

# Assignment 7: High Frequency Data

Yutao Gong

## 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/gongy/Documents/Hydrologic_Data_Analysis/Assignments"
```

```
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

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()

theme_set(theme_bw())
```

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`.

```
KansasData <- 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,Nitrate_mgL&flags=true
##
## Retrieved the following variables:
##   DO_mgL, Discharge_m3s, Nitrate_mgL
```

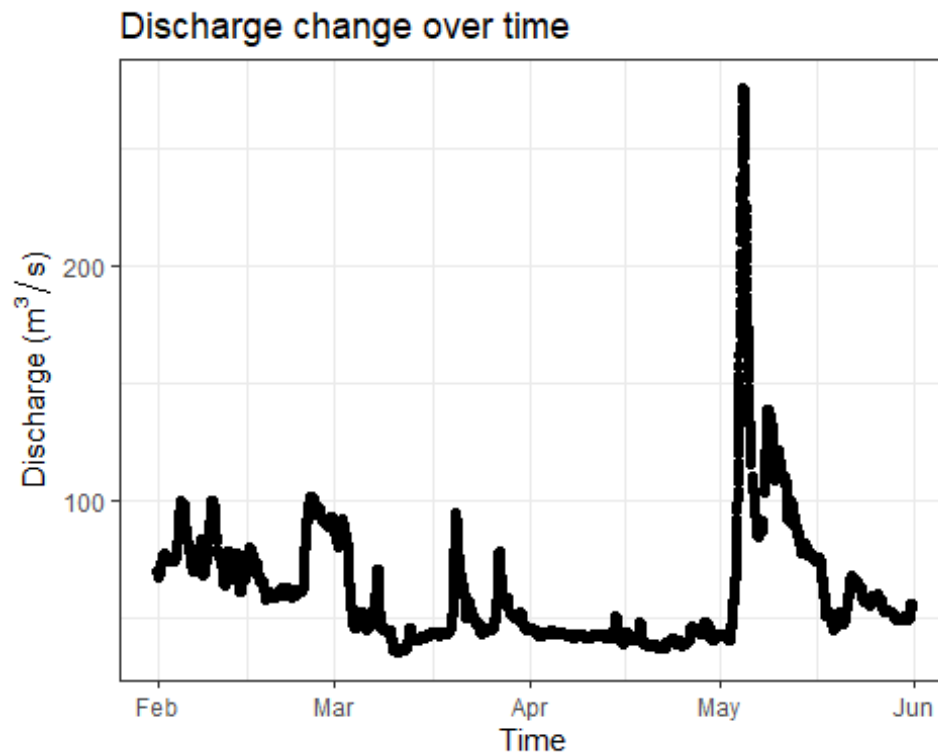
```
Kansas.lon <- KansasData[[2]]$lon

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

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

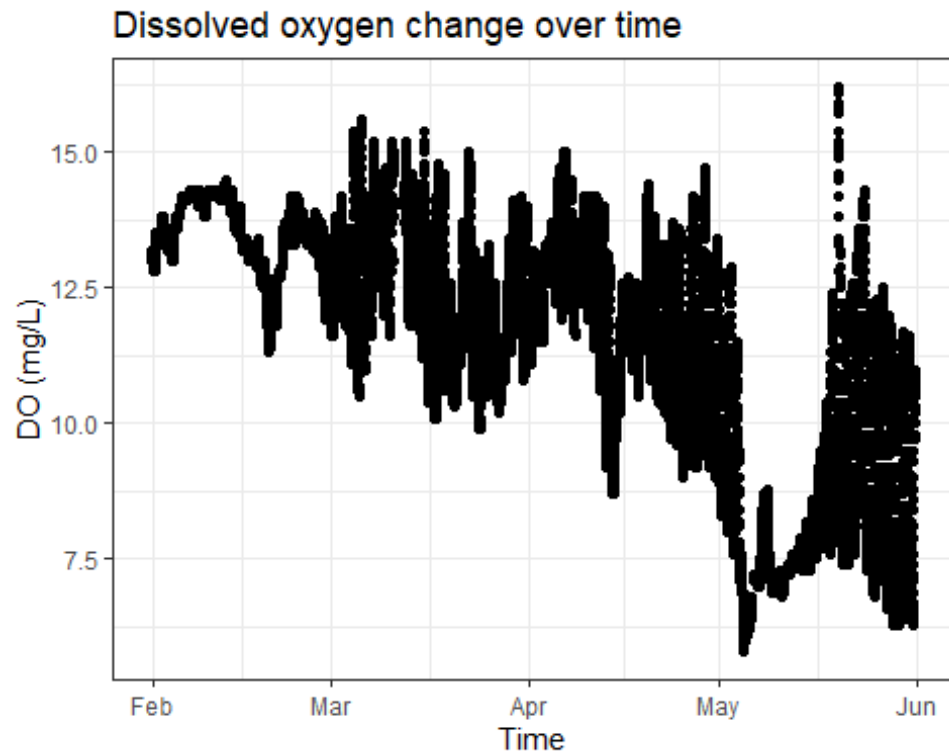
```
ggplot(Kansas.formatted,
  aes(x = DateTime_Solar, y = Discharge_m3s)) +
  labs(x = "Time", y = expression("Discharge (m3/s)"), title = "Discharge
change over time") +
  geom_point()
```

## Warning: Removed 3428 rows containing missing values (geom\_point).



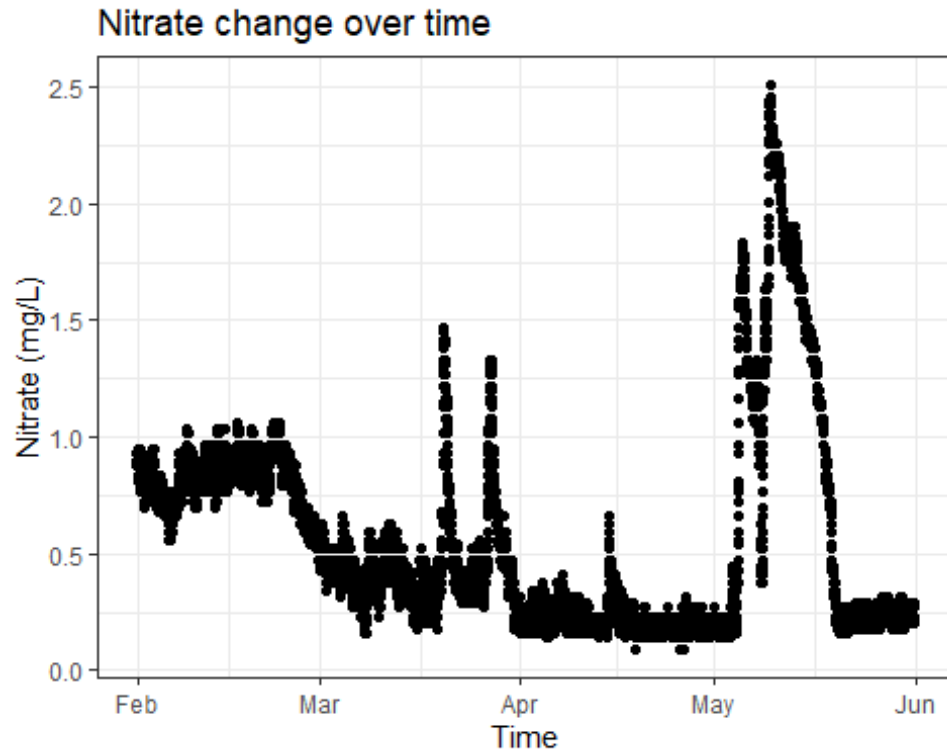
```
ggplot(Kansas.formatted,
  aes(x = DateTime_Solar, y = DO_mgL)) +
  labs(x = "Time", y = "DO (mg/L)", title = "Dissolved oxygen change over
time") +
  geom_point()
```

## Warning: Removed 3047 rows containing missing values (geom\_point).



```
ggplot(Kansas.formatted,
       aes(x = DateTime_Solar, y = Nitrate_mgL)) +
  labs(x = "Time", y = "Nitrate (mg/L)", title = "Nitrate change over time")
+
  geom_point()
```

## Warning: Removed 2908 rows containing missing values (geom\_point).



7. How will you address gaps in these dataserries?

I would either use linear interpolation to fill in the gaps or drop the rows with missing values (because the frequency of sampling is pretty high)

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

It increases from winter/early spring times (February) to summer times (June). In summer the temperature difference in a day may be more significant, and there would also be more precipitation, both of which lead to more significant swings of oxygen concentration in the 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.

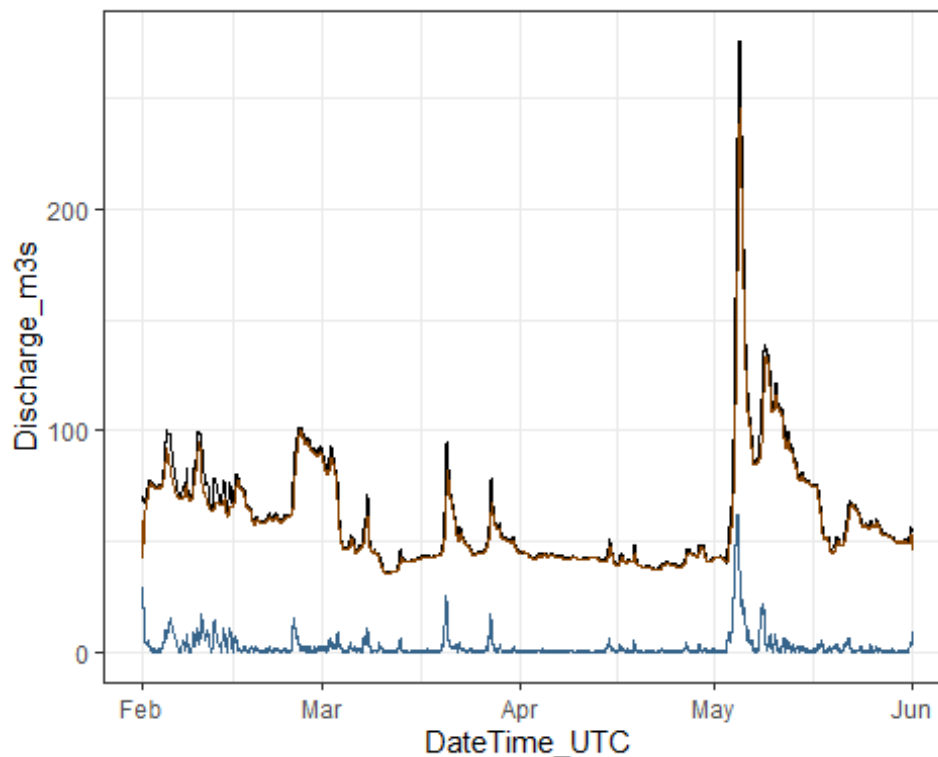
10. Create a ggplot showing total flow, baseflow, and quickflow together.

```
Kansas.discharge <- Kansas.formatted %>%
  select(DateTime_UTC, Discharge_m3s) %>%
  drop_na()
Kansasbaseflow <- EcoHydrology::BaseflowSeparation(
```

```
Kansas.discharge$Discharge_m3s,
filter_parameter = 0.925,
passes = 3
)
```

```
KansasFlow <- cbind(Kansas.discharge, Kansasbaseflow)
```

```
ggplot(KansasFlow, aes(x = DateTime.UTC, y = Discharge_m3s)) +
  geom_line() +
  # scale_y_log10() +
  geom_line(mapping = aes(x = DateTime.UTC, y = bt), color = "darkorange4") +
  geom_line(mapping = aes(x = DateTime.UTC, y = qft), color = "steelblue4")
```



```
Export <- KansasFlow %>%
  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 #baseflow percentage
## [1] 0.9568351
Export$QuickflowExport_cf/Export$TotalExport_cf #quickflow percentage
## [1] 0.04316493
```

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

Baseflow: 95.68%, quickflow: 4.32%

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 larger the watershed, the higher the percentage of baseflow takes up the total water flow of a river.

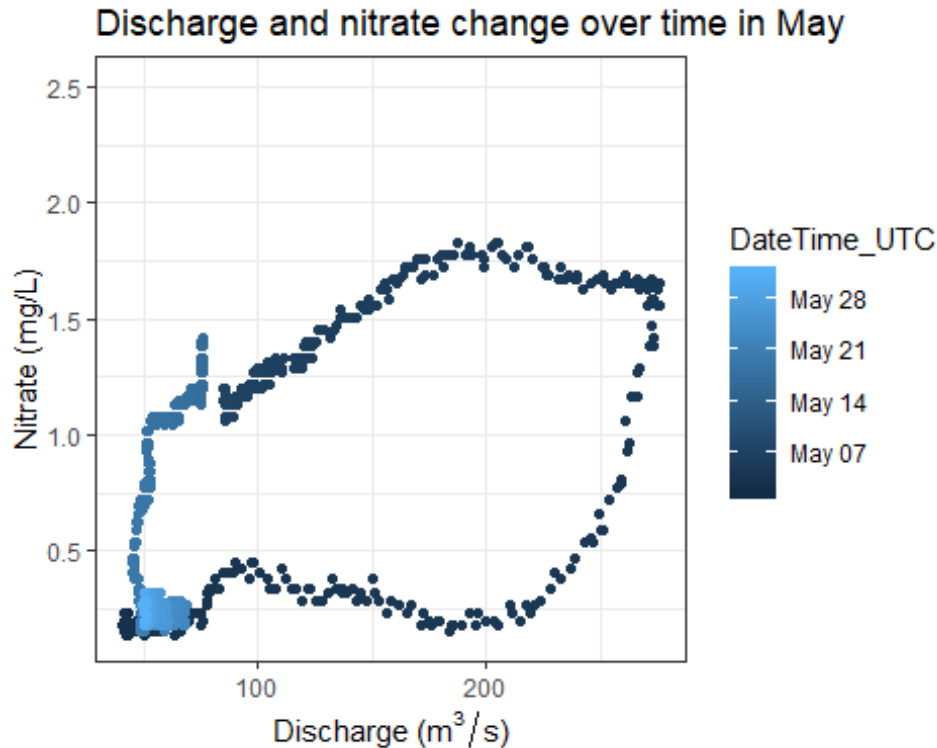
13. The site we are looking at is also further down in its river network (i.e. instead of being a headwater stream, this river has multiple tributaries that flow into it). How does this impact your interpretation of your results?

It could also be: rivers more downstream in the river network will tend to have a higher percentage of flow being baseflow.

## 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.

```
KansasMay <- Kansas.formatted %>%  
  filter(DateTime_Solar > "2018-05-01" & DateTime_Solar < "2018-05-31")  
ggplot(KansasMay, aes(x = Discharge_m3s, y = Nitrate_mgL, color =  
DateTime_UTC)) +  
  labs(x = expression("Discharge (m3/s)"), y = "Nitrate (mg/L)", title =  
"Discharge and nitrate change over time in May") +  
  geom_point()  
## Warning: Removed 1659 rows containing missing values (geom_point).
```



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

Counterclockwise hysteresis. The storm was a flushing storm.

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

It means that nitrate from the watershed gets into the river mainly through run-off from nearby (probably impervious) surfaces when there is a storm.

## Reflection

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

1. High frequency data provides rich resource for analysis but also calls for careful data wrangling techniques.
2. High frequency data allows for analysis on chemical hysteresis in water bodies.

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

I spent a lot of time figuring out how to deal with NAs when extracting baseflows. The last graph also supports chemical hysteresis analysis.

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

Yes, it presents some challenges that I did not expect to run into.

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

I thought high frequency data would be very convenient to analyze, but it turns out I need to put extra effort into data cleaning (like missing values.)