Assignment 7: High Frequency Data

Theo Cai

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] "Z:/Hydrologic_Data_Analysis"
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
install_github('streampulse/StreamPULSE')
```

```
## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install Rtools custom from http://cran.r-project.org/bin/windows/Rtools/.
```

```
## Downloading GitHub repo streampulse/StreamPULSE@master
```

```
## Cairo        (NA    -> 1.5-10     ) [CRAN]
## imputeTS     (NA    -> 3.0        ) [CRAN]
## ks           (NA    -> 1.11.5     ) [CRAN]
## geosphere    (NA    -> 1.5-10     ) [CRAN]
## zoo          (NA    -> 1.8-6      ) [CRAN]
## geoknife     (NA    -> 1.6.3      ) [CRAN]
## R.utils      (NA    -> 2.9.0      ) [CRAN]
## fastmap      (1.0.0 -> 1.0.1      ) [CRAN]
## stinepack    (NA    -> 1.4        ) [CRAN]
## forecast     (NA    -> 8.9        ) [CRAN]
## FNN          (NA    -> 1.1.3      ) [CRAN]
## kernlab      (NA    -> 0.9-27     ) [CRAN]
## mclust       (NA    -> 5.4.5      ) [CRAN]
## multicool    (NA    -> 0.1-10     ) [CRAN]
## mvtnorm      (NA    -> 1.0-11     ) [CRAN]
## sp           (NA    -> 1.3-1      ) [CRAN]
```

```
## R.oo          (NA    -> 1.22.0     ) [CRAN]
## R.methodsS3   (NA    -> 1.7.1      ) [CRAN]
## fracdiff      (NA    -> 1.4-2      ) [CRAN]
## lmtest        (NA    -> 0.9-37     ) [CRAN]
## timeDate      (NA    -> 3043.102   ) [CRAN]
## tseries       (NA    -> 0.10-47    ) [CRAN]
## urca          (NA    -> 1.3-0      ) [CRAN]
## RcppArmad...  (NA    -> 0.9.800.1.0) [CRAN]
## quadprog      (NA    -> 1.5-7      ) [CRAN]
## quantmod      (NA    -> 0.4-15     ) [CRAN]
## xts           (NA    -> 0.11-2     ) [CRAN]
## TTR           (NA    -> 0.23-5     ) [CRAN]

## Installing 28 packages: Cairo, imputeTS, ks, geosphere, zoo, geoknife, R.utils, fastmap, stinepack, f

## Installing packages into 'C:/Users/gyc4/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)

##
##   There are binary versions available but the source versions are
##   later:
##                     binary      source needs_compilation
## fastmap              1.0.0       1.0.1              TRUE
## RcppArmadillo 0.9.700.2.0 0.9.800.1.0              TRUE
##
##   Binaries will be installed
## package 'Cairo' successfully unpacked and MD5 sums checked
## package 'imputeTS' successfully unpacked and MD5 sums checked
## package 'ks' successfully unpacked and MD5 sums checked
## package 'geosphere' successfully unpacked and MD5 sums checked
## package 'zoo' successfully unpacked and MD5 sums checked
## package 'geoknife' successfully unpacked and MD5 sums checked
## package 'R.utils' successfully unpacked and MD5 sums checked
## package 'fastmap' successfully unpacked and MD5 sums checked
## package 'stinepack' successfully unpacked and MD5 sums checked
## package 'forecast' successfully unpacked and MD5 sums checked
## package 'FNN' successfully unpacked and MD5 sums checked
## package 'kernlab' successfully unpacked and MD5 sums checked
## package 'mclust' successfully unpacked and MD5 sums checked
## package 'multicool' successfully unpacked and MD5 sums checked
## package 'mvtnorm' successfully unpacked and MD5 sums checked
## package 'sp' successfully unpacked and MD5 sums checked
## package 'R.oo' successfully unpacked and MD5 sums checked
## package 'R.methodsS3' successfully unpacked and MD5 sums checked
## package 'fracdiff' successfully unpacked and MD5 sums checked
## package 'lmtest' successfully unpacked and MD5 sums checked
## package 'timeDate' successfully unpacked and MD5 sums checked
## package 'tseries' successfully unpacked and MD5 sums checked
## package 'urca' successfully unpacked and MD5 sums checked
## package 'RcppArmadillo' successfully unpacked and MD5 sums checked
## package 'quadprog' successfully unpacked and MD5 sums checked
## package 'quantmod' successfully unpacked and MD5 sums checked
## package 'xts' successfully unpacked and MD5 sums checked
## package 'TTR' successfully unpacked and MD5 sums checked
##
```

```
## The downloaded binary packages are in
##  C:\Users\gyc4\AppData\Local\Temp\RtmpyclKu9\downloaded_packages
## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install Rtools custom from http://cran.r-project.org/bin/windows/Rtools/.
##


   checking for file 'C:\Users\gyc4\AppData\Local\Temp\RtmpyclKu9\remotesc805a453c30\streampulse-Streaml

v  checking for file 'C:\Users\gyc4\AppData\Local\Temp\RtmpyclKu9\remotesc805a453c30\streampulse-Streaml
##



-  preparing 'StreamPULSE':
##    checking DESCRIPTION meta-information ...

   checking DESCRIPTION meta-information ...

v  checking DESCRIPTION meta-information
##



-  checking for LF line-endings in source and make files and shell scripts
##



-  checking for empty or unneeded directories
##



-  building 'StreamPULSE_0.0.0.9041.tar.gz'
##


##
## Installing package into 'C:/Users/gyc4/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("streamMetabolizer", dependencies=TRUE,
  repos=c("https://owi.usgs.gov/R","https://cran.rstudio.com"))
```

```
## Installing package into 'C:/Users/gyc4/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
## also installing the dependencies 'matrixStats', 'checkmate', 'rLakeAnalyzer', 'bitops', 'StanHeaders
## package 'matrixStats' successfully unpacked and MD5 sums checked
## package 'checkmate' successfully unpacked and MD5 sums checked
## package 'rLakeAnalyzer' successfully unpacked and MD5 sums checked
```

```
## package 'bitops' successfully unpacked and MD5 sums checked
## package 'StanHeaders' successfully unpacked and MD5 sums checked
## package 'inline' successfully unpacked and MD5 sums checked
## package 'loo' successfully unpacked and MD5 sums checked
## package 'RcppEigen' successfully unpacked and MD5 sums checked
## package 'deSolve' successfully unpacked and MD5 sums checked
## package 'LakeMetabolizer' successfully unpacked and MD5 sums checked
## package 'unitted' successfully unpacked and MD5 sums checked
## package 'chron' successfully unpacked and MD5 sums checked
## package 'dygraphs' successfully unpacked and MD5 sums checked
## package 'microbenchmark' successfully unpacked and MD5 sums checked
## package 'RCurl' successfully unpacked and MD5 sums checked
## package 'rstan' successfully unpacked and MD5 sums checked
## package 'streamMetabolizer' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\gyc4\AppData\Local\Temp\RtmpyclKu9\downloaded_packages
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------------------------ tidy
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(StreamPULSE)
```

```
## Loading required package: shiny
## Loading required package: Cairo
```

```
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(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(tidyverse)
library(cowplot)
```

```
##
## ************************************************************
## Note: As of version 1.0.0, cowplot does not change the
##   default ggplot2 theme anymore. To recover the previous
##   behavior, execute:
##   theme_set(theme_cowplot())
## ************************************************************
```

```
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.

```
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
```

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

Kansas.var <- Kansas.dat[[1]] %>%
  spread(value = value, key = variable) %>%
  mutate(DateTime_Solar = convert_UTC_to_solartime(DateTime_UTC, Kansas.lon)) %>%
  select(site, DateTime_UTC, DateTime_Solar, DO_mgL, Discharge_m3s, Nitrate_mgL)
```

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

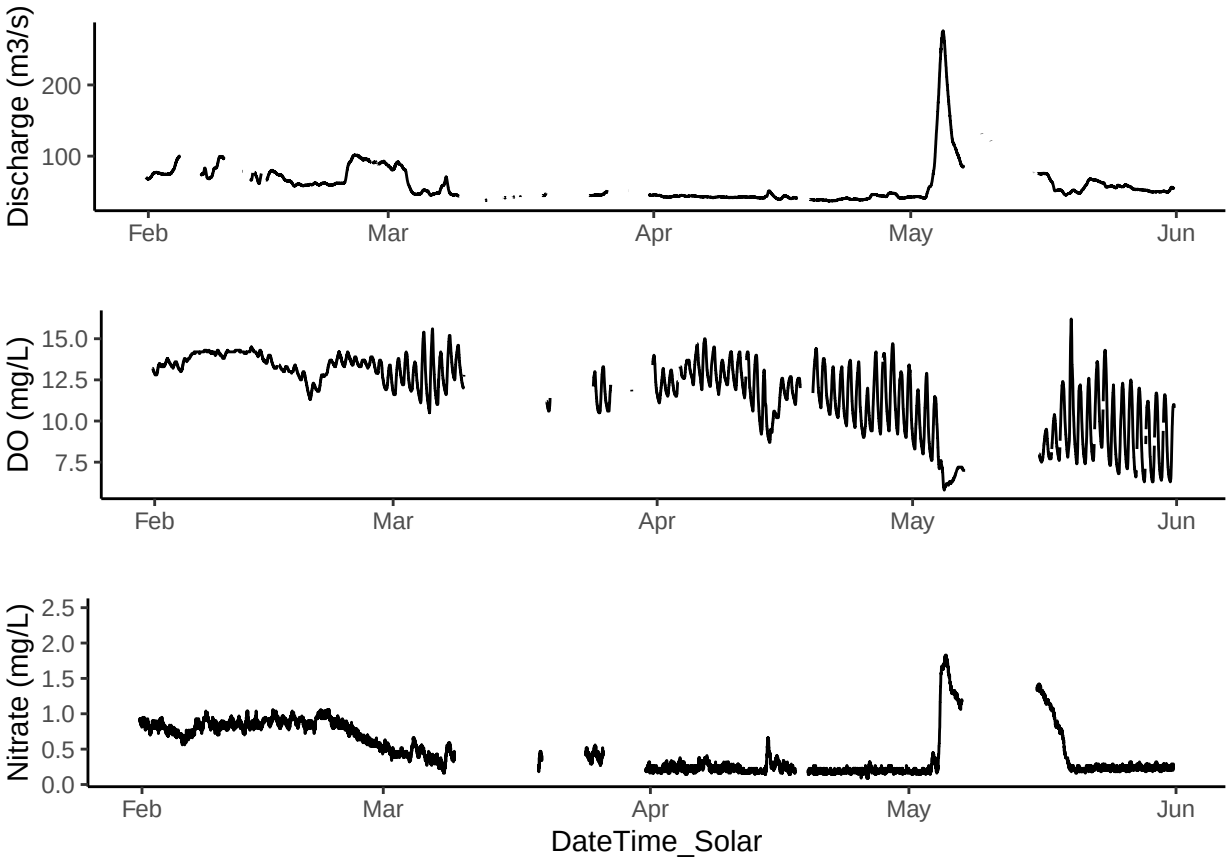
```
Discharge.plot <-
  ggplot(Kansas.var, aes(x = DateTime_Solar, y = Discharge_m3s)) +
  geom_line() +
  labs(x = "", y = "Discharge (m3/s)")

DO.plot <-
  ggplot(Kansas.var, aes(x = DateTime_Solar, y = DO_mgL)) +
  geom_line() +
  labs(x = "", y = "DO (mg/L)")

Nitrate.plot <-
  ggplot(Kansas.var, aes(x = DateTime_Solar, y = Nitrate_mgL)) +
```

```
  geom_line() +
  labs(y = "Nitrate (mg/L)")

Combined.plot <-
  plot_grid(Discharge.plot, DO.plot, Nitrate.plot,
            ncol = 1)
print(Combined.plot)
```



7. How will you address gaps in these dataseries?

   I think that I can address these gaps by analyzing each of the continuous chunks separately. I think this would technically be called a step-trend analysis (if I choose to do a time series analysis), even though the breaks in data seem to be arbitrary/do not seem to mark a distinct policy change.

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

   The daily amplitude of oxygen concentration seem to trend downwards over the course of the winter -> summer seasons. This is likely due to rising temperatures during that same time period - oxygen dissolves easier in cold water.

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.

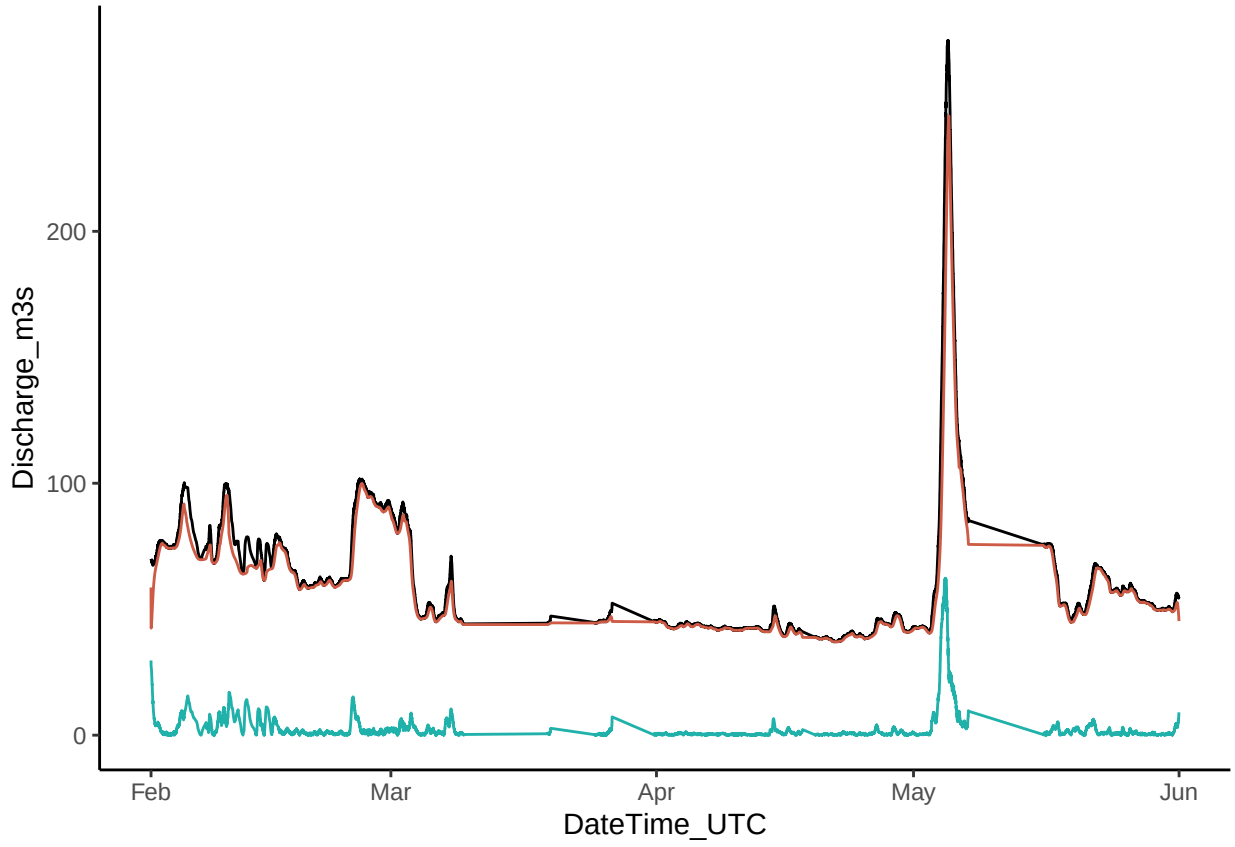
10. Create a ggplot showing total flow, baseflow, and quickflow together.

```
Kansas.base <-
  drop_na(Kansas.var)

Kansas.baseflow <- EcoHydRology::BaseflowSeparation(
  Kansas.base$Discharge_m3s
)

Kansas.flow <- cbind(Kansas.base, Kansas.baseflow)

ggplot(Kansas.flow, aes(x = DateTime_UTC, y = Discharge_m3s)) +
  geom_line() +
  geom_line(mapping = aes(x = DateTime_UTC, y = bt), color = "coral3") +
  geom_line(mapping = aes(x = DateTime_UTC, y = qft), color = "lightseagreen")
```




```
Export <- Kansas.flow %>%
  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)
```

```
Export$BaseflowExport_cf/Export$TotalExport_cf*100
```

```
## [1] 94.6137
```

```
Export$QuickflowExport_cf/Export$TotalExport_cf*100
```

```
## [1] 5.386295
```

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

    94.61% of the total water exported was baseflow, and 5.39% was 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?

    The size of the watershed impacts the amount of quickflow we see. The bigger the watershed, the more the overland flow needs to travel to reach the river. And of course, as this flow travels across the land, it has more opportunities to soak into the ground and percolate into the groundwater, later becoming base flow, the more land it has to cross.

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?

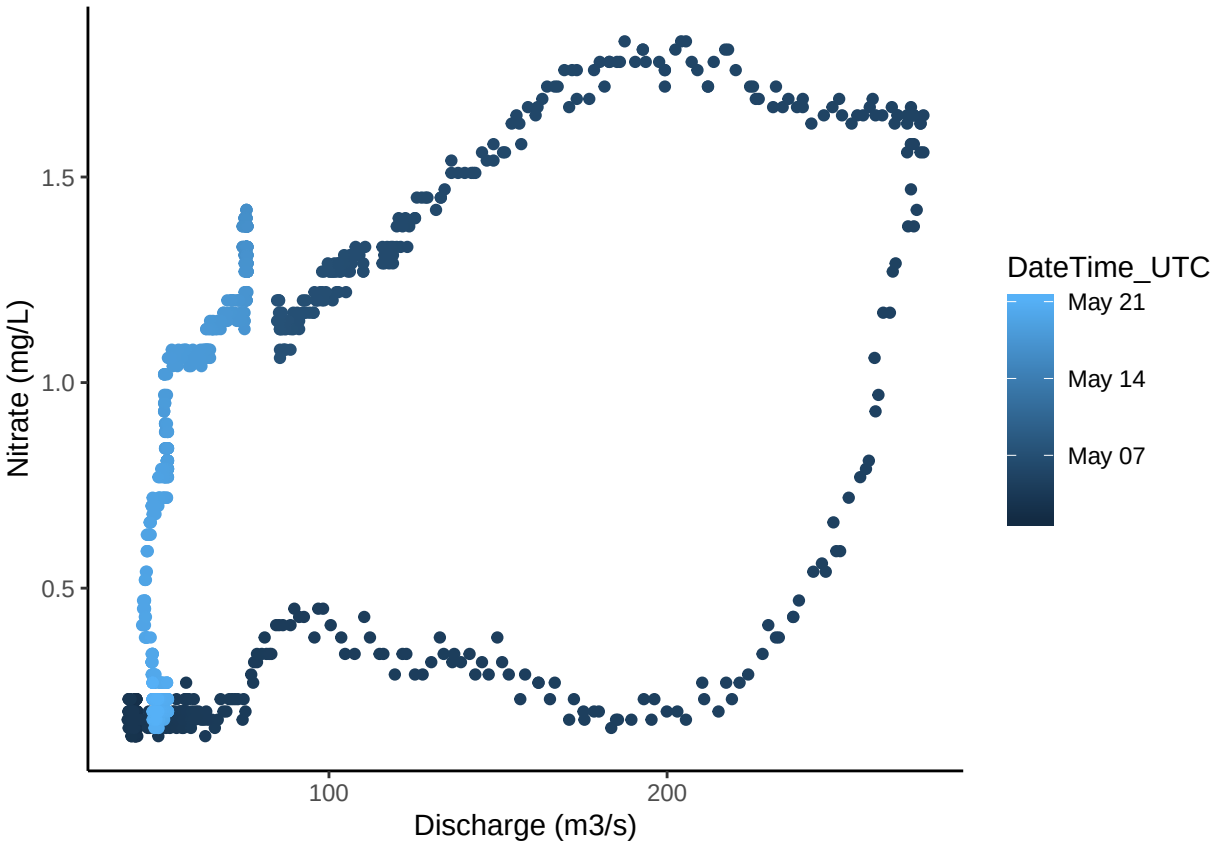
    I think this might imply that some of the baseflow we see is not just from the groundwater immediate to the site, but also from baseflows from other, further rivers. The calibrated baseflow of this one river might be lower than shown here.

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.

```
Kansas.storm <- Kansas.flow %>%
  filter(DateTime_UTC > "2018-05-01 00:00:00" & DateTime_UTC < "2018-05-20 23:00:00")

ggplot(Kansas.storm, aes(x = Discharge_m3s, y = Nitrate_mgL, color = DateTime_UTC)) +
  geom_point() +
  labs(x = "Discharge (m3/s)", y = "Nitrate (mg/L)")
```



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

    This storm shows a counterclockwise hysteresis. It was a flushing storm.

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

    This means that nitrate in this river comes primarily from overland flow, and it is not as present in the groundwater/baseflow.

Reflection

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

    I learned that breaks in high-frequency data make it harder to analyze, but not impossible. I also learned that high frequency data is extremely useful in analyzing the effects of individual storms, even if they only last a week or two

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

    My final ggplot/hysteresis plot supported both conclusions

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

    Yes, it did. It really helped me visualize concretely the theories about baseflow, quick flow, and hysteresis loops.

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

It matched up pretty well!