Assignment 5: Water Quality in Lakes

Yixin Wen

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] "/Users/yixinwen/Box/Duke/2019 Fall/Hydrologic Data Analysis/Hydrologic_Data_Analysis/Assignments
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------- tidyverse 1.2.1 --
```

```
## 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 -------------------------- tidyverse_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())
load(file = "/Users/yixinwen/Box/Duke/2019 Fall/Hydrologic Data Analysis/Hydrologic_Data_Analysis/Data/
LAGOStrophic <-read_csv("/Users/yixinwen/Box/Duke/2019 Fall/Hydrologic Data Analysis/Hydrologic_Data_Ana
```

```
## 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).

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

```
trophic.class <-
  LAGOStrophic %>%
    count(trophic.class)
unique(trophic.class)
```

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

```
trophic.class.secchi <-
  LAGOStrophic %>%
    count(trophic.class.secchi)
```

```
unique(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
```

```
trophic.class.tp <-
  LAGOStrophic %>%
    count(trophic.class.tp)
unique(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
```

for trophic.class, there are 41861 in Eutrophic, 14379 in Hpereutrophic, 15413 in Mesotrophic and 3298 in Oligotrophic; for trophic.class.secchi, there are 28659 in Eutrophic, 5099 in Hpereutrophic, 25083 in Mesotrophic and 16110 in Oligotrophic; for trophic.class.tp, there are 24839 in Eutrophic, 7228 in Hpereutrophic, 23023 in Mesotrophic and 19861 in Oligotrophic.

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

```
trophic.class.pro = (trophic.class$n[1]+trophic.class$n[2])/74951
trophic.class.secchi.pro = (trophic.class.secchi$n[1]+trophic.class.secchi$n[2])/74951
trophic.class.tp.pro = (trophic.class.tp$n[1]+trophic.class.tp$n[2])/74951
```

In trophic.class, the proportion is 0.75; in trophic.class.secchi, the proportion is 0.45; in trophic.class.tp, the proportion is 0.43.

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

trophic.class.tp is the most conservative metrics in eutrophic conditions. Since the limitation of phosphorus for phytoplankton growth only appears in summer.Thus, the data would be more conservative than the other two.

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.

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

```
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(lagoslakeid, sampledate, tn, tp,
         state, state_name) %>%
  mutate(sampleyear = year(sampledate),
         samplemonth = month(sampledate))
```

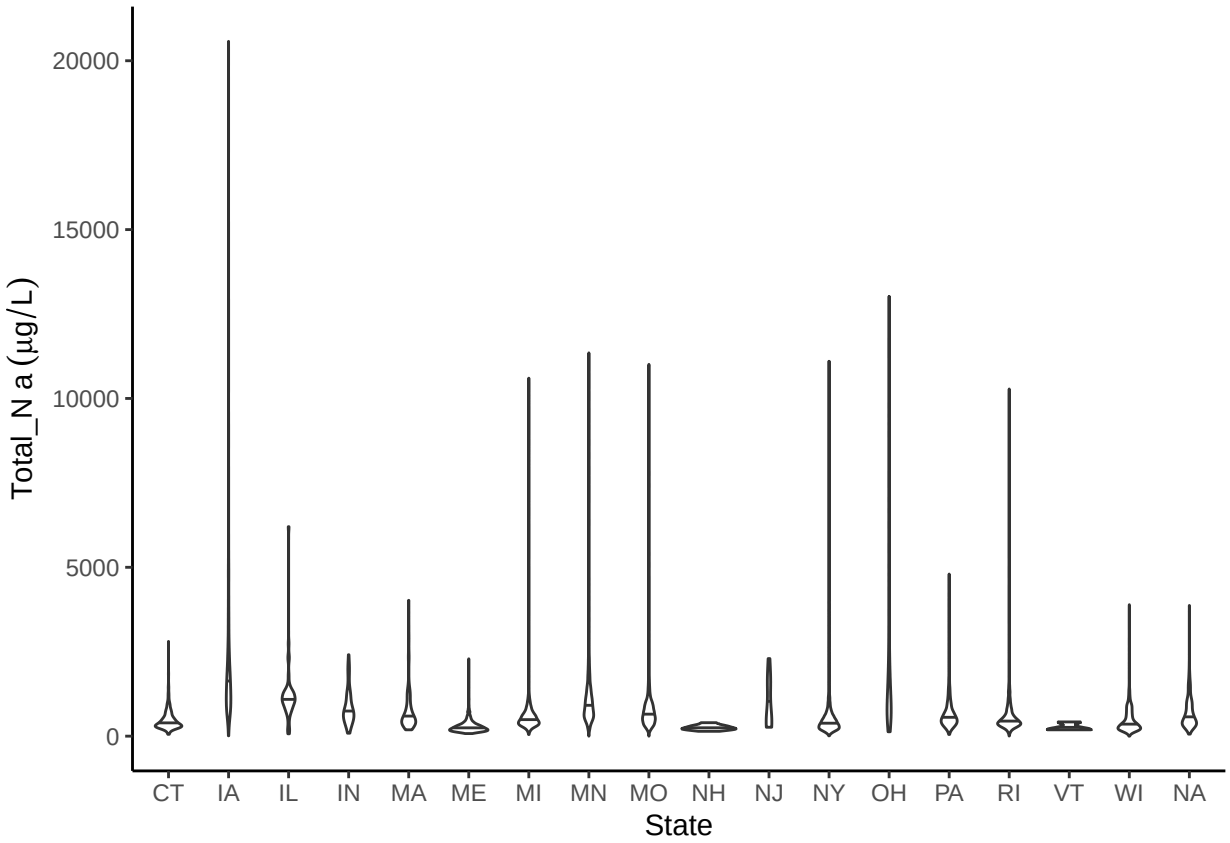
```
## Warning: Column `lagoslakeid` joining factors with different levels,
## coercing to character vector
```

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

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

```
## Warning: Removed 774226 rows containing non-finite values (stat_ydensity).
```

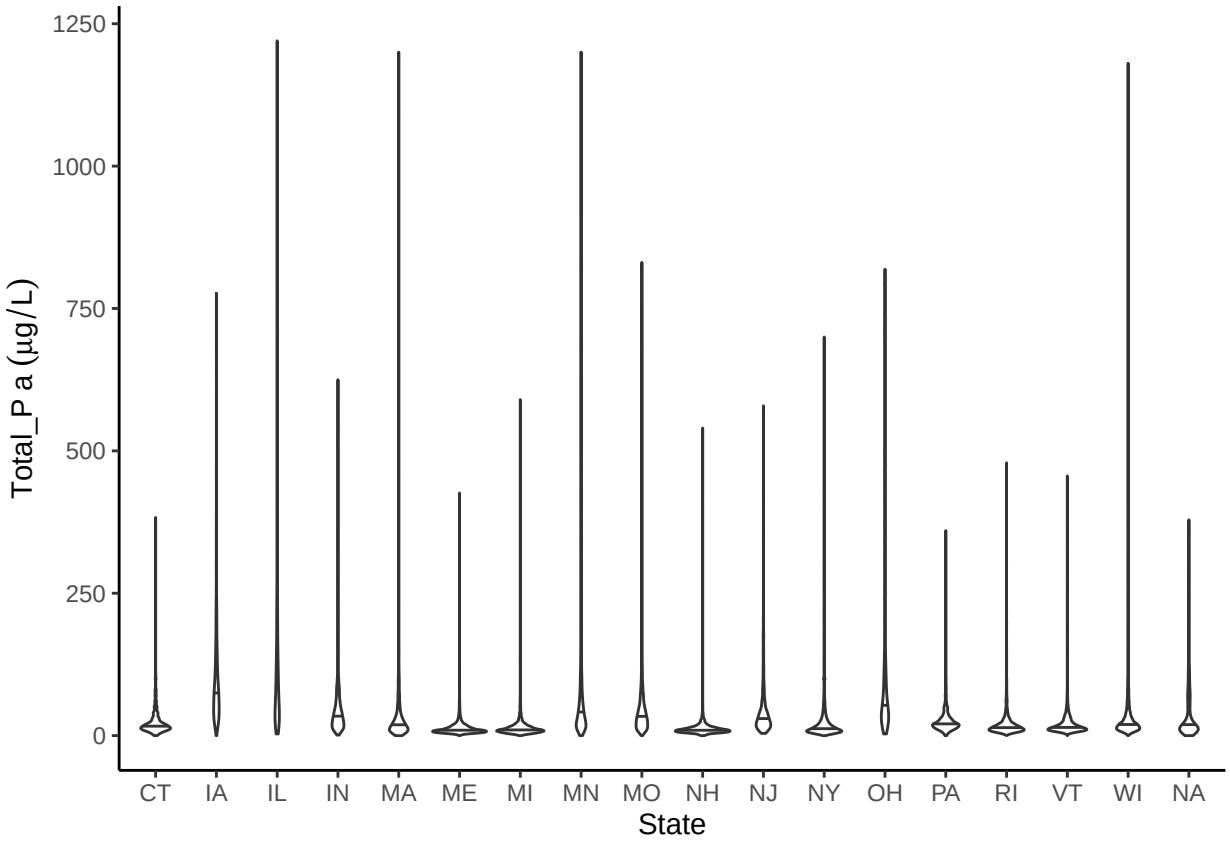
```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values

## 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
```




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

```
## Warning: Removed 672861 rows containing non-finite values (stat_ydensity).

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values
```

```
## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values
```




```
Nstate <-
  LAGOSNandP %>%
  select(state,tn)%>%
  group_by(state)%>%
  summarise(tn0.5 = quantile(tn, probs = 0.5, na.rm = TRUE))%>%
  mutate(tn0.5)

Pstate <-
  LAGOSNandP %>%
  select(state,tp)%>%
  group_by(state)%>%
  summarise(tp0.5 = quantile(tp, probs = 0.5, na.rm = TRUE))%>%
  mutate(tp0.5)
```

Which states have the highest and lowest median concentrations?

TN: IA has the highest median concentrations, VT and NH have the lowest median concentrations.

TP: IL has the highest median concentrations, MI,NH and ME have the lowest median concentrations.

Which states have the highest and lowest concentration ranges?

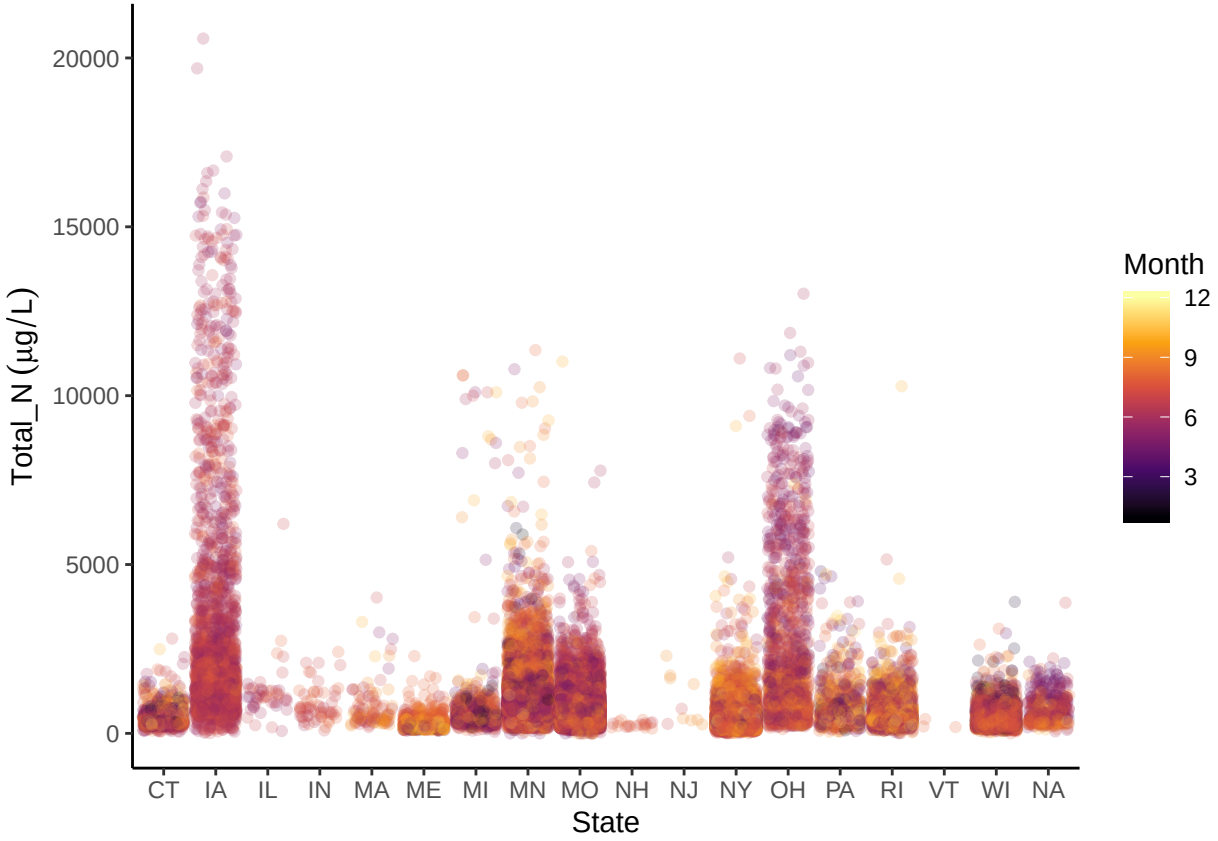
TN: IA has highest concentration ranges, VT and NH have the lowest concentration ranges.

TP: IL has highest concentration ranges, CT and PA have the lowest concentration ranges.

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.

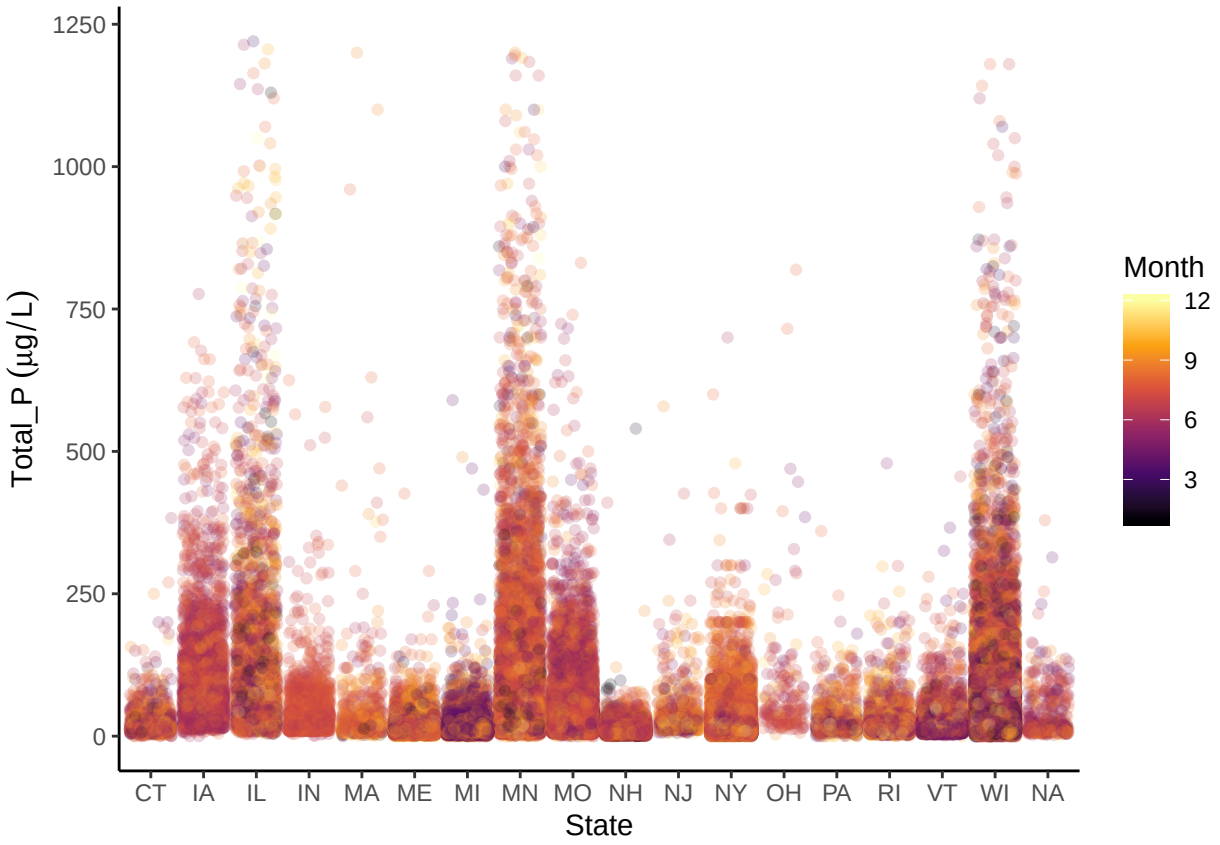
```
stateTNjitterbymonth <-
ggplot(LAGOSNandP,
       aes(x = state, y = tn, color = samplemonth)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "State", y = expression(Total_N ~ (mu*g / L)), color = "Month") +
  scale_color_viridis_c(option = "inferno")
print(stateTNjitterbymonth)
```

```
## Warning: Removed 774226 rows containing missing values (geom_point).
```

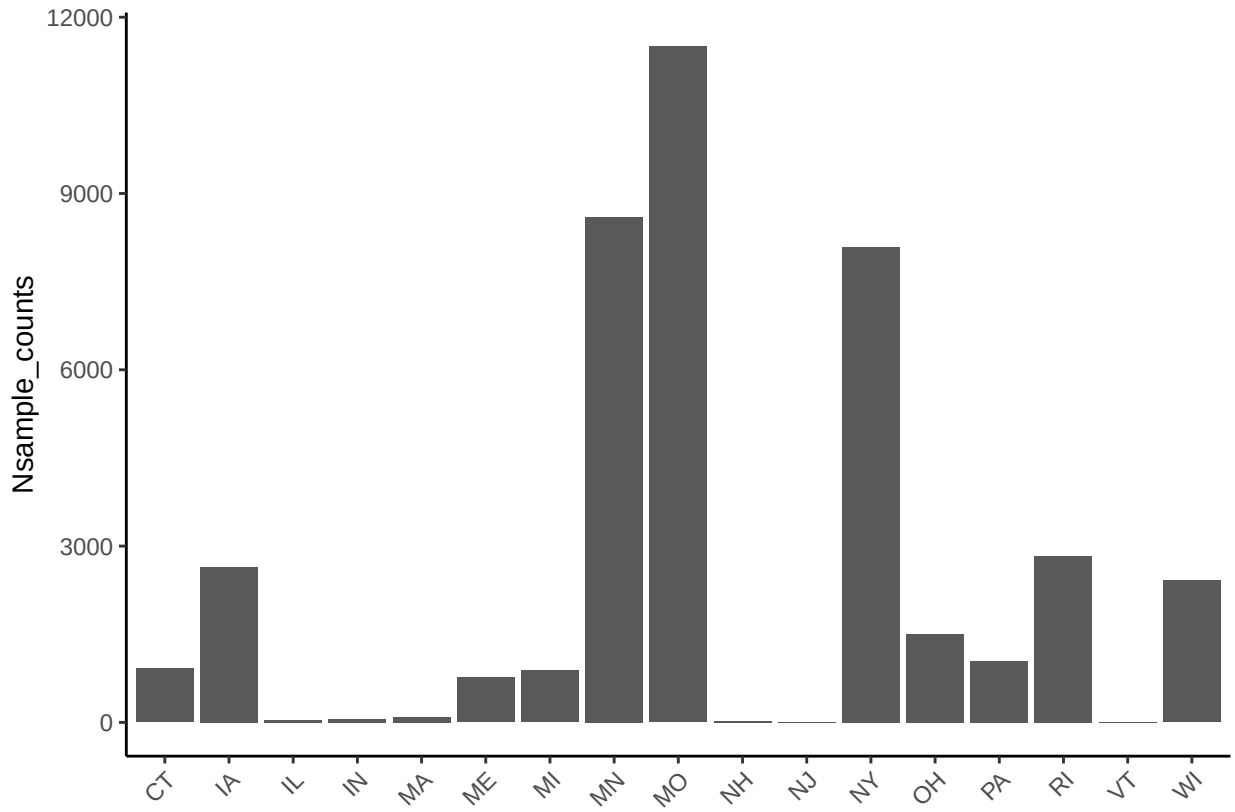



```
stateTPjitterbymonth <-
ggplot(LAGOSNandP,
       aes(x = state, y = tp, color = samplemonth)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "State", y = expression(Total_P ~ (mu*g / L)), color = "Month") +
  scale_color_viridis_c(option = "inferno")
print(stateTPjitterbymonth)
```

```
## Warning: Removed 672861 rows containing missing values (geom_point).
```




```
# count the sample data
Nmonth <-
  LAGOSNandP%>%
  select(state,tn)
Nmonth <- na.omit(Nmonth)

NCounts <- ggplot(Nmonth, aes(x = state)) +
  geom_bar(stat = "count") +
  labs(x = "", y = "Nsample_counts")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
print(NCounts)
```

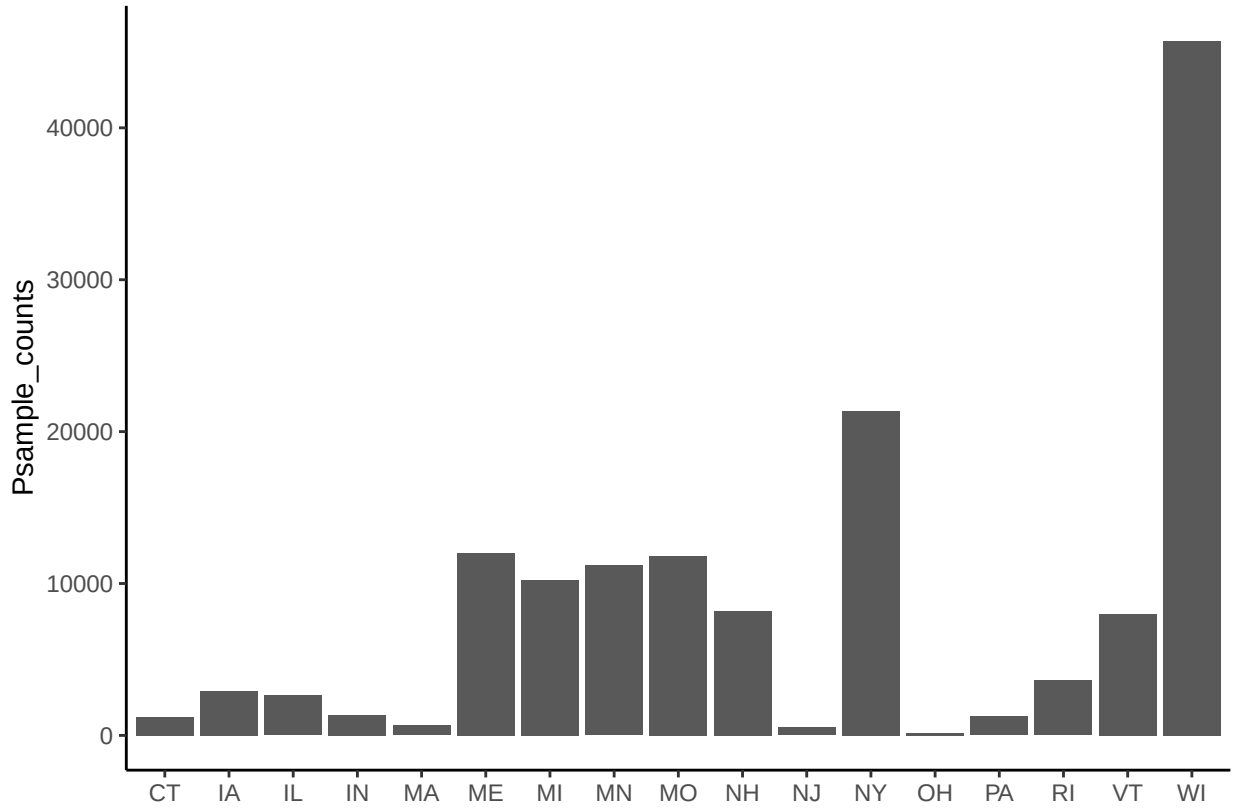



```
Pmonth <-
  LAGOSNandP%>%
  select(state,tp)
Pmonth <- na.omit(Pmonth)

PCounts <- ggplot(Pmonth, aes(x = state)) +
  geom_bar(stat = "count") +
  labs(x = "", y = "Psample_counts")
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
```

```
## List of 1
##  $ axis.text.x:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : num 45
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
```

```
##  - attr(*, "validate")= logi TRUE
```

```
print(PCounts)
```



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

> TN:MO has the most samples. The range of MO is much smaller than IA. When the data sample is large enough, it can give a more accurate result of concentration range. If the data sample is not enough, it has greater opportunity ot have error point, which cannot show the real range of N concentration.
>
> TP:WI has the most samples. WI still has a wide range of TP concentration. It may because that more data samples can includes different levels of data.

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

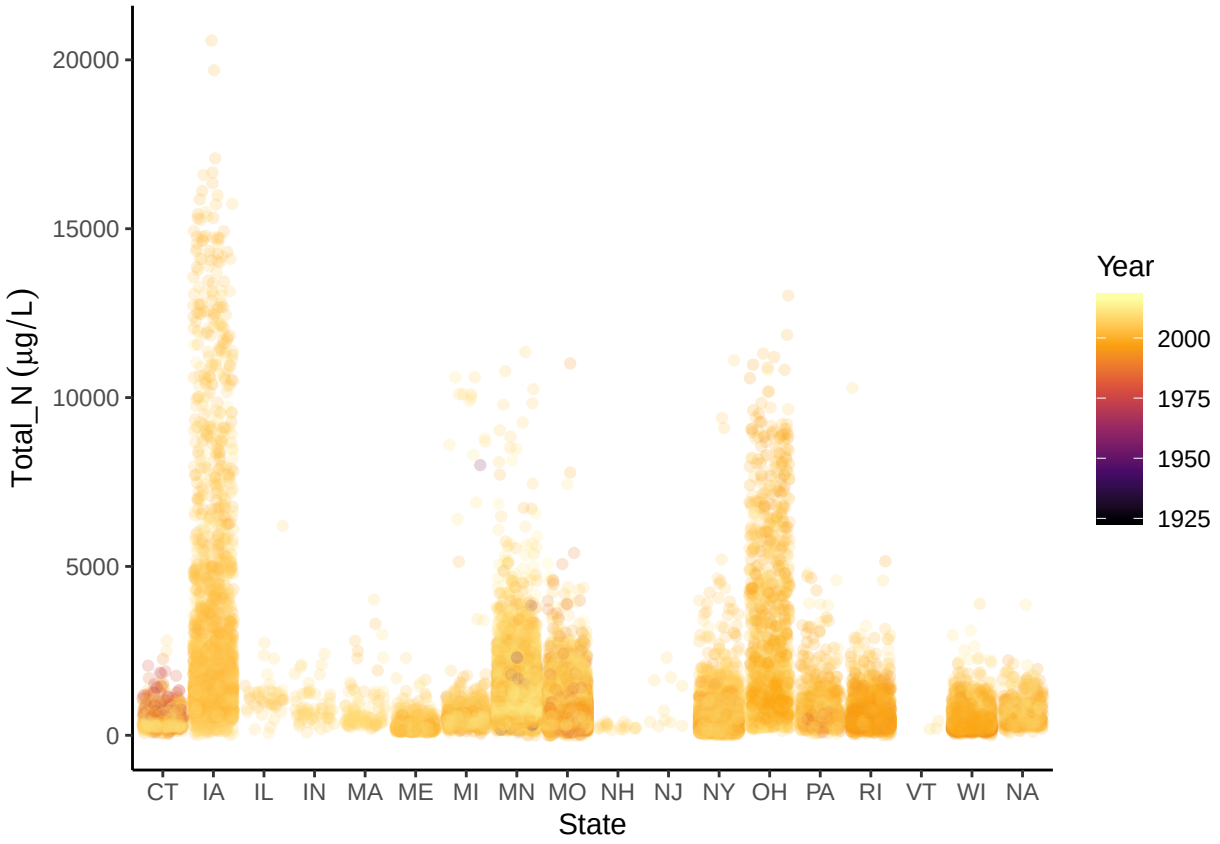
> TN:IA,OH,MO are sampled extensively in June or July. MA, ME, MN, NY,RI are sampled extensively in September or Octorber.
>
> TP: MI is sampled extensively in March or April; Other states are sampled extensively in July or August.

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.

```
stateTNjitterbyyear <-
ggplot(LAGOSNandP,
       aes(x = state, y = tn, color = sampleyear)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "State", y = expression(Total_N ~ (mu*g / L)), color = "Year") +
```

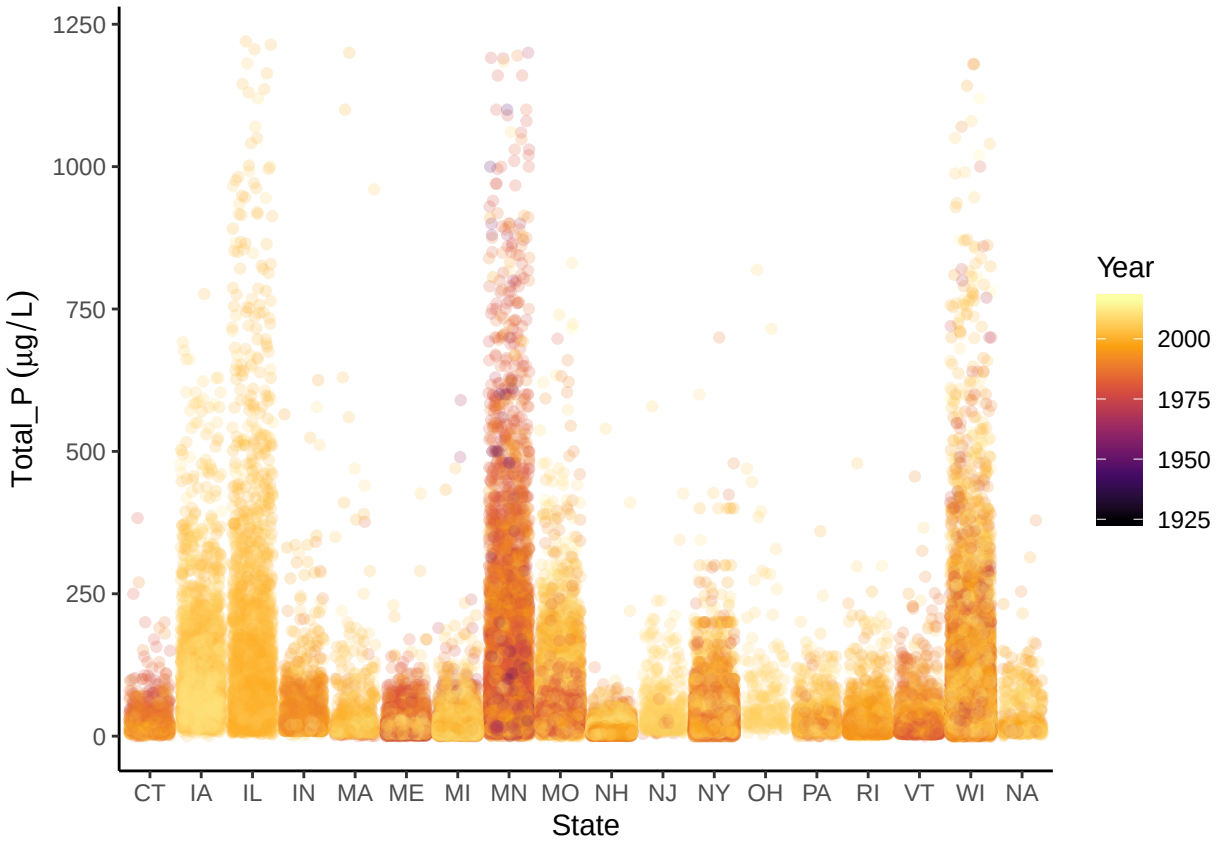
```
  scale_color_viridis_c(option = "inferno")
print(stateTNjitterbyyear)
```

```
## Warning: Removed 774226 rows containing missing values (geom_point).
```




```
stateTPjitterbyyear <-
ggplot(LAGOSNandP,
       aes(x = state, y = tp, color = sampleyear)) +
  geom_jitter(alpha = 0.2) +
  labs(x = "State", y = expression(Total_P ~ (mu*g / L)), color = "Year") +
  scale_color_viridis_c(option = "inferno")
print(stateTPjitterbyyear)
```

```
## Warning: Removed 672861 rows containing missing values (geom_point).
```



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

TN:All of the states are sampled most extensively in recent years, after 2010.

TP: MN and ME is sampled most extensively in around 1975, other states are sampled most extensively in recent years.

Reflection

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

    1. Using different metrix to evaluate TSI can give different results.
    2. Jitter plot can avoid overlapping each other, which can show the data distribution in cluster.

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

    1.the proportion of trophic.class, trophic.class.secchi and trophic.class.tp. 2. Jitter plot

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

    Hands-on data analysis can let me explore the theory on my own, and it can help me understand it better from examples. The only problem is it may not cover the whole theory from one or two examples, so it'll be better to summary main theories after hands-on data analysis.

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

    the real-world data is related to a lot of things. For example, we get Total-N data mostly from recent years, that may have some trouble in studying changing trend of N concentration over periods.