modeling.

```

```

##
## API call: https://data.streampulse.org/api?sitecode=KS_KANSASR&startdate=2018-02-01&enddate=2018-05-31
##
## Retrieved the following variables:
## DO_mgL, Discharge_m3s, Nitrate_mgL

```

```

Kansasdat.lon <- Kansasdat[[2]]$lon

```

```

Kansasdat.processed <- Kansasdat[[1]] %>%
  spread(value = value, key = variable) %>%
  mutate(DateTime_Solar = convert_utc_to_solartime(DateTime_UTC, Kansasdat.lon)) %>%
  select(DateTime_UTC, DateTime_Solar, site, DO_mgL, Discharge_m3s, Nitrate_mgL)

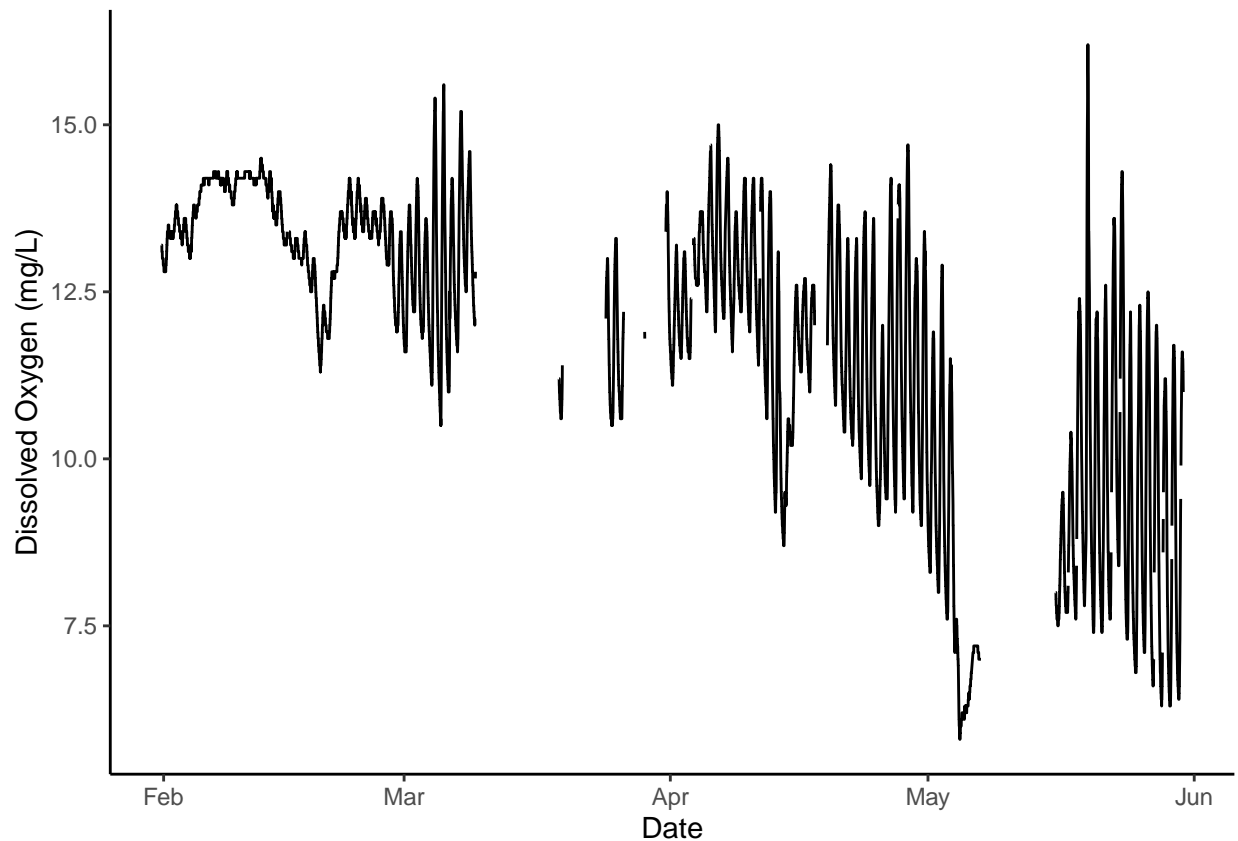
```

6. Plot each of the 3 variables against solar time for the period of record

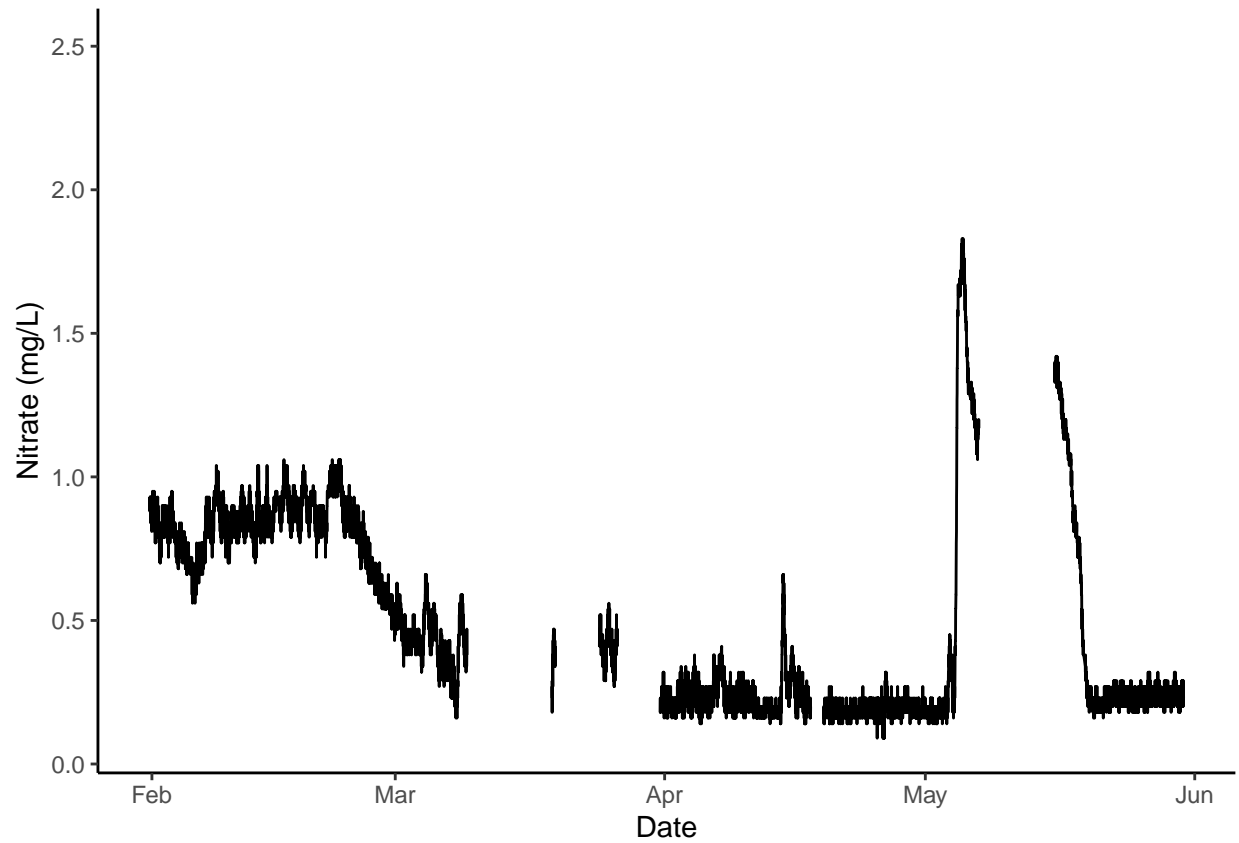
```

KansasDO <- ggplot(Kansasdat.processed) +
  geom_line(aes(x=DateTime_Solar, y = DO_mgL)) +
  labs(x= "Date", y = "Dissolved Oxygen (mg/L)")
print(KansasDO)

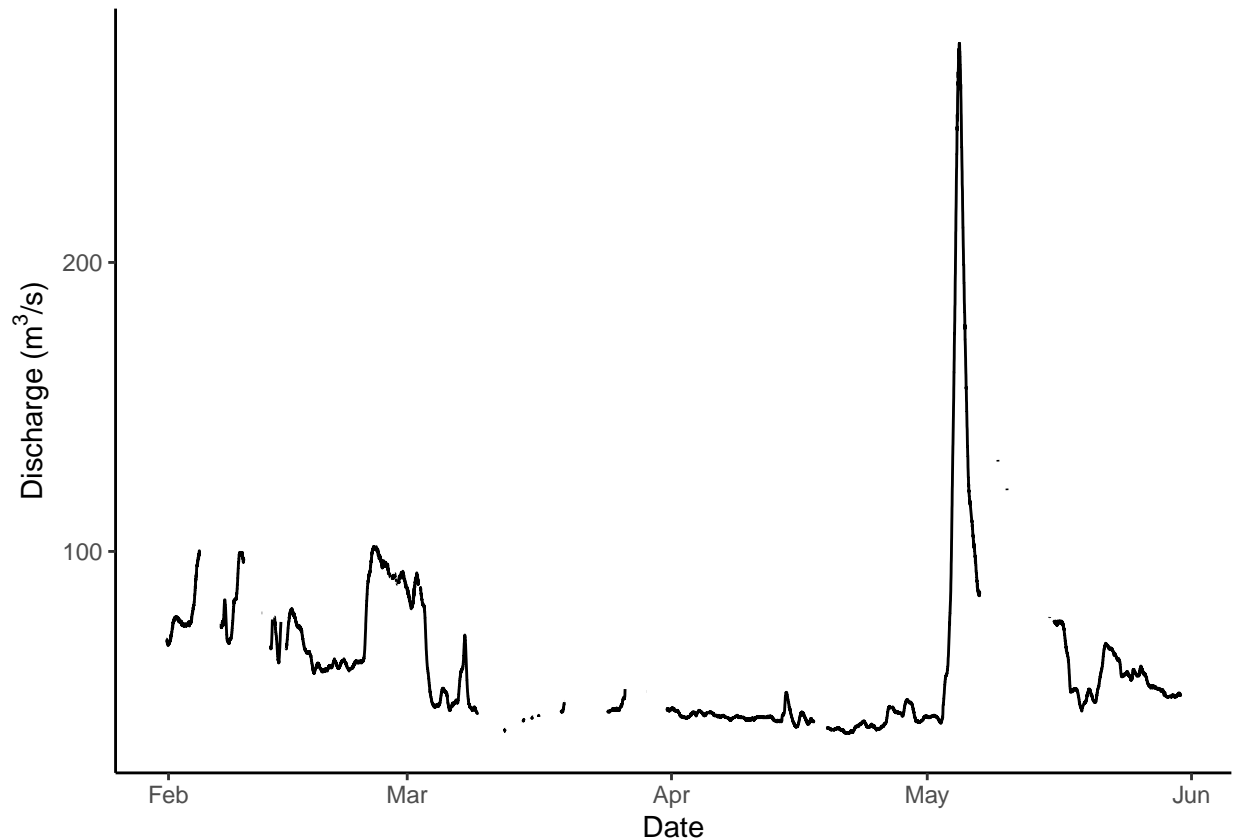
```



```
KansasNitrate <- ggplot(Kansasdat.processed) +
  geom_line(aes(x=DateTime_Solar, y = Nitrate_mgL)) +
  labs(x= "Date", y = "Nitrate (mg/L)")
print(KansasNitrate)
```



```
KansasDischarge <- ggplot(Kansasdat.processed) +
  geom_line(aes(x=DateTime_Solar, y = Discharge_m3s)) +
  labs(x= "Date", y = expression("Discharge (m3/s)"))
print(KansasDischarge)
```



7. How will you address gaps in these dataserries?

Interpolation.

8. How does the daily amplitude of oxygen concentration swings change over the season? What might cause this?

It appears that as the year progressed, oxygen concentration swings increased in size. This is because as it gets warmer out and it progresses to spring/summer, more plant growth occurs, thus photosynthesis and plant respiration. There is more activity as temp increases, causing a bigger swing in Oxygen levels.

## Baseflow separation

9. Use the `EcoHydRology::BaseflowSeparation()` function to partition discharge into baseflow and quickflow, and calculate how much water was exported as baseflow and quickflow for this time period. Use the `DateTime.UTC` column as your timestamps in this analysis.

The `package::function()` notation being asked here is a way to call a function without loading the library. Sometimes the `EcoHydRology` package can mask tidyverse functions like pipes, which will cause problems for knitting. In your script, instead of just typing `BaseflowSeparation()`, you will need to include the package and two colons as well.

10. Create a ggplot showing total flow, baseflow, and quickflow together.

```
#create small df so that interpolation can be done
Kansas.skinny <- Kansasdat.processed %>%
  select(DateTime.UTC, Discharge_m3s)
```

```

#determining number of time-steps
table(diff(Kansas.skinny$DateTime.UTC))

##
##      0   900  1800  4500 15300
## 2788 11291      1      2      7

#it looks like there are a substantial amount of observations taken at the same
#time, and a small portion of missing data

#therefore, n = # days * # timesteps in a day
as.Date("2018-05-31")-as.Date("2018-02-01")

## Time difference of 119 days
119*96

## [1] 11424

#interpolate by number of days in time period
linearinterpolation <- as.data.frame(approx(Kansas.skinny, n = 11424, method = "linear"))
linearinterpolation$x <- as.POSIXct(linearinterpolation$x, origin = "1970-01-01")
names(linearinterpolation) <- c("Date", "Discharge_m3s")

#plot interpolated data onto full data to see if it looks right
Kansas.interpolate.plot <-
  ggplot(Kansas.skinny, aes(x = DateTime.UTC, y = Discharge_m3s)) +
  geom_point() +
  geom_line() +
  geom_point(data = linearinterpolation, aes(x = Date, y = Discharge_m3s), color = "#c13d75ff")
#print(Kansas.interpolate.plot)

#separate baseflow and quickflow from total flow
Kansasbaseflow <- EcoHydRology::BaseflowSeparation(
  linearinterpolation$Discharge_m3s,
  filter_parameter = 0.925, #default parameter
  passes = 3 #default parameter
)

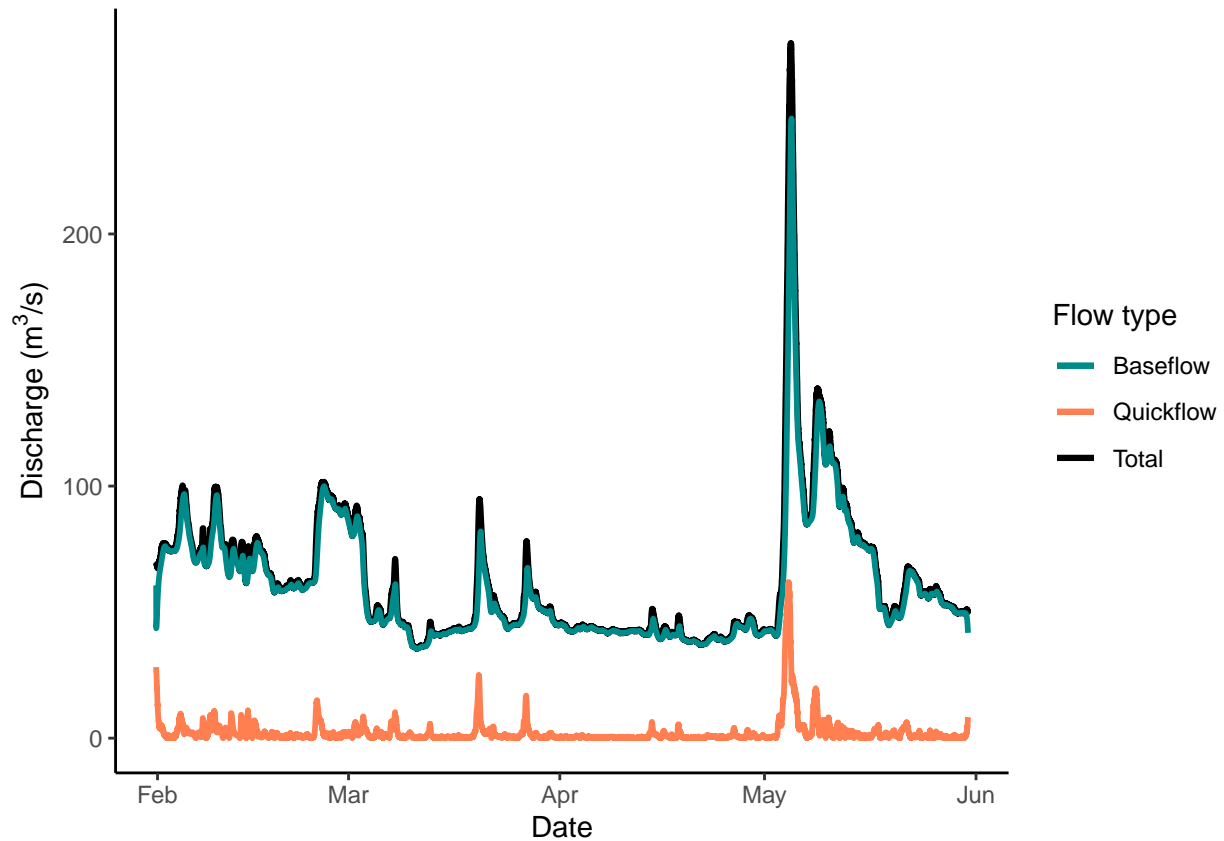
#add back quickflow and baseflow calculations to the total flow
Kansas.full <- cbind(linearinterpolation, Kansasbaseflow)

flowtype <- c("Total", "Quickflow", "Baseflow")

#plot the three lines on the same graph
Kansas.full.plot <- ggplot(Kansas.full, aes(x = Date)) +
  geom_line(aes(y = Discharge_m3s, color="Total"), size=1.1) +
  geom_line(mapping = aes(y = bt, color = "Baseflow"), size = 1.01) +
  geom_line(mapping = aes(y = qft, color="Quickflow"), size = 1.01) +
  labs(x="Date", y=expression("Discharge (m"3"/s)"), color="Flow type") +
  scale_colour_manual(values = c(
    'Total' = 'black',
    'Baseflow' = 'darkcyan',
    'Quickflow' = 'coral'))

print(Kansas.full.plot)

```



```
#Calculating total amount of water
Export <- Kansas.full %>%
  mutate(timestep = c(diff(as.numeric(Date))), NA_real_),
         baseflowexport = bt * timestep,
         quickflowexport = qft * timestep) %>%
  summarize(BaseflowExport_cf = sum(baseflowexport, na.rm = T),
            QuickflowExport_cf = sum(quickflowexport, na.rm = T),
            TotalExport_cf = BaseflowExport_cf + QuickflowExport_cf)

#baseflow proportion
Export$BaseflowExport_cf/Export$TotalExport_cf
```

```
## [1] 0.962135
```

```
#quickflow proportion
1-(Export$BaseflowExport_cf/Export$TotalExport_cf)
```

```
## [1] 0.03786495
```

11. What percentage of total water exported left as baseflow and quickflow from the Kansas River over this time period?

During this time period, 96.2% of total water exited as baseflow, while 3.8% of total water exited as quickflow.

12. This is a much larger river and watershed than the 2 we investigated in class. How does the size of the watershed impact how flow is partitioned into quickflow and baseflow?

In this situation, the larger river caused the majority of the total water to come from base flow.

This makes sense, as it would be harder for a large river to be supported by a large amount of rain over an extended period of time.

13. The site we are looking at is also further down in its river network (i.e. instead of being a headwater stream, this river has multiple tributaries that flow into it). How does this impact your interpretation of your results?

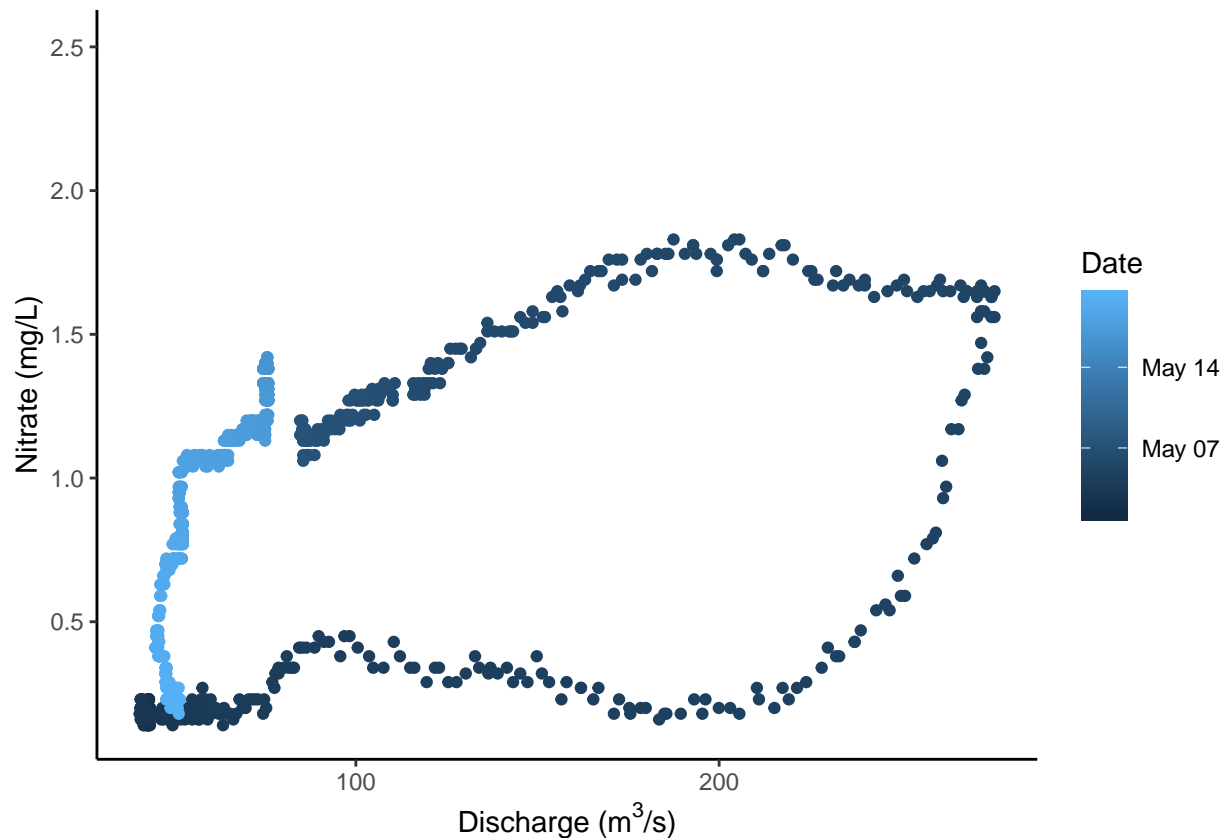
Multiple tributaries probably smooth out the results and contribute to the majority of the baseflow rate that is regularly being brought down the stream. Even if one tributary may experience a large storm, it would have a small effect on the larger river because of the multiple inputs to the river.

## Chemical Hysteresis

14. Create a ggplot of flow vs. nitrate for the large storm in May (~May 1 - May 20). Use color to represent Date and Time.

```
Kansasdat.hysteresis <- Kansasdat.processed %>%  
  filter(DateTime_UTC > "2018-05-01" & DateTime_UTC < "2018-05-20")  
  
Hysteresis.plot <- ggplot(Kansasdat.hysteresis) +  
  geom_point(aes(x=Discharge_m3s, y = Nitrate_mgL, color = DateTime_UTC)) +  
  labs(x=expression("Discharge (m3*/s)"), y = "Nitrate (mg/L)", color = "Date")  
print(Hysteresis.plot)
```

```
## Warning: Removed 1638 rows containing missing values (geom_point).
```



*#no scale color gradient will work except for the default*

15. Does this storm show clockwise or counterclockwise hysteresis? Was this storm a flushing or diluting storm?

Counterclockwise, flushing storm

16. What does this mean for how nitrate gets into the river from the watershed?

This means that nitrate concentrations increase at the beginning of the storm when quickflow is predominant, meaning that nitrate is coming from quickflow water. This is because nitrate is in the overland flow (likely from agriculture).

## Reflection

17. What are 2-3 conclusions or summary points about high frequency data you learned through your analysis?

There are 6 variations of possible hysteresis loops, and their direction and slope depends on where the nutrient is in the water system. I also learned that for large river systems, baseflow is the predominant source of water and is often steady, while quickflow from precipitation events causes the sharp increases in discharge volume/rate.

18. What data, visualizations, and/or models supported your conclusions from 17?

Visualizing the hysteresis loops are the only way to truly understand how and where nitrates come from in the system.

19. Did hands-on data analysis impact your learning about high frequency data relative to a theory-based lesson? If so, how?

Yes, definitely. Understanding the possible variations on hysteresis loops improved my understanding of them.

20. How did the real-world data compare with your expectations from theory?

Didn't change so much this time. They acted how I expected.