Assignment 8: Mapping

Theo Cai

OVERVIEW

This exercise accompanies the lessons in Hydrologic Data Analysis on mapping

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., "A08_Salk.html") prior to submission.

The completed exercise is due on 23 October 2019 at 9:00 am.

Setup

1. Verify your working directory is set to the R project file,
2. Load the tidyverse, lubridate, cowplot, LAGOSNE, sf, maps, and viridis packages.
3. Set your ggplot theme (can be theme_classic or something else)
4. Load the lagos database, the USA rivers water features shape file, and the HUC6 watershed shape file.

```
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(cowplot)
```

```
##
## ********************************************************

## Note: As of version 1.0.0, cowplot does not change the
```

```
##   default ggplot2 theme anymore. To recover the previous
##   behavior, execute:
##   theme_set(theme_cowplot())
## *******************************************************
##
## Attaching package: 'cowplot'
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```r
install.packages("LAGOSNE", repos = "https://cran.rstudio.com/bin/windows/contrib/3.6/LAGOSNE_2.0.1.zip
```

```
## Installing package into 'C:/Users/gyc4/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
## Warning: unable to access index for repository https://cran.rstudio.com/bin/windows/contrib/3.6/LAGOS
##   cannot open URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/LAGOSNE_2.0.1.zip/src/contrib/PA
## Warning: package 'LAGOSNE' is not available (for R version 3.6.1)
## Warning: unable to access index for repository https://cran.rstudio.com/bin/windows/contrib/3.6/LAGOS
##   cannot open URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/LAGOSNE_2.0.1.zip/bin/windows/co
```

```r
library(LAGOSNE)
library(sf)
```

```
## Linking to GEOS 3.6.1, GDAL 2.2.3, PROJ 4.9.3
```

```r
library(maps)
```

```
##
## Attaching package: 'maps'
## The following object is masked from 'package:purrr':
##
##     map
```

```r
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
theme_set(theme_classic())

#lagosne_get(dest_folder = LAGOSNE:::lagos_path(), overwrite = TRUE)
LAGOSdata <- lagosne_load()
```

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

```r
waterfeatures <- st_read("./Data/Raw/hydrogl020.dbf")
```

```
## Reading layer `hydrogl020' from data source `Z:\Hydrologic_Data_Analysis\Data\Raw\hydrogl020.dbf' us
## Simple feature collection with 76975 features and 12 fields
## geometry type:  LINESTRING
## dimension:      XY
## bbox:           xmin: -179.9982 ymin: 17.67469 xmax: 179.9831 ymax: 71.39819
## epsg (SRID):    NA
## proj4string:    NA
```

```r
HUC6 <- st_read("./Data/Raw/Watersheds_Spatial/WBDHU6.dbf")
```

```
## Reading layer `WBDHU6' from data source `Z:\Hydrologic_Data_Analysis\Data\Raw\Watersheds_Spatial\WBDI
## Simple feature collection with 33 features and 15 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -90.6235 ymin: 24.39533 xmax: -75.3981 ymax: 37.52103
## epsg (SRID):    4269
## proj4string:    +proj=longlat +datum=NAD83 +no_defs
```

Mapping water quality in lakes

Complete the in-class exercise from lesson 15, to map average secchi depth measurements across states in Maine, considering lake area and lake depth as predictors for water clarity. Steps here are identical to the lesson, with the following edits:

- Make sure all your wrangling is done in this document (this includes basic wrangling of the LAGOS database)
- In your cowplot, do not adjust the legend items (even though they look ugly). Rather, reflect on how you would improve them with additional coding.
- For item 9, **do** run a regression on secchi depth by lake area and a separate regression on secchi depth by lake depth. Make scatterplots of these relationships. Note that log-transforming one of these items may be necessary.

5. Filter the states and secchi depth datasets so that they contain Maine only. For the secchi depth dataset, create a summary dataset with just the mean secchi depth.

```r
LAGOSlocus <- LAGOSdata$locus
LAGOSstate <- LAGOSdata$state
LAGOSnutrient <- LAGOSdata$epi_nutr
LAGOSlimno <- LAGOSdata$lakes_limno

LAGOScombined <-
  left_join(LAGOSnutrient, LAGOSlocus) %>%
  left_join(., LAGOSlimno) %>%
  left_join(., LAGOSstate) %>%
  filter(!is.na(state)) %>%
  select(lagoslakeid, sampledate, secchi, lake_area_ha, maxdepth, nhd_lat, nhd_long, state)
```

```
## Joining, by = "lagoslakeid"

## Joining, by = c("lagoslakeid", "nhdid", "nhd_lat", "nhd_long")

## Joining, by = "state_zoneid"
```

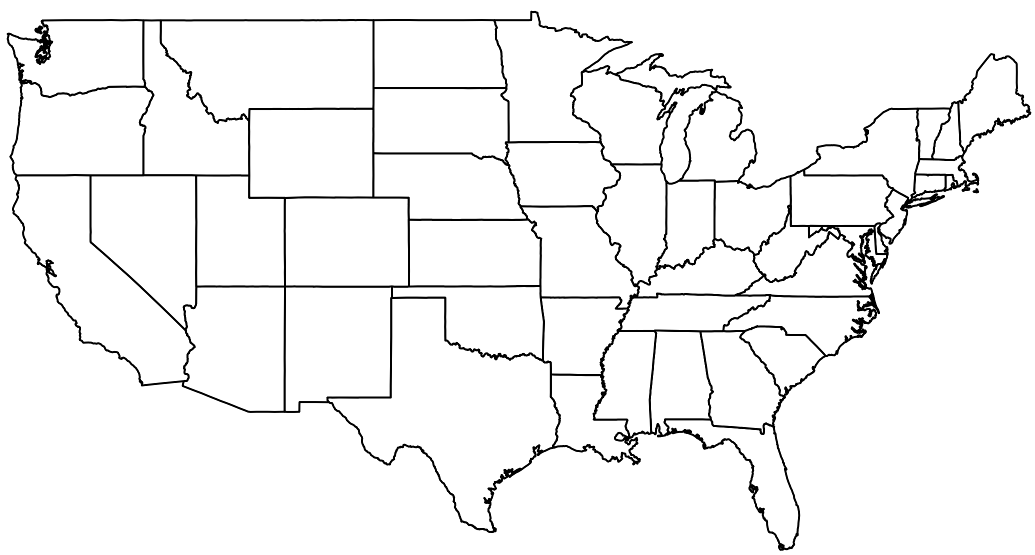
```r
LAGOSMaine <- LAGOScombined %>%
  filter(state == "ME")

secchiMaine.summary <- LAGOSMaine %>%
  group_by(lagoslakeid) %>%
  summarise(secchi.mean = mean(secchi), #summarise by mean to collapse all the multiple measurements
            area = mean(lake_area_ha),
            depth = mean(maxdepth),
            lat = mean(nhd_lat),
            long = mean(nhd_long)) %>%
  drop_na()
```

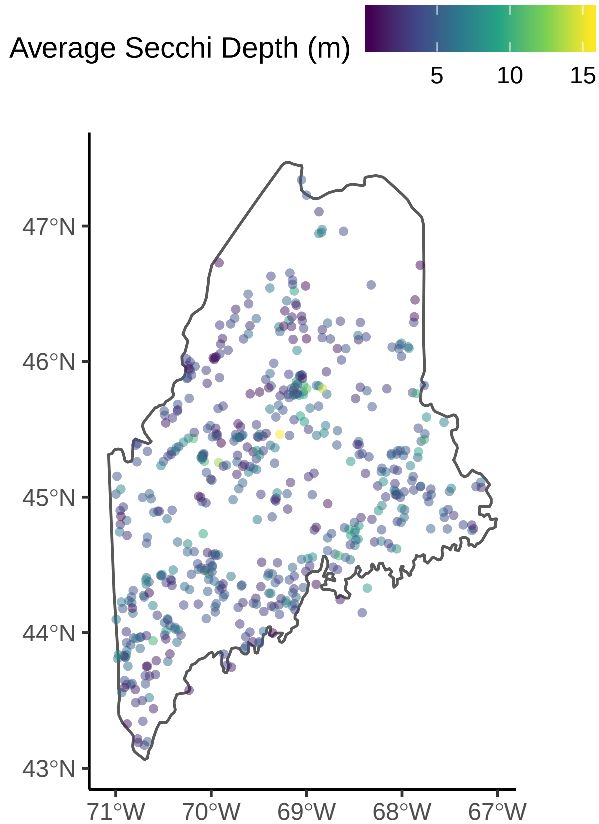
```r
secchiMaine.spatial <- st_as_sf(secchiMaine.summary, coords = c("long", "lat"), crs = 4326)
```

6. Create a plot of mean secchi depth for lakes in Maine, with mean secchi depth designated as color and the lake area as the size of the dot. Remember that you are using size in the aesthetics and should remove the size = 1 from the other part of the code. Adjust the transparency of points as needed.

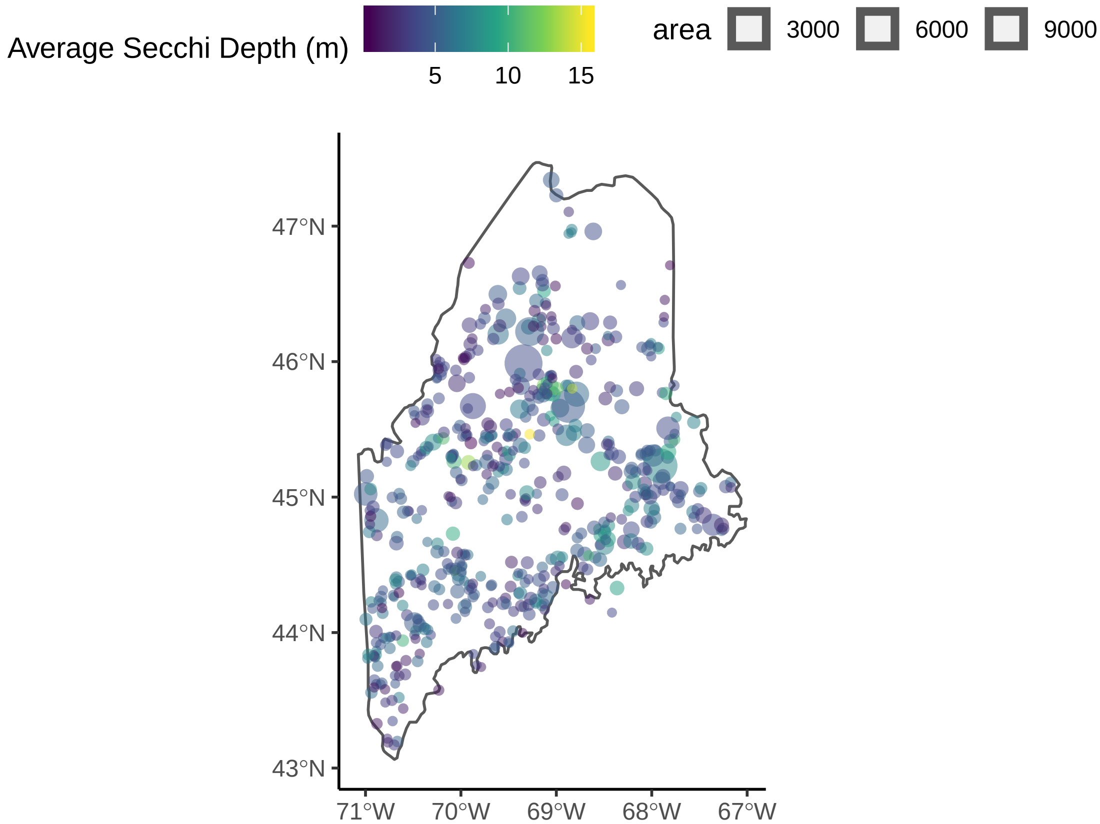
```r
states <- st_as_sf(map(database = "state", plot = TRUE, fill = TRUE, col = "white"))
```



```r
Maine.subset <- filter(states, ID %in%
                         "maine")
Secchiplot <- ggplot() +
  geom_sf(data = Maine.subset, fill = "white") +
  geom_sf(data = secchiMaine.spatial, aes(color = secchi.mean), #plot states first, then secchi depths
          alpha = 0.5, size = 1) +
  scale_color_viridis_c() +
  labs(color = "Average Secchi Depth (m)") +
  theme(legend.position = "top")
print(Secchiplot)
```

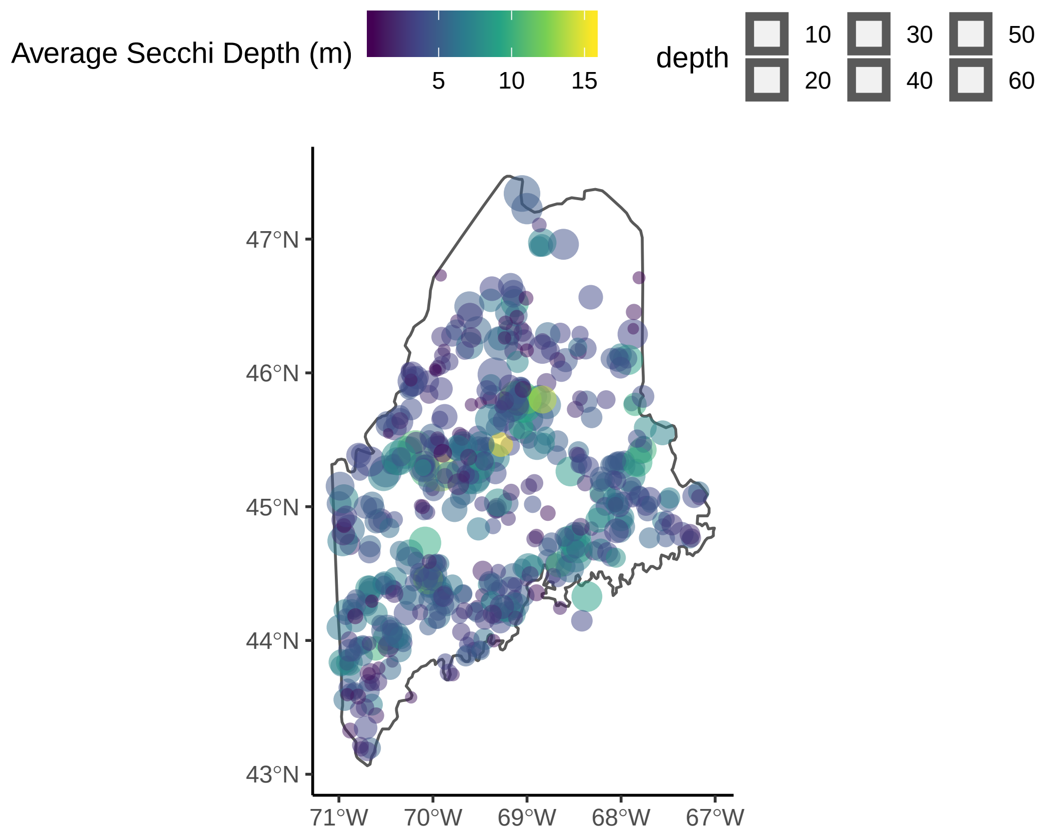



```r
MeanSecchiplot <- ggplot() +
  geom_sf(data = Maine.subset, fill = "white") +
  geom_sf(data = secchiMaine.spatial, aes(color = secchi.mean, size = area),
          alpha = 0.5) +
  scale_color_viridis_c() +
  labs(color = "Average Secchi Depth (m)") +
  theme(legend.position = "top")
print(MeanSecchiplot)
```



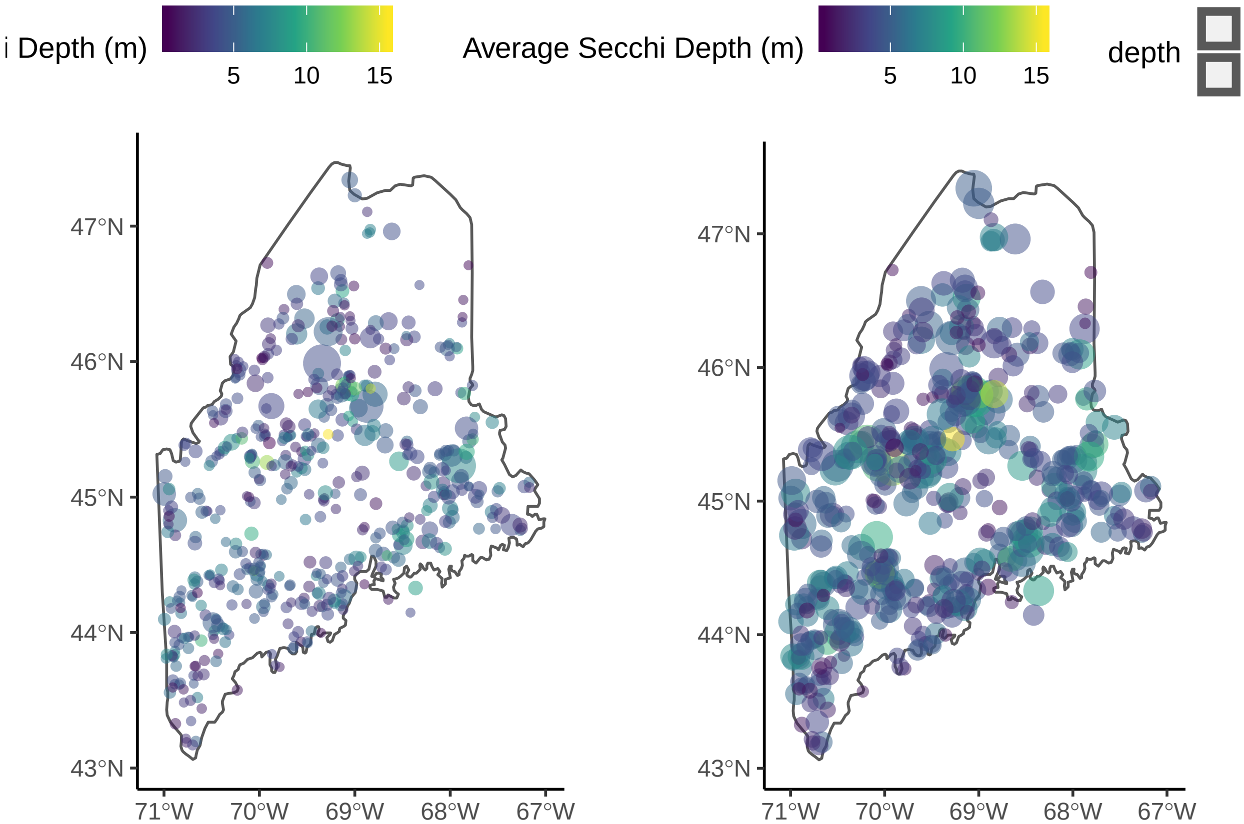
7. Create a second plot, but this time use maximum depth of the lake as the size of the dot.

```
DepthSecchiplot <- ggplot() +
  geom_sf(data = Maine.subset, fill = "white") +
  geom_sf(data = secchiMaine.spatial, aes(color = secchi.mean, size = depth),
          alpha = 0.5) +
  scale_color_viridis_c() +
  labs(color = "Average Secchi Depth (m)") +
  theme(legend.position = "top")
print(DepthSecchiplot)
```



8. Plot these maps in the same plot with the `plot_grid` function. Don't worry about adjusting the legends (if you have extra time this would be a good bonus task).

```
Secchi.combined <-
  plot_grid(MeanSecchiplot, DepthSecchiplot,
            ncol = 2)
print(Secchi.combined)
```



What would you change about the legend to make it a more effective visualization?

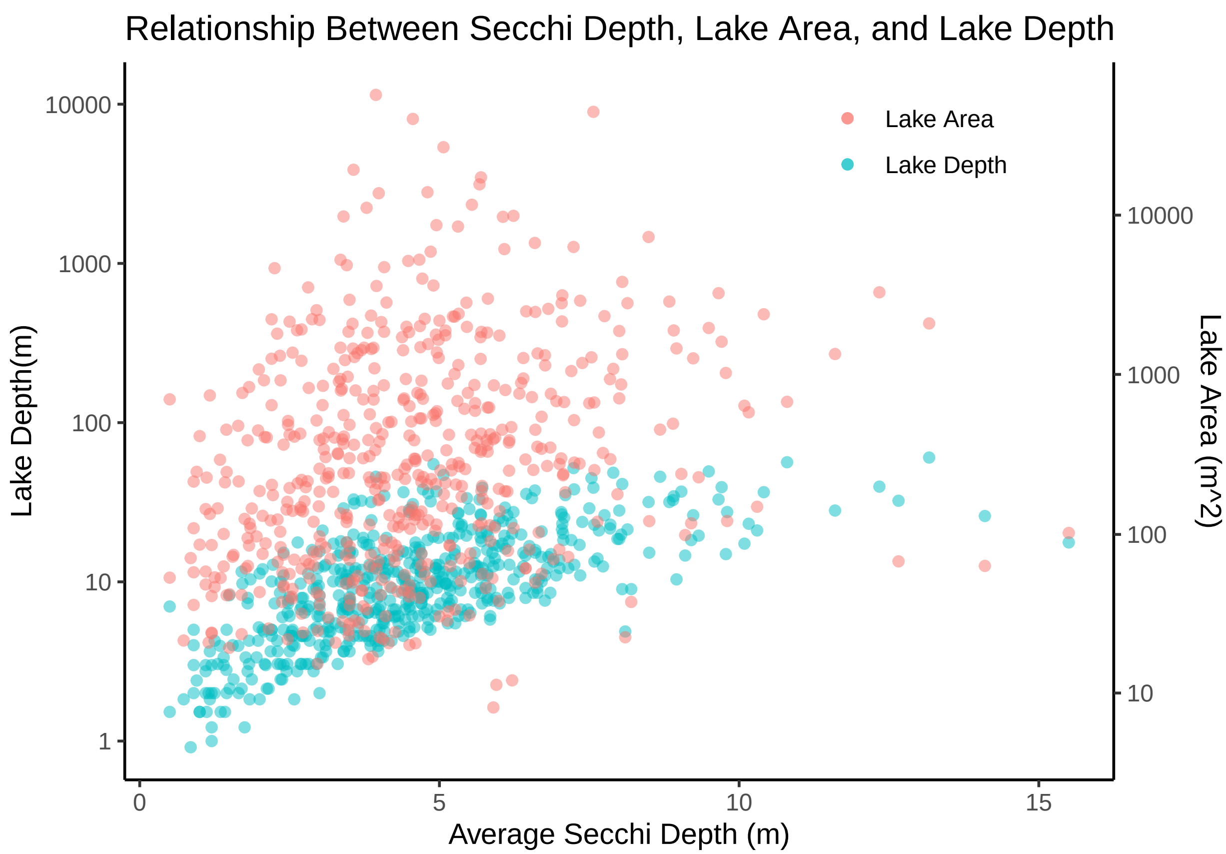
> I think I might at least change the area part of the legend so, instead of boxes, we can see the range of dot sizes and their meaning area-wise. Just something that mirrors the dots we see in the actual graph.

9. What relationships do you see between secchi depth, lake area, and lake depth? Which of the two lake variables seems to be a stronger determinant of secchi depth? (make a scatterplot and run a regression to test this)

Note: consider log-transforming a predictor variable if appropriate

```
secchiMaine <- secchiMaine.spatial %>%
  filter(area, depth, secchi.mean)


Secchi.scatter <- ggplot() +
  geom_point(data = secchiMaine, aes(x = secchi.mean, y = depth, color = "Lake Depth"), alpha = 0.5) +
  geom_point(data = secchiMaine,
             aes(x = secchi.mean, y = area, color = "Lake Area"), alpha = 0.5) +
  scale_y_log10(sec.axis = sec_axis(~.*5, name = "Lake Area (m^2)")) +
  labs(x = "Average Secchi Depth (m)", y = "Lake Depth(m)") +
  ggtitle("Relationship Between Secchi Depth, Lake Area, and Lake Depth")  +
  theme(legend.position = c(0.8, 0.9), legend.title = element_blank())
print(Secchi.scatter)
```




```r
Secchi.regression <- lm(data = secchiMaine, secchi.mean ~ depth + area)
summary(Secchi.regression)
```

```
##
## Call:
## lm(formula = secchi.mean ~ depth + area, data = secchiMaine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8481 -1.0782 -0.0957  0.9486 10.0549
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.781e+00  1.143e-01  24.328  < 2e-16 ***
## depth        1.512e-01  7.720e-03  19.583  < 2e-16 ***
## area        -4.518e-04  9.173e-05  -4.926 1.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.67 on 546 degrees of freedom
## Multiple R-squared:  0.4169, Adjusted R-squared:  0.4147
## F-statistic: 195.1 on 2 and 546 DF,  p-value: < 2.2e-16
```

It seems like Lake Depth is a better predictor of secchi depth, though Lake Area is also statistically significant. Visually, there seems to be a difference - when graphed, Lake Depth and Average Secchi Depth seem to form a more linear relationship with less noise than Lake Area and Average Secchi Depth.

Mapping water features and watershed boundaries

10. Wrangle the USA rivers and HUC6 watershed boundaries dataset so that they include only the features present in Florida (FL). Adjust the coordinate reference systems if necessary to ensure they use the same projection.

```
waterfeatures <- filter(waterfeatures, STATE == "FL")

summary(HUC6$States)
```

```
##       AL    AL,FL AL,FL,GA AL,GA,TN AL,LA,MS    AL,MS       FL    FL,GA
##        1        3        1        1        1        2        6        4
##       GA GA,NC,SC    LA,MS       NC    NC,SC NC,SC,VA    NC,VA       SC
##        2        1        1        4        2        1        2        1
```

```
HUC6.FL <- HUC6 %>%
  filter(States %in% c("Al,FL", "AL,FL,GA", "FL", "FL,GA"))

st_crs(waterfeatures)
```

```
## Coordinate Reference System: NA
```

```
st_crs(HUC6.FL)
```

```
## Coordinate Reference System:
##   EPSG: 4269
##   proj4string: "+proj=longlat +datum=NAD83 +no_defs"
```

11. Create a map of watershed boundaries in Florida, with the layer of water features on top. Color the watersheds gray (make sure the lines separating watersheds are still visible) and color the water features by type.

```
waterfeatures <- st_set_crs(waterfeatures, 4269)
st_crs(waterfeatures)
```

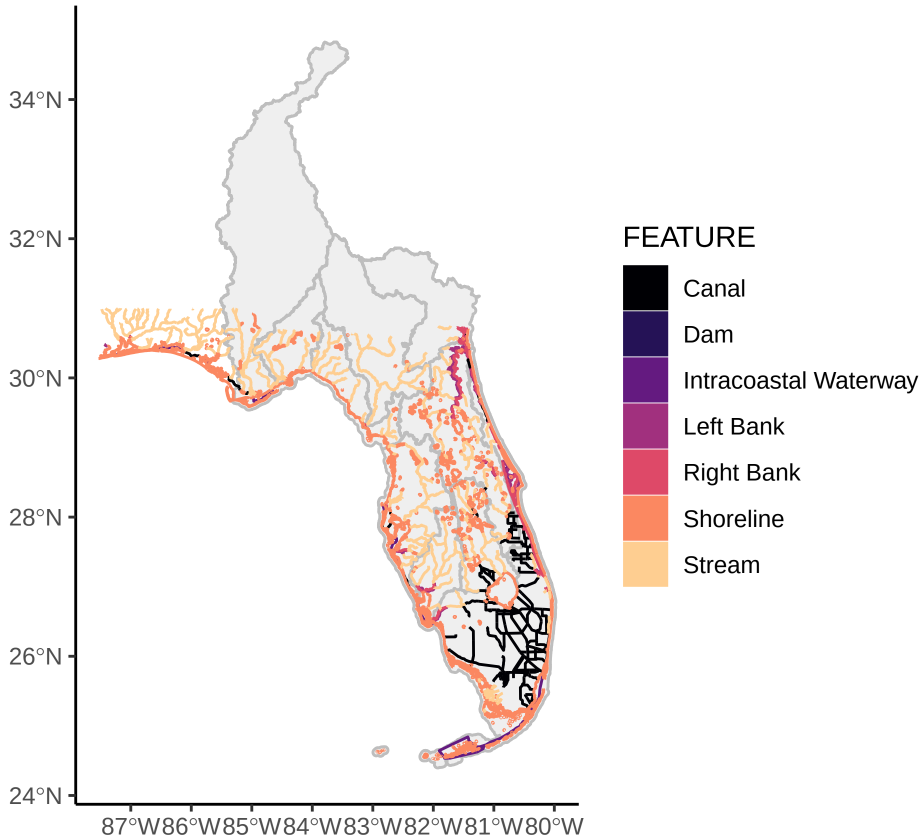
```
## Coordinate Reference System:
##   EPSG: 4269
##   proj4string: "+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs"
```

```
waterfeatures <- waterfeatures %>% st_set_crs(st_crs(HUC6.FL))
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform
## for that
```

```
waterfeatures <- filter(waterfeatures, FEATURE != "Apparent Limit" & FEATURE != "Closure Line")

FLlayers <- ggplot() +
  geom_sf(data = HUC6.FL, color = "gray", alpha = .6) +
  geom_sf(data = waterfeatures, aes(fill = FEATURE, color = FEATURE)) +
  scale_color_viridis_d(option = "magma", end = 0.9) +
  scale_fill_viridis_d(option = "magma", end = 0.9)
print(FLlayers)
```



12. What are the dominant water features in Florida? How does this distribution differ (or not) compared to North Carolina?

    The dominant water features in Florida seem to be streams and coastline, which is the same for North Carolina. In comparison, however, Florida has more canals and seemingly natural lakes than North Carolina.

Reflection

13. What are 2-3 conclusions or summary points about mapping you learned through your analysis?

    I learned that mapping can help visualize multiple variables at once (secchi.combined plot), like geography, average secchi depth, lake depth, and lake area, but it's still sometimes better to do a normal scatterplot to analyze the relationship they have to one another.(secchi.scatter) I also learned how helpful it is to just have a bird's eye view of an entire state or several states, especially when trying to conceptualize geographical features and less intuitive concepts like watersheds. (FLlayers)

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

    secchi.combined, Secchi.scatter, and FLlayers

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

    Yes, it did. I knew what a watershed was conceptually, but it wasn't until I plotted the water features and watersheds map that I understood visualizing how all the streams are interconnected and draining to the same place.

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

It compared pretty well!