Assignment 5: Water Quality in Lakes

Theo Cai

OVERVIEW

This exercise accompanies the lessons in Hydrologic Data Analysis on water quality in lakes

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 HTML file.
5. After Knitting, submit the completed exercise (HTML file) to the dropbox in Sakai. Add your last name into the file name (e.g., "A05_Salk.html") prior to submission.

The completed exercise is due on 2 October 2019 at 9:00 am.

Setup

1. Verify your working directory is set to the R project file,
2. Load the tidyverse, lubridate, and LAGOSNE packages.
3. Set your ggplot theme (can be theme_classic or something else)
4. Load the LAGOSdata database and the trophic state index csv file we created on 2019/09/27.

```
getwd()
```

```
## [1] "Z:/Hydrologic_Data_Analysis"
```

```
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   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
library(lubridate)
```

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

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

```
library(LAGOSNE)

theme_set(theme_classic())

lagosne_get(dest_folder = LAGOSNE:::lagos_path(), overwrite = TRUE)
```

```
## Downloading the 'locus' module ...
## Downloading LAGOSNE_lakeslocus101.csv ...
## Downloading the 'limno' module ...
## Downloading epi_waterquality10873.csv ...
## Downloading lakeslimno10873.csv ...
## Downloading sourceprogram10873.csv ...
## Downloading the 'geo' module ...
## Downloading LAGOSNE_buffer100m_105.csv ...
## Downloading LAGOSNE_buffer100m_lulc105.csv ...
## Downloading LAGOSNE_buffer500m_105.csv ...
## Downloading LAGOSNE_buffer500m_conn105.csv ...
## Downloading LAGOSNE_buffer500m_lulc105.csv ...
## Downloading LAGOSNE_county_105.csv ...
## Downloading LAGOSNE_county_chag105.csv ...
## Downloading LAGOSNE_county_conn105.csv ...
## Downloading LAGOSNE_county_lulc105.csv ...
## Downloading LAGOSNE_edu_105.csv ...
## Downloading LAGOSNE_edu_chag105.csv ...
## Downloading LAGOSNE_edu_conn105.csv ...
## Downloading LAGOSNE_edu_lulc105.csv ...
## Downloading LAGOSNE_hu4_105.csv ...
## Downloading LAGOSNE_hu4_chag105.csv ...
## Downloading LAGOSNE_hu4_conn105.csv ...
## Downloading LAGOSNE_hu4_lulc105.csv ...
## Downloading LAGOSNE_hu8_105.csv ...
## Downloading LAGOSNE_hu8_chag105.csv ...
## Downloading LAGOSNE_hu8_conn105.csv ...
## Downloading LAGOSNE_hu8_lulc105.csv ...
## Downloading LAGOSNE_hu12_105.csv ...
## Downloading LAGOSNE_hu12_chag105.csv ...
## Downloading LAGOSNE_hu12_conn105.csv ...
## Downloading LAGOSNE_hu12_lulc105.csv ...
## Downloading LAGOSNE_iws_105.csv ...
## Downloading LAGOSNE_iws_conn105.csv ...
## Downloading LAGOSNE_iws_lulc105.csv ...
## Downloading LAGOSNE_lakesgeo105.csv ...
```

```
## Downloading LAGOSNE_state_105.csv ...
## Downloading LAGOSNE_state_chag105.csv ...
## Downloading LAGOSNE_state_conn105.csv ...
## Downloading LAGOSNE_state_lulc105.csv ...
## LAGOSNE downloaded. Now compressing to native R object ...
## LAGOSNE compiled to C:\Users\gyc4\AppData\Local\LAGOSNE\LAGOSNE//data_1.087.3.rds
```

```
LAGOSdata <- lagosne_load()
```

```
## Warning in `_f`(version = version, fpath = fpath): LAGOSNE version
## unspecified, loading version: 1.087.3
```

```
library(readr)
LAGOStrophic <- read_csv("Data/LAGOStrophic.csv")
```

```
## Parsed with column specification:
## cols(
##   lagoslakeid = col_double(),
##   sampledate = col_date(format = ""),
##   chla = col_double(),
##   tp = col_double(),
##   secchi = col_double(),
##   gnis_name = col_character(),
##   lake_area_ha = col_double(),
##   state = col_character(),
##   state_name = col_character(),
##   sampleyear = col_double(),
##   samplemonth = col_double(),
##   season = col_character(),
##   TSI.chl = col_double(),
##   TSI.secchi = col_double(),
##   TSI.tp = col_double(),
##   trophic.class = col_character()
## )
```

Trophic State Index

5. Similar to the trophic.class column we created in class (determined from TSI.chl values), create two additional columns in the data frame that determine trophic class from TSI.secchi and TSI.tp (call these trophic.class.secchi and trophic.class.tp).

```
#Trophic class from TSI.chl
LAGOStrophic <-
  mutate(LAGOStrophic,
         trophic.class =
            ifelse(TSI.chl < 40, "Oligotrophic",
                   ifelse(TSI.chl < 50, "Mesotrophic",
                          ifelse(TSI.chl < 70, "Eutrophic", "Hypereutrophic"))))

#Trophic class from TSI.secchi
LAGOStrophic <-
  mutate(LAGOStrophic,
         trophic.class.secchi =
            ifelse(TSI.secchi < 40, "Oligotrophic",
```

```
                  ifelse(TSI.secchi < 50, "Mesotrophic",
                         ifelse(TSI.secchi < 70, "Eutrophic", "Hypereutrophic"))))

#Trophic class from TSI.tp
LAGOStrophic <-
  mutate(LAGOStrophic,
         trophic.class.tp =
           ifelse(TSI.tp < 40, "Oligotrophic",
                  ifelse(TSI.tp < 50, "Mesotrophic",
                         ifelse(TSI.tp < 70, "Eutrophic", "Hypereutrophic"))))
```

6. How many observations fall into the four trophic state categories for the three metrics (trophic.class, trophic.class.secchi, trophic.class.tp)? Hint: `count` function.

```
count(LAGOStrophic, trophic.class)
```

```
## # A tibble: 4 x 2
##   trophic.class      n
##   <chr>          <int>
## 1 Eutrophic      41861
## 2 Hypereutrophic 14379
## 3 Mesotrophic    15413
## 4 Oligotrophic    3298
```

```
count(LAGOStrophic, trophic.class.secchi)
```

```
## # A tibble: 4 x 2
##   trophic.class.secchi     n
##   <chr>                <int>
## 1 Eutrophic            28659
## 2 Hypereutrophic        5099
## 3 Mesotrophic          25083
## 4 Oligotrophic         16110
```

```
count(LAGOStrophic, trophic.class.tp)
```

```
## # A tibble: 4 x 2
##   trophic.class.tp     n
##   <chr>            <int>
## 1 Eutrophic        24839
## 2 Hypereutrophic    7228
## 3 Mesotrophic      23023
## 4 Oligotrophic     19861
```

7. What proportion of total observations are considered eutrophic or hypereutrophic according to the three different metrics (trophic.class, trophic.class.secchi, trophic.class.tp)?

```
#Proportion of trophic.class
(41861+14379)/74951
```

```
## [1] 0.7503569
```

```
#Proportion of trophic.class.secchi
(28659+5099)/74951
```

```
## [1] 0.4504009
```

```
#Proportion of trophic.class.tp
(24839+7228)/74951
```

```
## [1] 0.4278395
```

Which of these metrics is most conservative in its designation of eutrophic conditions? Why might this be?

> Total phosphorus is the most conservative in its designation of eutrophic conditions. This might be because just the presence of phosphorus is not enough to directly indicate the level of biomass in a system. The nutrient definitely feeds into it - being one of the nutrients that algae rely on to grow - but there are so many other factors that determine biomass level that phosphorus alone cannot predict eutrophic conditions. Plus, it would seem that, since phosphorus is the food source for biomass growth, higher levels of it might just mean it has not been depleted by said growth yet. (Basically, higher levels might indicate a future trend towards eutrophic conditions in the ecosystem, but say little about the present conditions.)

Note: To take this further, a researcher might determine which trophic classes are susceptible to being differently categorized by the different metrics and whether certain metrics are prone to categorizing trophic class as more or less eutrophic. This would entail more complex code.

Nutrient Concentrations

8. Create a data frame that includes the columns lagoslakeid, sampledate, tn, tp, state, and state_name. Mutate this data frame to include sampleyear and samplemonth columns as well. Call this data frame LAGOSNandP.

```
LAGOSnutrient <- LAGOSdata$epi_nutr
LAGOSlocus <- LAGOSdata$locus
LAGOSstate <- LAGOSdata$state

LAGOSlocus$lagoslakeid <- as.factor(LAGOSlocus$lagoslakeid)
LAGOSnutrient$lagoslakeid <- as.factor(LAGOSnutrient$lagoslakeid)

LAGOSlocations <- left_join(LAGOSlocus, LAGOSstate, by = "state_zoneid")

LAGOSNandP <-
  left_join(LAGOSnutrient, LAGOSlocations, by = "lagoslakeid") %>%
  select(sampledate, lagoslakeid, state, state_name, tn, tp) %>%
  mutate(sampleyear = year(sampledate),
         samplemonth = month(sampledate)) %>%
  drop_na()
```

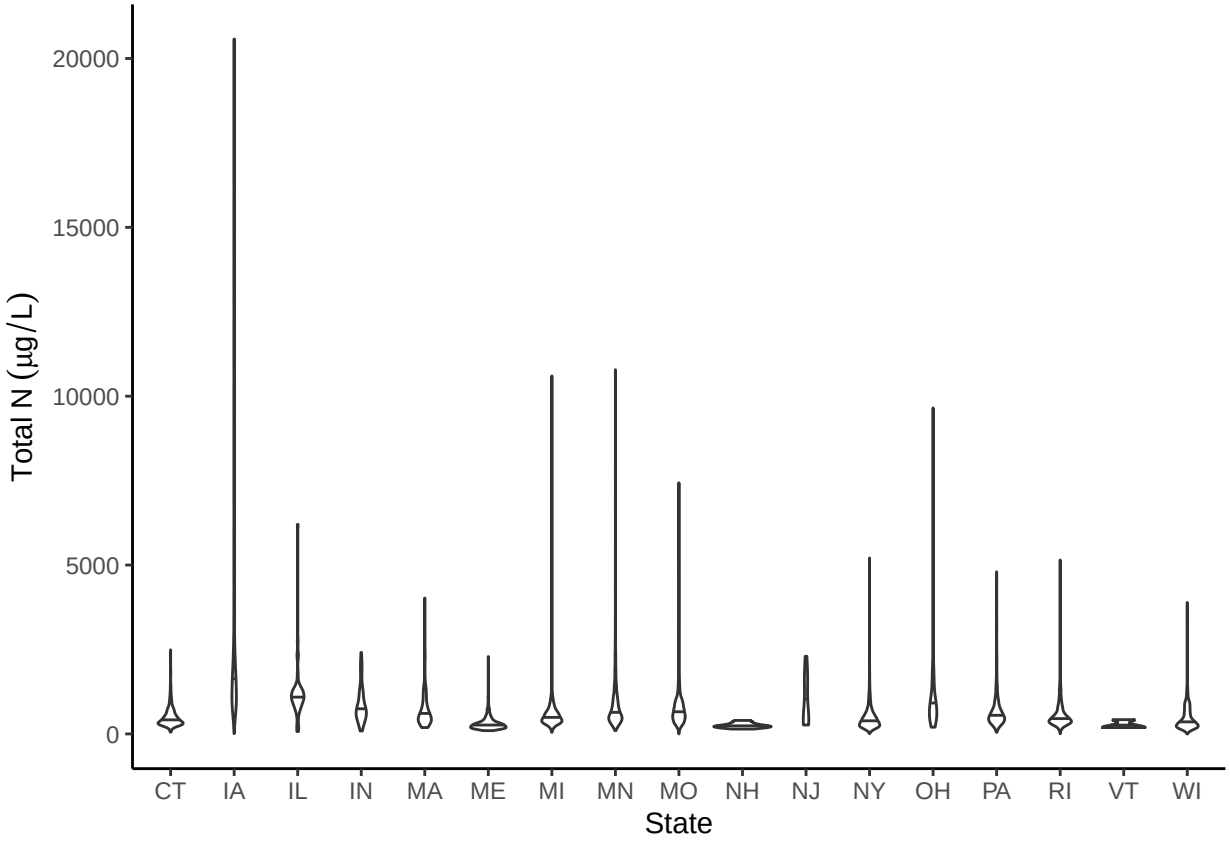
```
## Warning: Column `lagoslakeid` joining factors with different levels,
## coercing to character vector
```

9. Create two violin plots comparing TN and TP concentrations across states. Include a 50th percentile line inside the violins.

```
TNstateviolin <- ggplot(LAGOSNandP, aes(x = state, y = tn)) +
  geom_violin(draw_quantiles = 0.50) +
  labs(y = expression(Total ~ N ~ (mu*g / L)), x = "State")
print(TNstateviolin)
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```

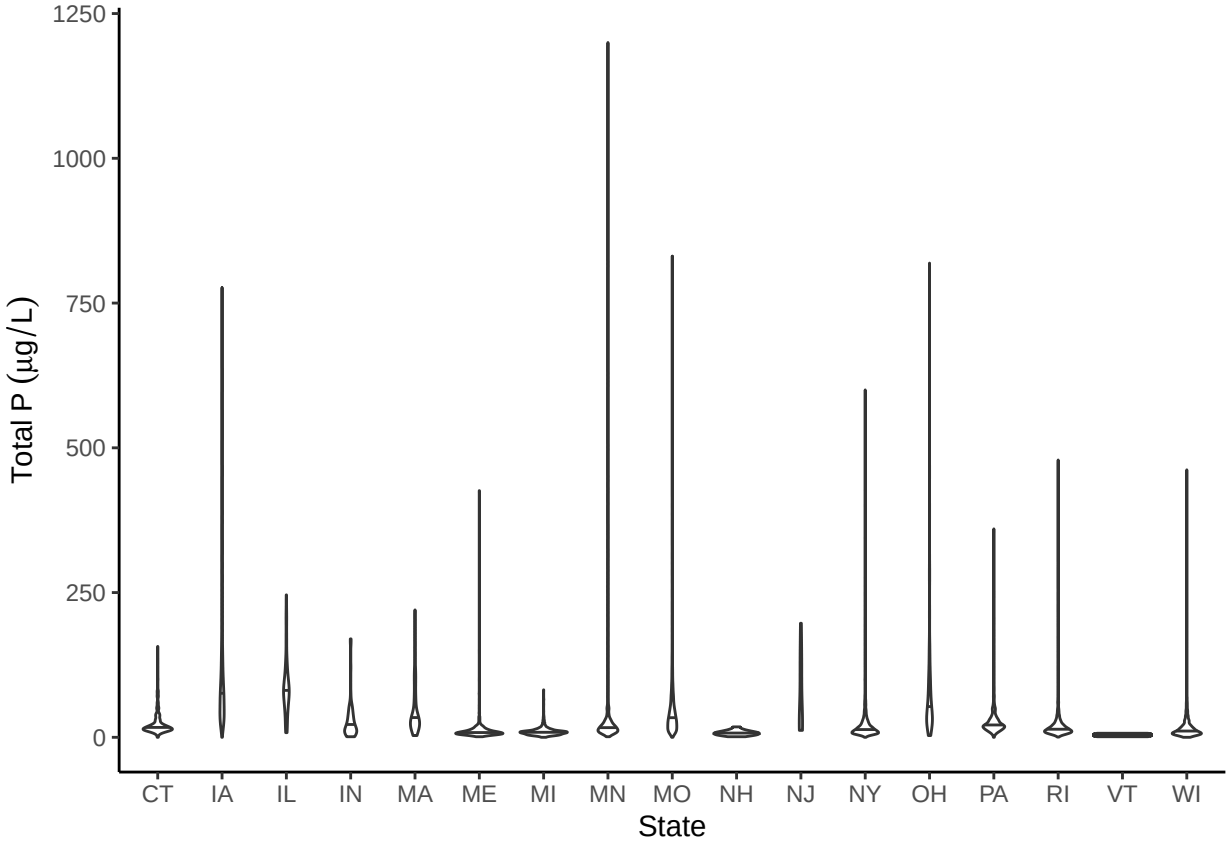
```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```




```
TPstateviolin <- ggplot(LAGOSNandP, aes(x = state, y = tp)) +
  geom_violin(draw_quantiles = 0.50) +
  labs(y = expression(Total ~ P ~ (mu*g / L)), x = "State")
print(TPstateviolin)
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```



Which states have the highest and lowest median concentrations?

TN: IA, highest. VT, lowest.

TP: IL, highest. VT, lowest.

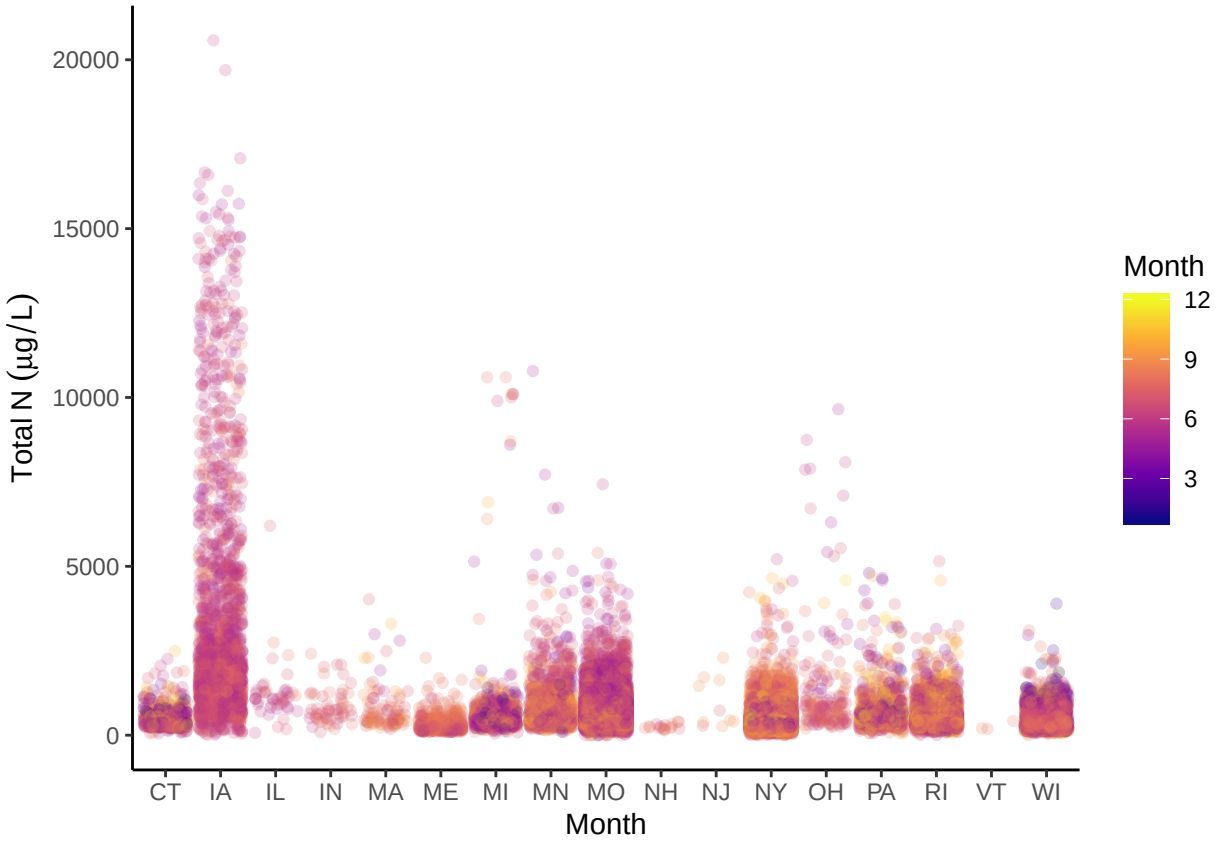
Which states have the highest and lowest concentration ranges?

TN: IA, highest. VT, lowest.

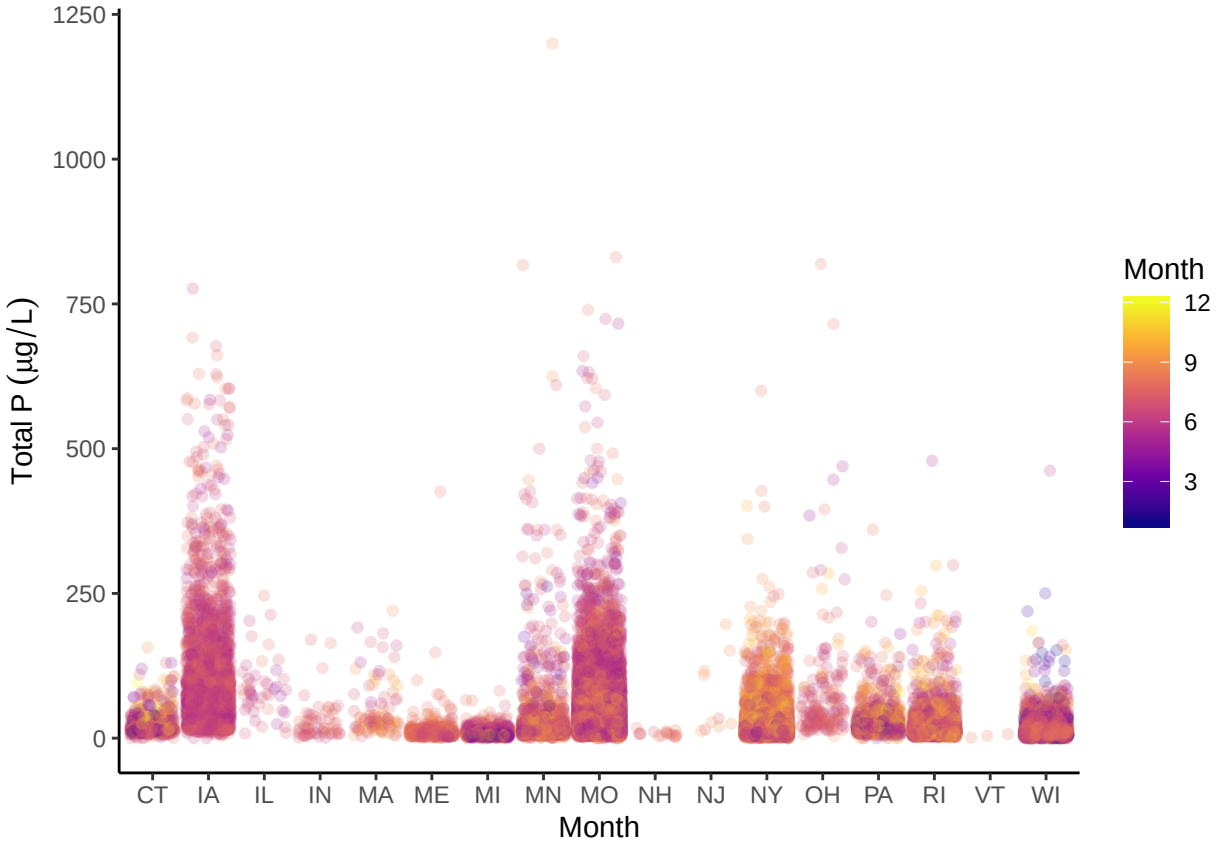
TP: MN, highest. VT, lowest.

10. Create two jitter plots comparing TN and TP concentrations across states, with samplemonth as the color. Choose a color palette other than the ggplot default.

```
TNstatejitter <-
ggplot(LAGOSNandP,
       aes(x = state, y = tn, color = samplemonth)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "Month", y = expression(Total ~ N ~ (mu*g / L)), color = "Month") +
  scale_color_viridis_c(option = "plasma")
print(TNstatejitter)
```




```
TPstatejitter <-
ggplot(LAGOSNandP,
       aes(x = state, y = tp, color = samplemonth)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "Month", y = expression(Total ~ P ~ (mu*g / L)), color = "Month") +
  scale_color_viridis_c(option = "plasma")
print(TPstatejitter)
```



Which states have the most samples? How might this have impacted total ranges from #9?

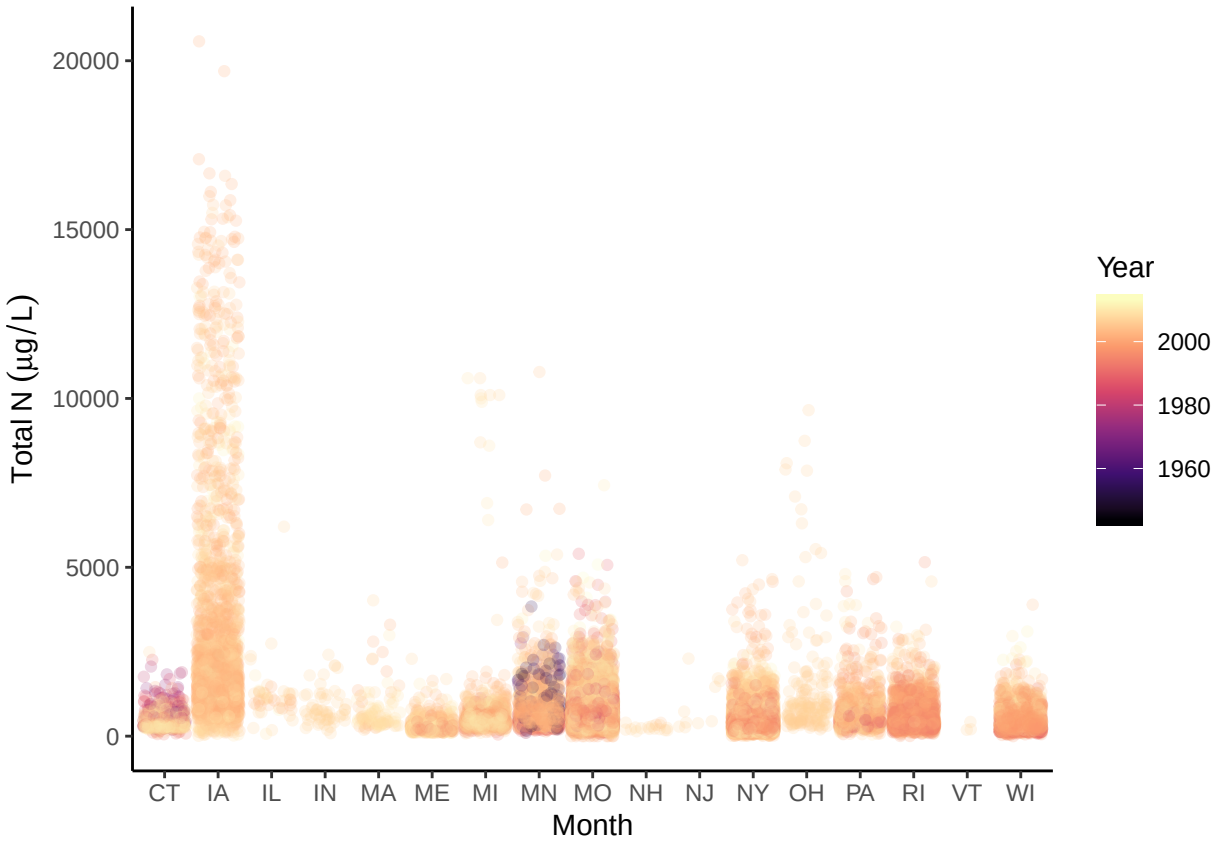
> TN: IA and MO have the most samples. I think that, the more samples a state has, the more I trust the total range of measurements that I saw in its violin plot. It seems to me that, the larger the sample size, the less likely the median, range, or even any one measurement is a fluke/unusual result due to limited sampling.
>
> TP: IA and MO have the most samples here again.

Which months are sampled most extensively? Does this differ among states?

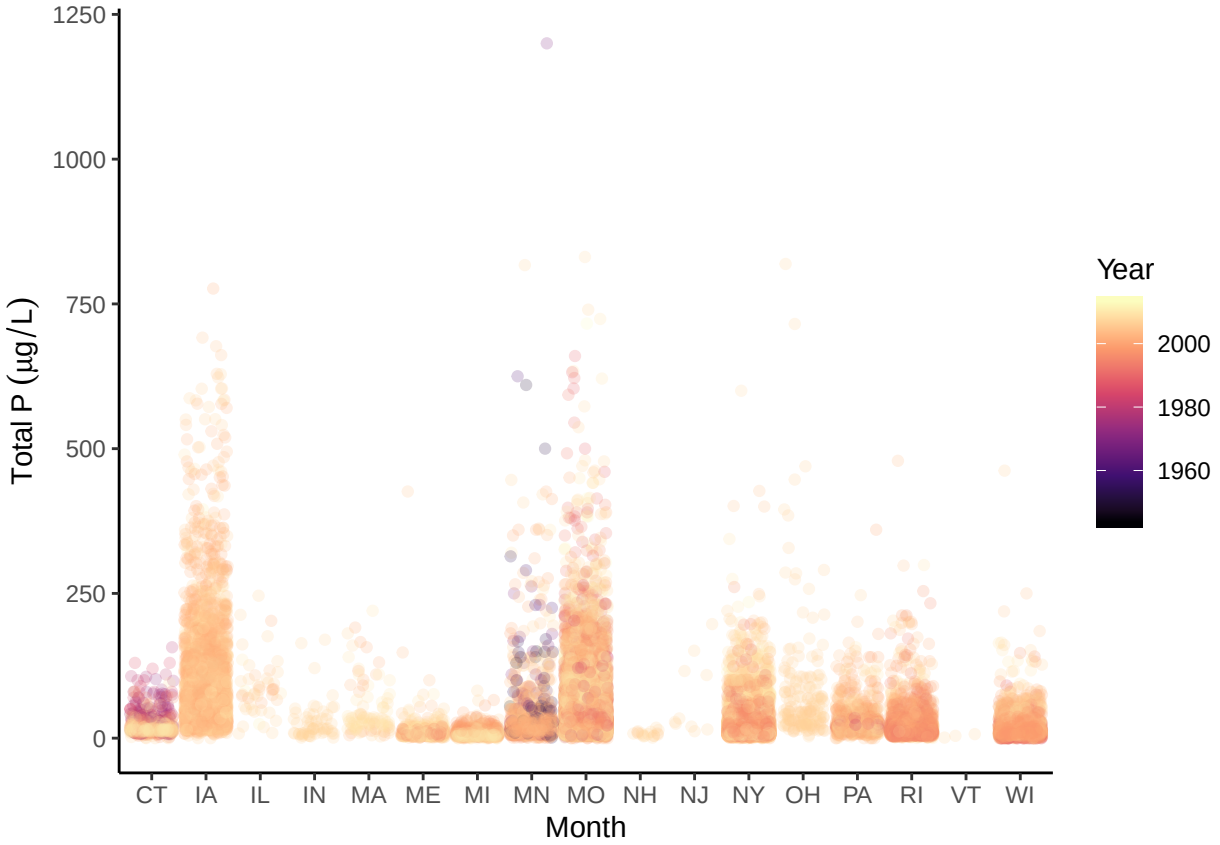
> TN: IA and Mo are sampled most extensively here. Yes, this differs widely among states; some, like ME, are sampling only in the fall-winter, and others, like NJ and VT, have barely been sampled at all.
>
> TP: It seems that MO is sampled the most extensively out of all the states - there are data points from all through the year.

11. Create two jitter plots comparing TN and TP concentrations across states, with sampleyear as the color. Choose a color palette other than the ggplot default.

```
TNyearstatejitter <-
ggplot(LAGOSNandP,
       aes(x = state, y = tn, color = sampleyear)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "Month", y = expression(Total ~ N ~ (mu*g / L)), color = "Year") +
  scale_color_viridis_c(option = "magma")
print(TNyearstatejitter)
```




```
TPyearstatejitter <-
ggplot(LAGOSNandP,
       aes(x = state, y = tp, color = sampleyear)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "Month", y = expression(Total ~ P ~ (mu*g / L)), color = "Year") +
  scale_color_viridis_c(option = "magma")
print(TPyearstatejitter)
```



Which years are sampled most extensively? Does this differ among states?

TN: The last two decades are sampled the most extensively across the board. This does differ slightly among states: MN and CT seem to have older samples from as far back as the 1960s.

TP: Same as with nitrogen, most of the samples from each state are from the most recent two decades. MN and Ct are, once again, different in that they have data points from further in the past ~1950s, ~1960s.

Reflection

12. What are 2-3 conclusions or summary points about lake water quality you learned through your analysis?

    I learned that chlorophyl is the most reactive/liberal indicator of trophic state in an ecosystem, and that certain measures of water quality like N and P were not prioritized until relatively recently.

13. What data, visualizations, and/or models supported your conclusions from 12?

    For the first one: the counts in question 6 and the proportions in question 7 indicated to me the conclusion. For the second one, the two jitter plots helped me.

14. Did hands-on data analysis impact your learning about water quality relative to a theory-based lesson? If so, how?

    Yes! As always, hands-on data analysis is better because it feels more personalized and concrete, whereas a theory-based lesson is abstract and sometimes difficult to grasp.

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

    It matched up pretty well! Not many surprises.