Assignment 7: High Frequency Data

Tristen Townsend

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] "/Users/Tristen/OneDrive - Duke University/Fall 2019/Hydrologic Data Analysis/Hydrologic_Data_Ana
```

```
library(streamMetabolizer)
```

```
## USGS Active Research Package:
## https://owi.usgs.gov/R/packages.html#research

## This package is in development. We are using it for our own
## applications and welcome flexible, resilient users who can help us
## make the package better. Details of the user interface and model
## implementations will change. Please give us feedback at
## https://github.com/USGS-R/streamMetabolizer/issues/new.

## Can't check GitHub for new package versions just now. We'll try again next time.
```

```
library(StreamPULSE)
```

```
## Loading required package: shiny

## Loading required package: Cairo

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v readr   1.3.1
## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   0.8.3     v stringr 1.4.0
## v ggplot2 3.2.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```r
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`), disolved 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.

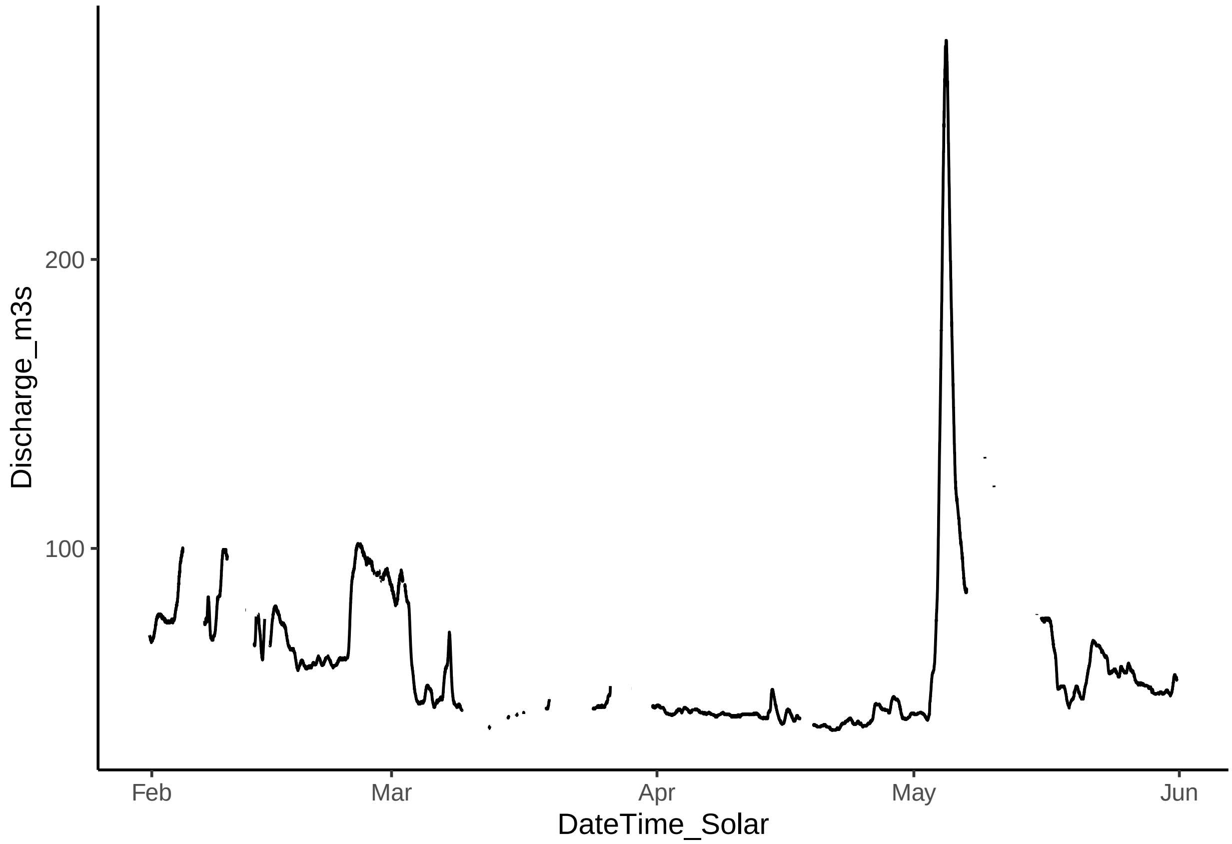
```r
Kansas.dat <- request_data(sitecode = "KS_KANSASR",
                 variables = c('Discharge_m3s', 'DO_mgL',
                               'Nitrate_mgL'))
```

```
## 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&variables=Discharge_m3s,DO_mgL,Nitrat
##
## Retrieved the following variables:
##   DO_mgL, Discharge_m3s, Nitrate_mgL
```

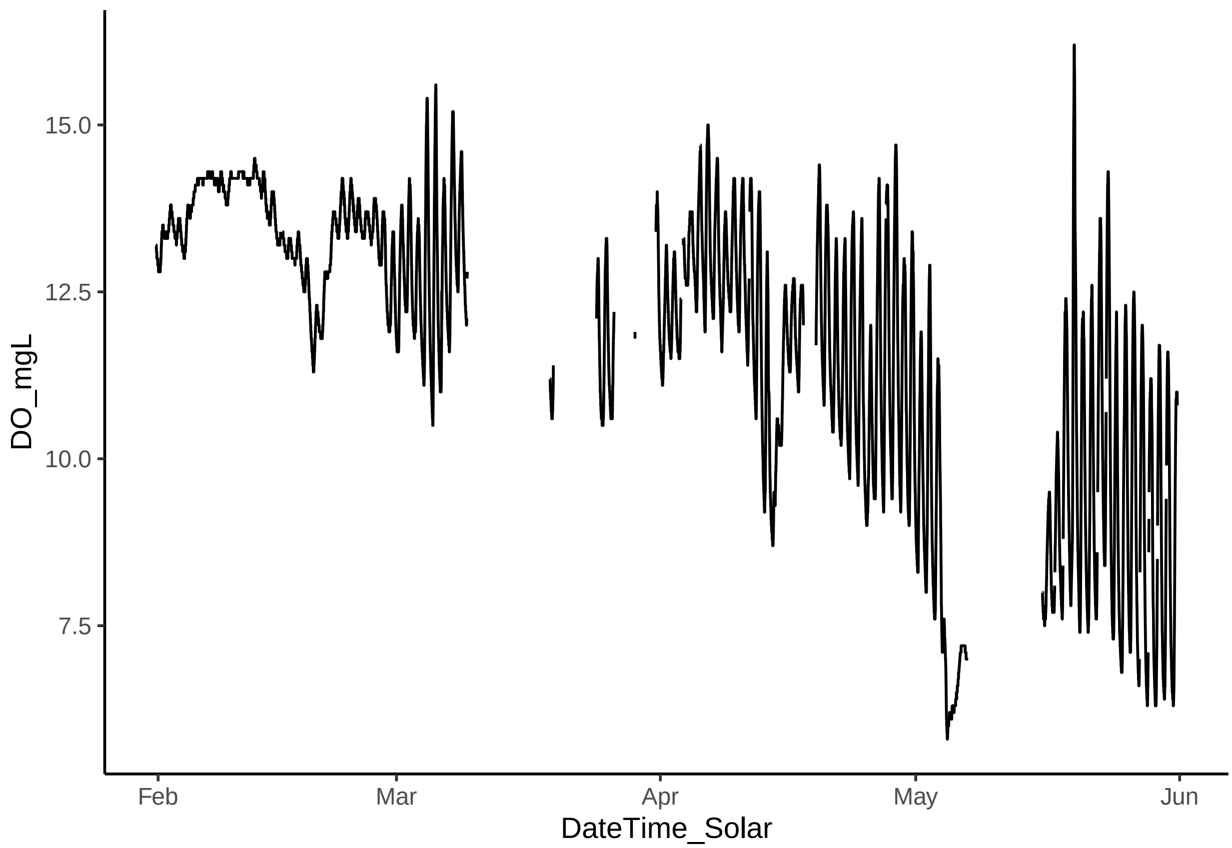
```r
Kansas.lon <- Kansas.dat[[2]]$lon

Kansas <- Kansas.dat[[1]] %>%
  spread(value = value, key = variable) %>%
  mutate(DateTime_Solar = convert_UTC_to_solartime(DateTime_UTC, Kansas.lon)) %>%
  select(-region, -site, -flagtype, -flagcomment)
Kansas <- Kansas[,c(1,5,2,3,4)]
```

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

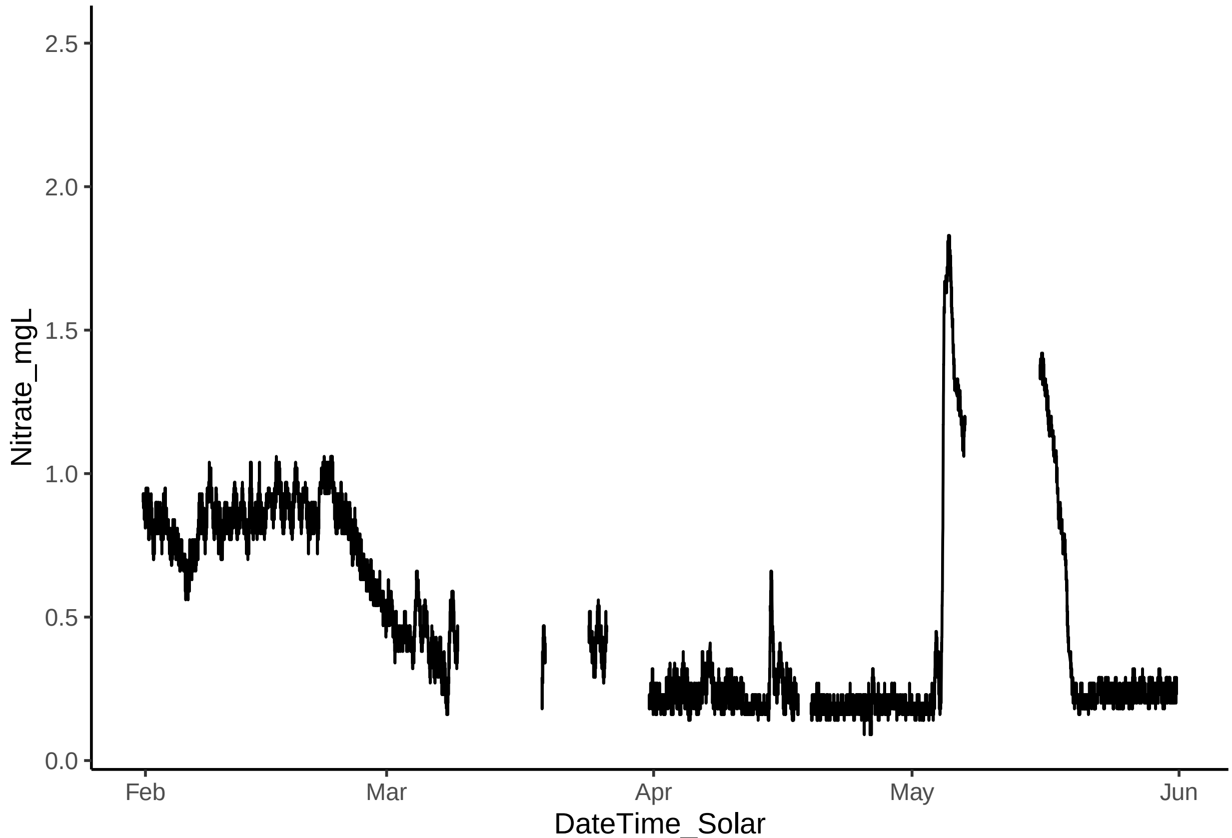
```r
ggplot(Kansas, aes(x = DateTime_Solar, y = Discharge_m3s)) + geom_line()
```




```
ggplot(Kansas, aes(x = DateTime_Solar, y = DO_mgL)) + geom_line()
```




```
ggplot(Kansas, aes(x = DateTime_Solar, y = Nitrate_mgL)) + geom_line()
```



7. How will you address gaps in these dataseries?

   Perform linear interpolation.

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

   Swings become more dramatic during late spring and early summer. This is likely due to increased primary productivity during these seasons, which would raise DO concentrations during the day and make decreases during the evening (when there's no sunlight) seem more pronounced. Also, this is the time algal blooms could occur and create hypoxic conditions.

Baseflow separation

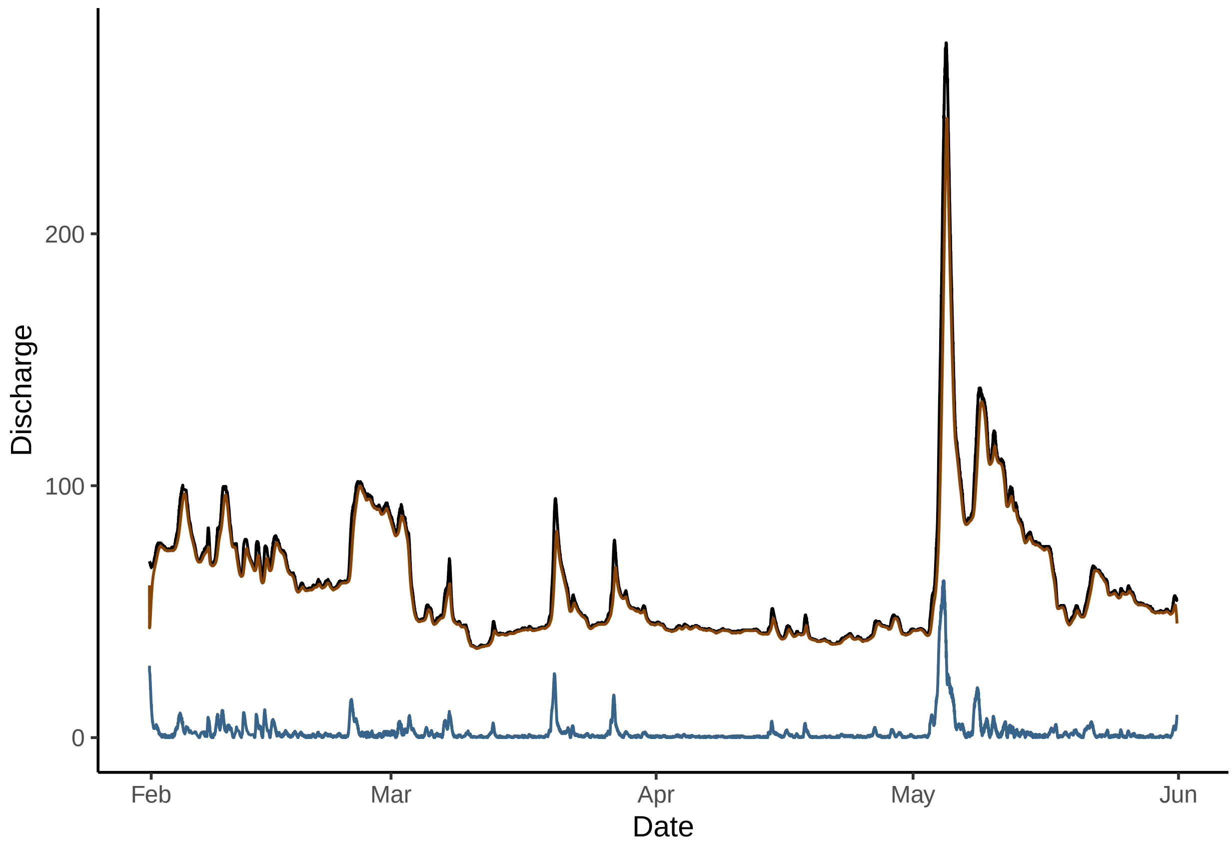
9. Use the `EcoHydRology::BaseflowSeparation()` function to partition discharge into baseflow and quickflow, and calculate how much water was exported as baseflow and quickflow for this time period. Use the DateTime_UTC column as your timestamps in this analysis.

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.

```
linearinterpolation <- as.data.frame(approx(x = Kansas$DateTime_UTC, y = Kansas$Discharge_m3s, n = 1152
linearinterpolation$x <-
  as.POSIXct(linearinterpolation$x, origin = "1970-01-01 00:00:00")
```

```r
names(linearinterpolation) <- c("Date", "Discharge")
#na.approx (zoo package for high frequency data)

Kansas.baseflow <- EcoHydRology::BaseflowSeparation(
  linearinterpolation$Discharge,
  filter_parameter = 0.925,
  passes = 3)

Kansas.full <- cbind(linearinterpolation, Kansas.baseflow)

ggplot(Kansas.full, aes(x = Date, y = Discharge)) +
  geom_line() +
  geom_line(mapping = aes(x = Date, y = bt), color = "darkorange4") +
  geom_line(mapping = aes(x = Date, y = qft), color = "steelblue4")
```




```r
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)
```

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

    Baseflow: 619464345/644091609 –> 96% Quickflow: = 24627264/644091609 –> 4%

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?

    Having a much larger watershed means the system is more stable and resistant to increases in quickflow.

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?

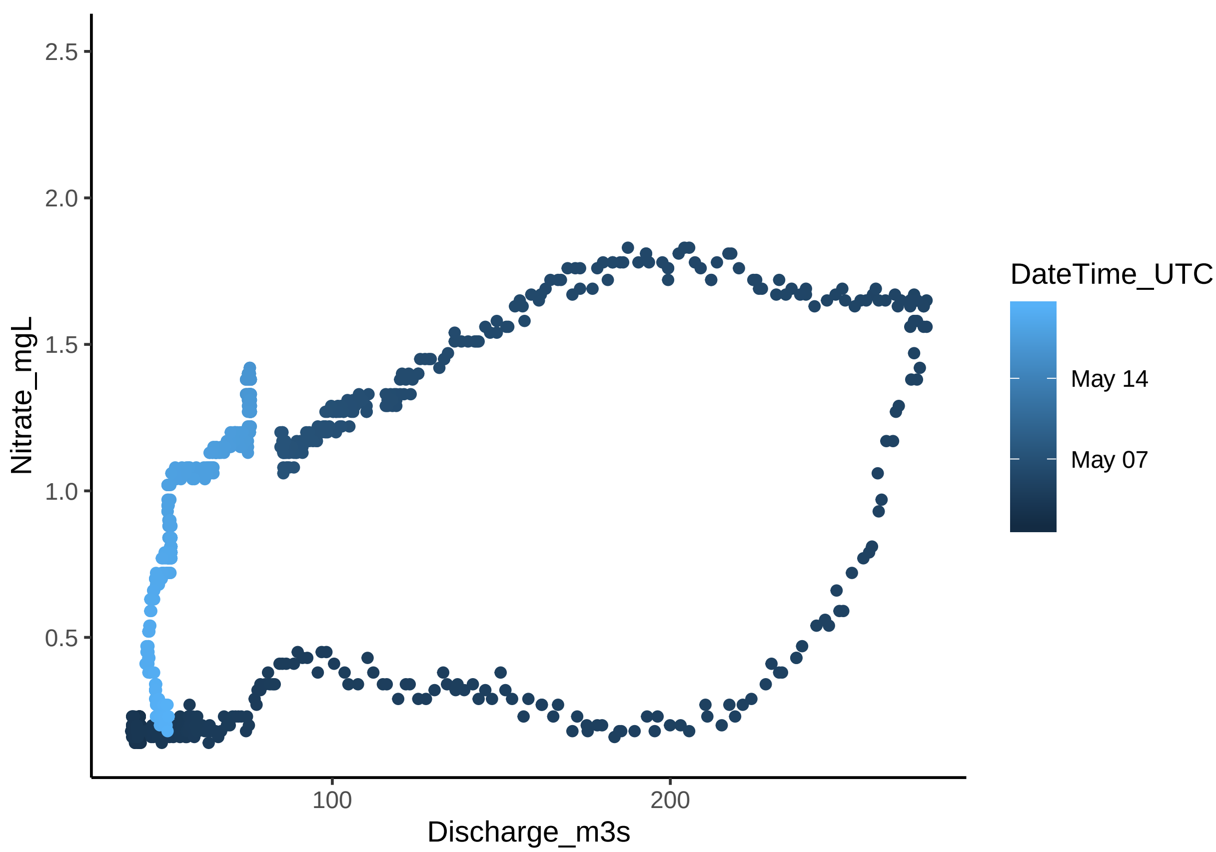
    Since it is further in the river network, it's likely that any increase in quickflow would eventually incorporate intself into the baseflow of the river by the time if made itself downstream. It supports the idea that the system is more resistant to large storms and quickflow.

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.

```
KansasStorm <- Kansas %>%
  filter(DateTime_UTC > "2018-05-01 00:00:00" & DateTime_UTC < "2018-05-20 00:00:00")

ggplot(KansasStorm, aes(x = Discharge_m3s, y = Nitrate_mgL, color = DateTime_UTC)) +
  geom_point()
```

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



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

    This storm shows clockwise hysteresis and is a flushing storm since nitrate concentrations increased with flow.

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

    This means that nitrate is entering the river from higher in the watershed via runoff.

Reflection

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

    High frequency data is hard to manage! We can use high frequency data to better understand how riverine systems respond to storms

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

    There's A LOT of observations for high frequency data (see Kansas discharge data measured every fifteen minutes) and it means that you are to take care in how it's analyzed and be as efficient as possible with wrangling The hysteresis plot allowed us to understand the storm was "flushing" which provides insights into how different pollutants make their way into the river.

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

    Yes, working with the actual data always helps. Better understand the format data will be in and the challenges that accompany real world data.

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

    Roughly matches with theory, with some slight deviations