Assignment 5: Water Quality in Lakes

Jack Carpenter

OVERVIEW

This exercise accompanies the lessons in Water Data Analytics 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 PDF file.
5. After Knitting, check your PDF against the key and then submit your assignment completion survey at https://forms.gle/fSe18vMhgzcjUKM39

Having trouble? See the assignment’s answer key if you need a hint. Please try to complete the assignment without the key as much as possible - this is where the learning happens!

Target due date: 2022-02-22

Setup

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

```
#1 session setup
getwd()
```

```
## [1] "/Users/Jack/Documents/Duke/Spring 2022/Water Data Analytics/Water_Data_Analytics_2022"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

```
library(lubridate)
```

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

```
## The following objects are masked from 'package:base':
##
```

```
##     date, intersect, setdiff, union
```

```
library(LAGOSNE)

theme_set(theme_classic(base_size = 10) +
          theme(axis.text = element_text(color = "black"),
                legend.position = "right"))
options(scipen = 10)

#2 load data
LAGOSdata <- lagosne_load()
```

```
## Warning in (function (version = NULL, fpath = NA) : LAGOSNE version unspecified,
## loading version: 1.087.3
```

```
LAGOStrophic <- read.csv("./Data/Processed/LAGOStrophic.csv", stringsAsFactors = TRUE)
```

Trophic State Index

3. 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 <- LAGOStrophic %>%
  mutate(trophic.class.secchi = case_when(TSI.secchi<40 ~ "Oligotrophic",
                                          TSI.secchi>=40 & TSI.secchi<50 ~ "Mesotrophic",
                                          TSI.secchi>=50 & TSI.secchi<70 ~ "Eutrophic",
                                          TSI.secchi>=70 ~ "Hypereutrophic"),
         trophic.class.tp = case_when(TSI.tp<40 ~ "Oligotrophic",
                                      TSI.tp>=40 & TSI.tp<50 ~ "Mesotrophic",
                                      TSI.tp>=50 & TSI.tp<70 ~ "Eutrophic",
                                      TSI.tp>=70 ~ "Hypereutrophic"))

class(LAGOStrophic$trophic.class.secchi)
```

```
## [1] "character"
```

```
class(LAGOStrophic$trophic.class.tp)
```

```
## [1] "character"
```

```
LAGOStrophic$trophic.class.secchi <- factor(LAGOStrophic$trophic.class.secchi,
                                            levels = c("Oligotrophic", "Mesotrophic",
                                                       "Eutrophic", "Hypereutrophic"))
LAGOStrophic$trophic.class.tp <- factor(LAGOStrophic$trophic.class.tp,
                                        levels = c("Oligotrophic", "Mesotrophic",
                                                   "Eutrophic", "Hypereutrophic"))
```

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

```
#unique(LAGOStrophic$trophic.class)
#unique(LAGOStrophic$trophic.class.secchi)
#unique(LAGOStrophic$trophic.class.tp)

LAGOStrophic %>%
  group_by(trophic.class) %>%
  summarise(count = n())
```

```
## # A tibble: 4 x 2
##   trophic.class  count
##   <fct>          <int>
## 1 Eutrophic      37457
## 2 Hypereutrophic 13234
## 3 Mesotrophic    13964
## 4 Oligotrophic    2762
```

```
LAGOStrophic %>%
  group_by(trophic.class.secchi) %>%
  summarise(count = n())
```

```
## # A tibble: 4 x 2
##   trophic.class.secchi count
##   <fct>                <int>
## 1 Oligotrophic         14559
## 2 Mesotrophic          22344
## 3 Eutrophic            25793
## 4 Hypereutrophic        4721
```

```
LAGOStrophic %>%
  group_by(trophic.class.tp) %>%
  summarise(count = n())
```

```
## # A tibble: 4 x 2
##   trophic.class.tp count
##   <fct>            <int>
## 1 Oligotrophic     17984
## 2 Mesotrophic      20607
## 3 Eutrophic        22419
## 4 Hypereutrophic    6407
```

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

```
#trophic class chl
LAGOStrophic %>%
  group_by(trophic.class) %>%
  summarise(count = n()) %>%
  mutate(prop = count/sum(count))
```

```
## # A tibble: 4 x 3
##   trophic.class  count   prop
##   <fct>          <int>  <dbl>
## 1 Eutrophic      37457 0.556
## 2 Hypereutrophic 13234 0.196
## 3 Mesotrophic    13964 0.207
## 4 Oligotrophic    2762 0.0410
```

```
LAGOStrophic %>%
  group_by(trophic.class.secchi) %>%
  summarise(count = n()) %>%
  mutate(prop = count/sum(count))
```

```
## # A tibble: 4 x 3
##   trophic.class.secchi count   prop
##   <fct>                <int>  <dbl>
## 1 Oligotrophic         14559 0.216
## 2 Mesotrophic          22344 0.331
## 3 Eutrophic            25793 0.383
## 4 Hypereutrophic        4721 0.0700
```

```
LAGOStrophic %>%
  group_by(trophic.class.tp) %>%
  summarise(count = n()) %>%
  mutate(prop = count/sum(count))
```

```
## # A tibble: 4 x 3
##   trophic.class.tp count   prop
##   <fct>            <int>  <dbl>
## 1 Oligotrophic     17984 0.267
## 2 Mesotrophic      20607 0.306
## 3 Eutrophic        22419 0.333
## 4 Hypereutrophic    6407 0.0950
```

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

> The chla method is most conservative estimate of eutrophic conditions. This is probably because it measures the results of extra nutrients in primary production (phytoplankton) and not the concentration of nutrients available like tp or the visible water quality like sechhi disks. Chla, depending on sensing method, may also be limited to the surface water and not truly reflect how much eutrophication is also happening at depth.

Nutrient Concentrations

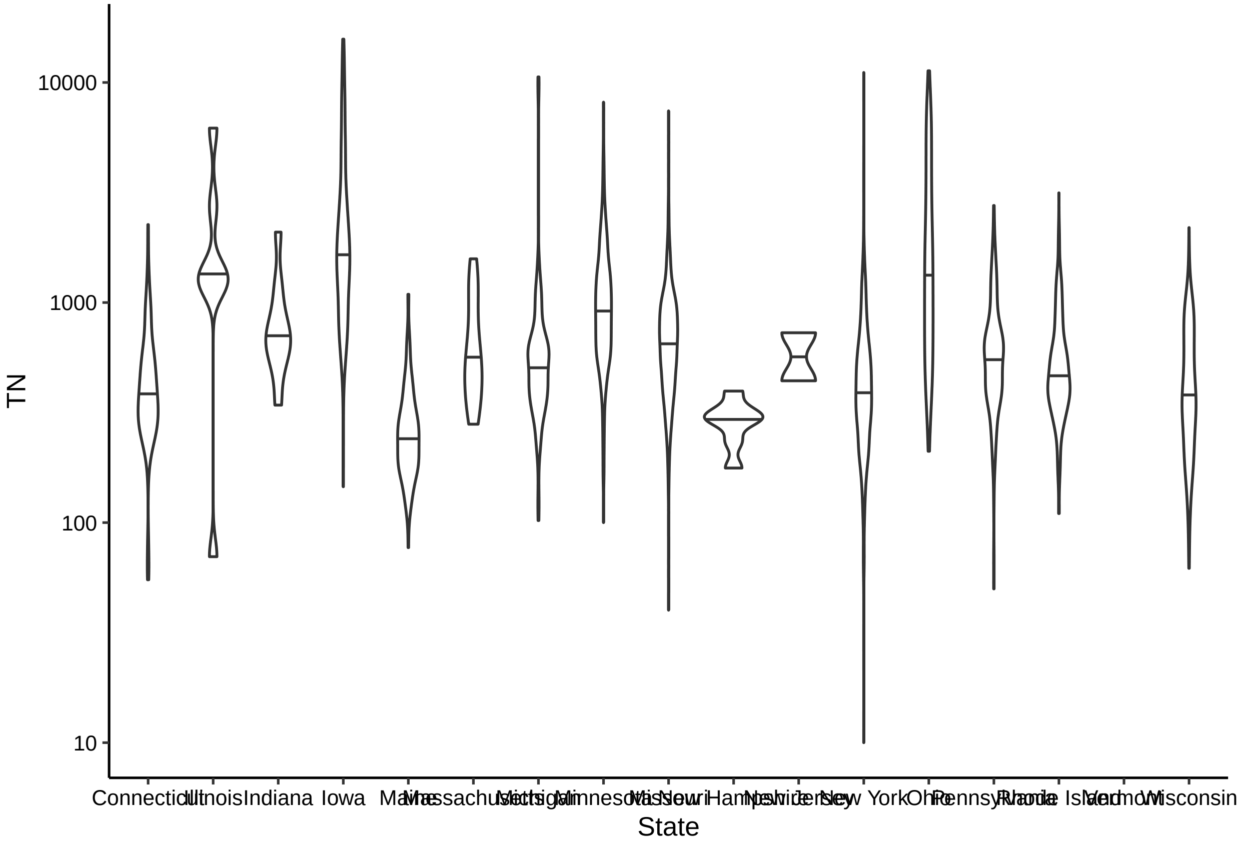
6. Create a data frame that includes the columns lagoslakeid, sampledate, tn, tp, state, and state_name. Mutate this data frame to include sampleyear and samplemonth columns as well. Filter the data frame for May-September. Call this data frame LAGOSNandP.

```
LAGOSlocus <- LAGOSdata$locus #spatial information
LAGOSstate <- LAGOSdata$state #state information
LAGOSnutrient <- LAGOSdata$epi_nutr #nutrient parameters

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

# join locus and state to put those together for locations
LAGOSlocals <- left_join(LAGOSstate, LAGOSlocus, by = "state_zoneid")
# now join the locations to the nutrients
LAGOSNandP <- left_join(LAGOSlocals, LAGOSnutrient, by = "lagoslakeid")

#now we select columns and mutate to create the ones we want
LAGOSNandP <- LAGOSNandP %>%
  select(lagoslakeid, sampledate, tn, tp, state, state_name) %>%
  mutate(sampleyear = year(sampledate),
         samplemonth = month(sampledate)) %>%
  filter(samplemonth == c(5,6,7,8,9))
```

7. Create two violin plots comparing TN and TP concentrations across states. Include a 50th percentile line inside the violins. Create a logged y axis and relabel axes.

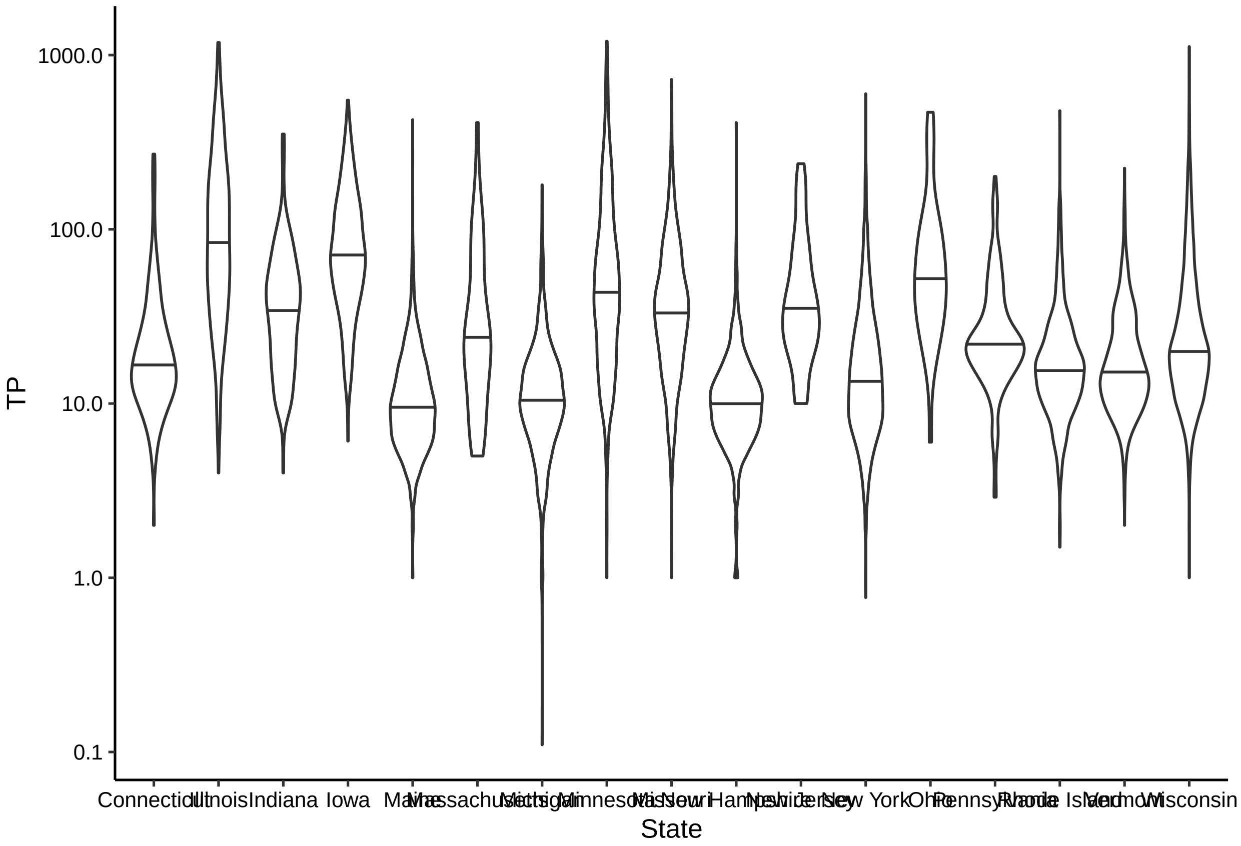
```
ggplot(LAGOSNandP, aes(x = state_name, y = tn)) +
  geom_violin(draw_quantiles = 0.50) +
  scale_y_log10() +
  labs(x = "State", y = "TN")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 138852 rows containing non-finite values (stat_ydensity).
```



```
ggplot(LAGOSNandP, aes(x = state_name, y = tp)) +
  geom_violin(draw_quantiles = 0.50) +
  scale_y_log10() +
  labs(x = "State", y = "TP")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 123250 rows containing non-finite values (stat_ydensity).

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



Which states have the highest and lowest median concentrations?

TN: highest is Iowa, lowest is Maine

TP: highest is Illinois, lowest is Maine again

Which states have the largest and smallest concentration ranges?

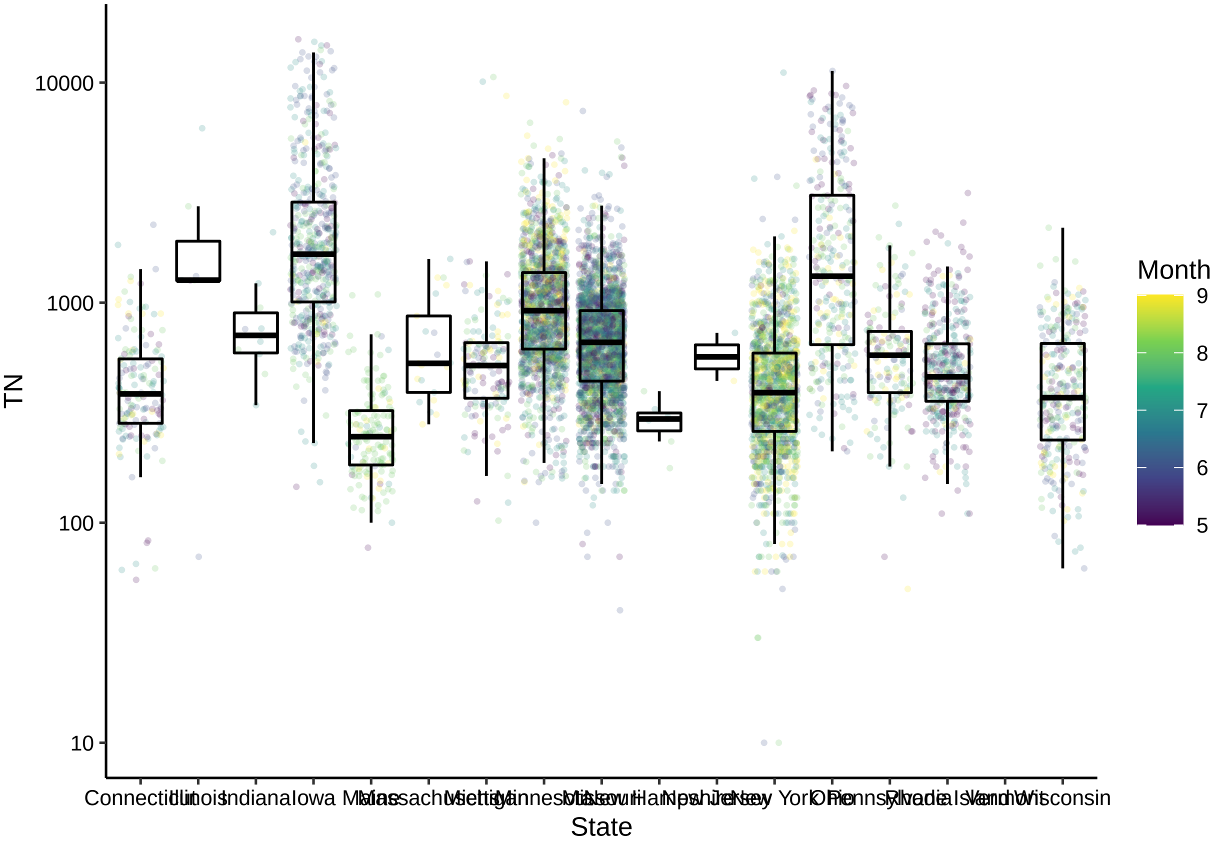
TN: largest range is New York, smallest is New Jersey

TP: largest is Michigan, smallest is New Jersey

8. 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.

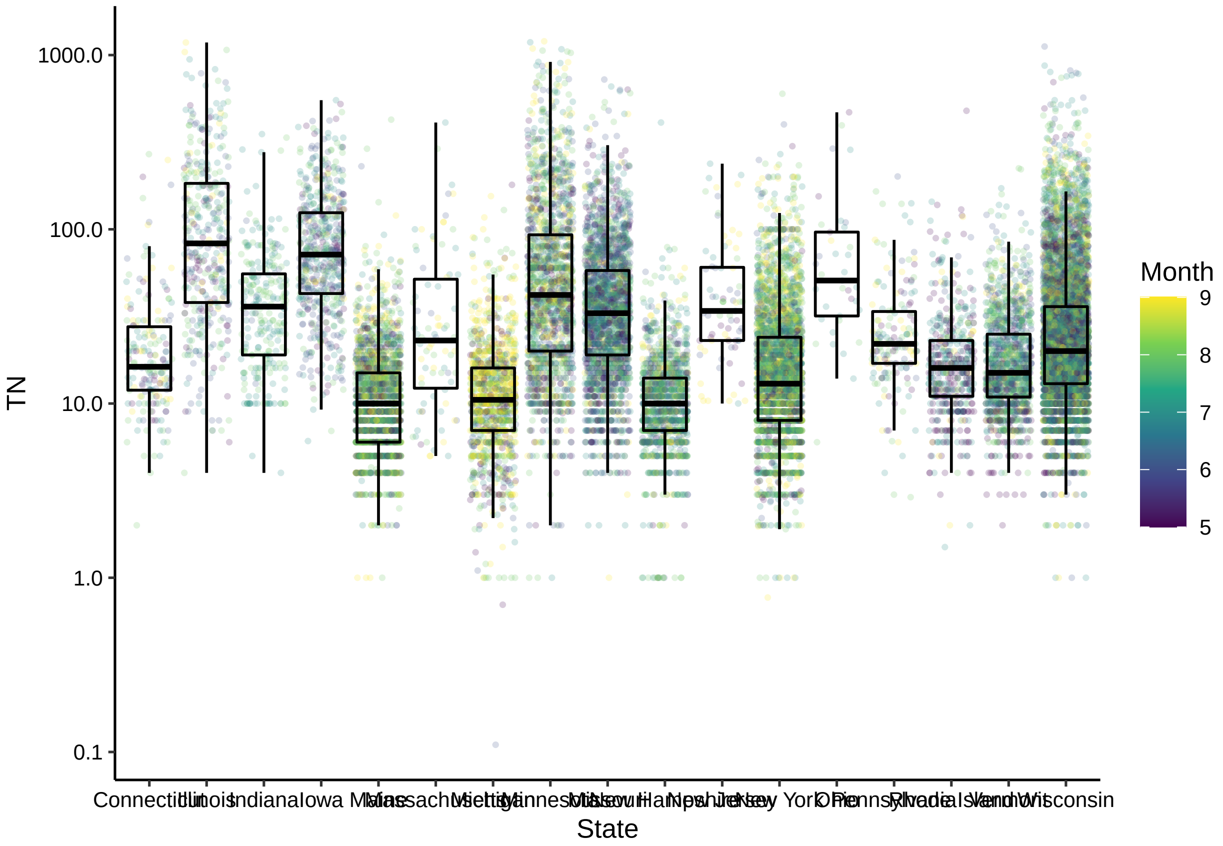
```
ggplot(LAGOSNandP, aes(x = state_name, y = tn, color = samplemonth)) +
  geom_jitter(alpha = 0.2, size = 0.5) +
  geom_boxplot(outlier.shape = NA, color = "black", fill = NA) +
  scale_y_log10() +
  scale_color_viridis_c() +
  labs(x = "State", y = "TN", color = "Month")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 138852 rows containing non-finite values (stat_boxplot).

## Warning: Removed 138852 rows containing missing values (geom_point).
```



```
ggplot(LAGOSNandP, aes(x = state_name, y = tp, color = samplemonth)) +
  geom_jitter(alpha = 0.2, size = 0.5) +
  geom_boxplot(outlier.shape = NA, color = "black", fill = NA) +
  scale_y_log10() +
  scale_color_viridis_c() +
  labs(x = "State", y = "TN", color = "Month")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 123250 rows containing non-finite values (stat_boxplot).

## Warning: Removed 123250 rows containing missing values (geom_point).
```



Which states have the most samples? How might this have impacted total ranges from #7?

> TN: Missouri, Minnesota and New York have the most samples, and likely is partly why they have larger ranges than other states. More samples means more chances to have very high and very low sample values.
>
> TP: Wisconsin, New York, and Missouri have the most samples here. Similar impact as with TN, where these states have larger ranges than most.