Assignment 7: High Frequency Data

Keqi He

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/kh382/Documents/Hydrologic_Data_Analysis/Assignments"
```

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

# library(devtools)
# install_github('streampulse/StreamPULSE')

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`), 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
Kandischarge <- request_data(
  sitecode = "KS_KANSASR",
  variables = c('Discharge_m3s')
  )
```

```
## 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&flags=true
##
## Retrieved the following variables:
##   Discharge_m3s
```

```r
KanDO <- request_data(
  sitecode = "KS_KANSASR",
  variables = c('DO_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&flags=true
##
## Retrieved the following variables:
##   DO_mgL
## Could not find:
##   DO_mgl
```

```r
Kannitrate <- request_data(
  sitecode = "KS_KANSASR",
  variables = c('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=Nitrate_mgL&flags=true
##
## Retrieved the following variables:
##   Nitrate_mgL
```

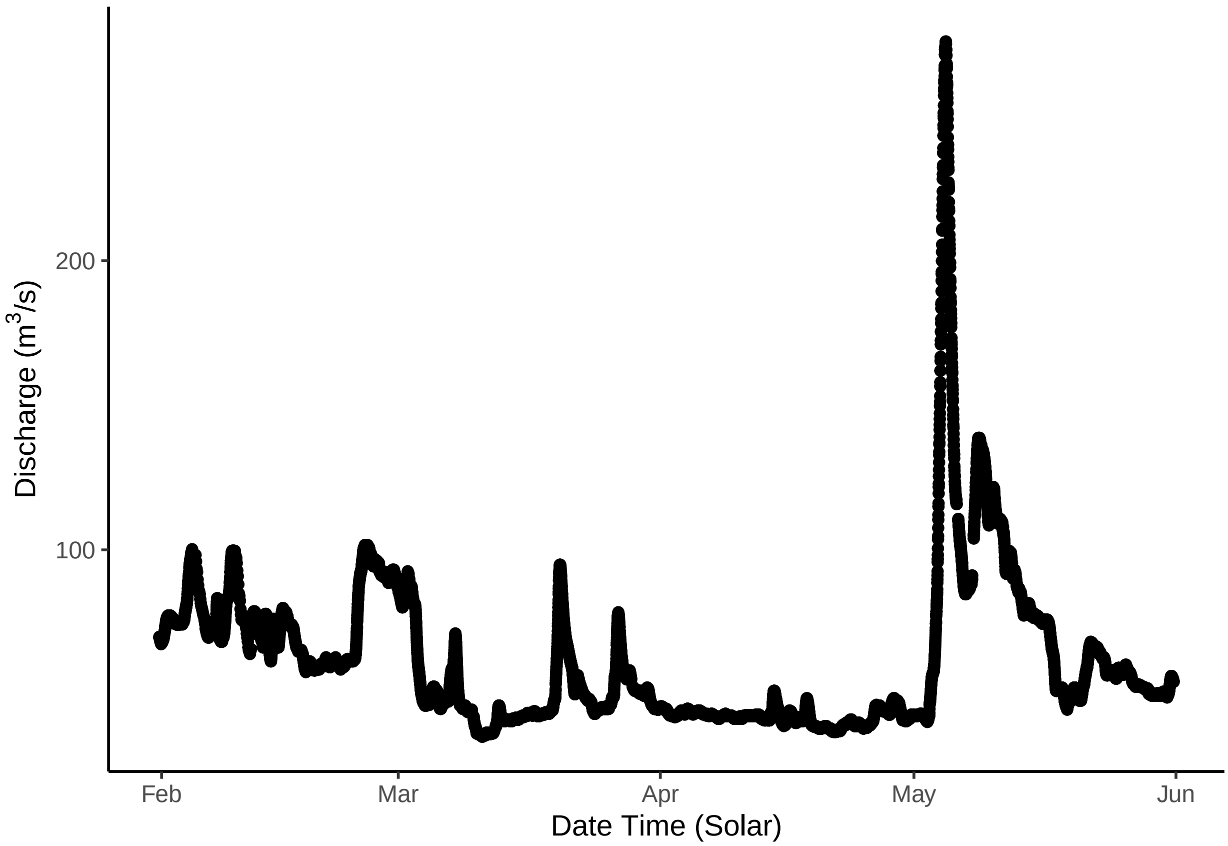
```r
Kandischargedat <- Kandischarge[[1]] %>%
  spread(value = value, key = variable)
KanDOdat <- KanDO[[1]] %>%
  spread(value = value, key = variable)
Kannitratedat <- Kannitrate[[1]] %>%
  spread(value = value, key = variable)

Kan.lon <- KanDO[[2]]$lon

Kandat1 <- left_join(Kannitratedat, Kandischargedat, by = "DateTime_UTC")
Kandat <- left_join(Kandat1, KanDOdat, by = "DateTime_UTC") %>%
  select(DateTime_UTC, site.x, DO_mgL, Discharge_m3s, Nitrate_mgL) %>%
  mutate(DateTime_Solar = convert_UTC_to_solartime(DateTime_UTC, Kan.lon))
```

```r
names(Kandat)[2] <- c("Sitename")
```

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

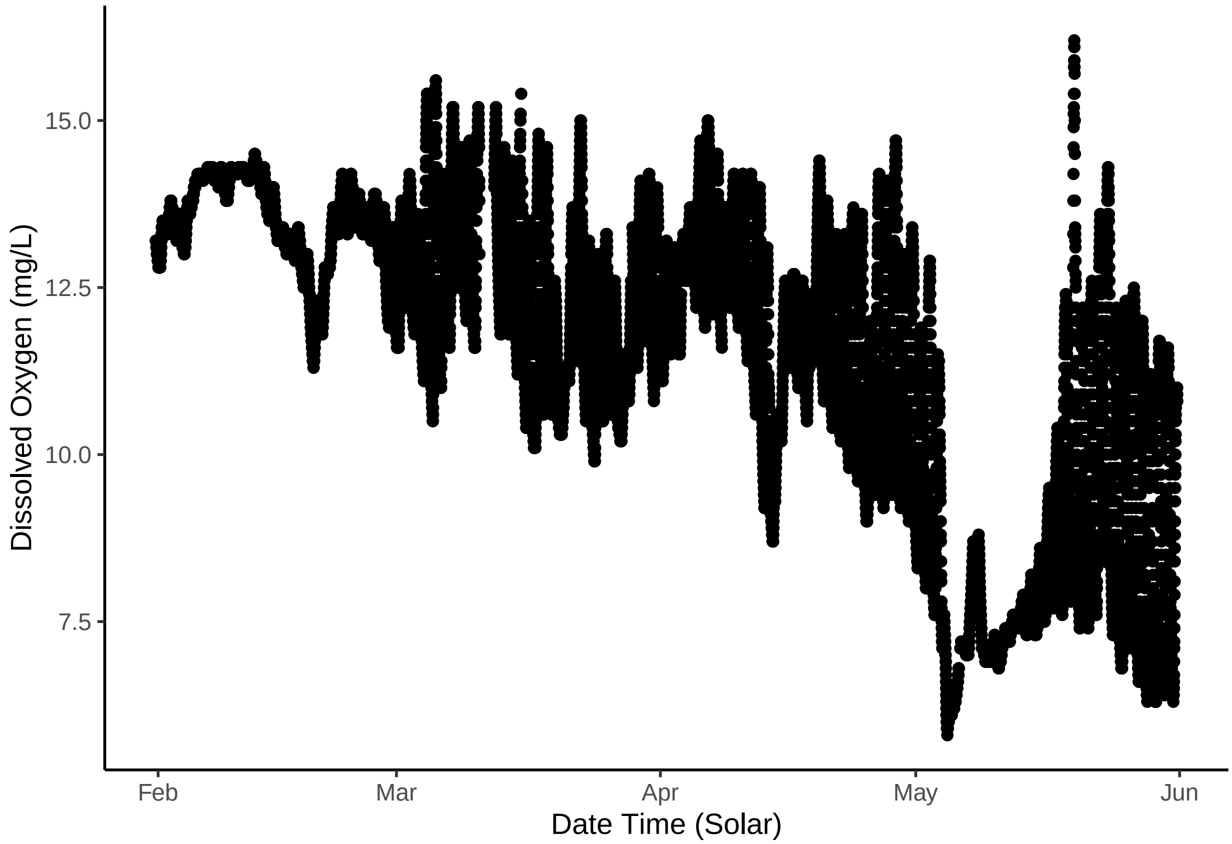
```r
KanDischargeplot <- ggplot(Kandat,
       aes(x = DateTime_Solar, y = Discharge_m3s)) +
  geom_point() +
  labs(x = "Date Time (Solar)", y = expression("Discharge (m"^3*"/s)"))
print(KanDischargeplot)
```

```
## Warning: Removed 640 rows containing missing values (geom_point).
```

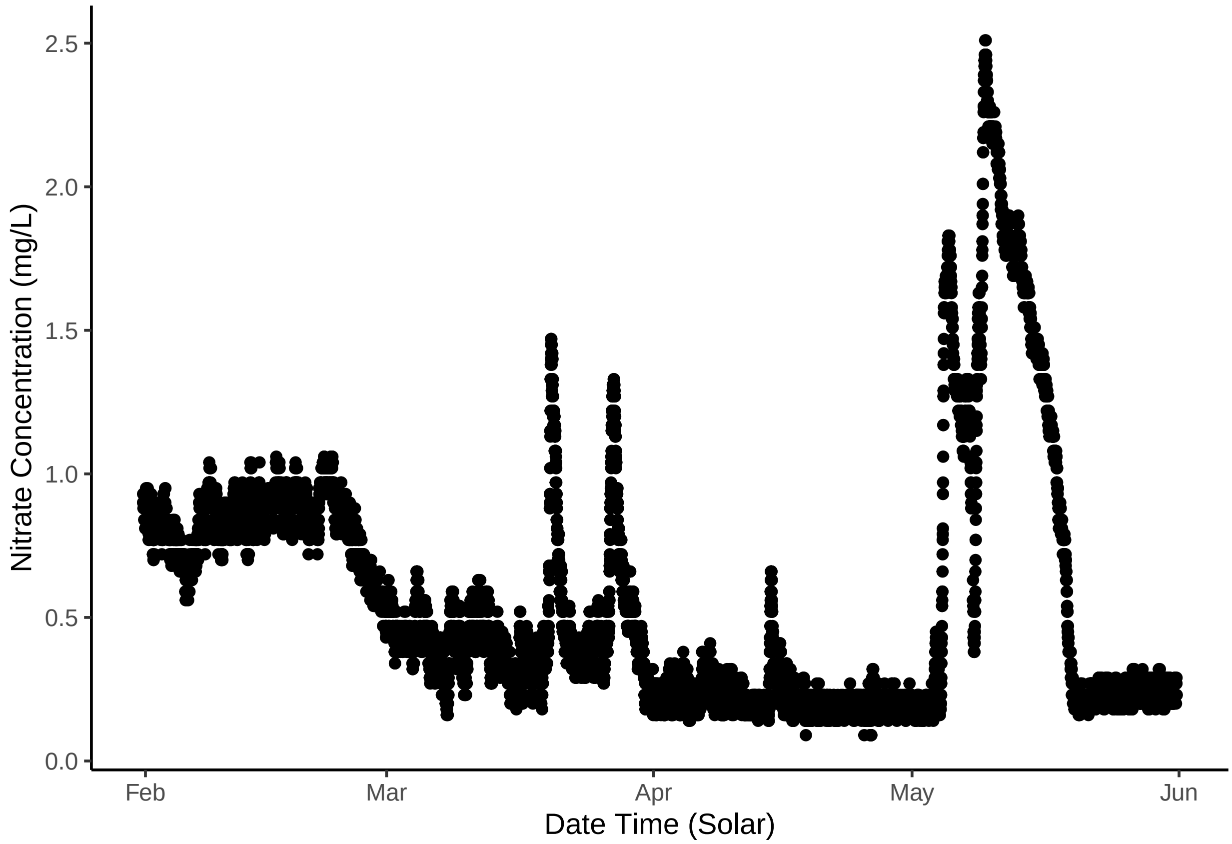



```r
KanDOplot <- ggplot(Kandat,
       aes(x = DateTime_Solar, y = DO_mgL)) +
  geom_point() +
  xlab("Date Time (Solar)") +
  ylab("Dissolved Oxygen (mg/L)")
print(KanDOplot)
```

```
## Warning: Removed 162 rows containing missing values (geom_point).
```




```r
KanNitrateplot <- ggplot(Kandat,
       aes(x = DateTime_Solar, y = Nitrate_mgL)) +
  geom_point() +
  xlab("Date Time (Solar)") +
  ylab("Nitrate Concentration (mg/L)")
print(KanNitrateplot)
```



7. How will you address gaps in these dataseries?

   I will interpolate the data to generate the continuous data, such as piecewise constant interpolation, linear interpolation and spline interpolation.

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

   The daily amplitude of oxygen concentration swings in winter is small, while it is bigger in spring. In summer, the daily amplitude of oxygen concentration swings is much bigger. Because the temperature rises from winter to summer, the solubility of oxygen in water decreases with increasing temperature, while the intensity of photosynthesis increases with increasing temperature. Therefore, the daily amplitude of oxygen concentration swings is bigger with higher temperature.

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.

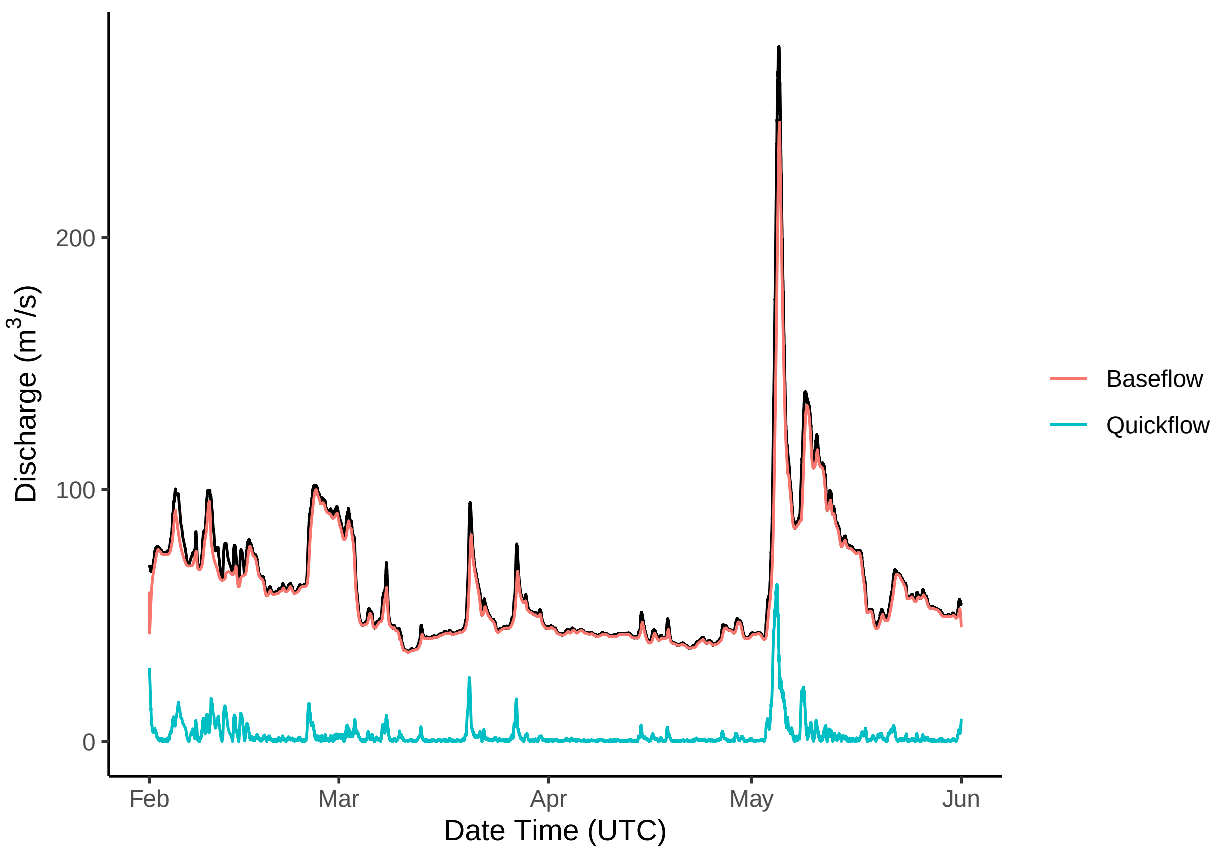
10. Create a ggplot showing total flow, baseflow, and quickflow together.

```
Kanbaseflow <- EcoHydRology::BaseflowSeparation(
  Kandischargedat$Discharge_m3s,
```

```r
  filter_parameter = 0.925,
  passes = 3
  )

Kan2018 <- cbind(Kandischargedat, Kanbaseflow)

Export <- Kan2018 %>%
  mutate(timestep = c(diff(as.numeric(DateTime_UTC)), NA_real_),
         baseflowexport = bt * timestep,
         quickflowexport = qft * timestep) %>%
  summarize(BaseflowExport_KAN = sum(baseflowexport, na.rm = T),
            QuickflowExport_KAN = sum(quickflowexport, na.rm = T),
            TotalExport_KAN = BaseflowExport_KAN + QuickflowExport_KAN)
Export
```

```
##   BaseflowExport_KAN QuickflowExport_KAN TotalExport_KAN
## 1          616413493            27807768       644221261
```

```r
Kanplot <- ggplot(Kan2018, aes(x = DateTime_UTC, y = Discharge_m3s)) +
  geom_line() +
  geom_line(mapping = aes(x = DateTime_UTC, y = bt, color = "darkorange4")) +
  geom_line(mapping = aes(x = DateTime_UTC, y = qft, color = "steelblue4")) +
  labs(x = "Date Time (UTC)", y = expression("Discharge (m"^3*"/s)")) +
  theme(legend.title = element_blank()) +
  scale_color_discrete(labels = c("Baseflow", "Quickflow"))
print(Kanplot)
```



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 baseflow of the Kansas River is much larger than its quickflow, while the baseflow of Third Fork Creek and Ellerbe Creek are similar with their corresponding quickflow. Watershed with bigger area is more likely to have larger quickflow than baseflow.

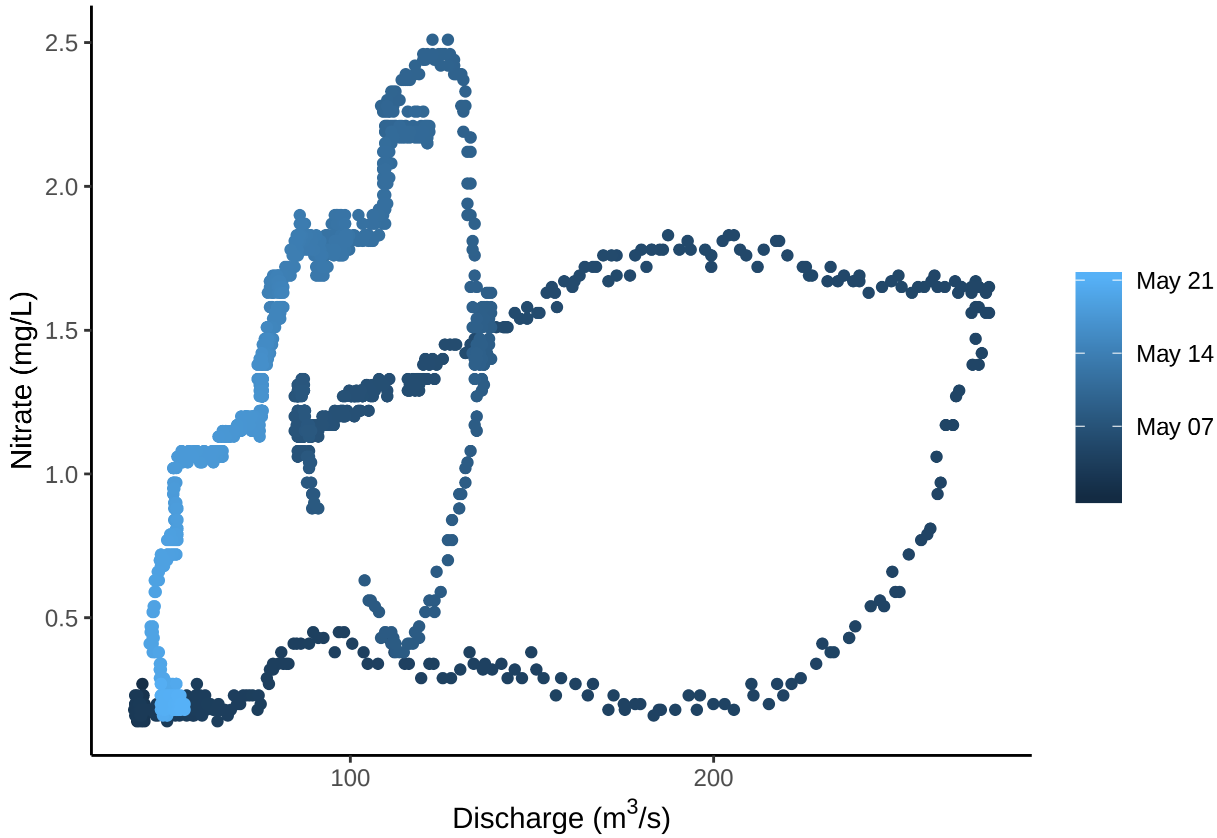
13. The site we are looking at is also further down in its river network (i.e. instead of being a headwater stream, this river has multiple tributaries that flow into it). How does this impact your interpretation of your results?

    Because the site we are looking at is further down its river network, it costs more time for water to aggregate into the river. Therefore, more water will infiltrate and become groundwater, which makes the baseflow of the Kansas River is much larger thanthe 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.

```
KanStorm <- Kandat %>%
  filter(DateTime_Solar > "2018-04-30" & DateTime_Solar < "2018-05-21")

KanNitrateDis <- ggplot(KanStorm,
       aes(x = Discharge_m3s, y = Nitrate_mgL, color = DateTime_Solar)) +
  labs(x = expression("Discharge (m"^3*"/s)"), y = "Nitrate (mg/L)") +
  geom_point() +
  theme(legend.title = element_blank())
print(KanNitrateDis)
```



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

    This storm showed counterclockwise hysteresis. This storm was a flushing storm.

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

    This means that nitrate gets into the river from the watershed by groundwater.

Reflection

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

    Streamflow can be divided into baseflow and quickflow. Baseflow usually comes from groundwater while quickflow usually comes from overlandflow and the water falling into the river channel directly. Hydrologic Flashiness can reflect whether a river responds to a precepitation event quickly. Chemical concentrations in rivers can be either highly variable with discharge (flushing or dilution), or "chemostatic", which means the concentration barely changes with discharge.

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

    Data used includes discharge data, area of watershed and nitrate data from Third Fork Creek at Woodcroft Parkway Near Blands and Ellerbe Creek at Club Boulevard at Durham, NC. Ggplot and dygraph are used to draw those conclusions.

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 convinced me that high frequency data analysis is very useful in the research of rivers or lakes. Also it is more vivid and can help me understand the theory and master some useful tools of processing high frequency data better.

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

    Real-world data is complicated and imperfect because there are lots of factors can affect the value of real-world data. But in general, real-world data has a theoretical law.