Water Quality Data Report Stony Creek

Data Exploration with Water Quality Data 2020-2021 Hudson River

Loading in the Data

The first step in the exploration was to load in the datasets from 2020 and 2021, though I began with just working with 2020. Because I began with only 2020, datasets were not combined until later. The datasets originate from the National Estuarine Research Reserve System and the datasets contain water quality measurements for Stony Creek station of the Hudson River. The files were loaded in using read_csv() from tidyverse. The tidyverse package was used often in order to provide a consistent framework for data wrangling.

Separating Date and Time Variables

After successfully loading in the datasets, the DateTimeStamp variable was separated into two new components: date and time. Using the separate() function, I could then independently look at date or time, even though my main focus was date. The date_clean was created thereafter in order to ensure standardization in the variable. This also made sure that visualizations, using date like time series plots would properly interpret the date variable.

Month and Season Variables

After dates were clean and standardized, month was extracted from the new date variable using mutate() and month(). This categorization by month allows for an examination of monthly and later seasonal trends. After the monthly variable was created, I created a seasonal variable using case_when() to organize months into seasonal groups of Spring, Summer, Fall, and Winter. Although Winter was included in the data, its observations are infrequent due to collection gaps. Because of this, the season was later excluded for clarity.

Missing Values

Missing values were an especially prominent issue in this exploration. Counts of missing values by season and by month using group_by() and summarise(). Based on the outputs, Winter certainly had a consistently large number of missing values for both years. Monthly observations further highlighted gaps in missing values. Interestingly, for 2021, there is missing data around July/August in the Summer. I was unable to account for why this data was missing, however, this missing data is important to note for further analysis in 2021, especially in the Summer. This is because the gap in data may impact interpretation of seasonal trends, especially for variables which fluctuate in Summer in particular. In other words, this could skew the representation of variables during this period in 2021.

Summary Statistics

To begin exploring the variables after removing Winter, I created summary statistics for both years. These tables provide general information on the spread of measurements for the variables. The tables help provide basic overview of the central tendencies before visualizations for the summary statistics. After creating tables for overall summary statistics, I produced the same information except aggregated by season to look at seasonal spreads. In order to better show this information, I made box plot visualizations for both years. In 2020, variables including Depth, DO_mgl, DO_Pct, pH, SpCond, and Temp are quite fairly distributed. In contrast, the salinity variable is much less evenly distributed. Additionally, ChlFluor and Turbidity are even less evenly distributed and includes many high outliers. Also, the plot highlights the lack of information for the Level variable. Due to this observation, the Level variable was ignored for the rest of the exploration. These trends are similar for 2021, however, in 2021 the box plots reveal far more outliers. These boxplots were then organized seasonally by year in order to examine similar patterns on a more granular level.

```
[1] "/Users/josephinewhite/Documents/GitHub/git-one/Environmental_Data"
```

```
The downloaded binary packages are in
    /var/folders/5x/yc7bfwrx5cg9vgr47hqkkq_00000gn/T//RtmpXLXjue/downloaded_packages
```

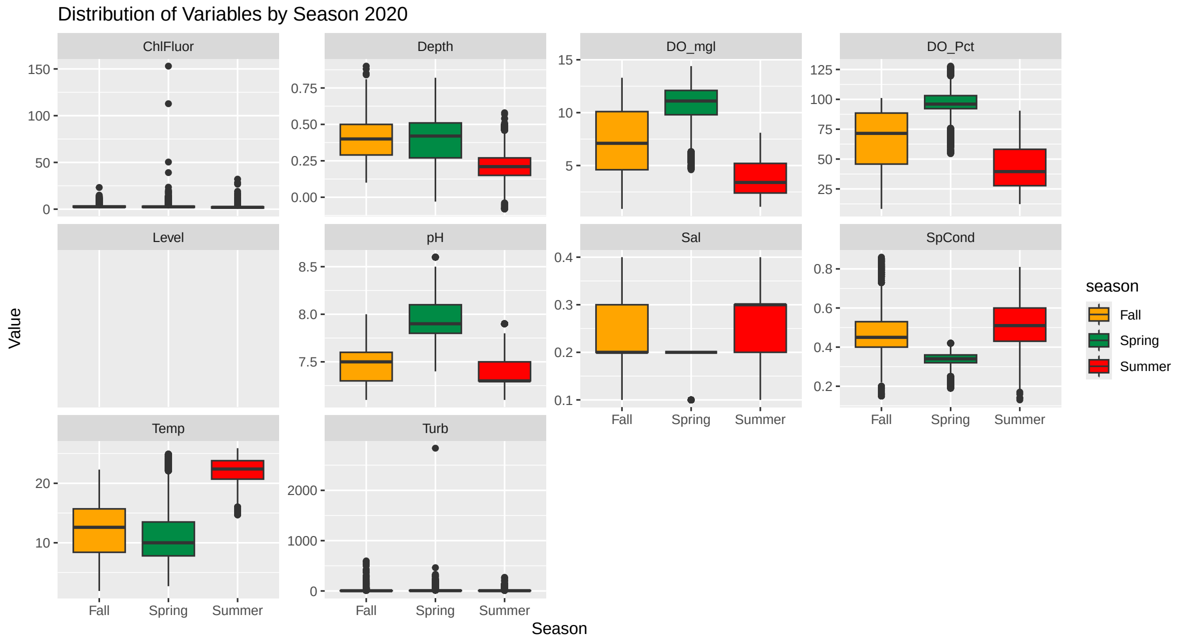
```
here() starts at /Users/josephinewhite/Documents/GitHub/git-one/Environmental_Data
```

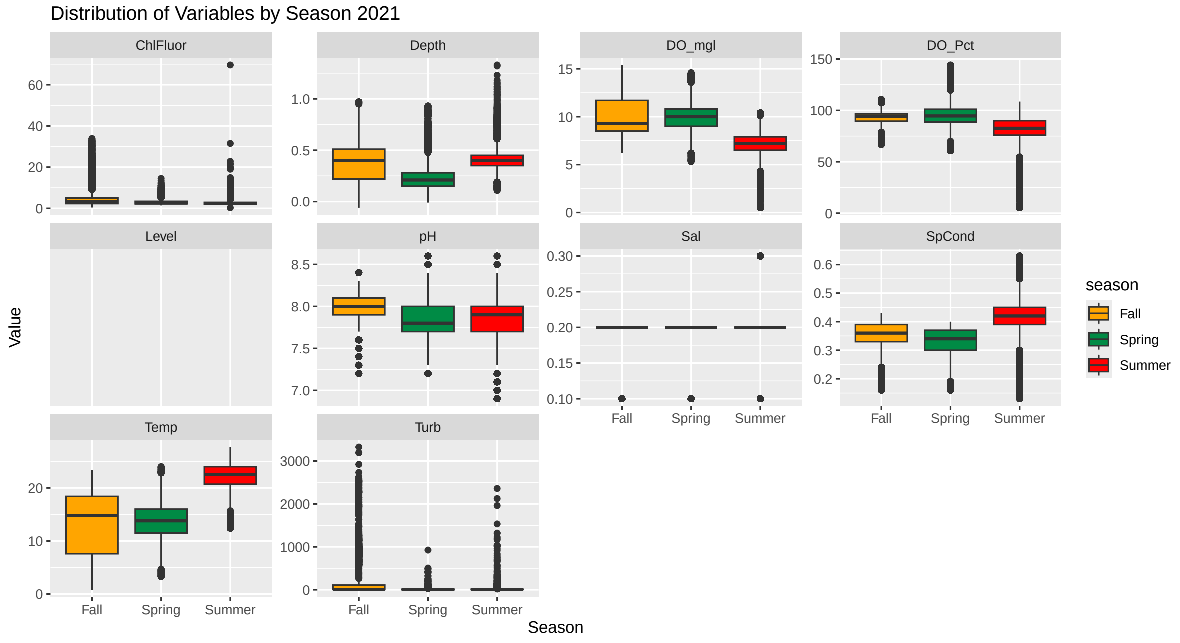
```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.1     v stringr   1.5.2
```

```
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
New names:
Rows: 35136 Columns: 31
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (14): StationCode, isSWMP, DateTimeStamp, F_Temp, F_SpCond, F_Sal, F_DO_...
dbl (12): Historical, ProvisionalPlus, Temp, SpCond, Sal, DO_Pct, DO_mgl, De...
lgl  (5): F_Record, Level, cLevel, F_cLevel, ...31

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
New names:
Rows: 35040 Columns: 31
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (15): StationCode, isSWMP, DateTimeStamp, F_Record, F_Temp, F_SpCond, F_...
dbl (12): Historical, ProvisionalPlus, Temp, SpCond, Sal, DO_Pct, DO_mgl, De...
lgl  (4): Level, cLevel, F_cLevel, ...31

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```





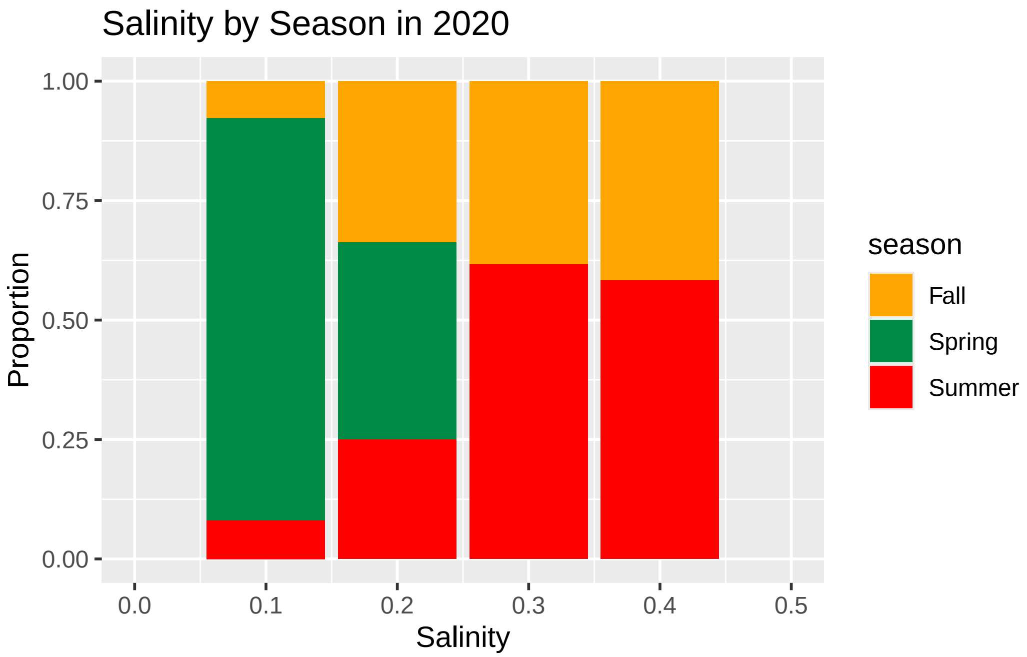
Temperature Visualization- Initial Visual Exploration

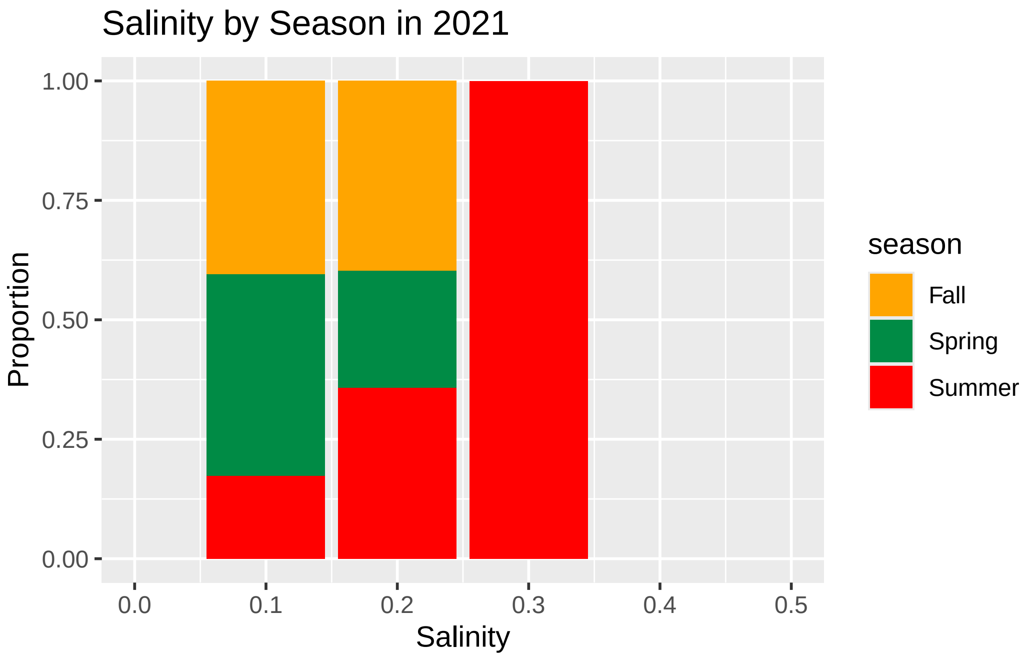
Using geom_point(), I firstly explored the temperature variable to look at general temperature trends. These plots showed predictable seasonal variation as temperatures peaked in Summer months and dropped in Fall. I later used geom_line() and colored temperature measurements by season to better visualize the seasonal changes while also acknowledging the limitations due to the nature of the Winter variable. The plot in 2021 visualized the missing data in the middle of the Summer. Importantly, both the 2020 and 2021 plots show that there was some unidentified data which is represented at the end of the plots. This is likely extra data from winter months that happened to be collected in the range of winter- for both years, there were a few data points that were collected for winter. However, overall, measurements in winter were very limited in comparison to other seasons, and therefore ignored because this would not create fair comparisons.

Next, I produced seasonal temperature plots, faceted by month in order to display trends across months. Based on the plots. July and August seemed to experience the most consistent (high) temperatures, while other months fluctuated in 2020. In 2021 there was a bit less consistency for these months, but similar patterns overall. In the 2021 plot is is also shown that March generally had missing values as well. This likely reflects inconsistency due to wintry weather conditions.

Visualizing Salinity and pH

Other key variables like Salinity (Sal) and pH were examined using bar plots, density plots, and boxplots. Salinity was visualized using the counts with geom_bar() and also proportionality using position = "fill" in order to examine both the absolute and relative distributions by season. Monthly visualizations showed interesting differences in salinity levels, particularly the spread of them. For example, in November, 2020 salinity levels concentrate almost entirely at 0.2. In contrast, in September, levels were much more spread between 0.1 and 0.4. The boxplots allowed for a comparison of central tendencies and the presence of outliers. On the other hand, density plots showed subtle nuances in distributional shapes. For pH, density plots were faceted seasonally to showcase the specific differences by period in the year and boxplots explicitly showed ranges and medians. Since Salinity is not a raw variable, but rather, a derived variable, (from conductivity, in part), it was later mutated into bins for further analysis. Because of the nature of the variable, in the later half of my study, I focused on conductivity as a replacement for salinity.





Daily Average Calculations and Variable Time Series

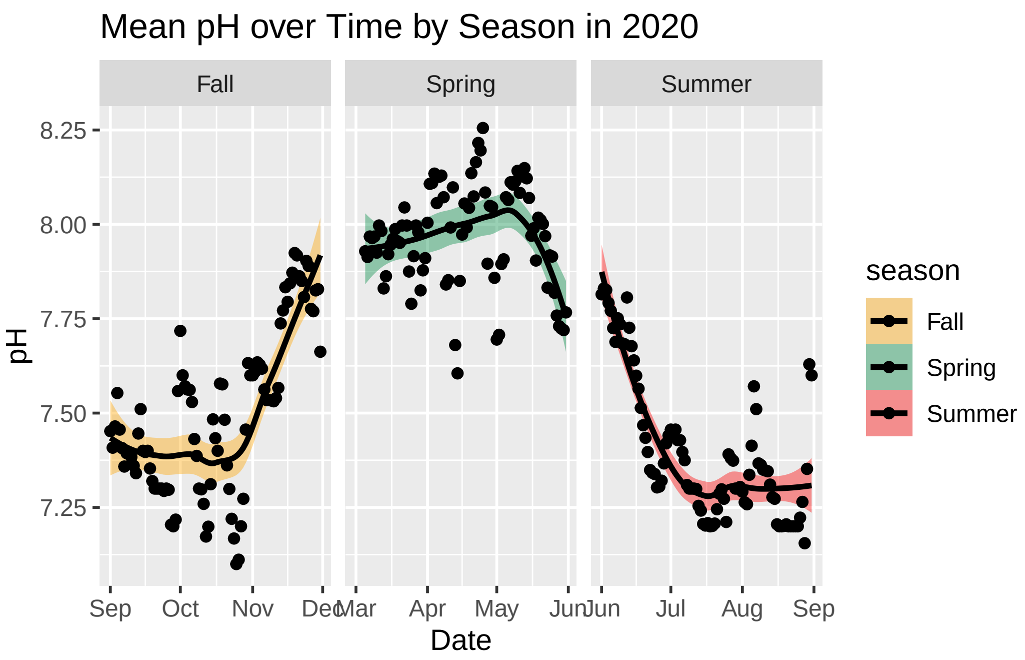
Daily averages for each variable were calculated by season using group_by and summarize().

This reduced noise from individual measurements and allowed a view of overall trends while also considering seasonal variation. These daily averages provided a solid foundation for subsequent visualizations including time series plots for each variable for both years. Plots were also colored by season. Interestingly, the seasonal variable time series plots show significant discrepancies when comparing 2020 to 2021, especially daily mean pH plots. Also when I later looked at pH more closely, faceted by season and smoothed using geom_smooth(method = "loess"), the differences between 2020 and 2021 became even more apparent.

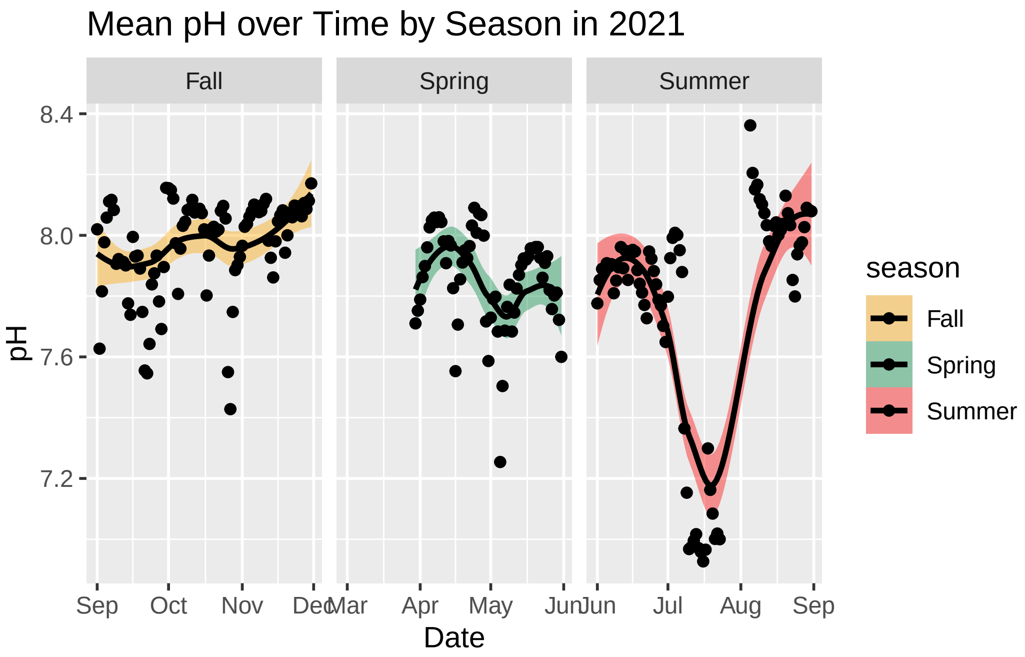
Some plots do follow similar patterns though, such as the temperature plots. Also, the dissolved oxygen plots (Pct and Mgl) vary significantly, even though these are similar variables. These differences likely reflect other environmental influences like temperature or salinity since warmer or saltier water may hold less oxygen. The percent dissolved oxygen measure accounts for such conditions, while the mg/L measure only shows the absolute amount of oxygen. The salinity plots were not as useful for analysis considering the nature of the variable and its narrow range. Also, the plots for turbidity reveal interesting spikes in the variable during different times for 2020 and 2021. Perhaps the turbidity variable ought to be examined more closely as it may reflect short-term weather disruptions.

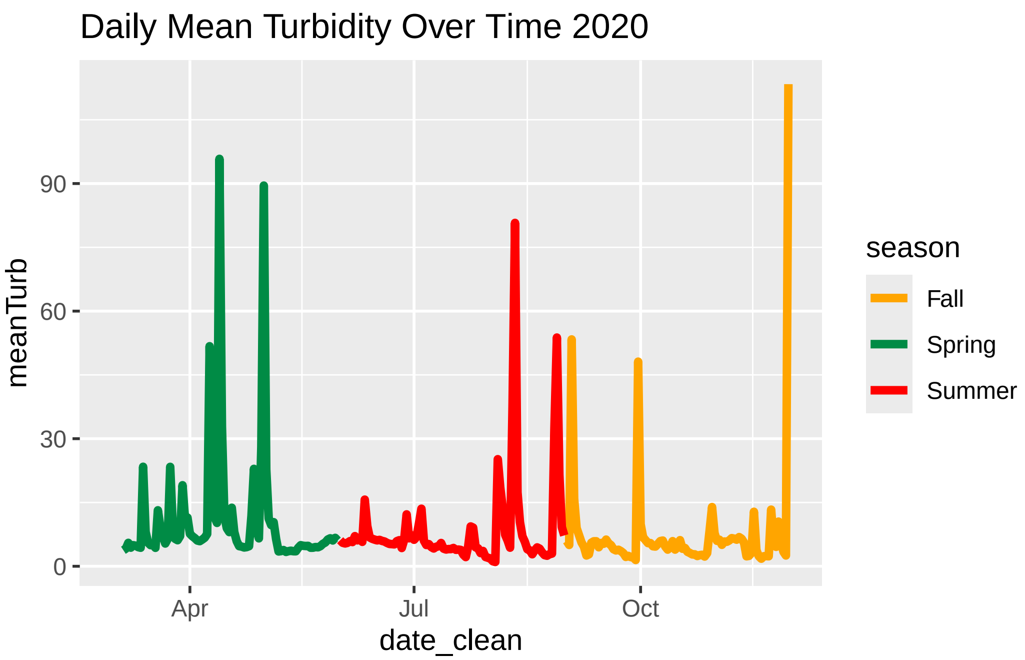
```
`summarise()` has grouped output by 'date_clean'. You can override using the
`.groups` argument.
`summarise()` has grouped output by 'date_clean'. You can override using the
`.groups` argument.
```

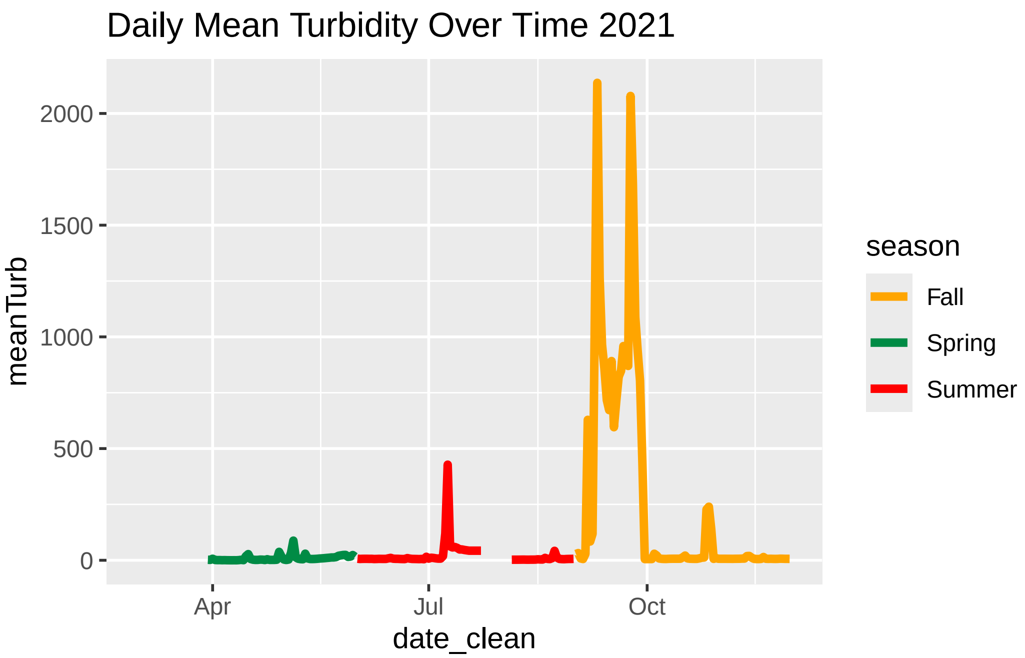
```
`geom_smooth()` using formula = 'y ~ x'
```



```
`geom_smooth()` using formula = 'y ~ x'
```





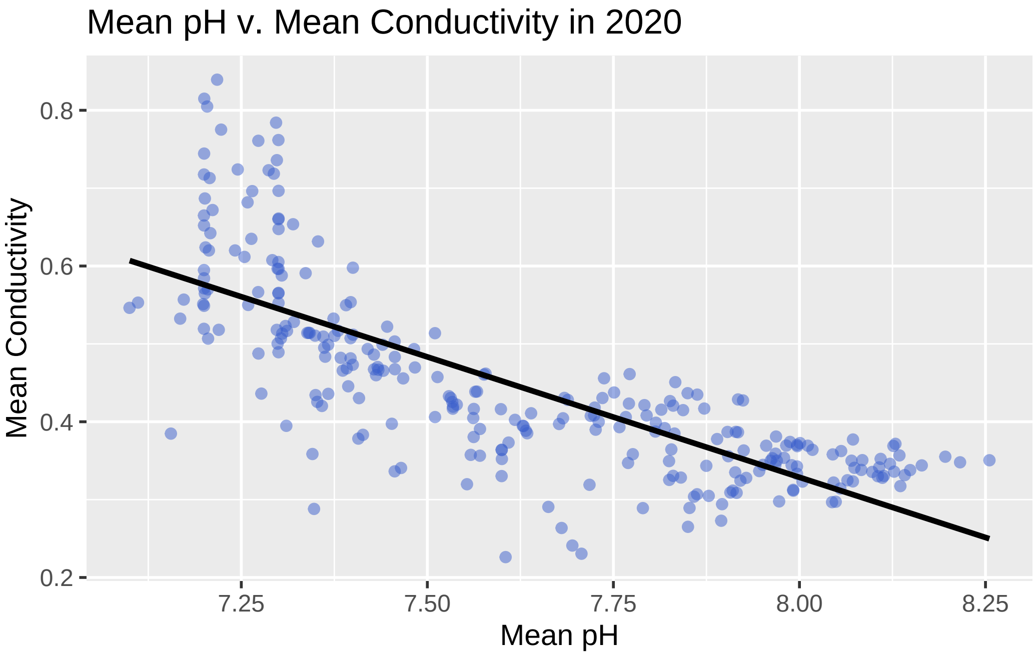


Correlation Plots

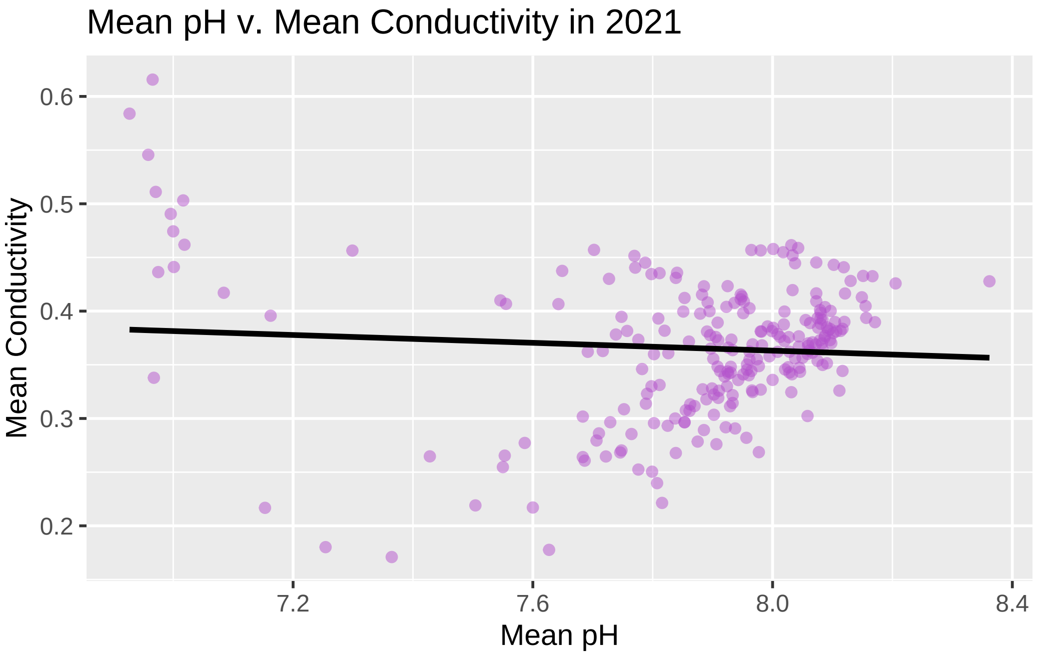
I firstly explored the relationship between mean salinity and mean temperature with jitter plots across both years. However, since salinity was still not yet binned, the plots were not visually striking. To fix this, salinity was binned at this point through a mutation. This allowed for discrete categories of salinity. Box plots colored by season were created using mean salinity bins to look at its relation to other variables and the influence of seasonality.

Also, since salinity is in part derived from conductivity, the same plots were replicated using conductivity, a raw measure. I also created additional plots between salinity and conductivity in order to verify the consistency of the two variables. Ultimately, the conductivity variable provided wider variability and allowed for better interpretation of the relationships. Due to this, conductivity was considered a practical replacement for salinity for the rest of my analysis. For both years in comparing mean pH and mean conductivity, there was a negative relationship (though this was much stronger in 2020). After faceting this plot by season, seasonal influence between the two variables became evident, as patterns shifted across seasons for both years.

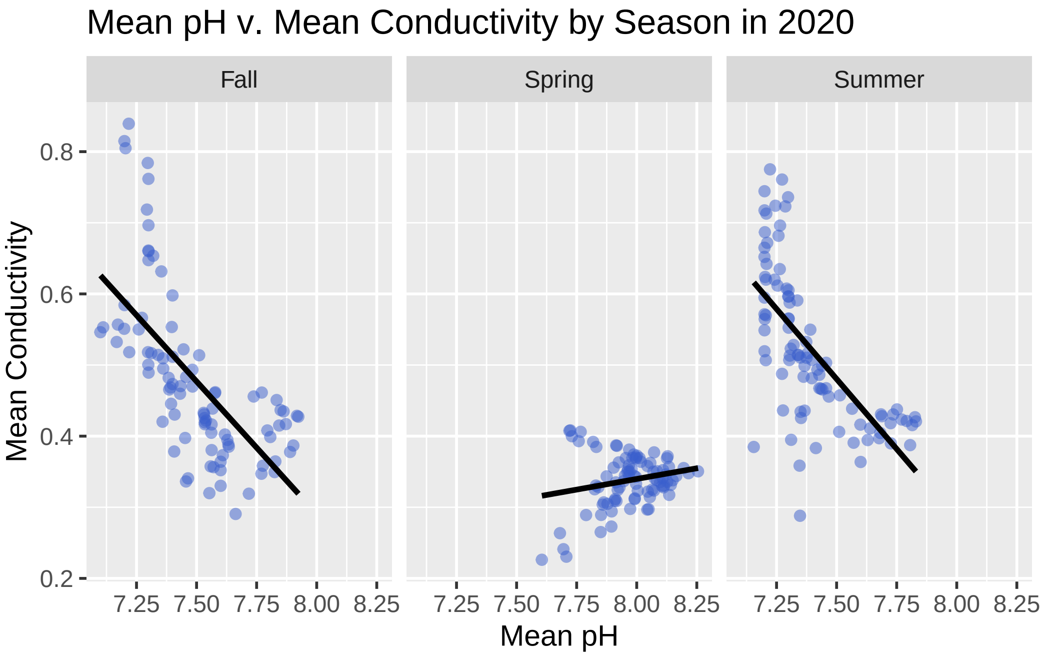
```
`geom_smooth()` using formula = 'y ~ x'
```



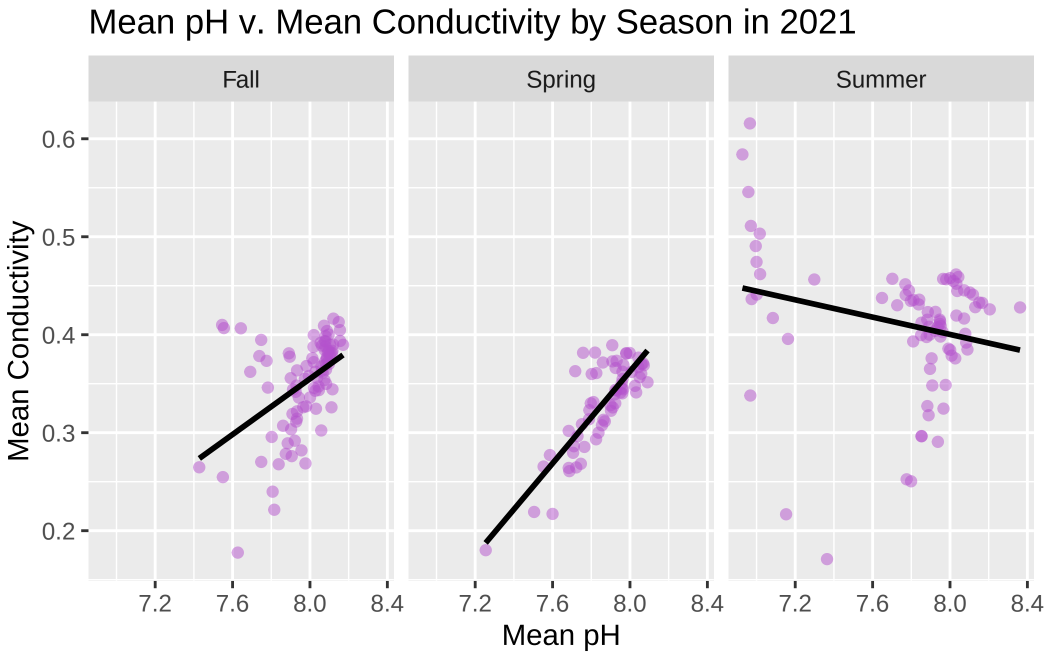
```
`geom_smooth()` using formula = 'y ~ x'
```



```
`geom_smooth()` using formula = 'y ~ x'
```

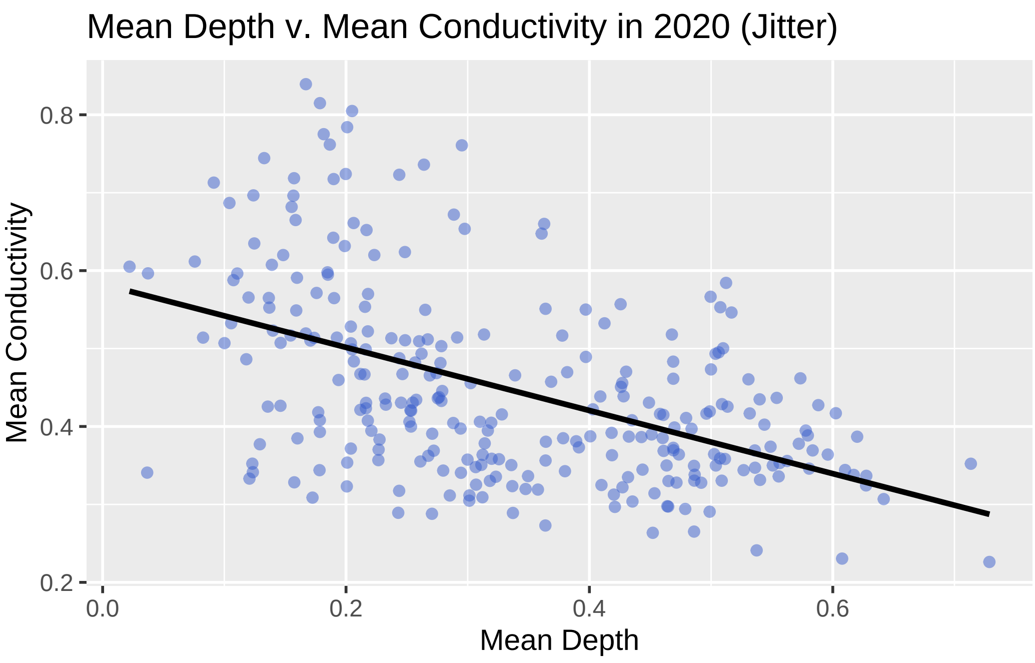


```
`geom_smooth()` using formula = 'y ~ x'
```

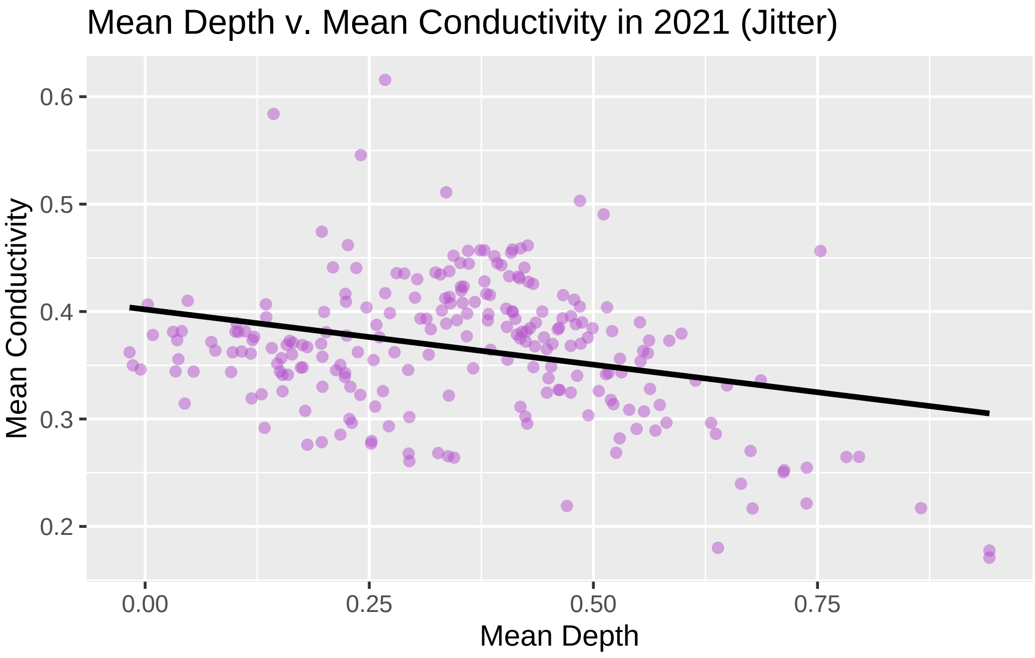


Next, I looked at the relationship between conductivity and temperature. These jitter plots of temperature versus conductivity produced an extremely dense cloud of overlapping points, creating a clustered pattern which were not visually effective for analysis. These plots suggest that there may be a distinctive relationship between conductivity and temperature. I also looked at the relationship between depth and conductivity for both years and found a consistent negative relationship. Temperature and depth for both years also shared a similar negative pattern.

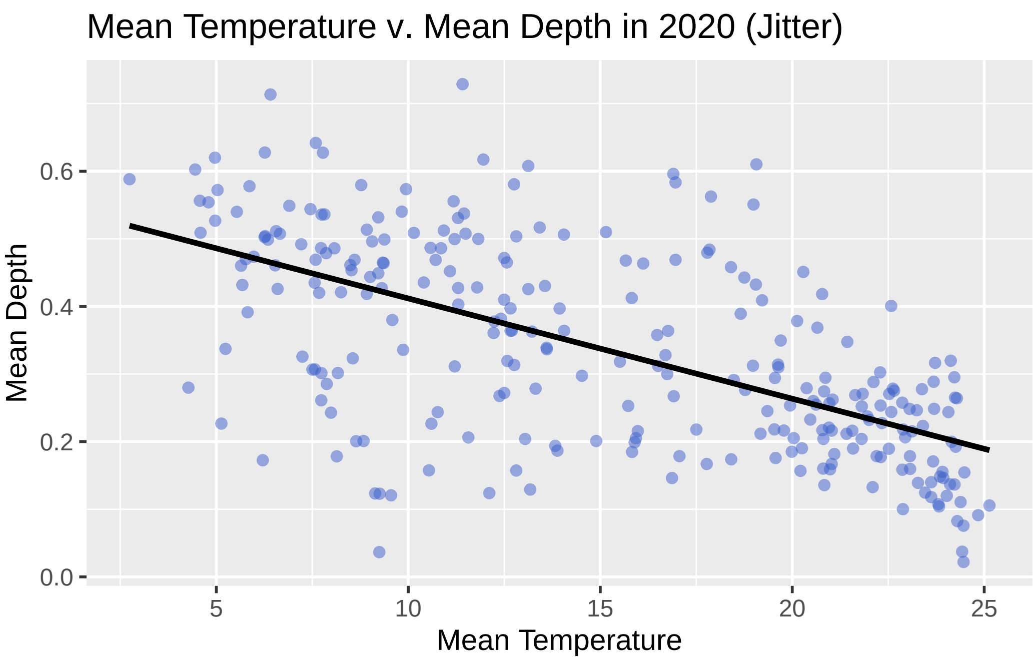
```
`geom_smooth()` using formula = 'y ~ x'
```



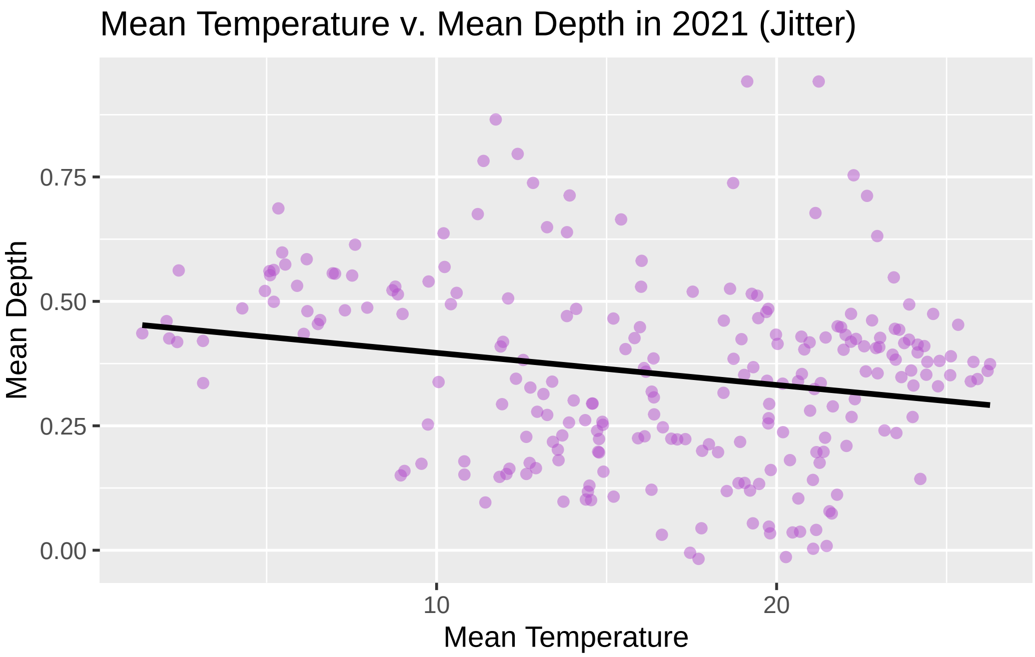
```
`geom_smooth()` using formula = 'y ~ x'
```



```
`geom_smooth()` using formula = 'y ~ x'
```

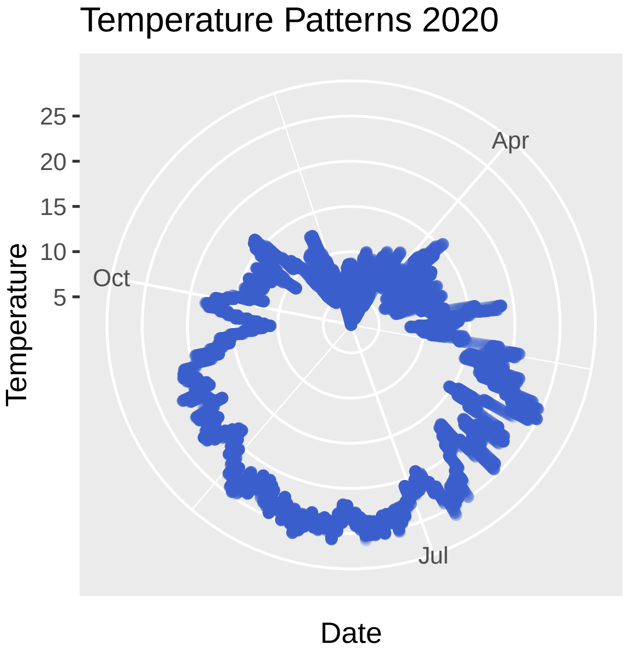


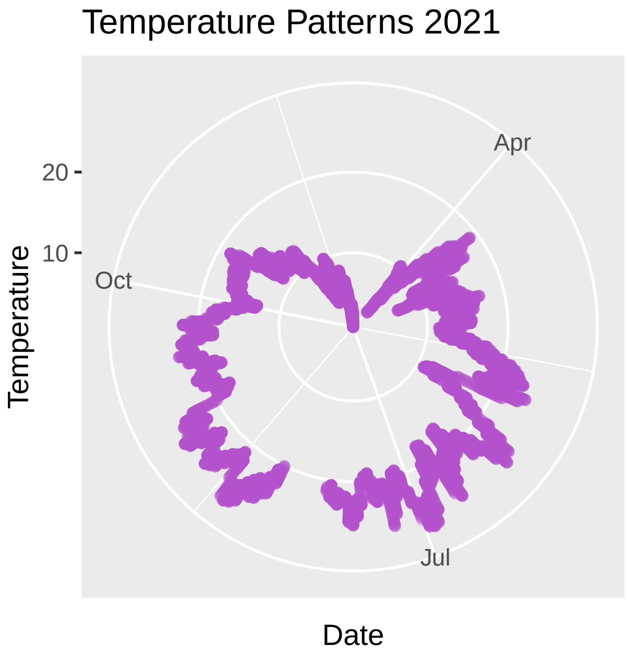
```
`geom_smooth()` using formula = 'y ~ x'
```



Polar Plots Exploration

I experimented with polar plots as a new way of visualizing temperature over time in a cyclical manner. This plot style was chosen to visualize temperature over time as an environmental cycle. By plotting temperature and date in the polar plots, I wanted to highlight possible recurring patterns in temperature. These plots offered an interesting visual perspective, but did not necessarily reveal new information. I also thought that the polar plot may clarify the relationship between conductivity and temperature, but the plot does not offer much insight. Ultimately, I found that polar plots worked most effectively looking at one variable over time.





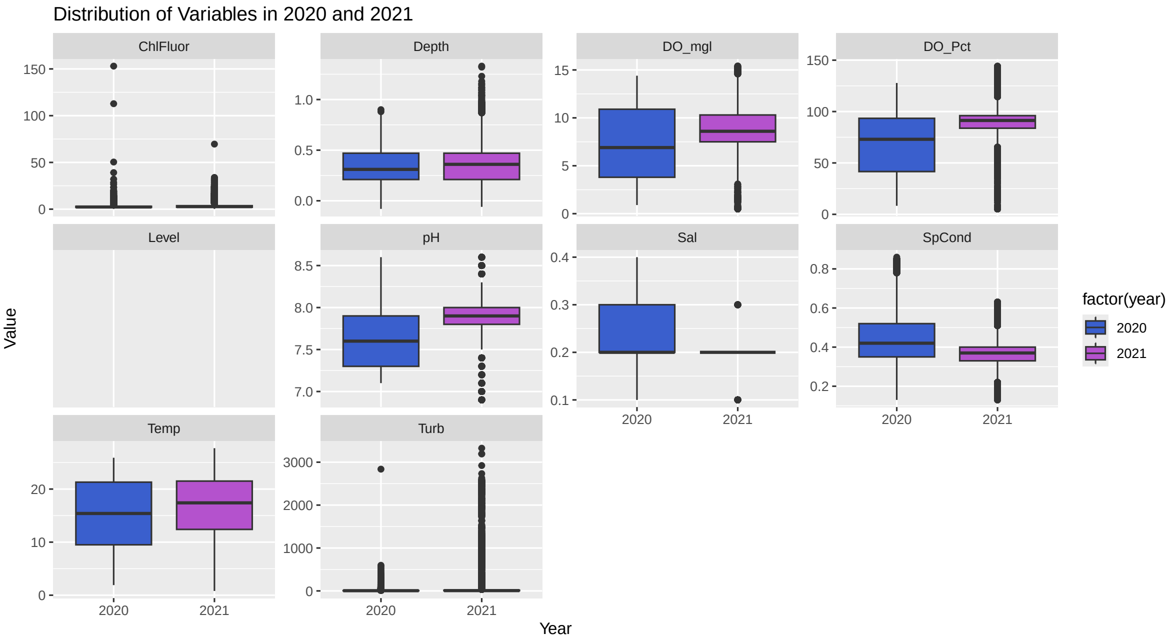
Combined Dataset Exploration

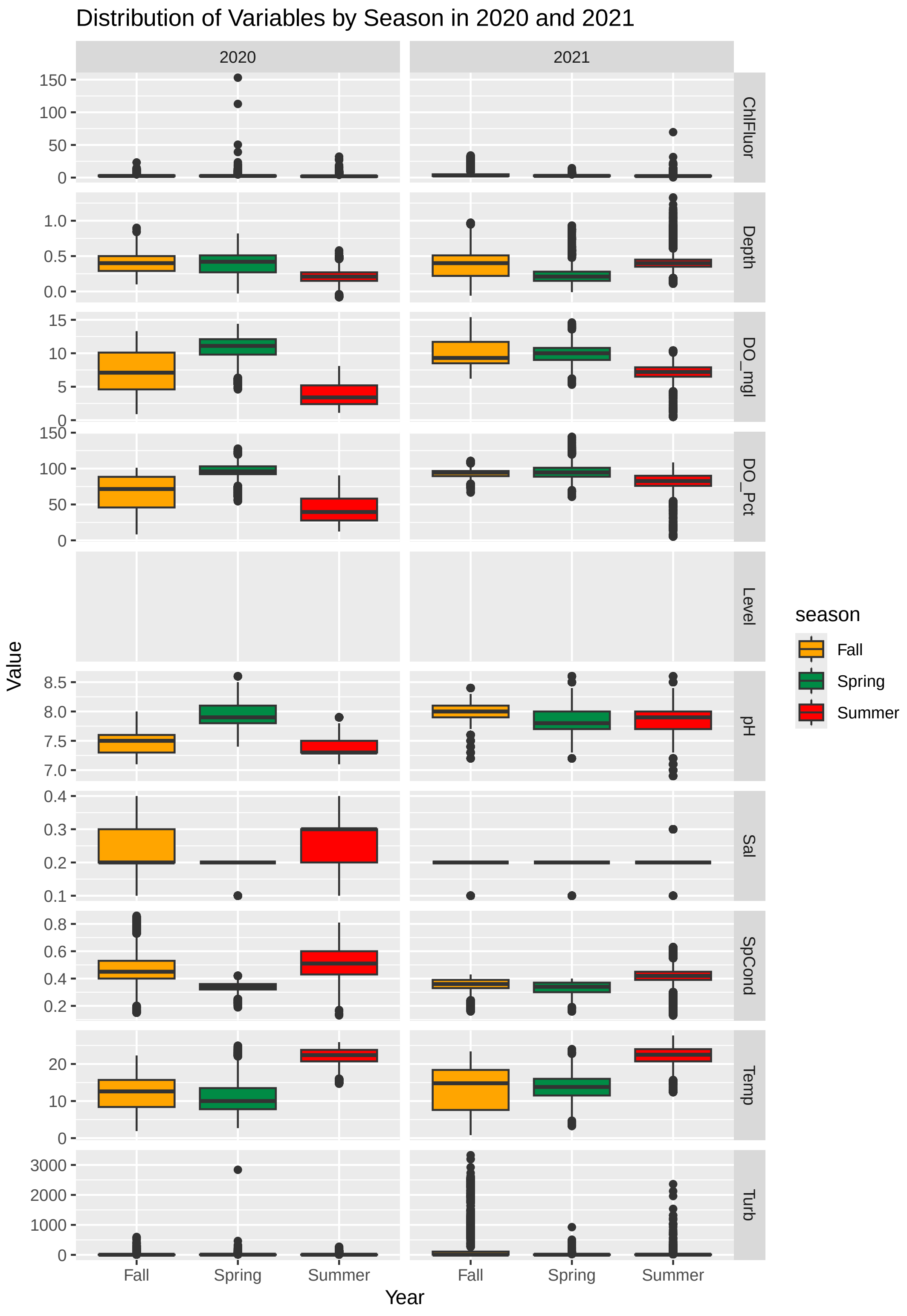
```
New names:
New names:
* `...31` -> `...32`
```

For the last part of my study, I combined the two datasets from 2020 and 2021 for a more effective comparison. By restructuring the combined dataset into a long format, I created boxplots that display the same distribution of key variables, except side by side for each year. For the next plot, I created similar boxplots except colored by season in order to show how the variables varied both annually and seasonally.

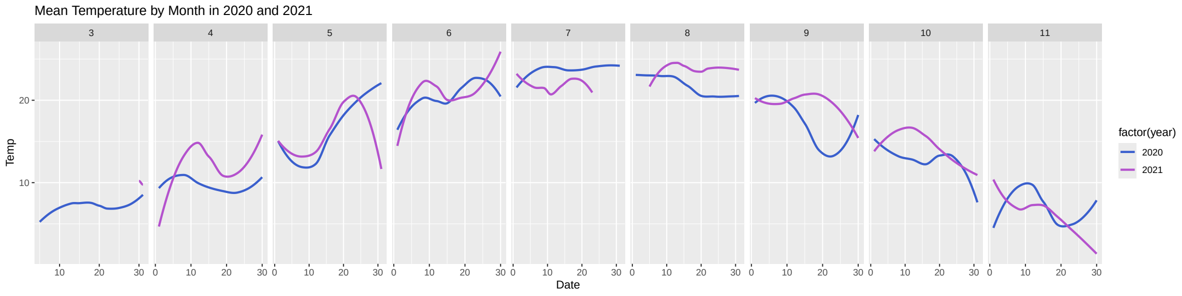
After creating plots to visualize the distributions of variables side by side, I then created time series plots for individual variables with a yearly comparison, similar to those created at the beginning of the study. However, this time, I explored using smoothed plots with a loess regression. I decided to use this approach in order to reduce noise and gain more clarity in overall yearly patterns for each variable. Additionally since two lines would be present in the same plot, without smoothing, interpretability could be challenging. However, results could be even more clear by increasing the width of each monthly plot, if possible.

```
New names:
New names:
* `...31` -> `...32`
```

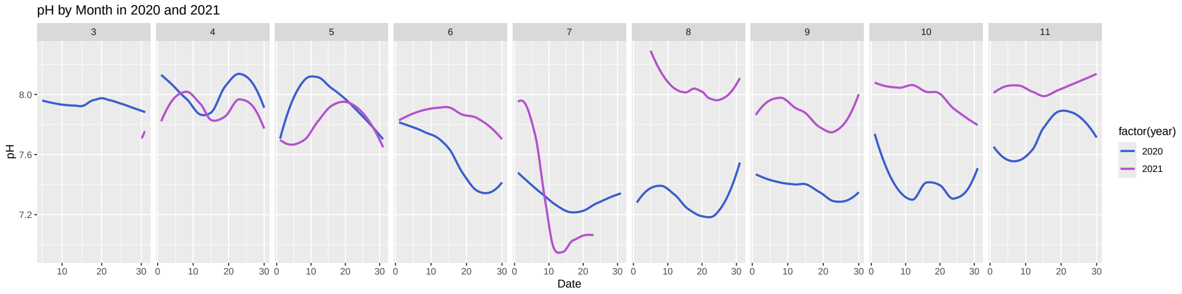

Distribution of Variables in 2020 and 2021
ChlFluor
Depth
DO_mgl
DO_Pct
Level
pH
Sal
SpCond
Temp
Turb
Value
Year
2020
2021
factor(year)
2020
2021
0
50
100
150
0.0
0.5
1.0
0
5
10
15
0
50
100
150
7.0
7.5
8.0
8.5
0.1
0.2
0.3
0.4
0.2
0.4
0.6
0.8
0
10
20
0
1000
2000
3000


Distribution of Variables by Season in 2020 and 2021
2020
2021
ChlFluor
Depth
DO_mgl
DO_Pct
Level
pH
Sal
SpCond
Temp
Turb
150
100
50
0
1.0
0.5
0.0
15
10
5
0
150
100
50
0
8.5
8.0
7.5
7.0
0.4
0.3
0.2
0.1
0.8
0.6
0.4
0.2
20
10
0
3000
2000
1000
0
Value
Fall
Spring
Summer
Fall
Spring
Summer
Year
season
Fall
Spring
Summer

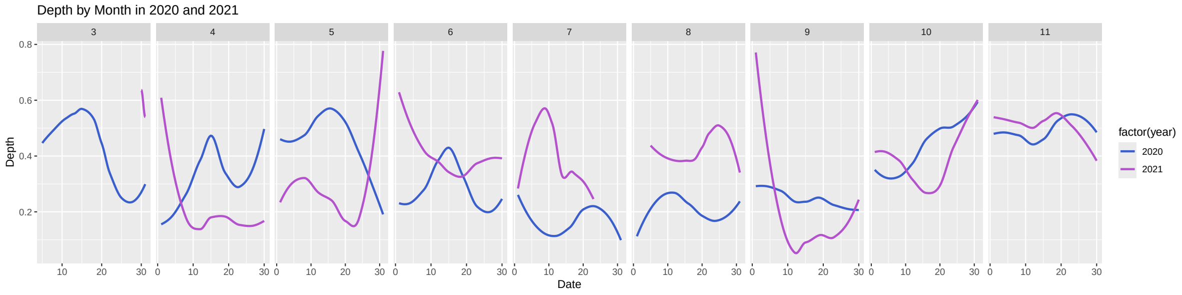
```
`geom_smooth()` using formula = 'y ~ x'
```



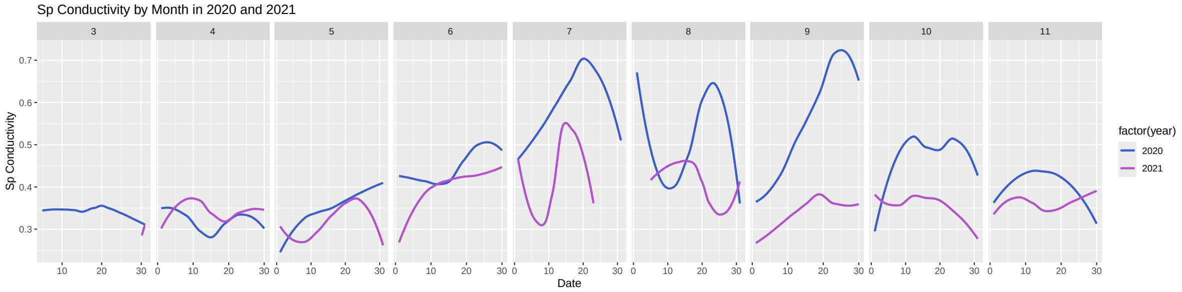
```
`geom_smooth()` using formula = 'y ~ x'
```



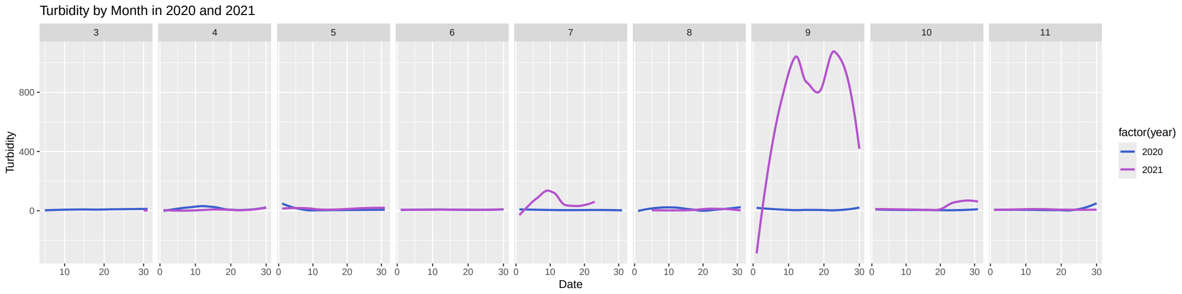
```
`geom_smooth()` using formula = 'y ~ x'
```



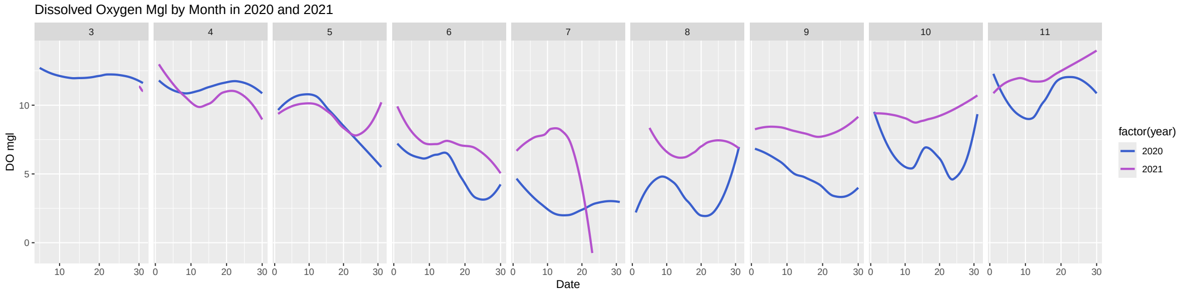
```
`geom_smooth()` using formula = 'y ~ x'
```



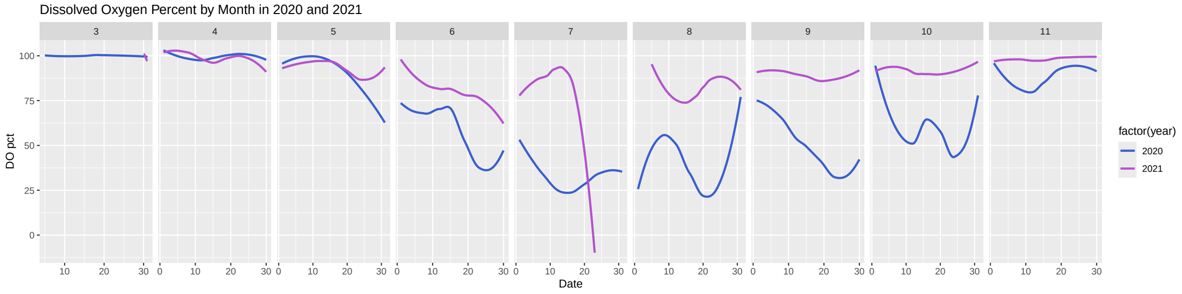
```
`geom_smooth()` using formula = 'y ~ x'
```



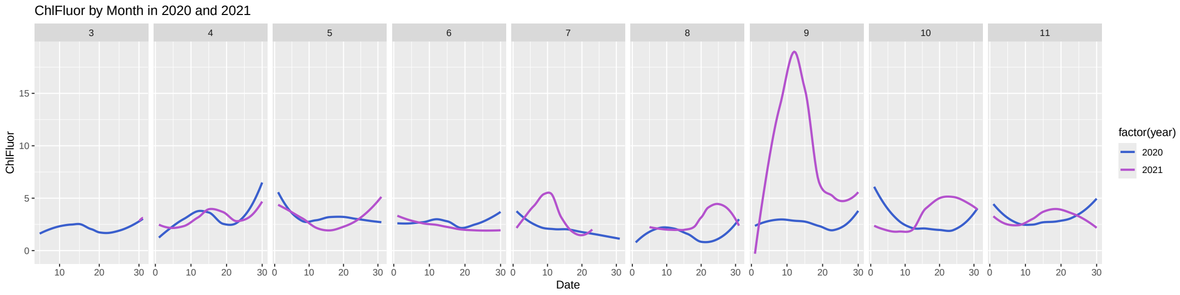
```
`geom_smooth()` using formula = 'y ~ x'
```



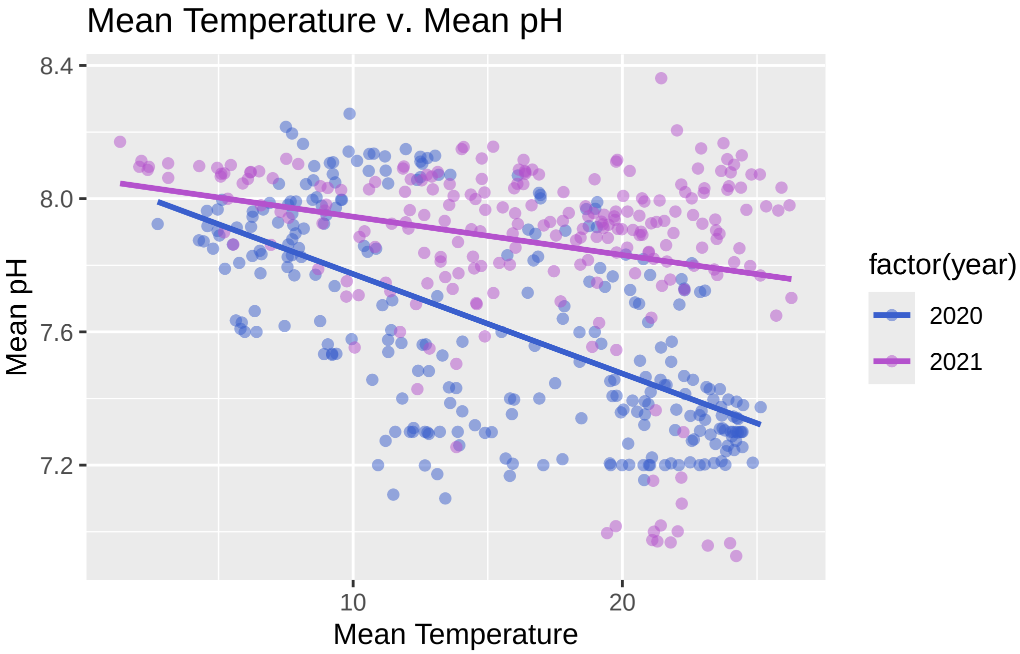
```
`geom_smooth()` using formula = 'y ~ x'
```



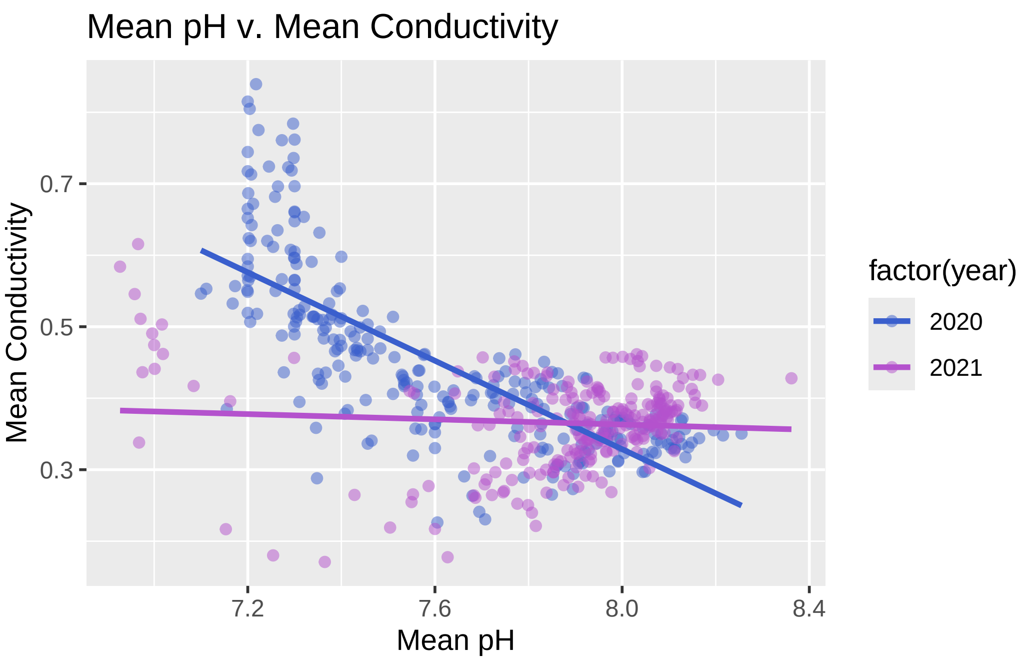
```
`geom_smooth()` using formula = 'y ~ x'
```



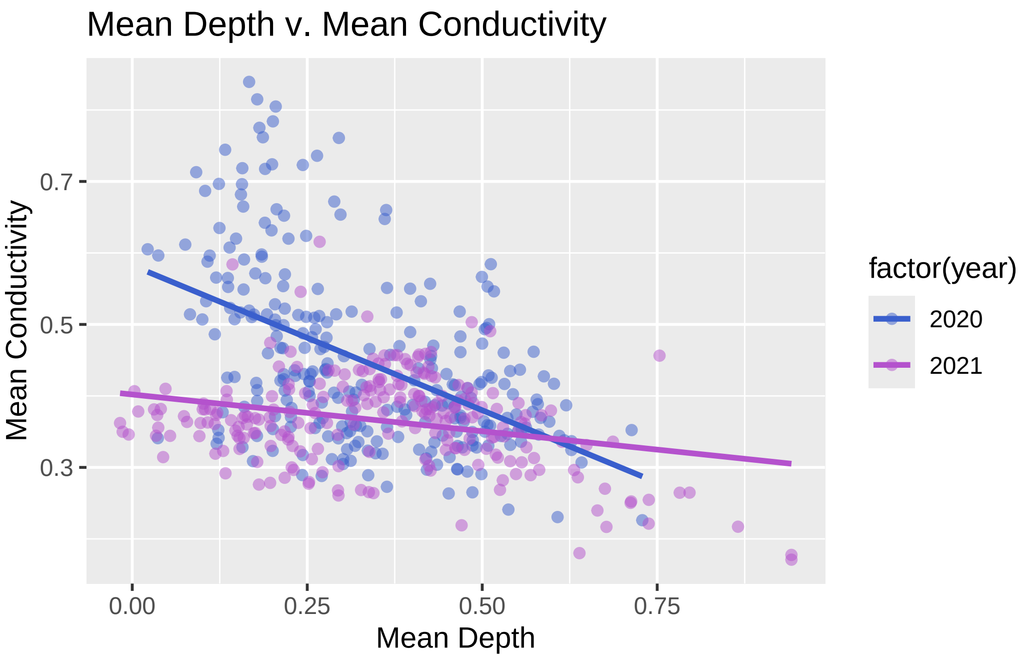
```
`geom_smooth()` using formula = 'y ~ x'
```



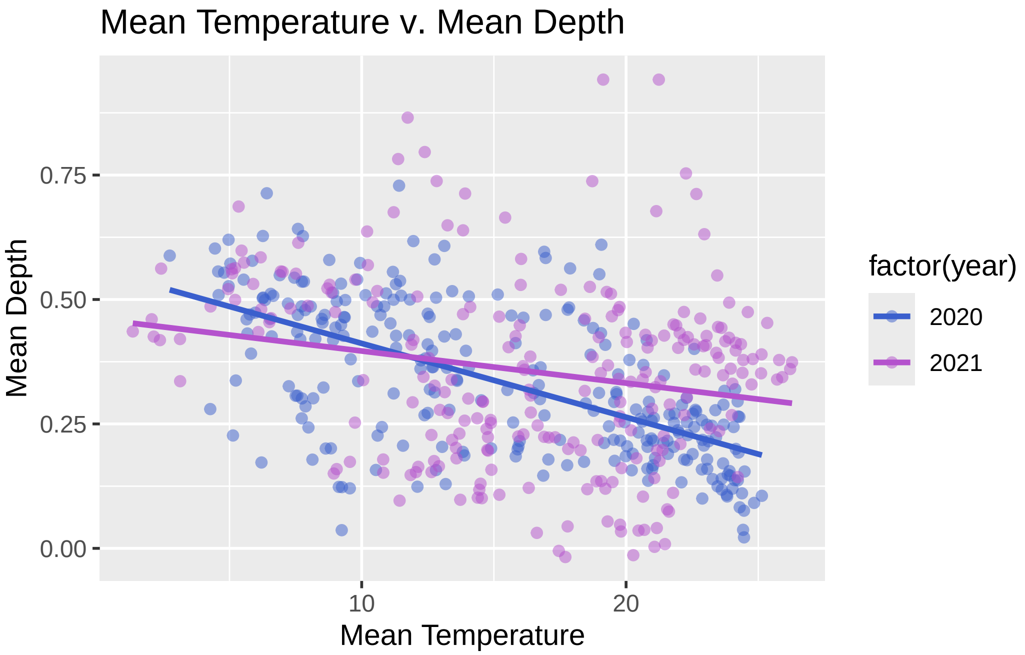
```
`geom_smooth()` using formula = 'y ~ x'
```



```
`geom_smooth()` using formula = 'y ~ x'
```



```
`geom_smooth()` using formula = 'y ~ x'
```



Note on Warnings

Throughout my study, my code produced several warnings. While generating boxplots and smoothed time series plots, R produced warnings about the removal of rows due to non-finite values. These warnings are expected in this dataset, since there are various gaps in data which vary by variable. The number of removed rows for each plot is higher for 2021, there are more missing values in the dataset. Likewise, combined dataset has more removed rows since the dataset contains both sets of missing values.

Additionally, removals are slightly smaller for plots using daily means because these contain fewer points than the non-averaged plots. Also, the loess warnings likely reflect the smoothing algorithm adjusting for the gaps in the data. Ultimately, these warnings persisted because they generally stem from the structural issue of missing values in the datasets. Though, also, when combining the datasets, R produced a "new names" warning (…31 -> ..32). This occurred because this variable did not align across both. However, these columns were not relevant to the analysis and were ignored. To keep the report clean, these warnings were suppressed.

Conclusions & Future Opportunities

The exploration of water quality from Stony Creek in the Hudson River across 2020 and 2021 highlighted seasonal patterns, monthly patterns, distributions of variables, and discrepancies between years. Also, the combination of the datsets more effectively compared trends among years. This could be strengthened by adding additional yearly datasets. However, missing data and the uneven timing of some missing data between both years may limit the findings. In the future, development of more robust methods for handling in-completion within the datasets. Also, it would be interesting to explore shifts in the variables over time by linking findings to external conditions at the time, such as climate or tides. Lastly, future work could include the development of predictive models in order to anticipate fluctuations or patterns.