Assignment 7: High Frequency Data

Caroline Watson

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)

```r
#checking working directory
getwd()
```

```
## [1] "/Users/carolinewatson/Documents/Fall 2019/Hydrologic_Data_Analysis/Assignments"
```

```r
#loading packages into R
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(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.
```

```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   1.0.0     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(dplyr)

#setting ggplot theme
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.

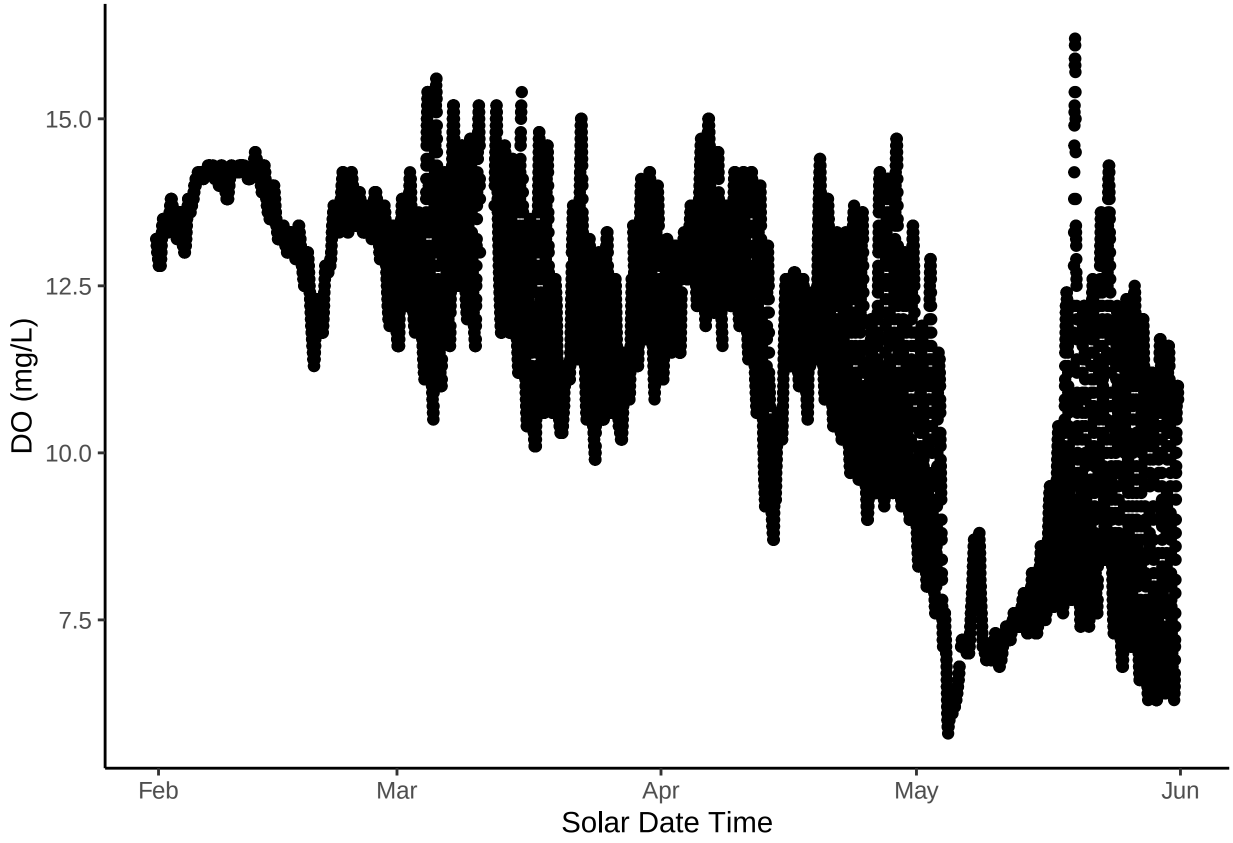
```r
Kansasdat <- request_data(
  sitecode = "KS_KANSASR",
  variables = c('DO_mgL',
                'Discharge_m3s',
                'Nitrate_mgL'))
```

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

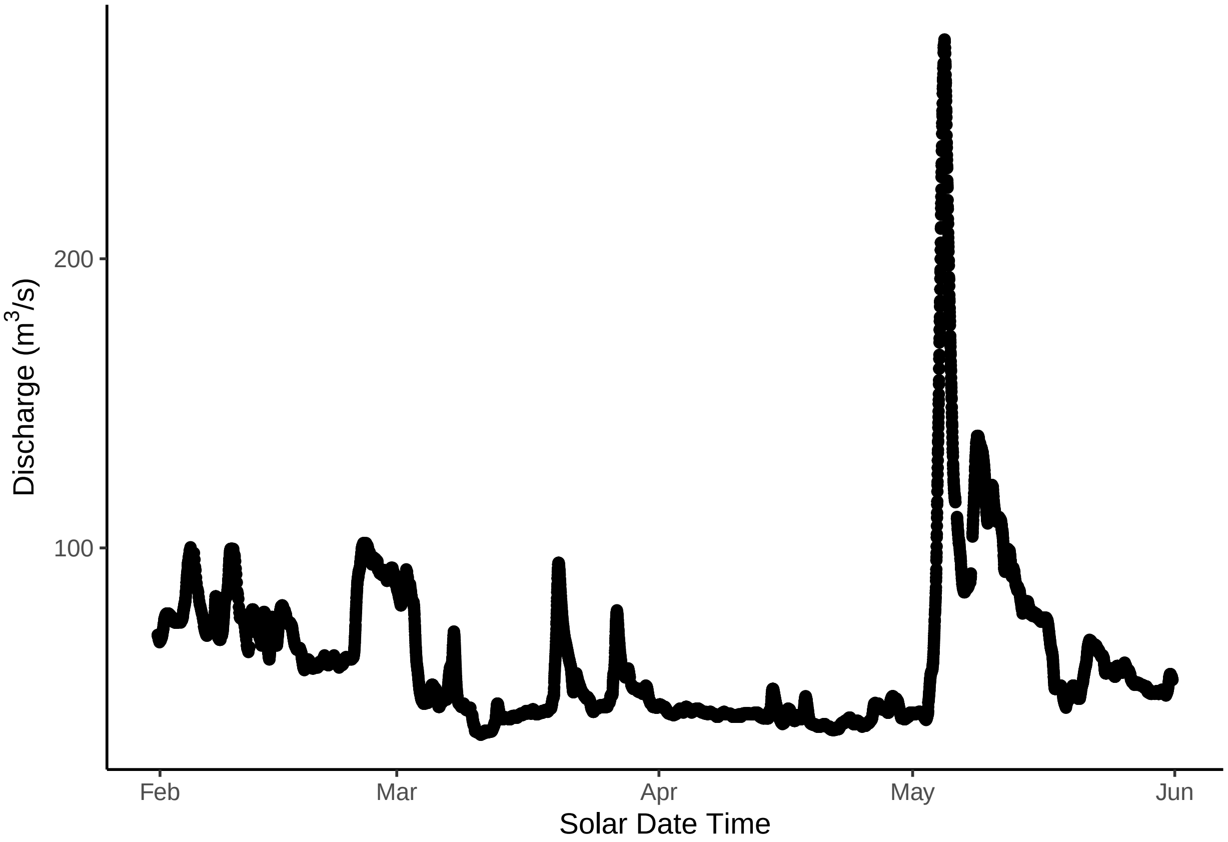
```r
#indexing longitude data
Kansas.lon <- Kansasdat[[2]]$lon

#converting data into one dataframe with columns
Kansas.data <- Kansasdat[[1]] %>%
  spread(value = value, key = variable) %>%
  mutate(DateTime_Solar = convert_UTC_to_solartime(DateTime_UTC, Kansas.lon))
```

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

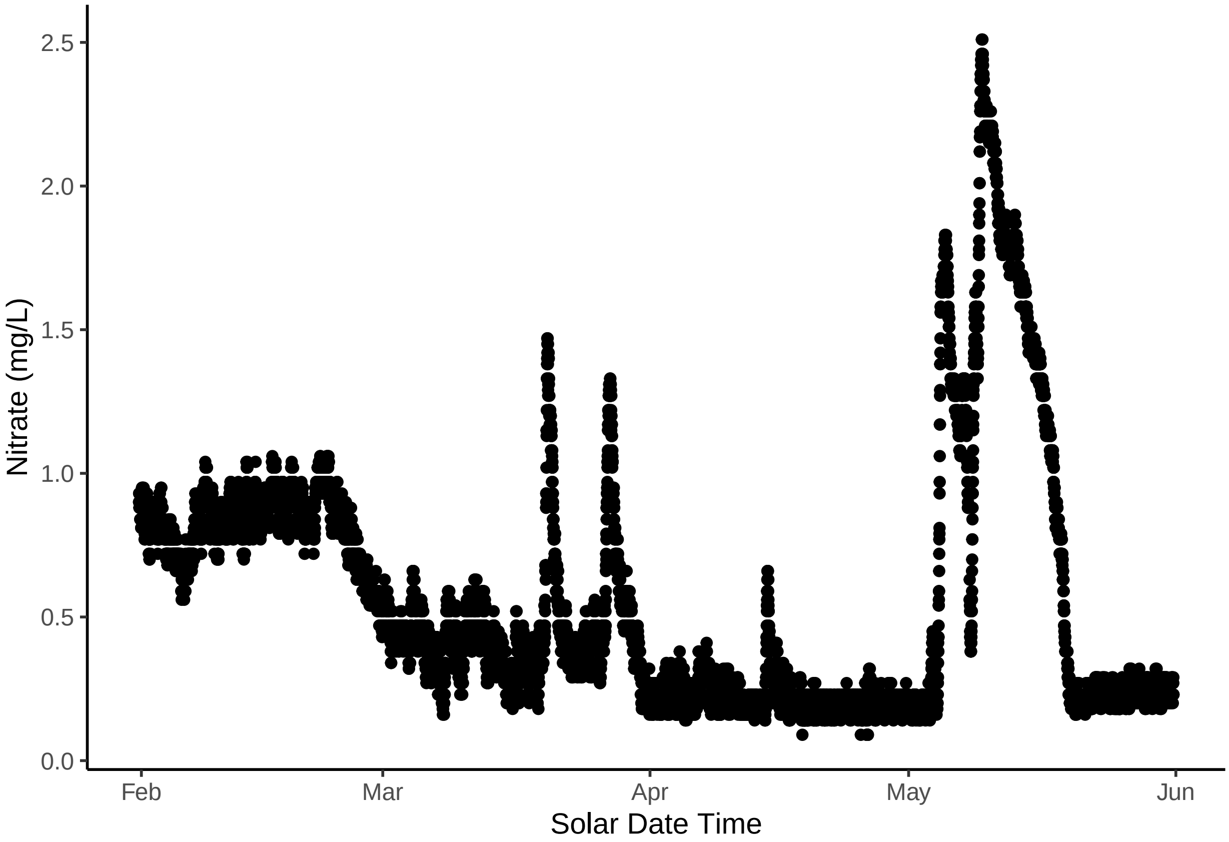
```r
#DO plot against solar time for the period of record
DO.Kansas <- ggplot(Kansas.data, aes(x = DateTime_Solar, y = DO_mgL)) +
  geom_point() +
  labs(x = "Solar Date Time", y = "DO (mg/L)")
print(DO.Kansas)
```



```
#discharge plot
Discharge.Kansas <- ggplot(Kansas.data, aes(x = DateTime_Solar, y = Discharge_m3s)) +
  geom_point() +
  labs(x = "Solar Date Time", y = (expression("Discharge (m"^3*"/s)")))
print(Discharge.Kansas)
```



```
#Nitrate plot
Nitrate.Kansas <- ggplot(Kansas.data, aes(x = DateTime_Solar, y = Nitrate_mgL)) +
  geom_point() +
  labs(x = "Solar Date Time", y = "Nitrate (mg/L)")
print(Nitrate.Kansas)
```



7. How will you address gaps in these dataseries?

   Gaps in the dataseries will be addressed by using the approx function to do a linear interpolation of the data to determine what the data might be for the gaps depending on the data that is collected. The linear interpolation will be better at dealing with gaps in the dataset than just omitting the NAs from the dataset because the interpolation will take into account the collected data to predict possible data points for the times when data was not collected.

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

   The daily amplitude of oxygen concentrations swings over the season are higher in the winder months and lower in the summer months. This is likely caused by the increased temperatures in the summer which leads to less DO in the water since colder water can hold more DO.

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.

```r
#figuring out the n to use by multiplying the number of timesteps(15) by 24 hours in a day to get 96 ho
as.Date("2018-05-31") - as.Date("2018-02-01") #119 days in the dataset = n
```

```
## Time difference of 119 days
```

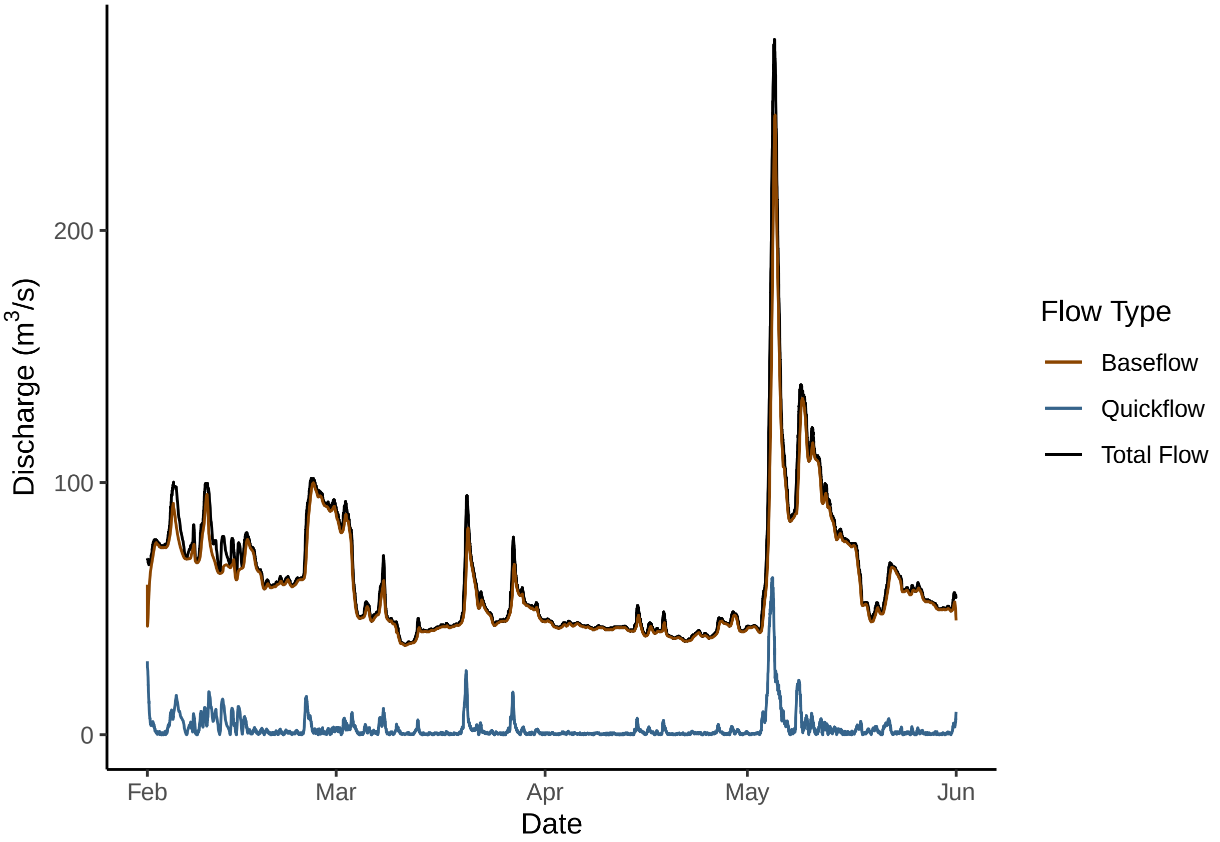
```r
#creating skinny dataset
KSdata.skinny <- Kansas.data %>%
  select(DateTime_UTC, Discharge_m3s) %>%
  arrange(DateTime_UTC) %>%
  na.omit()

#interpolating data
linearinterpolation.ks <- as.data.frame(approx(KSdata.skinny, n = 11424, method = "linear"))
linearinterpolation.ks$x <- as.POSIXct.Date(linearinterpolation.ks$x, origin = "1970-01-01")

#renaming the columns in the interpolation dataset
names(linearinterpolation.ks) <- c("Date", "Discharge")

#partitioning discharge into baseflow and quickflow
KSbaseflow <- EcoHydRology::BaseflowSeparation(
  KSdata.skinny$Discharge_m3s,
  filter_parameter = 0.925,
  passes = 3
)

#combining the dataframes into one
Kansas.dataframe <- cbind(KSdata.skinny, KSbaseflow)

Export_Kansas <- Kansas.dataframe %>%
  mutate(timestep = c(diff(as.numeric(DateTime_UTC)), 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)

#percentage of baseflow leaving in 2018 at Kansas
(Export_Kansas$BaseflowExport_cf/Export_Kansas$TotalExport_cf) * 100 #95.7%
```

```
## [1] 95.68351
```

```r
#percentage of quickflow leaving in 2018 in Kansas River
(Export_Kansas$QuickflowExport_cf/Export_Kansas$TotalExport_cf) * 100 #4.3
```

```
## [1] 4.316493
```

```r
#plotting the baseflow and quickflow
Kansas.plot <- ggplot(Kansas.dataframe, aes(x = DateTime_UTC, y = Discharge_m3s,
                             color = "Total Flow")) +
  geom_line() +
  geom_line(mapping = aes(x = DateTime_UTC, y = bt, color = "Baseflow" )) +
  geom_line(mapping = aes(x = DateTime_UTC, y = qft, color = "Quickflow")) +
  labs(x = "Date", y = expression("Discharge (m"^3*"/s)"), color = "Flow Type") +
  scale_color_manual(values = c(
    "Total Flow" = 'black',
    "Baseflow" = 'darkorange4',
    "Quickflow" = 'steelblue4'))
print(Kansas.plot)
```



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

    95.7% of the water exported left as baseflow from the Kansas River and 4.3% of water exported left as quickflow from the Kansas River.

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?

    The size of the watershed can have an affect on the baseflow and the quickflow because there is either more land for the water to flow over, or less land. In the 2 examples we looked at in class, the percentage of baseflow and quikflow leaving the rivers were very close (~ 10% difference), whereas this analysis shows most of the water was exported as baseflow. Because this watershed is larger, there is more room for the water to flow over land and could be flowing over more non-impervious sufaces as compared to in Ellerbee Creek or Third Fork Creek. The smaller the watershed, the more likely there will be more quickflow due to the size and likely significant amount of impervious surfaces.

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?

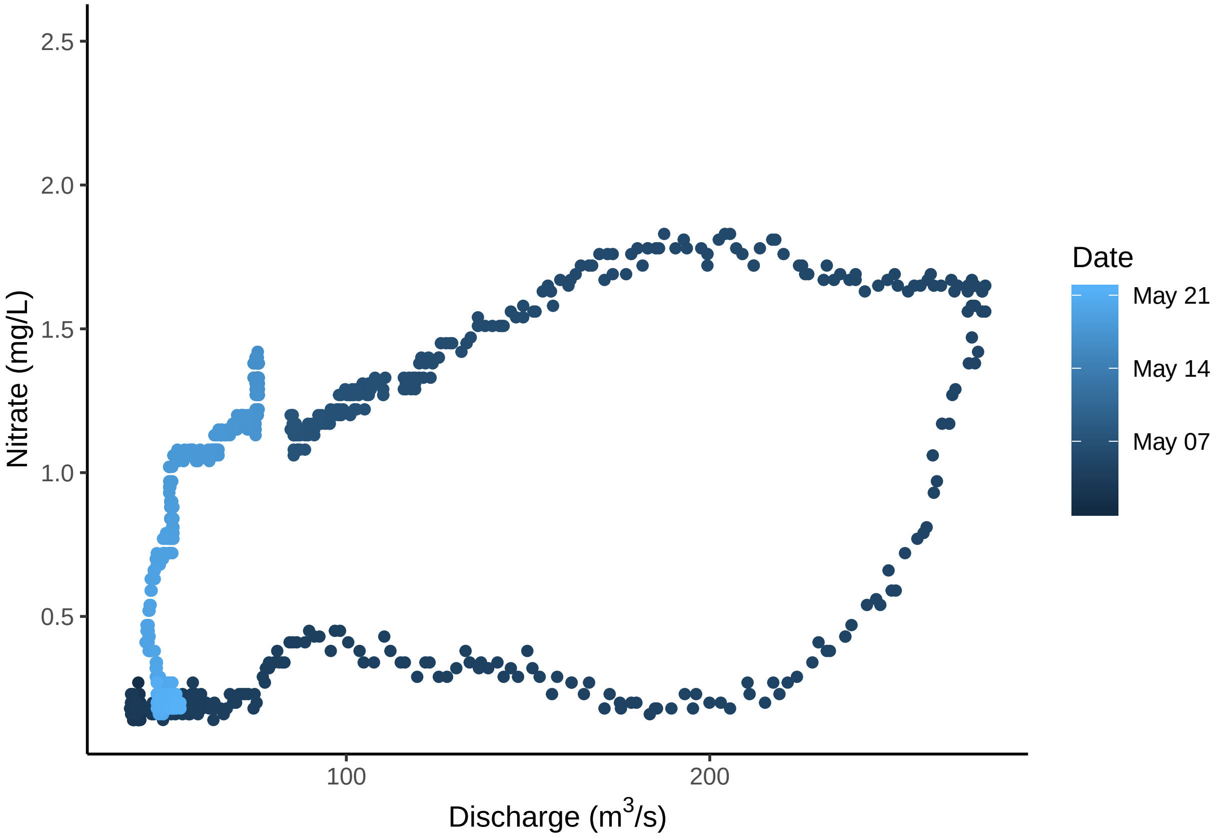
    This impacts the interpretation of the results because most of the water likely flows through the tributaries and slows down the flow and amount of quickflow. This could be why the percentage of quickflow is lower than what it would be if Kansas River was a headwater stream.

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.

```r
#filtering for storm from May 1 - May 20
KansasStorm <- Kansas.data %>%
  filter(DateTime_Solar > "2018-04-30" & DateTime_Solar < "2018-05-21")

#ggplot of flow vs nitrate
Nitrate <- ggplot(KansasStorm, aes(x = Discharge_m3s, y = Nitrate_mgL,
              color = DateTime_UTC)) +
  geom_point() +
  labs(x = expression("Discharge (m"^3*"/s)"), y = "Nitrate (mg/L)",
       color = "Date")

print(Nitrate)
```

```
## 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 counterclockwise hysteresis and as discharge increases during the storm, the nitrate concentration also increases. This is a flushing storm because as the discharge is increasing, the nitrate concentrations are increasing because the nutrients are being flushed with high amounts of water. large amounts of water are flushing the nutrients.

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

Nitrate concentrations increase on the increasing limb, which is generally when quickflow occurs during a storm. Nitrate concentrations decrease on the decreasing limb, which is representative of baseflow conditions. Thus, nitrate likely gets into the river from the watershed via overland flow.

Reflection

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

    One summary point I learned from my analysis is that baseflow and quickflow can tell a story about a river and you can determine storms that occured. Another summary point is that hysteresis loops have two nitrate concentration values for every one value of discharge.

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

    The ggplot with baseflow, quickflow, and total flow support my conclusions for how baseflow and quickflow tell a story about a river ands storms. The hysteresis plot supported my conclusion that each discharge value has two nitrate concentration values.

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

    Yes, hands-on data analysis did impact my learning about high frequencey data relative to a theory-based lesson because I had to interpret the results which led me to better grasp the concepts.

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

    Real-world data compares to my expectations from theory in that most of the data corresponds to what the theory suggests.