Water Quality Exploratory Data Analysis

Begin working with file for 2020

```
library(tidyverse)
#Load in tidyverse package for cleaning and wrangling
```

```
#Load in 2020 file
dataset_2020 <- read_csv("data/hudscwq2020.csv")
```

Read in the data file from the National Estuarine Research Reserve System at cdmo.baruch.sc.edu/dges/ This file is data specific to the water quality of the Stony Creek station of the Hudson River in 2020.

In the next code we can load in another dataset for the next year to see if all the codes run the same.

Begin working with file for 2021

```
#Copy for 2021
dataset_2021 <- read_csv("data/hudscwq2021.csv")
```

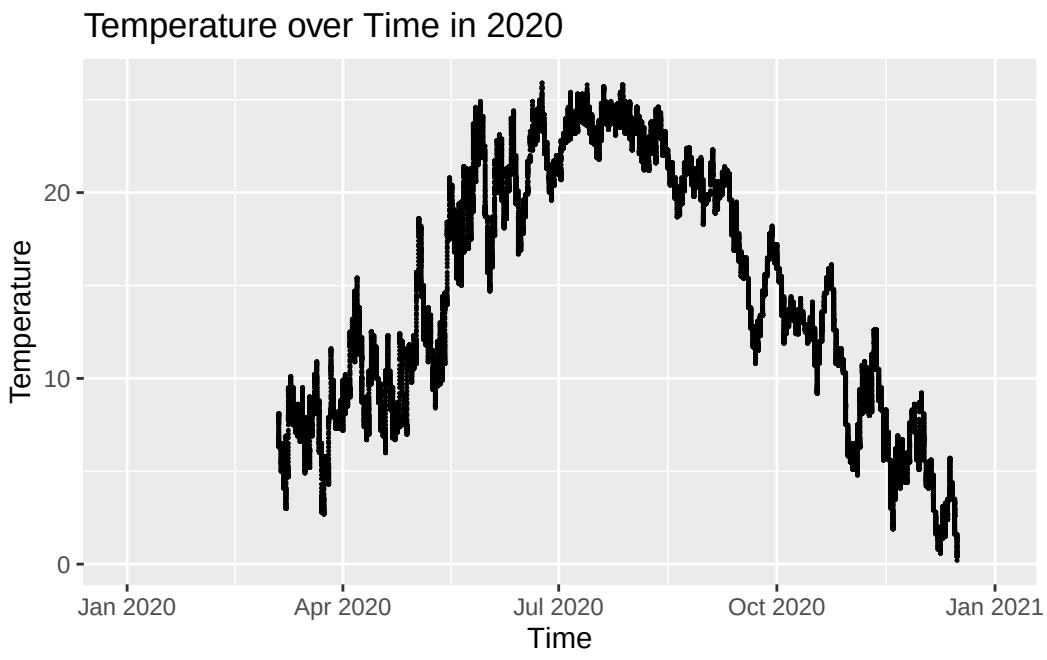
```
New names:
Rows: 35040 Columns: 31
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(15): StationCode, isSWMP, DateTimeStamp, F_Record, F_Temp, F_SpCond, F_... dbl
(12): Historical, ProvisionalPlus, Temp, SpCond, Sal, DO_Pct, DO_mgl, De... lgl
(4): 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.
* `` -> `...31`
```

```r
dataset_2020 <- dataset_2020 %>%
  separate(DateTimeStamp, into = c("date", "time"), sep = " ")%>%
  mutate(date_clean = mdy(str_trim(date)))
```

This separates the datetime variable into date and time. This step will allow for us to analyze date and time separately. This also cleans the date column by removing any non-numerics and ensures that the date values will be consistent.

```r
#Copy for 2021
dataset_2021 <- dataset_2021 %>%
  separate(DateTimeStamp, into = c("date", "time"), sep = " ")%>%
  mutate(date_clean = mdy(str_trim(date)))
```

```r
library(ggplot2)

ggplot(data = dataset_2020, aes(x = date_clean, y = Temp))+
         geom_point(size = 0.2)+
  labs(title = "Temperature over Time in 2020",
       x = "Time",
       y = "Temperature")
```

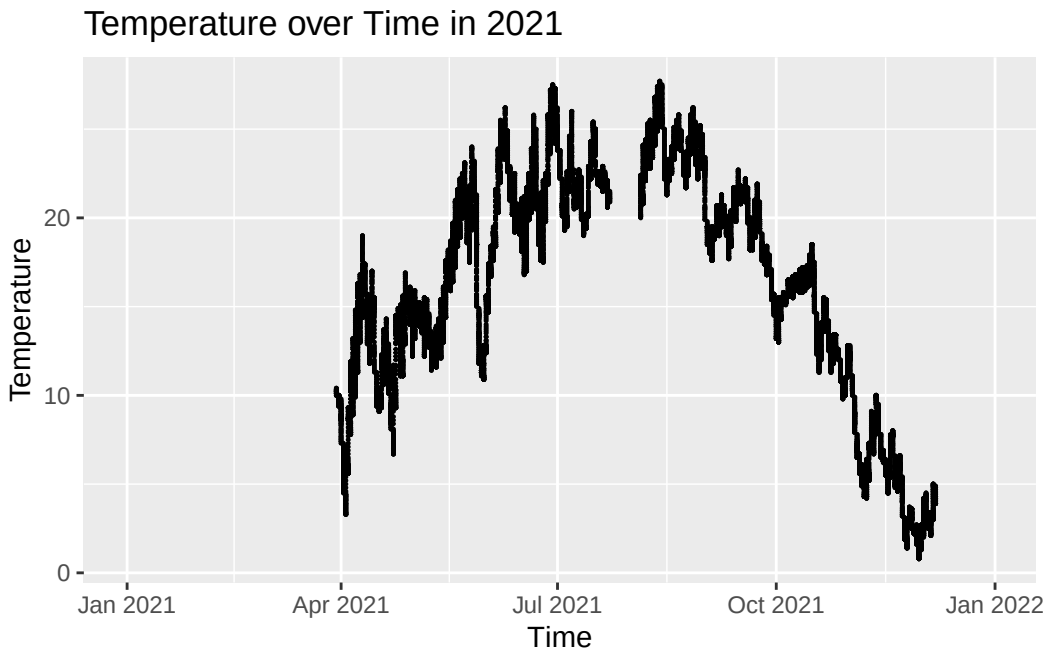


Plot to firstly observe general temperature patterns over time. This plot shows that temperature spikes during late Spring and Summer, and it declines around Fall. Though there are

some significant fluctuations during Spring and Fall, but temperature during Summer looks to be more consistent at the peak temperatures during July through September.

```
library(ggplot2)
#Copy for 2021

ggplot(data = dataset_2021, aes(x = date_clean, y = Temp))+
         geom_point(size = 0.2)+
  labs(title = "Temperature over Time in 2021",
       x = "Time",
       y = "Temperature")
```

```
Warning: Removed 12142 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
dataset_2020 <- dataset_2020 %>%
  mutate(month = month(date_clean))
```

This extracts month from the new date_clean column so we can categorize the data by months.

```r
#copy for 2021
dataset_2021 <- dataset_2021 %>%
  mutate(month = month(date_clean))
```

```r
dataset_2020 <- dataset_2020 %>%
  mutate(date_clean = as.Date(date_clean))
```

```r
#copy for 2021
dataset_2021 <- dataset_2021 %>%
  mutate(date_clean = as.Date(date_clean))
```

This converts date_clean to a Date object to further ensure date is consistently formatted.

```r
dataset_2020 <- dataset_2020 %>%
  mutate(
    month = as.numeric(month), #make sure its considered a numeric because maybe not all mont
    season = case_when(
      month %in% 3:5 ~ "Spring",
      month %in% 6:8 ~ "Summer",
      month %in% 9:11 ~ "Fall",
      month %in% c(12, 1, 2) ~ "Winter",
      TRUE ~ NA_character_ #clarifies that only values of 12,1 and 2 should go into winter to
    )
  )
```

This creates a seasonal variable based on the month so data can be organized seasonally This chunk can be edited to get rid of or edit seasonal variables (may wanted to get rid of Winter/ change to NAs).

```r
#copy for 2021
dataset_2021 <- dataset_2021 %>%
  mutate(
    season = case_when(
      month %in% 3:5 ~ "Spring",
      month %in% 6:8 ~ "Summer",
      month %in% 9:11 ~ "Fall",
      month %in% c(12, 1, 2) ~ "Winter",
      TRUE ~ NA_character_ #clarifies that only values of 12,1 and 2 should go into winter to
    )
  )
```

This output will count all of the NA values (missing values) for variables in each season. Fall has the least missing variables outside of F_Record, Level, cLevel, F_cLevel, and the …31 variable which can be disregarded. These variables all have missing values for other seasons (between 8700 and 8800). The large number of missing NAs in the Level variable may be a concern, depending upon its value for the purposes of our study. Fall does have 2 missing variables for cDepth (corrected depth) as well, though it is not missing any raw depth values.

Many variables for spring have between 439 and 441 missing values including significant variables such as Temp, Sal, pH, Turb, SpCond, and ChlFlour. The moderate number of missing values come from variables measured using sensors, so the loss of data may be attributed to sensor issues.

Summer often has between 2 to 4 missing values for variables, so summer is mostly complete, with only minor missing data.

Winter is missing the most variables with thousands of missing values for variables of study. This is likely attributed to a data collection gap during the season.

This code will group by month rather than season to examine counts of missing values by month.

The output shows that months of December, January, February, and March consistently have missing values, which can be attributed to the data collection gap. On occasion, other months such as July have 2 missing values for many raw variables, which suggests a minor inconsistency or data entry issue. Some variables such as cLevel, F_cLevel, …31, Level, and F_Record consistently have missing values across all months. This indicates a structural absence in collection of Level measurements.

This creates a new dataset without Winter.

Summary Statistics

```
#install.packages("DT")
#library(DT)
summary_stats_no_winter_2020 <- no_winter_2020 %>%
  summarise(
    across(
      where(is.numeric),
      list(
        mean = ~mean(.x, na.rm = TRUE),
        median = ~median(.x, na.rm = TRUE),
        sd = ~sd(.x, na.rm = TRUE),
        min = ~min(.x, na.rm = TRUE),
        max = ~max(.x, na.rm = TRUE)
      )
```

```
    )
  )%>%
  pivot_longer(
    everything(),
    names_to = c("variable", "metric"),
    names_sep = "_",
    values_to = "stat_value"
  )
```

```
Warning: Expected 2 pieces. Additional pieces discarded in 10 rows [26, 27, 28, 29, 30,
31, 32, 33, 34, 35].
```

```
#DT :: datatable(summary_stats_no_winter_2020)
#To produce a datatable instead of a tibble
#commenting out so pdf can render
```

```
#copy for 2021
summary_stats_no_winter_2021 <- no_winter_2021 %>%
  summarise(
    across(
      where(is.numeric),
      list(
        mean = ~mean(.x, na.rm = TRUE),
        median = ~median(.x, na.rm = TRUE),
        sd = ~sd(.x, na.rm = TRUE),
        min = ~min(.x, na.rm = TRUE),
        max = ~max(.x, na.rm = TRUE)
      )
    )
  )%>%
  pivot_longer(
    everything(),
    names_to = c("variable", "metric"),
    names_sep = "_",
    values_to = "stat_value"
  )
```

```
Warning: Expected 2 pieces. Additional pieces discarded in 10 rows [26, 27, 28, 29, 30,
31, 32, 33, 34, 35].
```

```
#DT :: datatable(summary_stats_no_winter_2021) commented out so pdf can render
```

This code produces summary statistics for all of our variables using our dataset without winter.

Summary Stats by Season

```
summary_stats_no_winter_season_2020 <- no_winter_2020 %>%
  group_by(season)%>%
  summarise(
    across(
      where(is.numeric),
      list(
        mean = ~mean(.x, na.rm = TRUE),
        median = ~median(.x, na.rm = TRUE),
        sd = ~sd(.x, na.rm = TRUE),
        min = ~min(.x, na.rm = TRUE),
        max = ~max(.x, na.rm = TRUE)
      )
    )
  )%>%
  pivot_longer(
    -season,
    names_to = c("variable", "metric"),
    names_sep = "_",
    values_to = "stat_value"
  )
```

```
Warning: Expected 2 pieces. Additional pieces discarded in 10 rows [26, 27, 28, 29, 30,
31, 32, 33, 34, 35].
```

```
#DT :: datatable(summary_stats_no_winter_season_2020)
```

```
#for 2021
summary_stats_no_winter_season_2021 <- no_winter_2021 %>%
  group_by(season)%>%
  summarise(
    across(
      where(is.numeric),
      list(
```

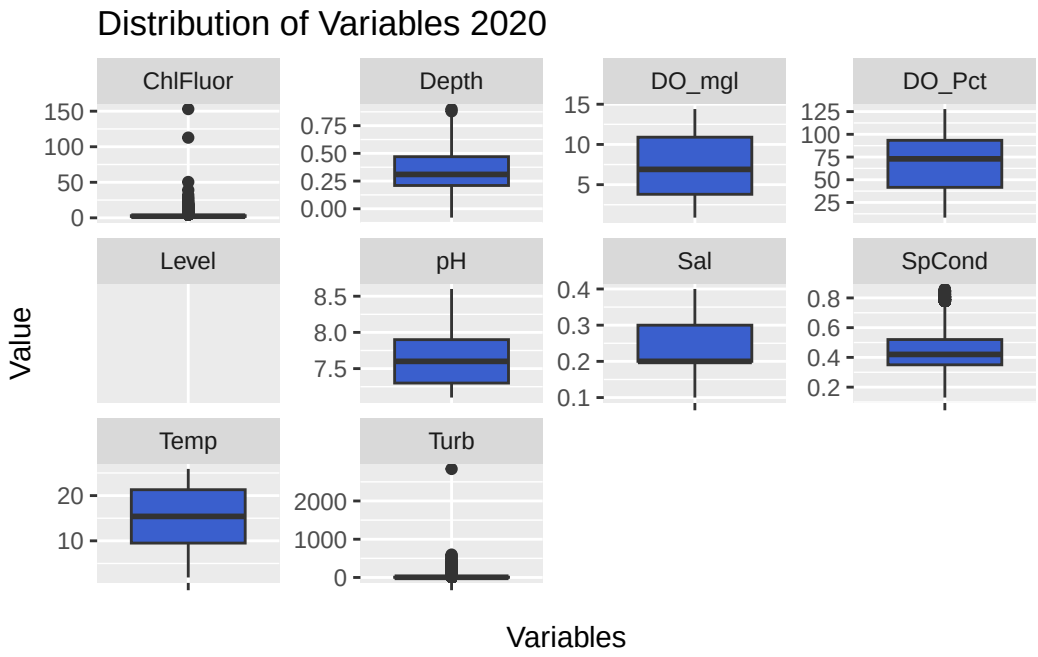
```
        mean = ~mean(.x, na.rm = TRUE),
        median = ~median(.x, na.rm = TRUE),
        sd = ~sd(.x, na.rm = TRUE),
        min = ~min(.x, na.rm = TRUE),
        max = ~max(.x, na.rm = TRUE)
      )
    )
  )%>%
  pivot_longer(
    -season,
    names_to = c("variable", "metric"),
    names_sep = "_",
    values_to = "stat_value"
  )
```

```
Warning: Expected 2 pieces. Additional pieces discarded in 10 rows [26, 27, 28, 29, 30,
31, 32, 33, 34, 35].
```

```
#DT :: datatable(summary_stats_no_winter_season_2021)
```

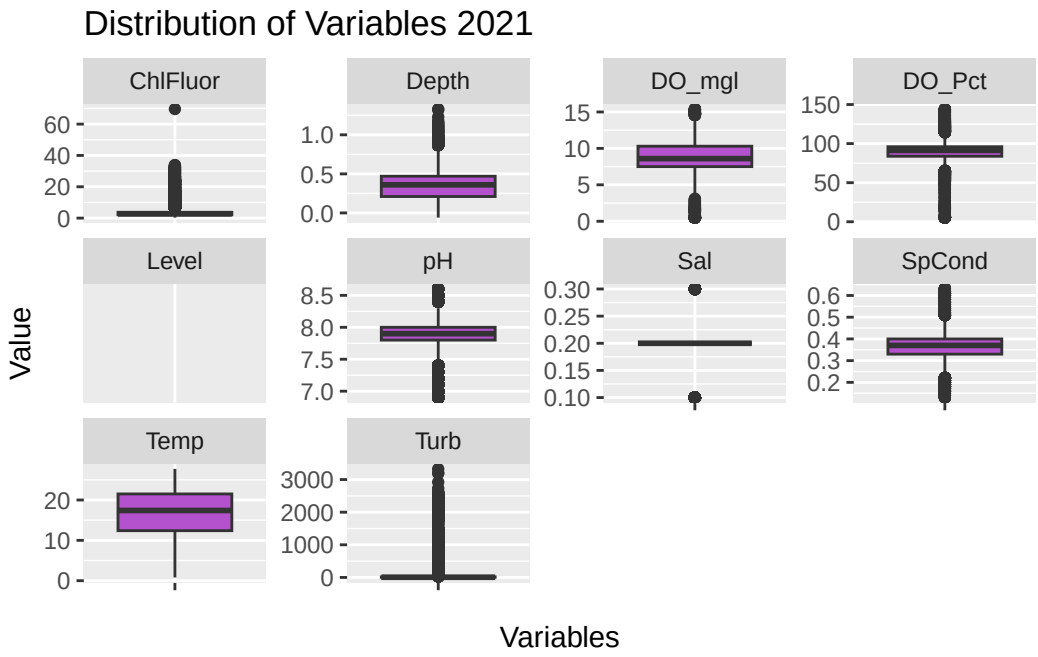
Variable Summaries (Box Plots)

```
no_winter_2020 %>%
  pivot_longer(
    cols = c(Temp, Sal, Turb, ChlFluor, Level, Depth, pH, SpCond, DO_Pct, DO_mgl),
    names_to = "variable",
    values_to = "value") %>%
  ggplot(aes(x = "", y = value))+
  geom_boxplot( fill = "royalblue3")+
  facet_wrap(~variable, scales = "free_y")+
  labs(y = NULL)+
  labs(title = "Distribution of Variables 2020",
       x = "Variables",
       y = "Value")
```



This code produces boxplot distributions for each of our variables of interest after using pivot_longer to select particular variables. The output shows a warning that 30369 rows were skipped because they were missing- this number is so large because it is the sum of all missing values

```
#copy for 2021
no_winter_2021 %>%
  pivot_longer(
    cols = c(Temp, Sal, Turb, ChlFluor, Level, Depth, pH, SpCond, DO_Pct, DO_mgl),
    names_to = "variable",
    values_to = "value") %>%
  ggplot(aes(x = "", y = value))+
  geom_boxplot(fill = "mediumorchid3")+
  facet_wrap(~variable, scales = "free_y")+
  labs(y = NULL)+
  labs(title = "Distribution of Variables 2021",
       x = "Variables",
       y = "Value")
```

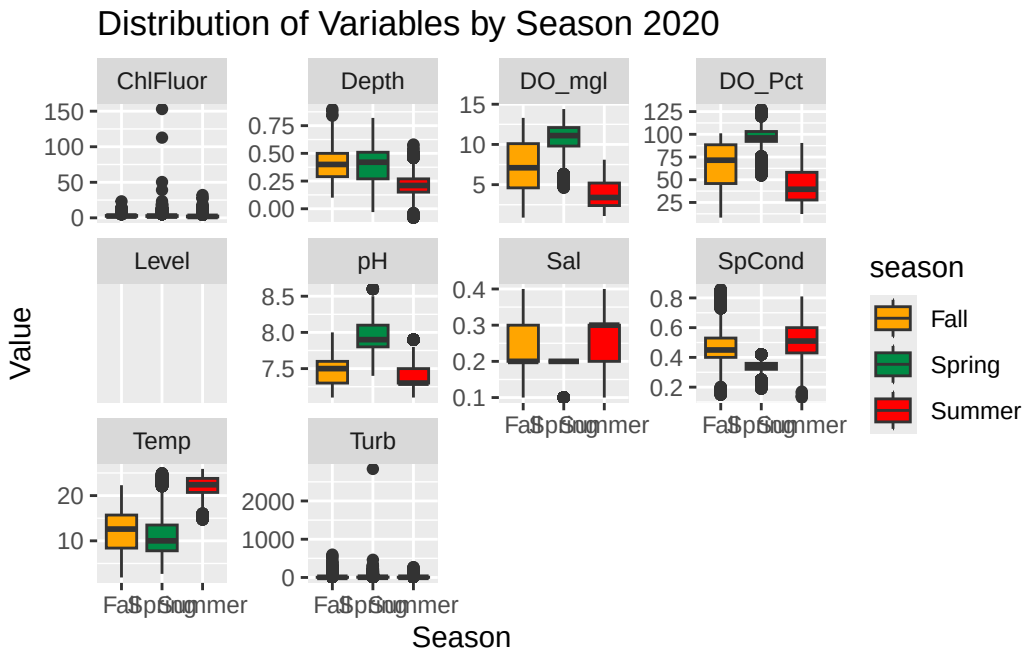


An even larger number of rows lost due to pivoting.

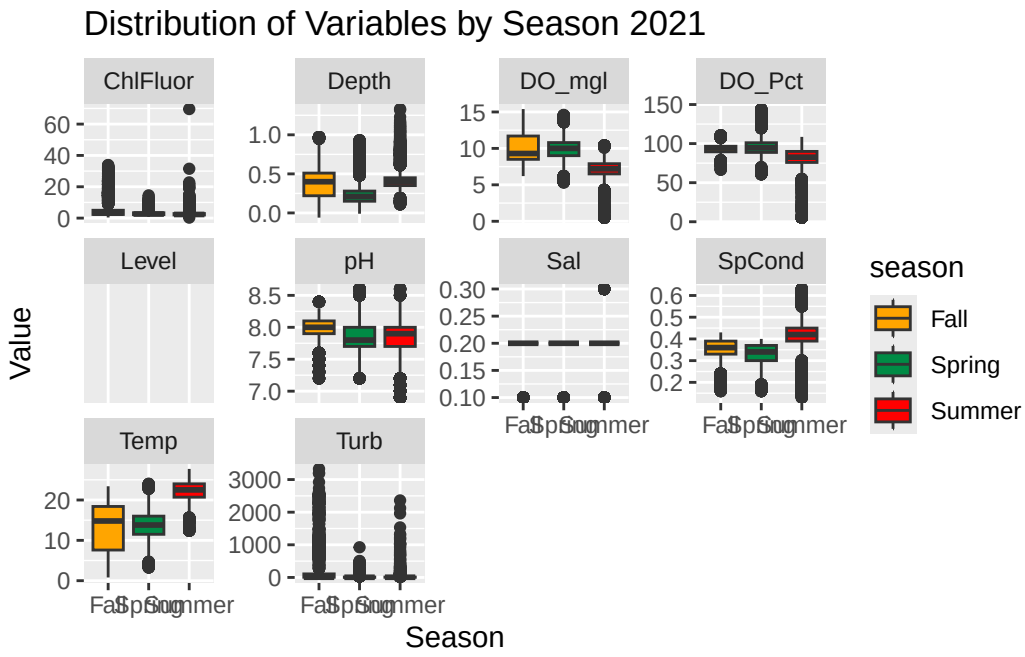
The next code uses the same variables to show summaries except by season as well

Variable Summaries by Season (Box Plots)

```
no_winter_2020 %>%
  pivot_longer(
    cols = c(Temp, Sal, Turb, ChlFluor, Level, Depth, pH, SpCond, DO_Pct, DO_mgl),
    names_to = "variable",
    values_to = "value") %>%
  ggplot(aes(x = season, y = value, fill = season))+
  geom_boxplot()+
  facet_wrap(~ variable, scales = "free_y")+
  scale_fill_manual(values = c(
    "Fall" = "orange",
    "Spring" = "springgreen4",
    "Summer" = "red"
  ))+
  labs(title = "Distribution of Variables by Season 2020",
       x = "Season",
       y = "Value")
```



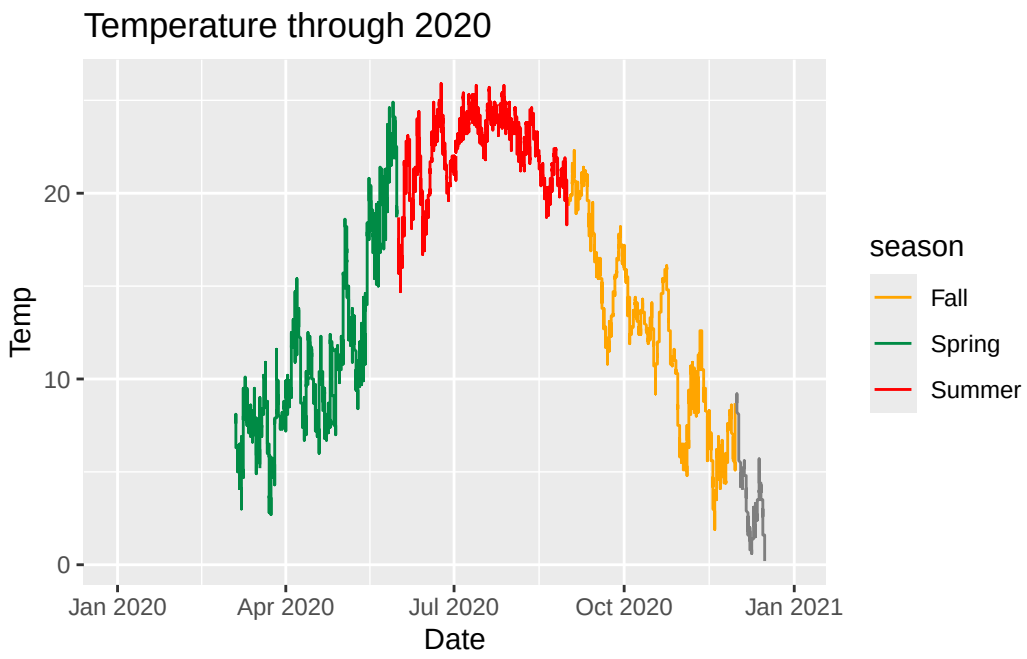

```
#copy for 2021
no_winter_2021 %>%
  pivot_longer(
    cols = c(Temp, Sal, Turb, ChlFluor, Level, Depth, pH, SpCond, DO_Pct, DO_mgl),
    names_to = "variable",
    values_to = "value") %>%
  ggplot(aes(x = season, y = value, fill = season))+
  geom_boxplot()+
  facet_wrap(~ variable, scales = "free_y")+
  scale_fill_manual(values = c(
    "Fall" = "orange",
    "Spring" = "springgreen4",
    "Summer" = "red"
  ))+
  labs(title = "Distribution of Variables by Season 2021",
       x = "Season",
       y = "Value")
```



Temperature Plots

```
ggplot(data = dataset_2020, aes(x= date_clean, y= Temp, color = season, group = season))+
  geom_line()+
scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Temperature through 2020",
       x = "Date",
       y = "Temp")
```

```
Warning: Removed 7692 rows containing missing values or values outside the scale range
(`geom_line()`).
```

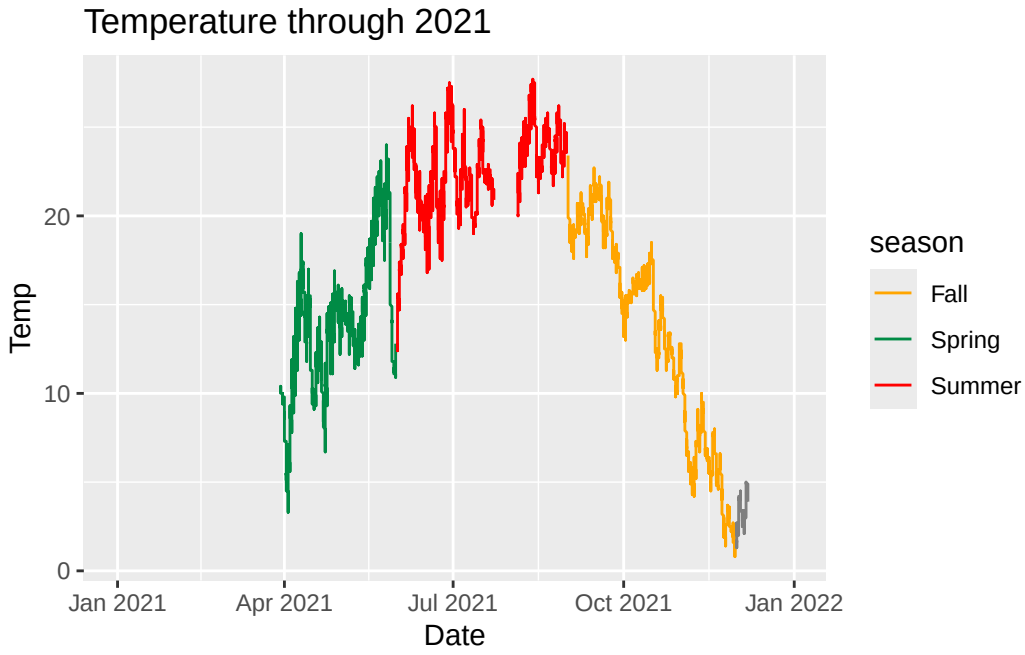


Similar to first plot, this plot observes temperature over time, but with added seasonal label, indicated by color. This plot gives a bit more clarity than the first plot as it adds color by each season, so we can more clearly see fluctuations in temperature based on the season, reinforcing prior findings. But this plot also clarifies that in early Summer, we still have some significant fluctuations in temperature, but this becomes more consistent through the rest of Summer. We do not immediately concern ourselves with the short length of Winter because this is a limitation of the dataset, as much less data is recorded during Winter months. This may impact seasonal comparisons, so we can primarily focus on Spring, Summer, and Fall which have more complete data.

The resulting plot induces a message that 7,692 rows contain missing values or values outside of `geom_line()`, so we can inspect the missing values.

```
#copy for 2021
ggplot(data = dataset_2021, aes(x= date_clean, y= Temp, color = season, group = season))+
  geom_line()+
scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Temperature through 2021",
       x = "Date",
       y = "Temp")
```

```
Warning: Removed 10864 rows containing missing values or values outside the scale range
(`geom_line()`).
```



Looks like a small gap of unusually missing data in 2021 during Summer- also some NAs at the end in Winter.

```
colSums(is.na(dataset_2020))
```

```
    StationCode           isSWMP             date             time       Historical
              0                0                0                0                0
ProvisionalPlus       F_Record             Temp           F_Temp           SpCond
              0            35136             7695                0             7695
       F_SpCond              Sal           F_Sal           DO_Pct         F_DO_Pct
              0             7695                0             7695                0
          DO_mgl        F_DO_mgl            Depth          F_Depth           cDepth
           7695                0             7695                0             7715
        F_cDepth            Level          F_Level           cLevel         F_cLevel
           7695            35136                0            35136            35136
             pH             F_pH             Turb          F_Turb         ChlFluor
           7695                0             7695                0             7695
      F_ChlFluor             ...31       date_clean            month           season
              0            35136                0                0                0
```

This code shows that we have several columns with missing data.

We do not need to concern ourselves with missing data from any of the F_(variable) columns. However, there is missing data from Temp, Turbidity (Turb), Depth, DO_pct, pH, DO_mgl, Level, ChlFlour, SpCond, and date_clean. All are variables that we may use for analysis. These are limitations of the original dataset as there are only missing values from original variables, likely due to Winter limitation.

```
#copy for 2021
colSums(is.na(dataset_2021))
```

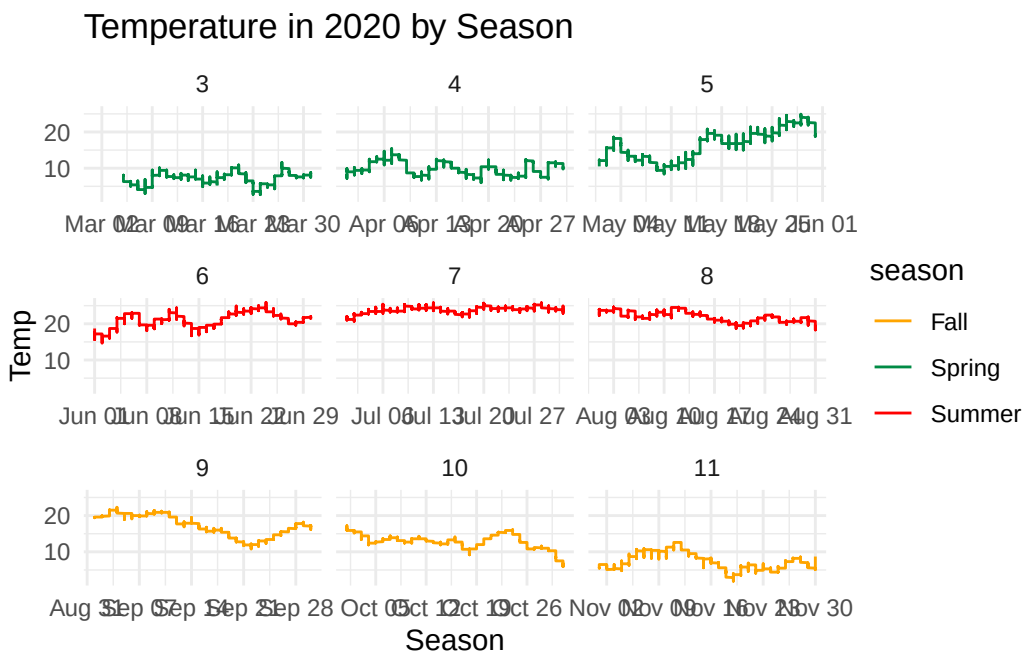
```
    StationCode          isSWMP            date            time      Historical
              0               0               0               0               0
ProvisionalPlus        F_Record            Temp          F_Temp          SpCond
              0           27714           12142               0           12142
       F_SpCond             Sal          F_Sal          DO_Pct        F_DO_Pct
              0           12142               0           12603               0
         DO_mgl        F_DO_mgl           Depth         F_Depth          cDepth
          12603               0           12142               0           12343
       F_cDepth           Level         F_Level          cLevel        F_cLevel
          12142           35040               0           35040           35040
             pH            F_pH            Turb          F_Turb        ChlFluor
          12142               0           12142               0           12142
     F_ChlFluor           ...31      date_clean           month          season
              0           35040               0               0               0
```

Slightly more missing values for each notable variable.

Temperature Plots Seasonally

```
ggplot(data = no_winter_2020, aes(x = date_clean, y = Temp, color = season))+
  geom_line()+
  facet_wrap(~month, scales = "free_x")+
  theme_minimal()+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Temperature in 2020 by Season",
       x = "Season",
       y = "Temp")
```

```
Warning: Removed 438 rows containing missing values or values outside the scale range
(`geom_line()`).
```

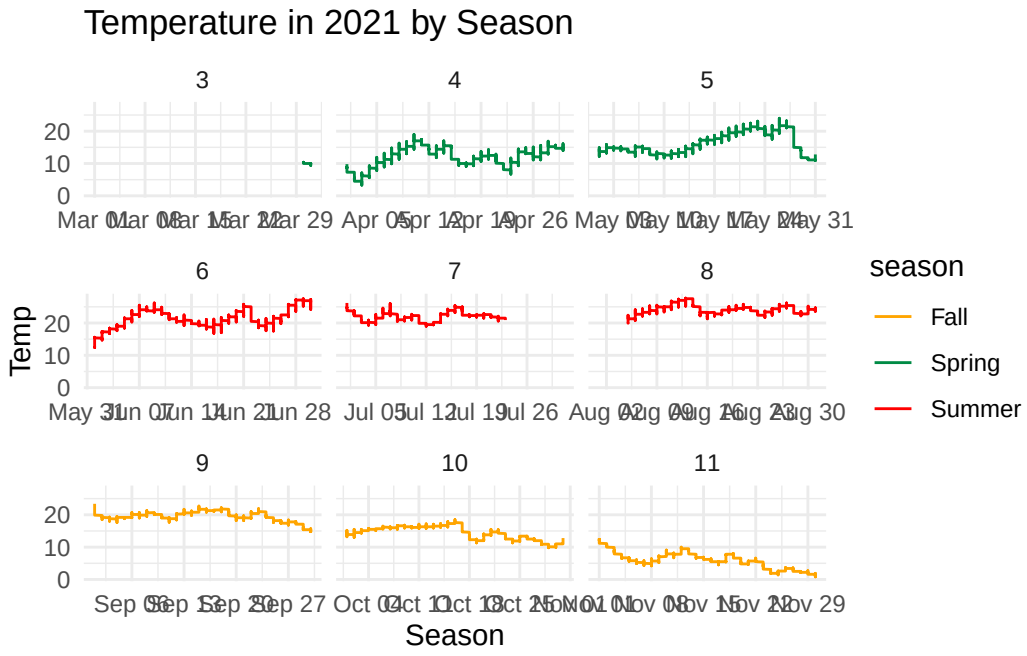


This plot describes further detail on temperature, displaying trends by month in order to show monthly trends and/or outliers. This plot more specifically shows temperature over time as it separates by Month and colors by season. From this, we can distinctly see that July and August appear to have the most consistent and high temperatures. We can also see that months such as May and all of the Fall months experience more significant fluctuations compared to others. These fluctuations may indicate seasonal effects so it may be helpful to look at other variables during these periods.

```
#copy for 2021
ggplot(data = no_winter_2021, aes(x = date_clean, y = Temp, color = season))+
  geom_line()+
  facet_wrap(~month, scales = "free_x")+
  theme_minimal()+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Temperature in 2021 by Season",
```

```
       x = "Season",
       y = "Temp")
```

```
Warning: Removed 2841 rows containing missing values or values outside the scale range
(`geom_line()`).
```

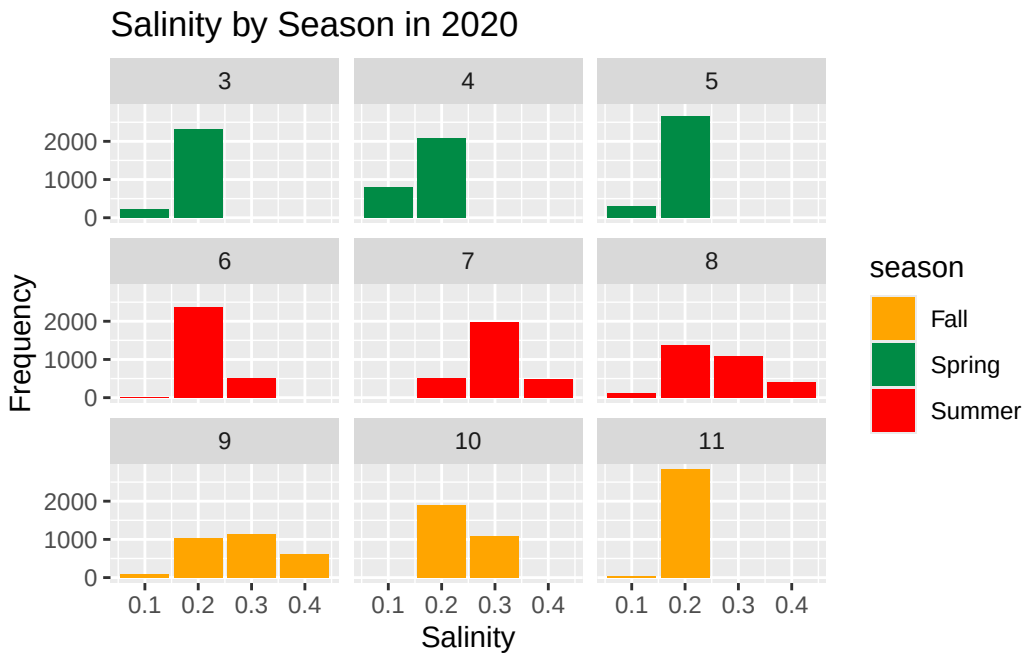


Looks like the month of March had majority of missing values and also the later end of July for 2021

Salinity by Season and Month

```
ggplot(data = no_winter_2020, aes(x = Sal, fill = season, group = season))+
  geom_bar()+
  facet_wrap(~month)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Salinity by Season in 2020",
       x = "Salinity",
       y = "Frequency")
```

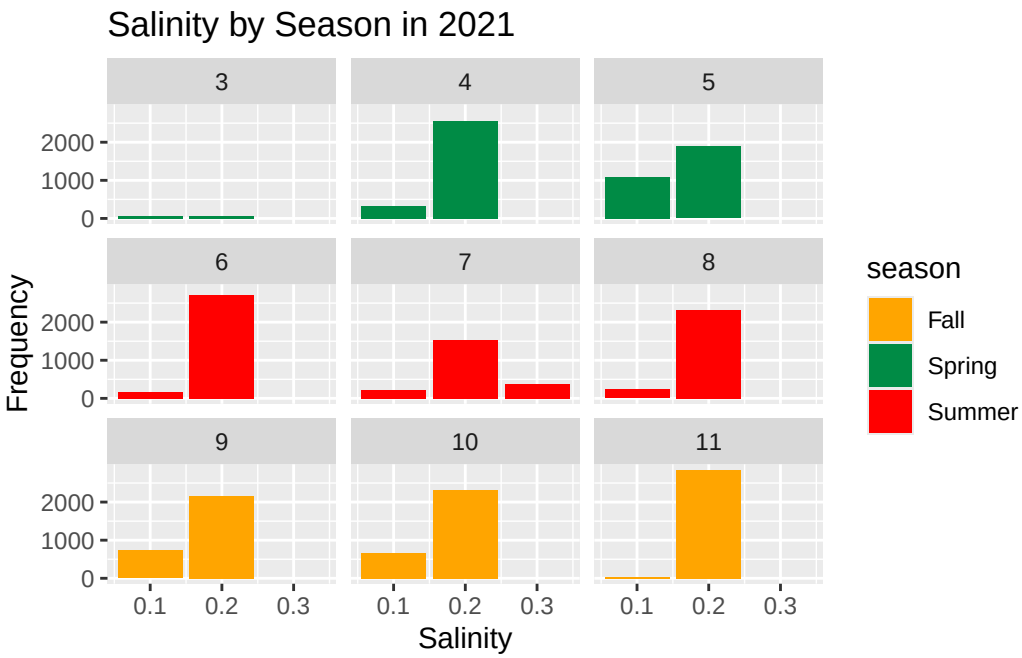
```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_count()`).
```



This visualization creates a bar plot in order visualize the frequency of salinity levels across both seasons and months. We can use this in order to observe seasonal and monthly measures of salinity. For Spring months, salinity levels generally stayed between 0.1 and 0.2. For Summer months, salinity levels varied much more, leveling between 0.1 and 0.4. In addition, variation increased as temperature increased. For fall months, a similar pattern happens except in reverse- variation gradually declines and salinity decreases as it gets cooler. These results indicate that salinity and temperature may be variables worth examining in conjunction.

```
#copy for 2021
ggplot(data = no_winter_2021, aes(x = Sal, fill = season, group = season))+
  geom_bar()+
  facet_wrap(~month)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Salinity by Season in 2021",
       x = "Salinity",
       y = "Frequency")
```

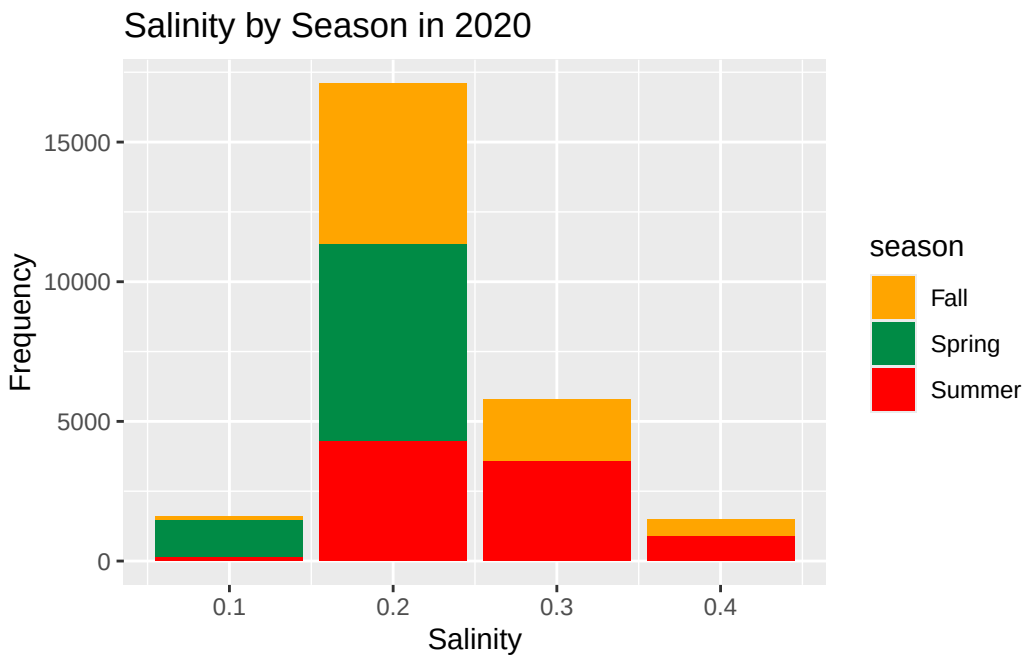
```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_count()`).
```



Salinity by Season

```
library(ggplot2)
ggplot(data = no_winter_2020, aes(x = Sal, fill = season))+
  geom_bar()+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+ # fixed Josie's error
  labs(title = "Salinity by Season in 2020",
       x = "Salinity",
       y = "Frequency")
```

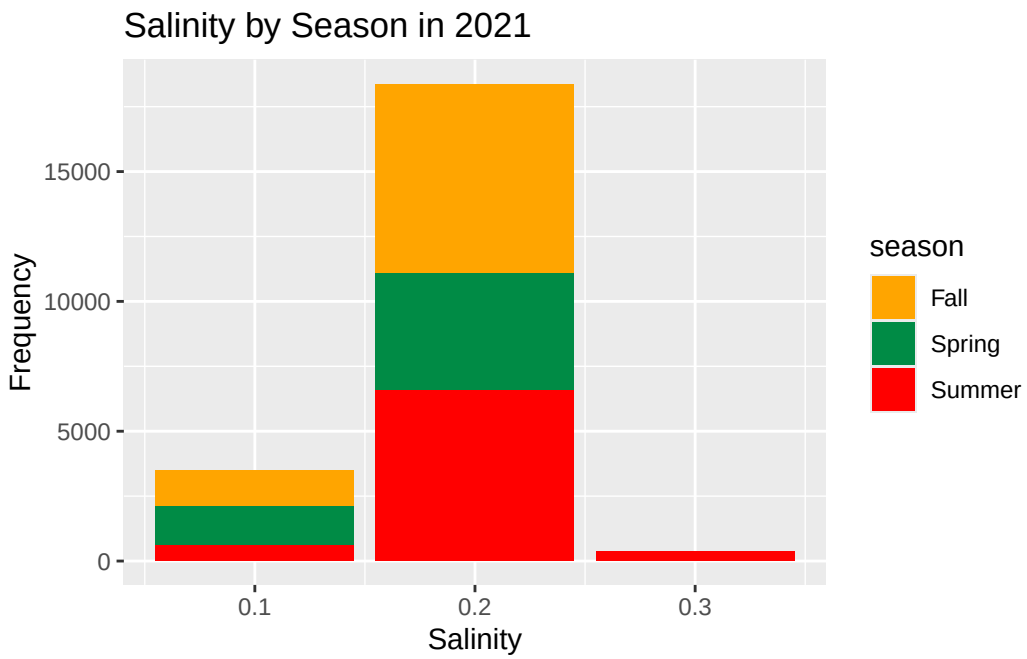
```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_count()`).
```



This visualizes a similar plot, but without facet wrap so all bars can be more closely compared by season within a single plot. Thus, focusing on seasonal change rather than individual monthly change. Based on the plot, we can see more immediately the ranges of salinity for each month. Additionally, we can se that 0.2 level of Salinity is the most frequent and 0.4 is the least common. It may be useful to extract Winter from plots which arent faceted like this because it can impact our interpretation.

```
#copy for 2021
library(ggplot2)
ggplot(data = no_winter_2021, aes(x = Sal, fill = season))+
  geom_bar()+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Salinity by Season in 2021",
       x = "Salinity",
       y = "Frequency")
```

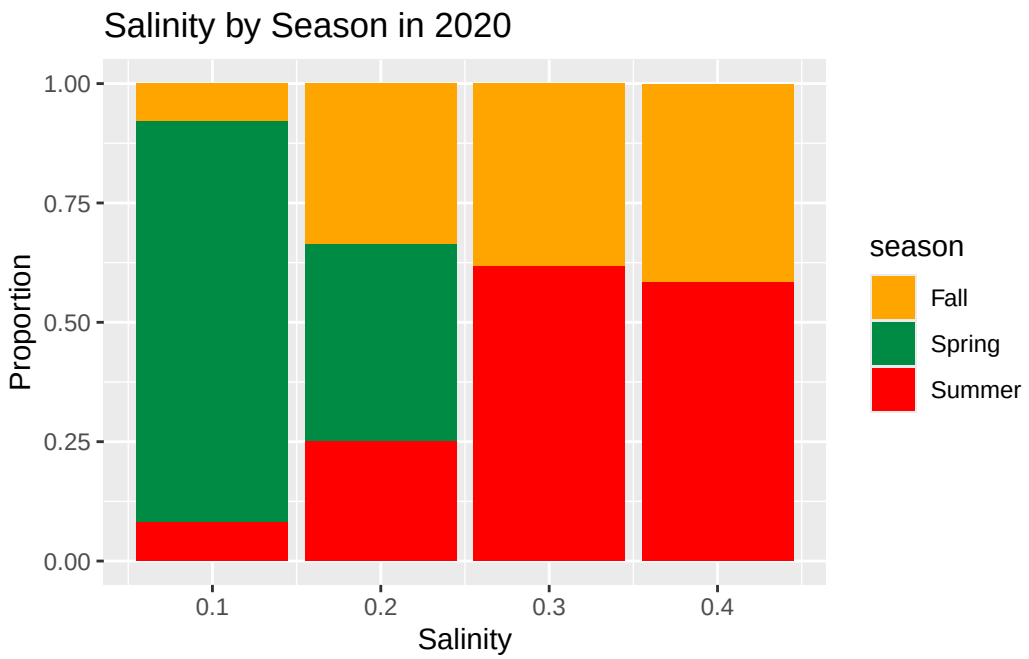
```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_count()`).
```



Proportional Salinity by Season

```
ggplot(data = no_winter_2020, aes(x = Sal, fill = season))+ geom_bar(position = "fill")+
  ylab("Proportion")+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+

  labs(title = "Salinity by Season in 2020",
       x = "Salinity",
       y = "Proportion")
```

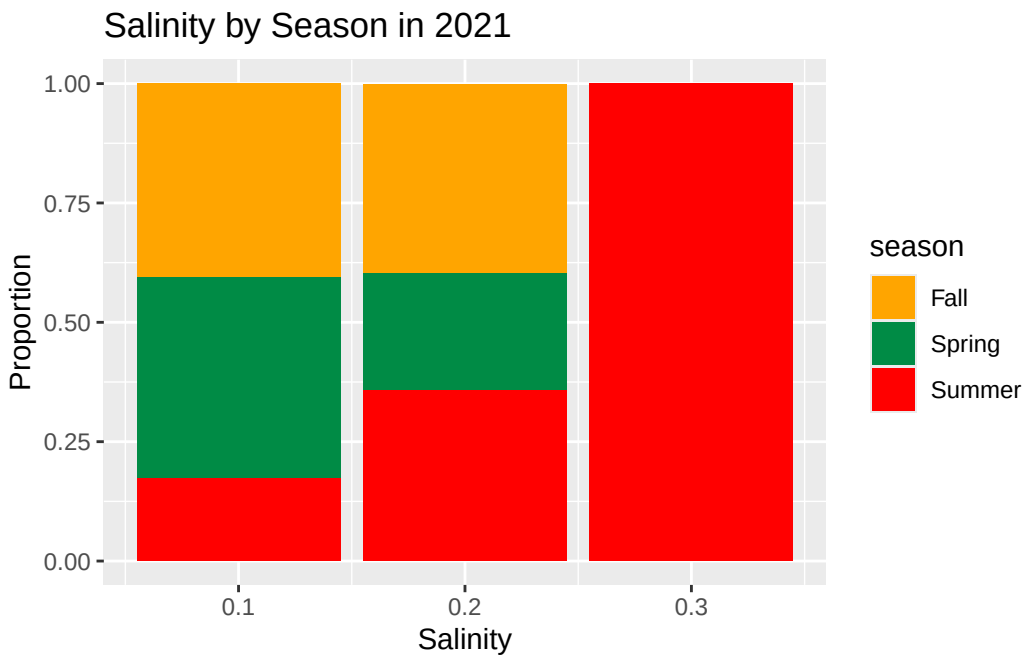
```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_count()`).
```



This code chunk creates a plot using the same information from prior plot, but considering proportionality for Salinity.

```
#copy for 2021
ggplot(data = no_winter_2021, aes(x = Sal, fill = season))+ geom_bar(position = "fill")+
  ylab("Proportion")+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Salinity by Season in 2021",
       x = "Salinity",
       y = "Proportion")
```

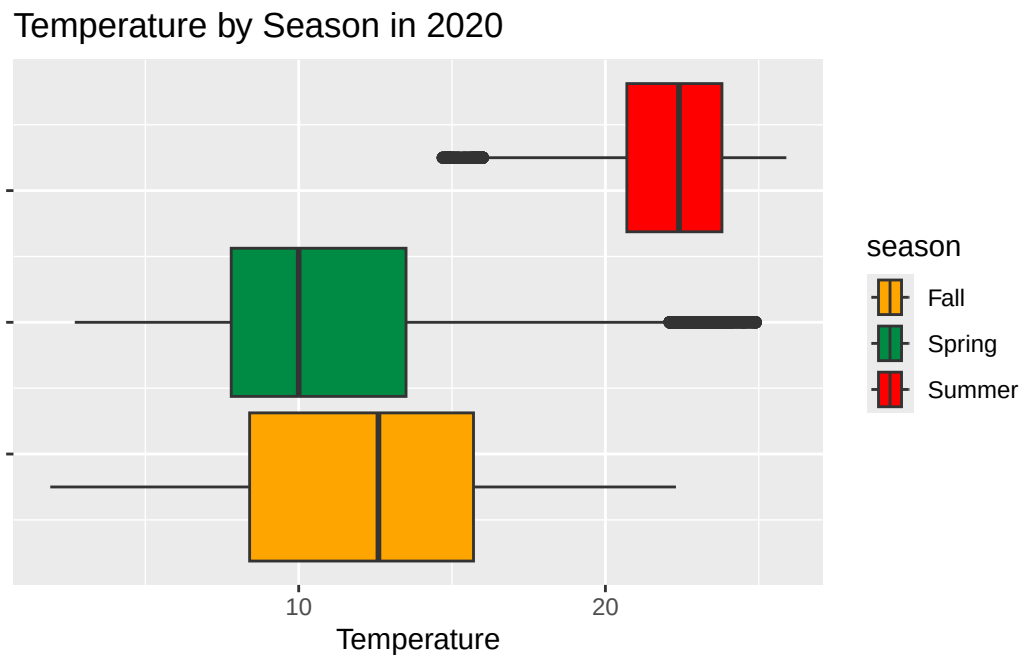
```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_count()`).
```



Temperature by Season

```
ggplot(data = no_winter_2020, aes(x = Temp, fill = season))+
  geom_boxplot()+
  theme(axis.text.y = element_blank())+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Temperature by Season in 2020",
       x = "Temperature")
```

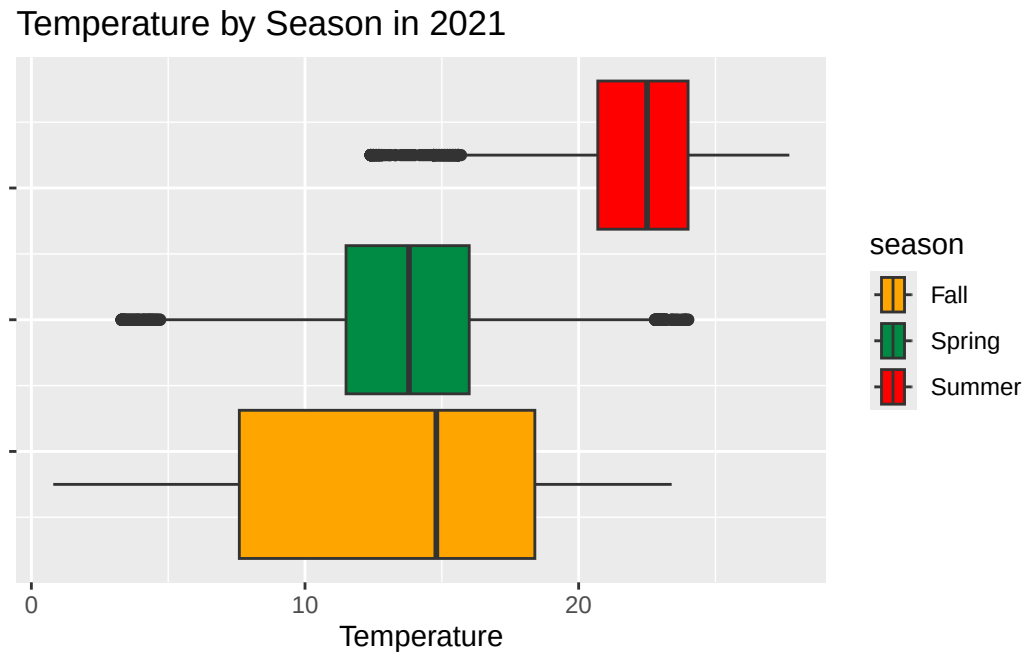
```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



This plot shows the distribution of temperature by season using boxplots in order to compare temperature distributions by season. This plot is pretty effective in showing the distributions of temperature by season. We can especially see in this plot that Spring and Summer have outliers. However, Summer has the least varied spread in temperature. However, Spring has both outliers and a more varied spread. Fall has the most variation, though no serious outliers.This plot shows the distribution of temperature by season using boxplots in order to compare temperature distributions by season. This plot is pretty effective in showing the distributions of temperature by season. We can especially see in this plot that Spring and Summer have outliers. However, Summer has the least varied spread in temperature. However, Spring has both outliers and a more varied spread. Fall has the most variation, though no serious outliers.

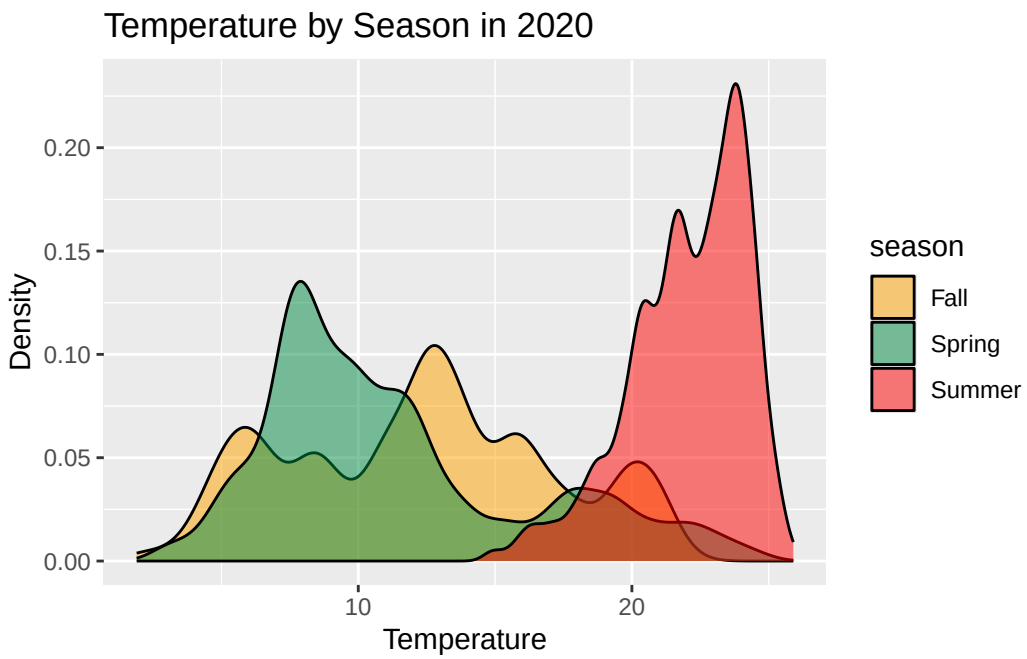
```
#copy for 2021
ggplot(data = no_winter_2021, aes(x = Temp, fill = season))+
  geom_boxplot()+
  theme(axis.text.y = element_blank())+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Temperature by Season in 2021",
       x = "Temperature")
```

```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```




```
ggplot(data = no_winter_2020, aes(x = Temp, fill = season))+
  geom_density(alpha = 0.5)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Temperature by Season in 2020",
       x = "Temperature",
       y = "Density")
```

```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_density()`).
```

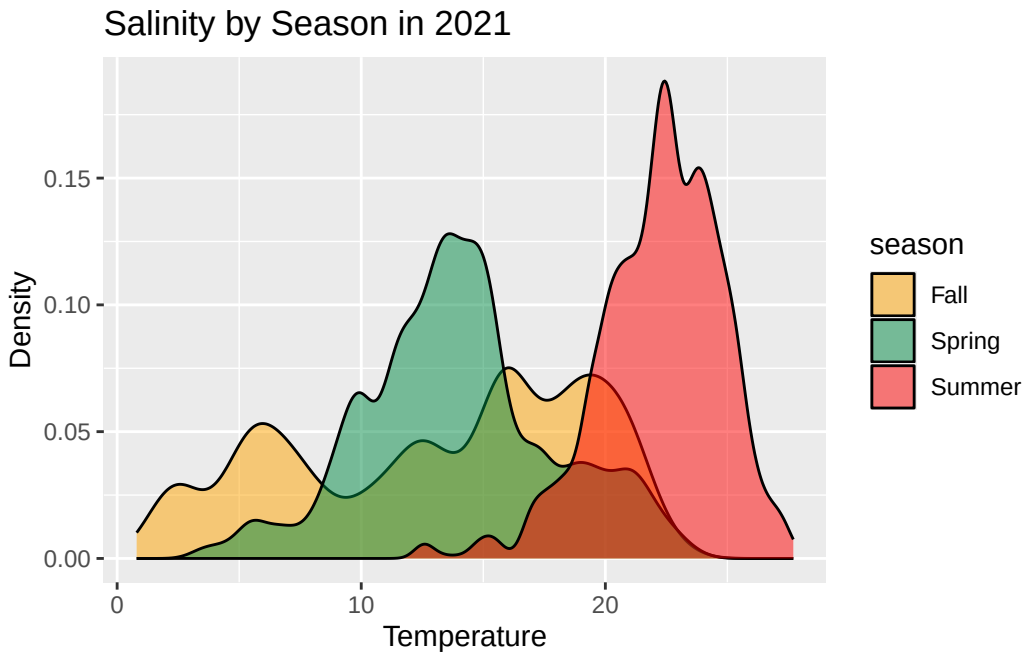


This plot uses the same data except for a density plot with slight transparency to view distributions in their full shape rather than in summary using boxplot (prior plot). This plot shows a full spread of the temperature patterns by season, so we can see more small nuances of where exactly are the peaks and valleys of the variable. Based on the plot, we can immediately recognize that temperature tends to be more consistent in the Summer, though we see a pretty uneven distribution. The spread of Fall temperatures is particularly interesting as it has a with a fairly balanced distribution which is also widely spread. Spring temperatures are also very spread out, although the distribution of temperatures is far more skewed. This plot in particular shows how the three seasons of focus follow temperature distribution patterns in order: Spring has a right skew with a very wide range, Summer with a strong left skew and narrow range, and then Fall shows a wide range and an even spread. Therefore, seasonality has a very strong effect on temperature variability.

```
#copy for 2021
ggplot(data = no_winter_2021, aes(x = Temp, fill = season))+
  geom_density(alpha = 0.5)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Salinity by Season in 2021",
       x = "Temperature",
```

```
       y = "Density")
```

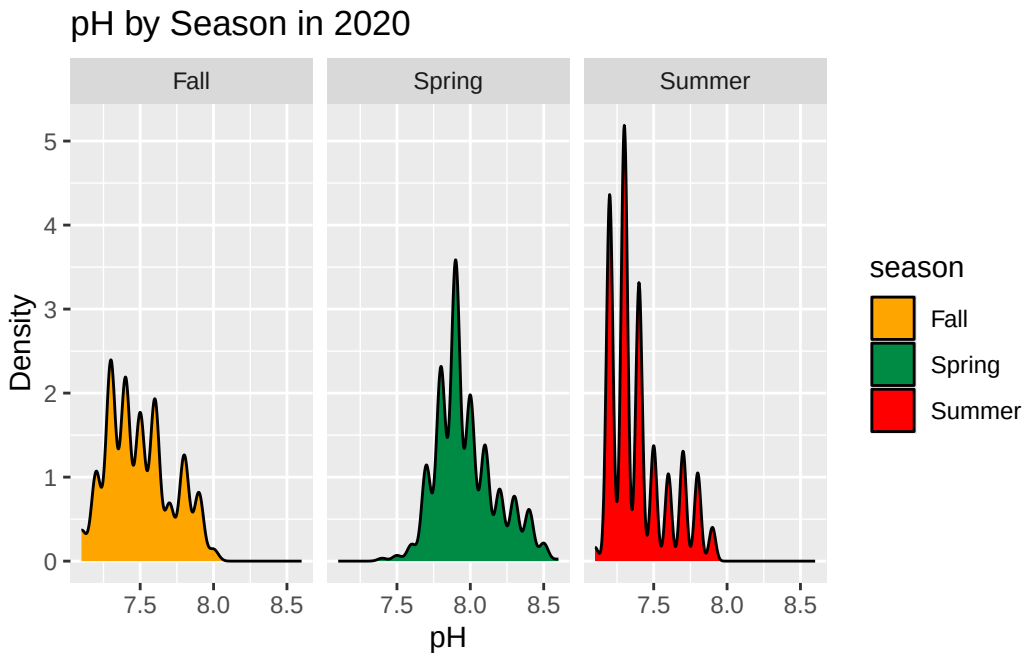
```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_density()`).
```



PH by Season

```
ggplot(data = no_winter_2020, aes(x = pH, fill =season)) +
  geom_density()+
  facet_wrap(~season)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "pH by Season in 2020",
       x = "pH",
       y = "Density")
```

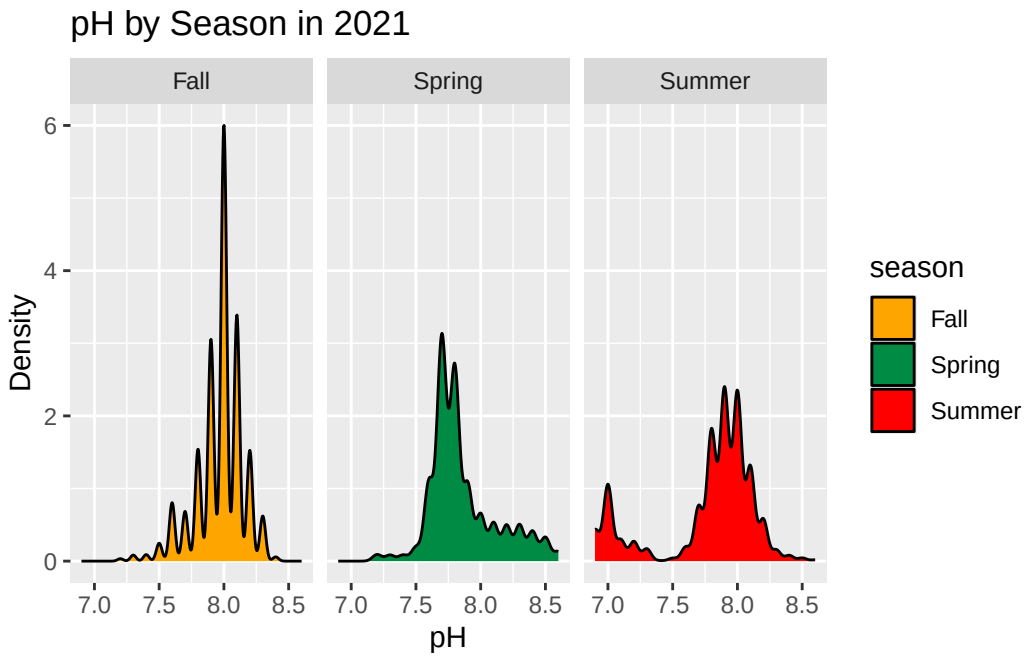
```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_density()`).
```



This plot explores the distribution of pH levels across seasons in order to identify seasonal trends in pH levels. This plot shows us significant detail into the overall frequencies of pH levels by season. Interestingly, for Summer we can see significant fluctuations and a right skew in the pH level. For Spring, there is a right skew as well, though not as strong. And for Fall, we can see that there is the most consistency in pH levels. Therefore, seasonality also has a strong effect on the size and the spread of pH levels.

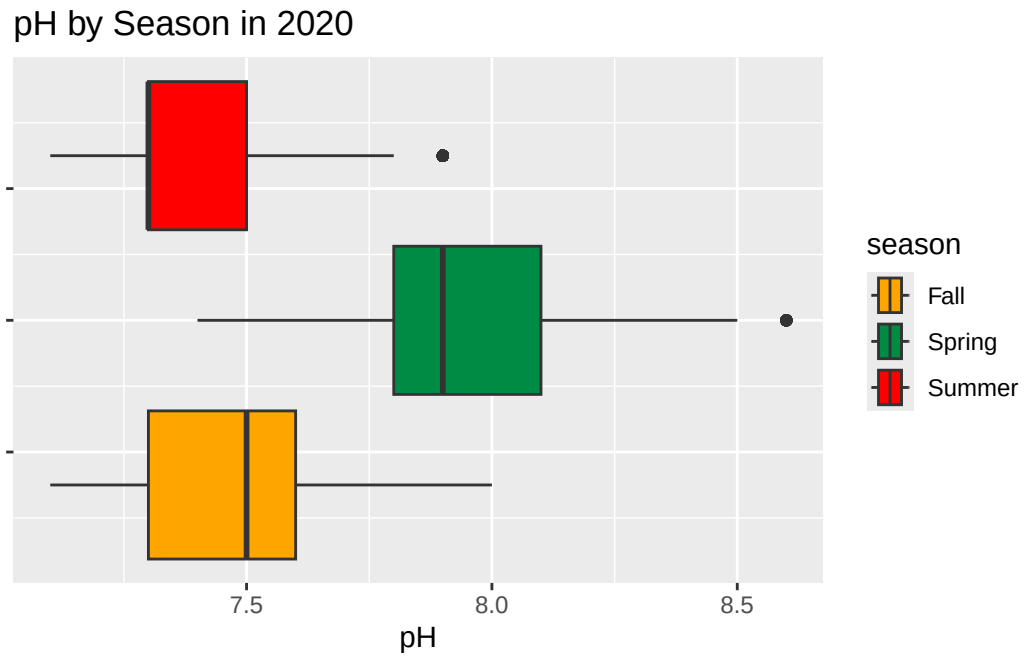
```
#copy for 2021
ggplot(data = no_winter_2021, aes(x = pH, fill =season)) +
  geom_density()+
  facet_wrap(~season)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "pH by Season in 2021",
       x = "pH",
       y = "Density")
```

```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_density()`).
```




```
ggplot(data = no_winter_2020, aes(x = pH, fill = season))+ geom_boxplot()+
  theme(axis.text.y = element_blank())+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "pH by Season in 2020",
       x = "pH")
```

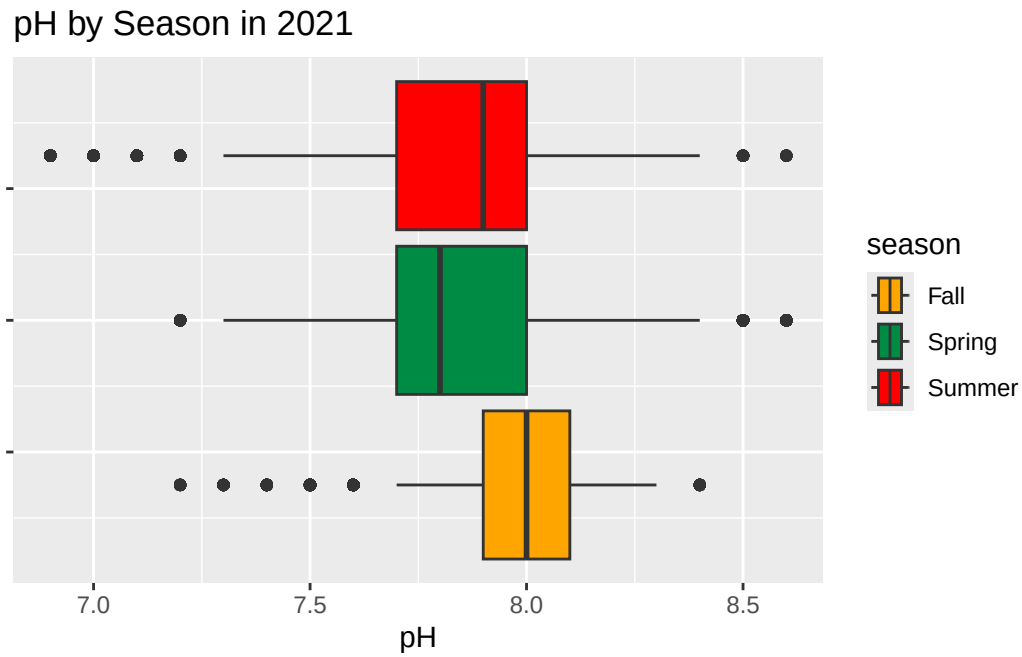
```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



This plot shows the same variables as the plot before except using a boxplot to see summary distributions in pH by season. Based on the plot, pH ranges for Fall and Summer are similar, though we can once again see that Summer has a very strong skew, shown through the extreme median line. And the boxplots for Fall and Spring also reinforce trends from the density plots. With the boxplots, we can immediately recognize that the other considered months (Fall and Spring) have even distributions. The boxplot in comparison to the faceted density plot can more easily show the exact ranges of each season and exactly how skewed they are in constrast to one another because they all use the exact same axes. The boxplot also visualizes that Spring and Summer both have an extreme outlier.

```
#copy for 2021
ggplot(data = no_winter_2021, aes(x = pH, fill = season))+ geom_boxplot()+
  theme(axis.text.y = element_blank())+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "pH by Season in 2021",
       x = "pH")
```

```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



Daily Averages for each Variable

```
#This code calculates the daily average of our variables by  season in order to look at trend
daily_2020 <- no_winter_2020 %>%
  group_by(date_clean, season)%>%
  summarise(
    meanpH = mean(pH, na.rm = TRUE),
    meanSal = mean(Sal, na.rm = TRUE),
    meanTemp = mean(Temp, na.rm = TRUE),
    meanDepth = mean(Depth, na.rm = TRUE),
    meanSpCond = mean(SpCond, na.rm = TRUE),
    meanTurb = mean(Turb, na.rm = TRUE),
    meanChlFluor = mean(ChlFluor, na.rm = TRUE),
    meanDO_Pct = mean(DO_Pct, na.rm = TRUE),
    meanDO_mgl = mean(DO_mgl, na.rm = TRUE))
```

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

Here can observe all of the variables over time using time series plots, seasonally.

```
#copy for 2021
#This code calculates the daily average of our variables by  season in order to look at trend
daily_2021 <- no_winter_2021 %>%
  group_by(date_clean, season)%>%
  summarise(
    meanpH = mean(pH, na.rm = TRUE),
    meanSal = mean(Sal, na.rm = TRUE),
    meanTemp = mean(Temp, na.rm = TRUE),
    meanDepth = mean(Depth, na.rm = TRUE),
    meanSpCond = mean(SpCond, na.rm = TRUE),
    meanTurb = mean(Turb, na.rm = TRUE),
    meanChlFluor = mean(ChlFluor, na.rm = TRUE),
    meanDO_Pct = mean(DO_Pct, na.rm = TRUE),
    meanDO_mgl = mean(DO_mgl, na.rm = TRUE))
```

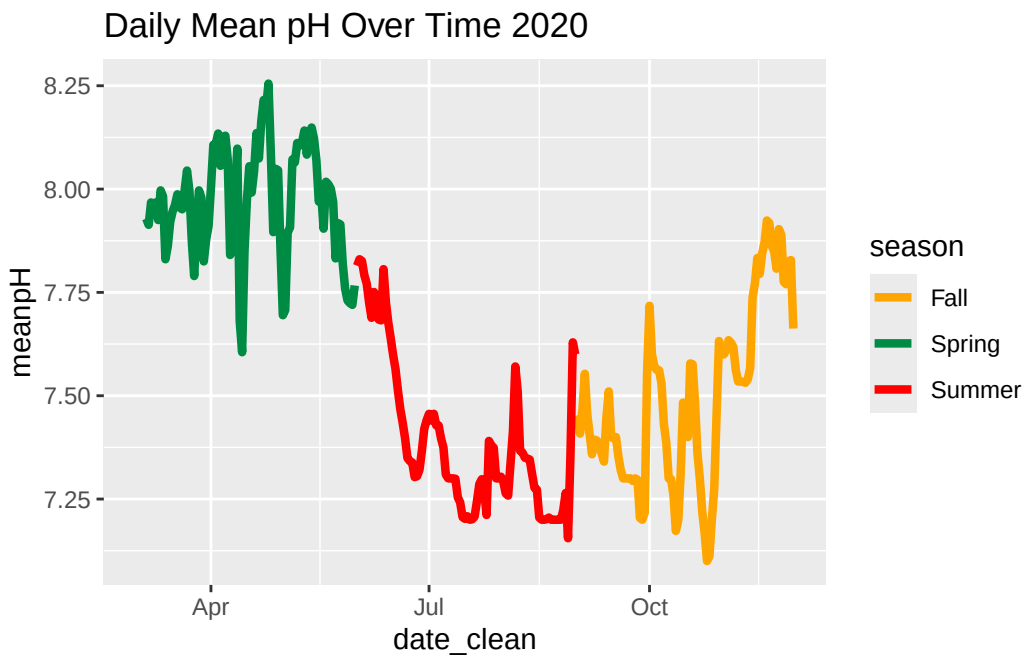
```
`summarise()` has grouped output by 'date_clean'. You can override using the
`.groups` argument.
```

Variable Time Series

```
ggplot(daily_2020, aes(x = date_clean, y = meanpH, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean pH Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

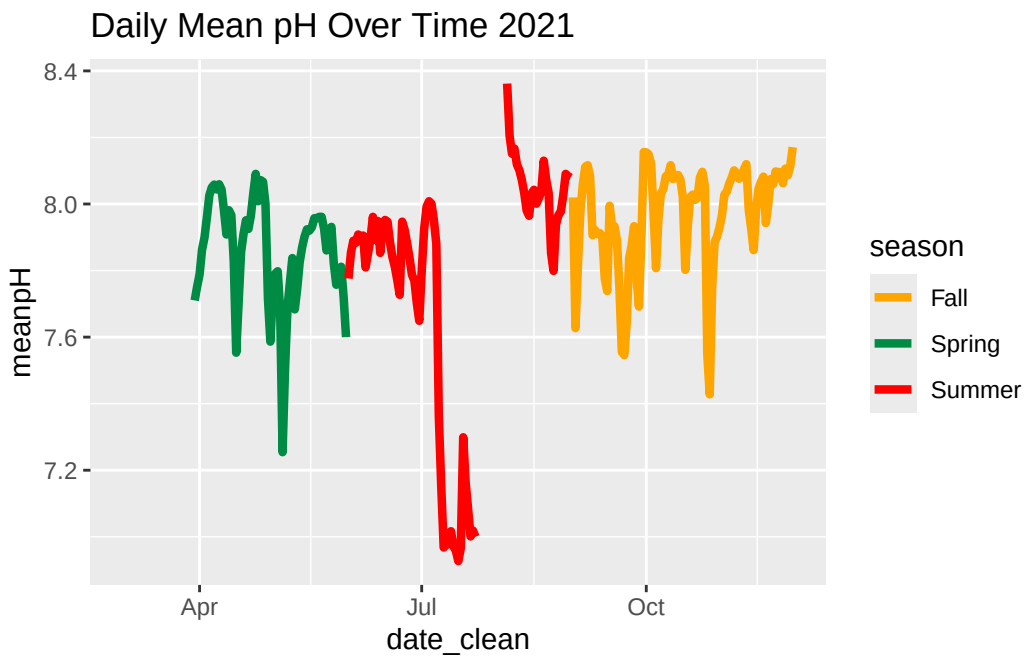
```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



The warning shows that a few rows were missing or outside of the axis limits. This is likely from the make of the dataset which includes a few missing rows.

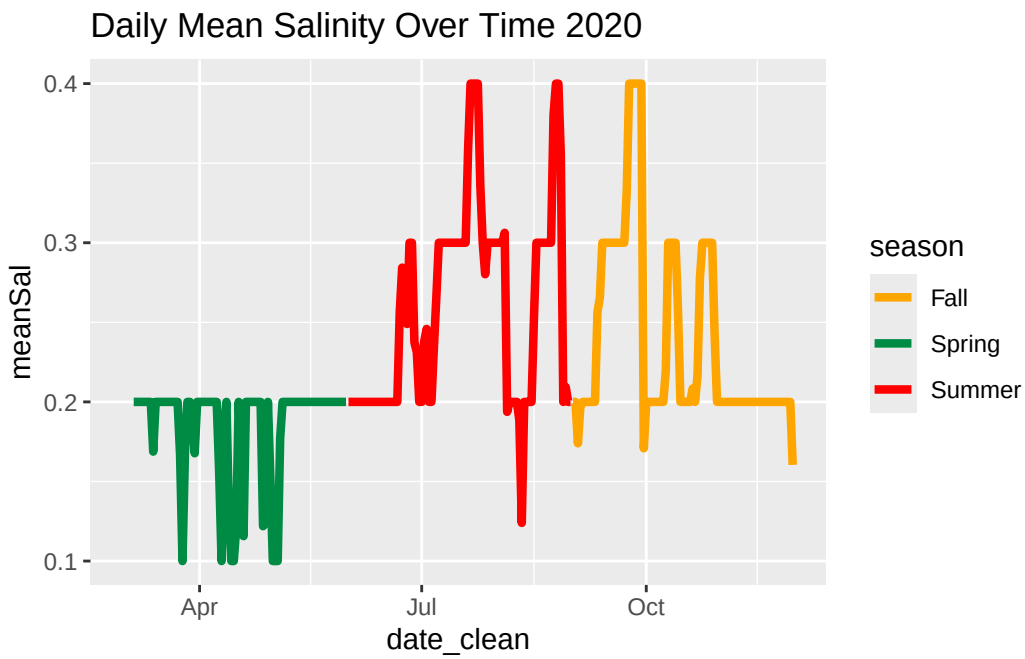
```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanpH, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean pH Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



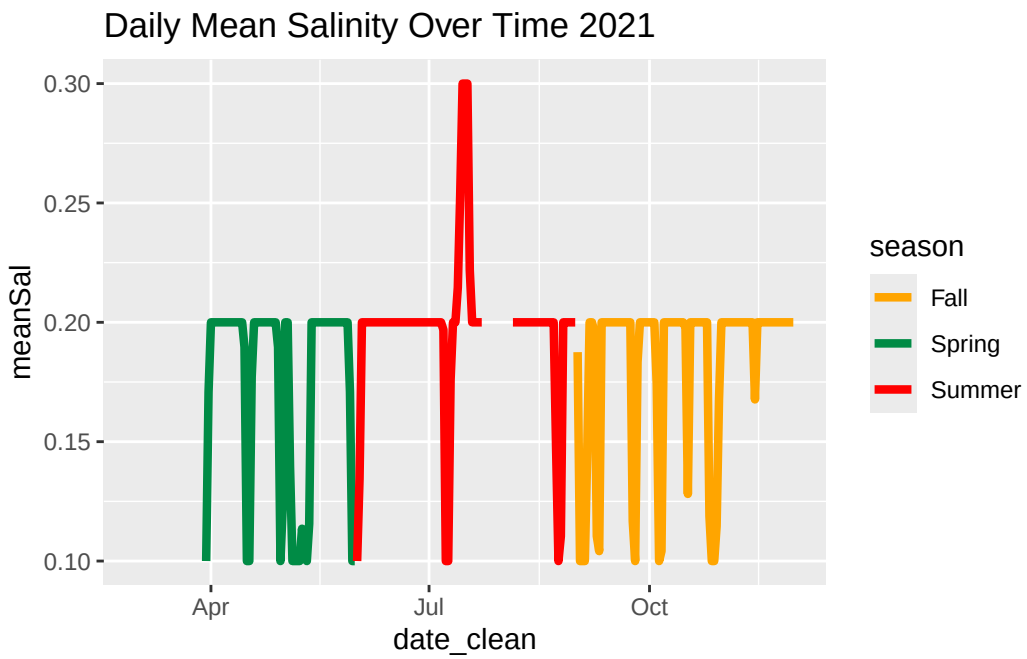

```
ggplot(daily_2020, aes(x = date_clean, y = meanSal, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Salinity Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



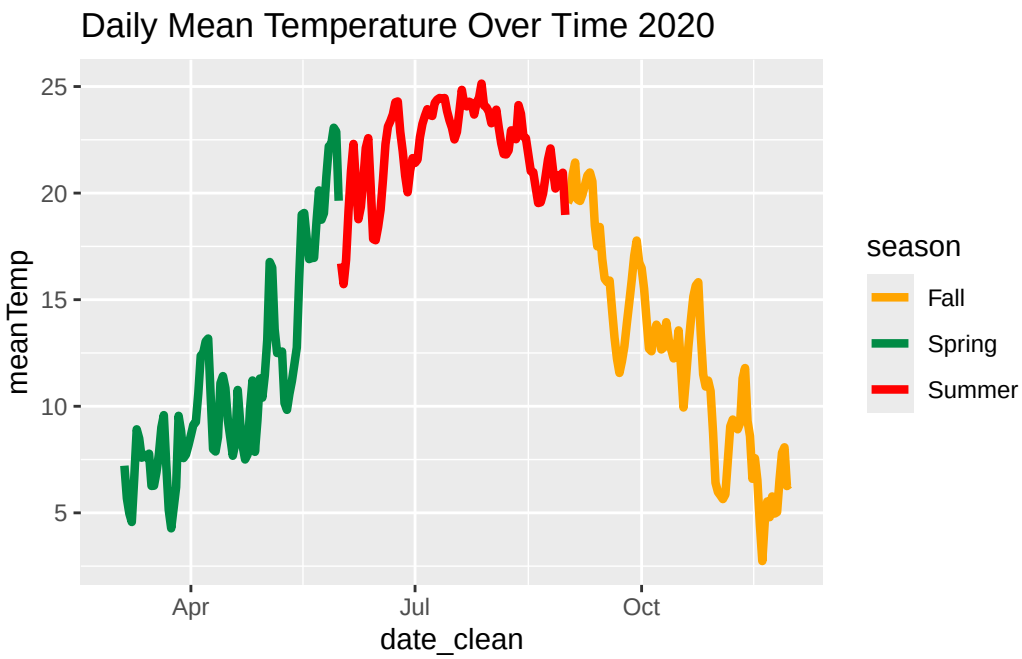

```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanSal, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Salinity Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



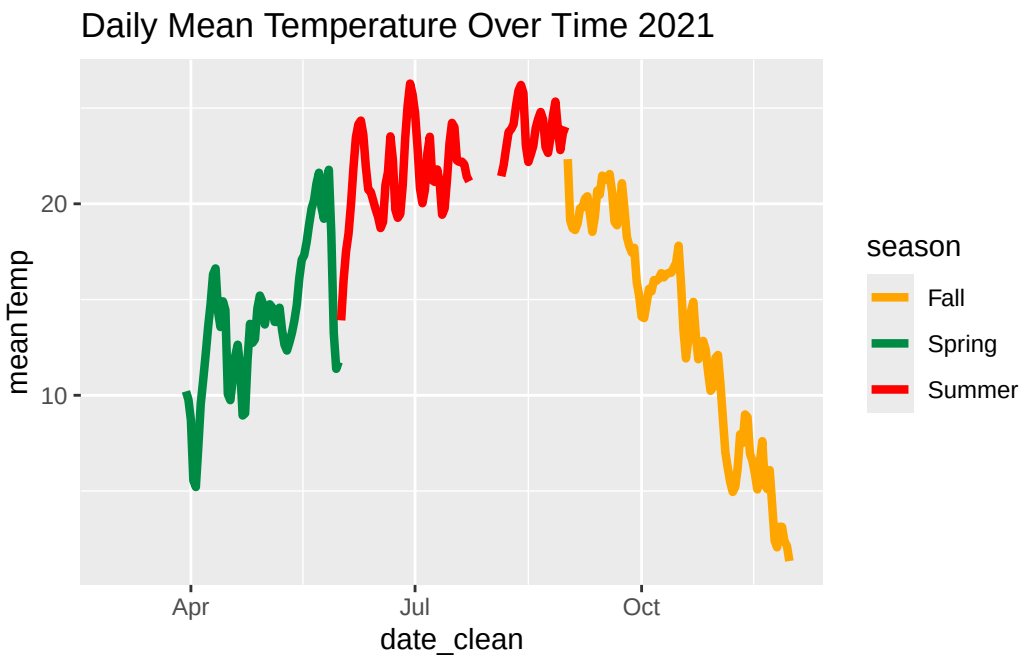

```
ggplot(daily_2020, aes(x = date_clean, y = meanTemp, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Temperature Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



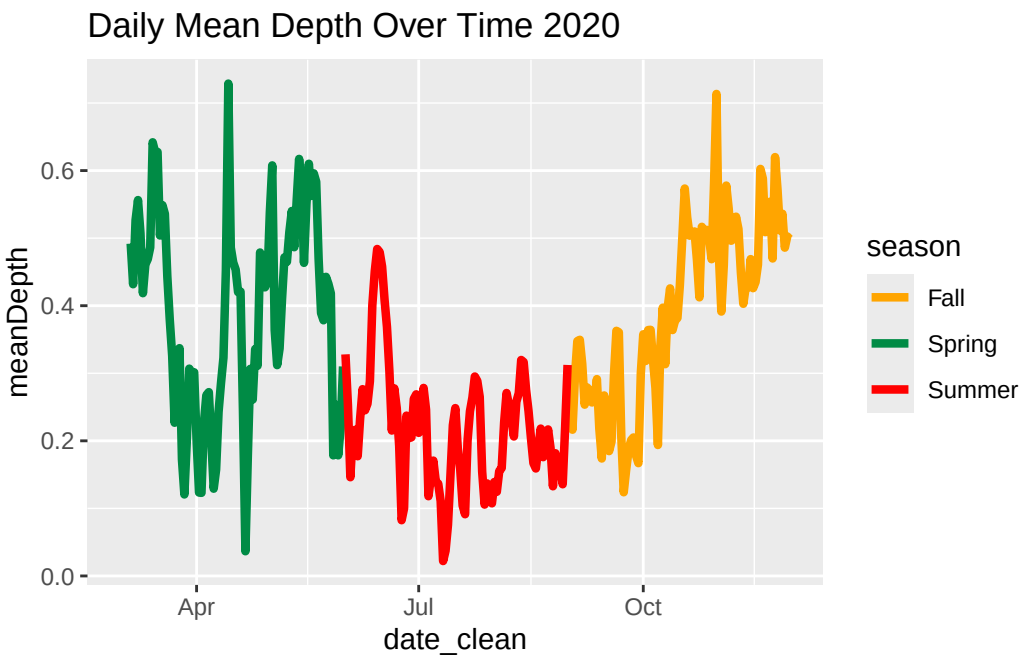

```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanTemp, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Temperature Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



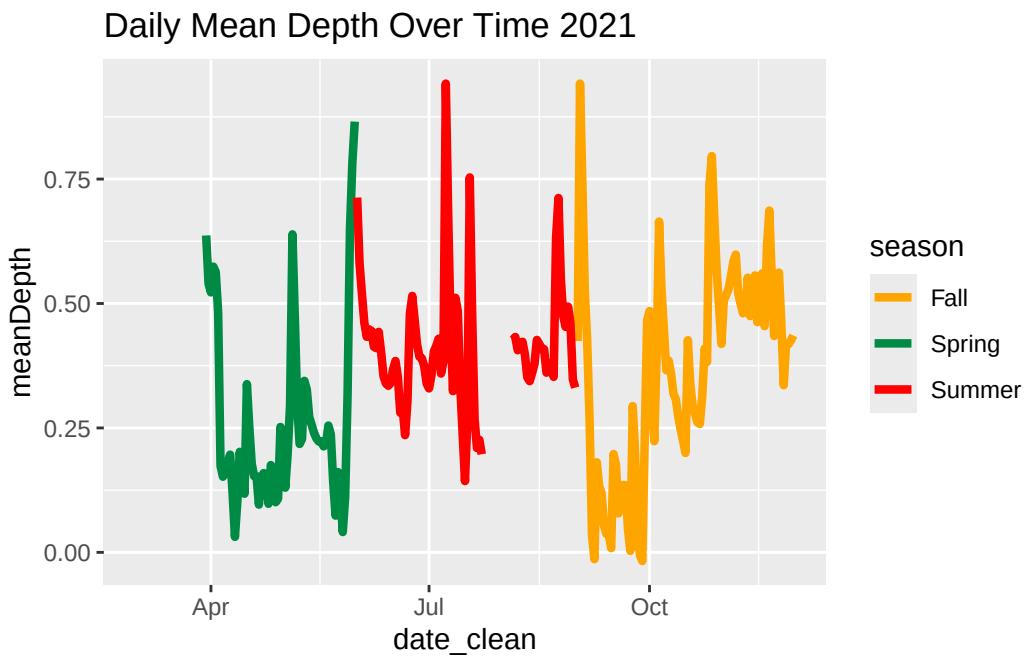

```
ggplot(daily_2020, aes(x = date_clean, y = meanDepth, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Depth Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



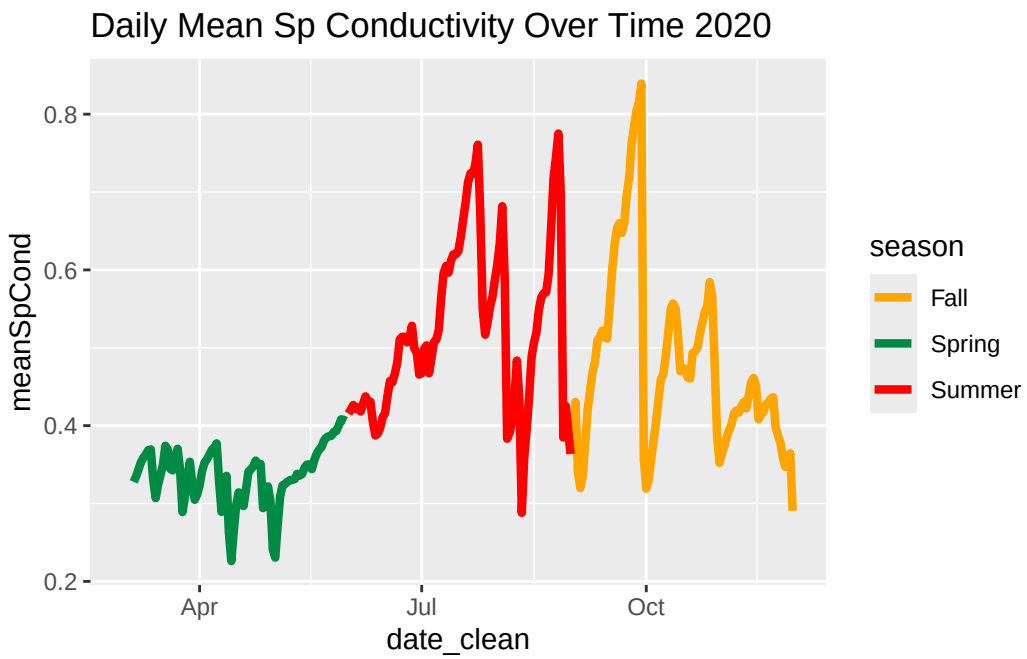

```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanDepth, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Depth Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



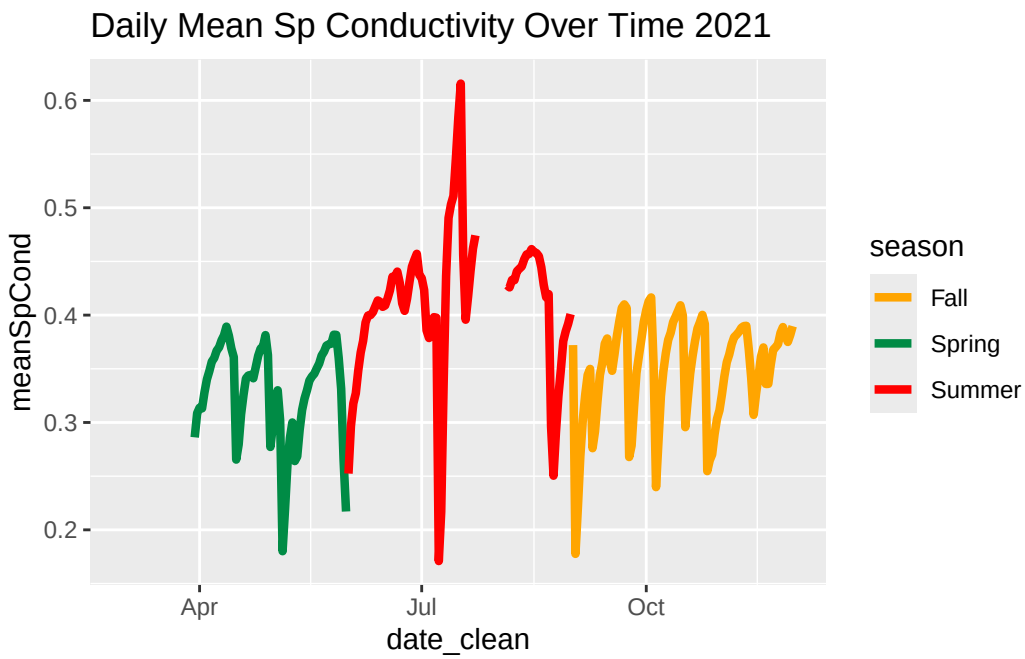

```
ggplot(daily_2020, aes(x = date_clean, y = meanSpCond, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Sp Conductivity Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



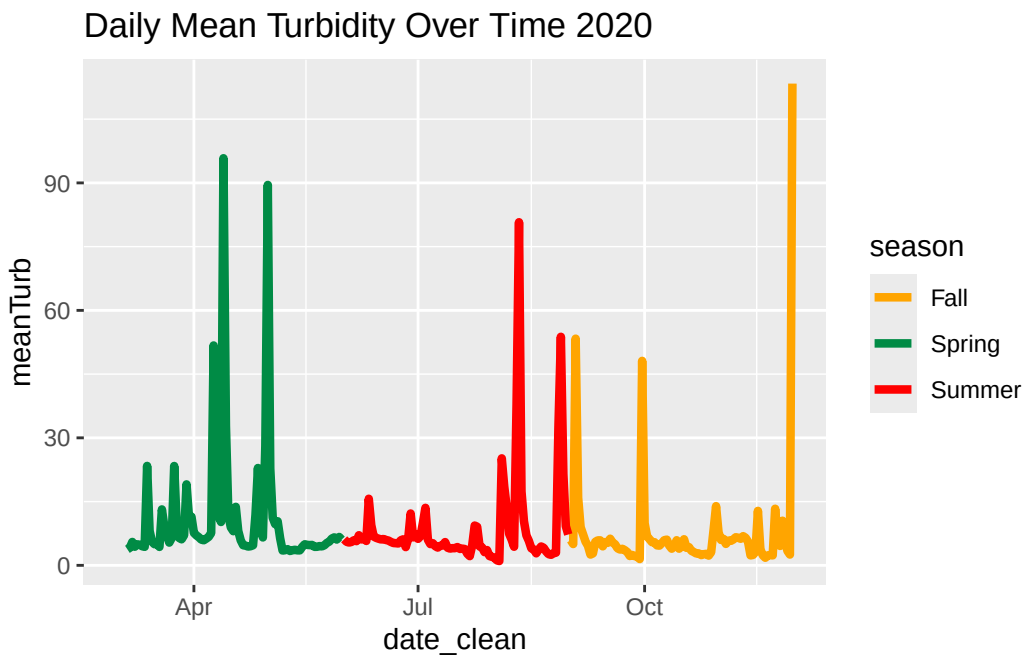

```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanSpCond, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Sp Conductivity Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



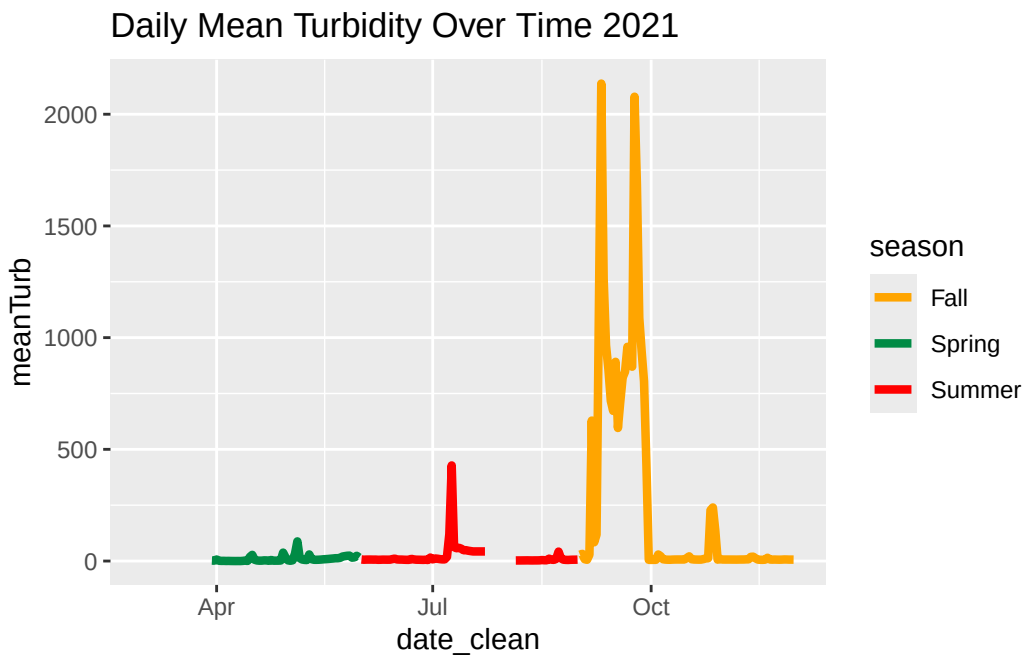

```
ggplot(daily_2020, aes(x = date_clean, y = meanTurb, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Turbidity Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



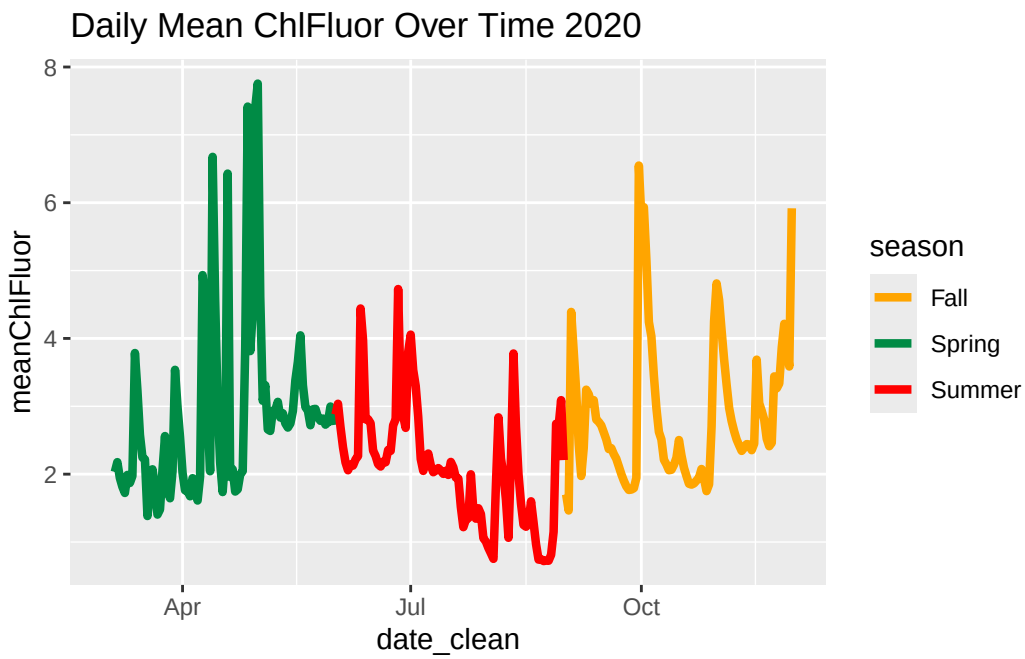

```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanTurb, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Turbidity Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



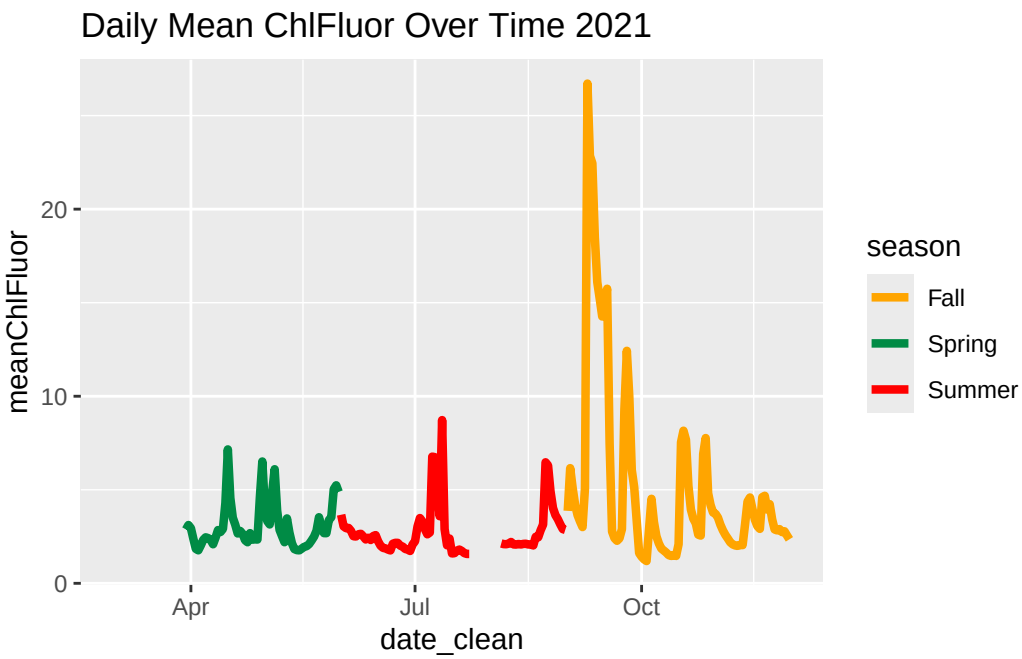

```
ggplot(daily_2020, aes(x = date_clean, y = meanChlFluor, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean ChlFluor Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



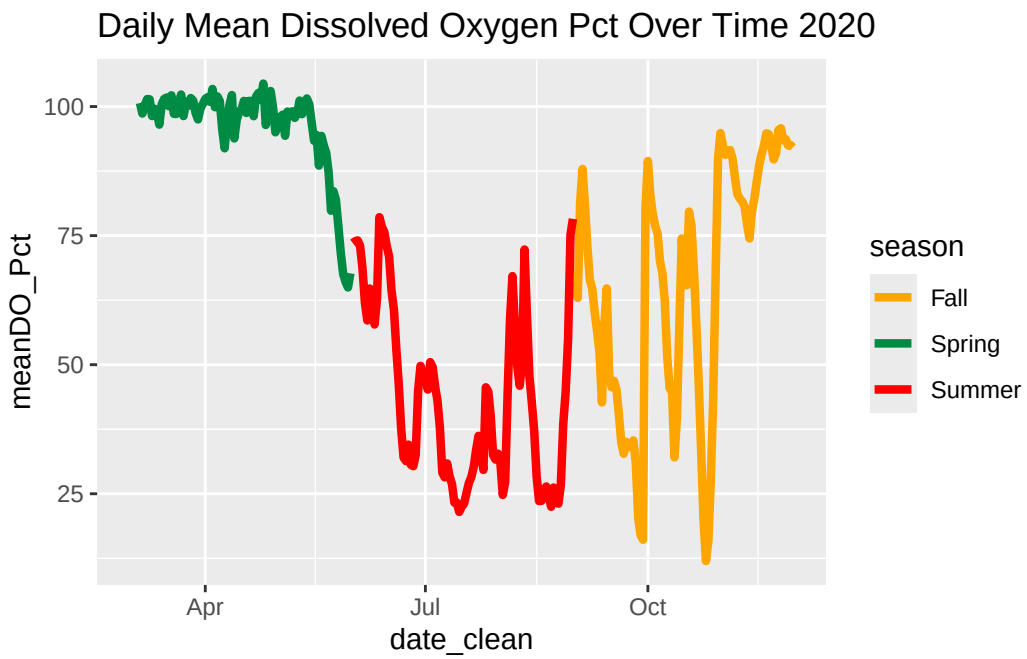

```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanChlFluor, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean ChlFluor Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



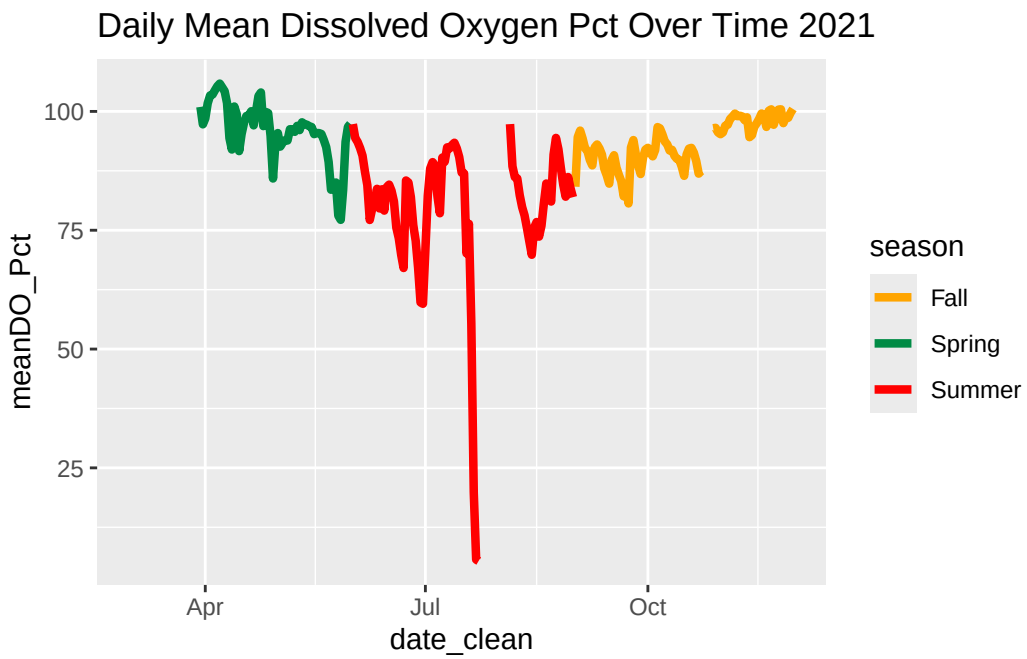

```
ggplot(daily_2020, aes(x = date_clean, y = meanDO_Pct, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Dissolved Oxygen Pct Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```



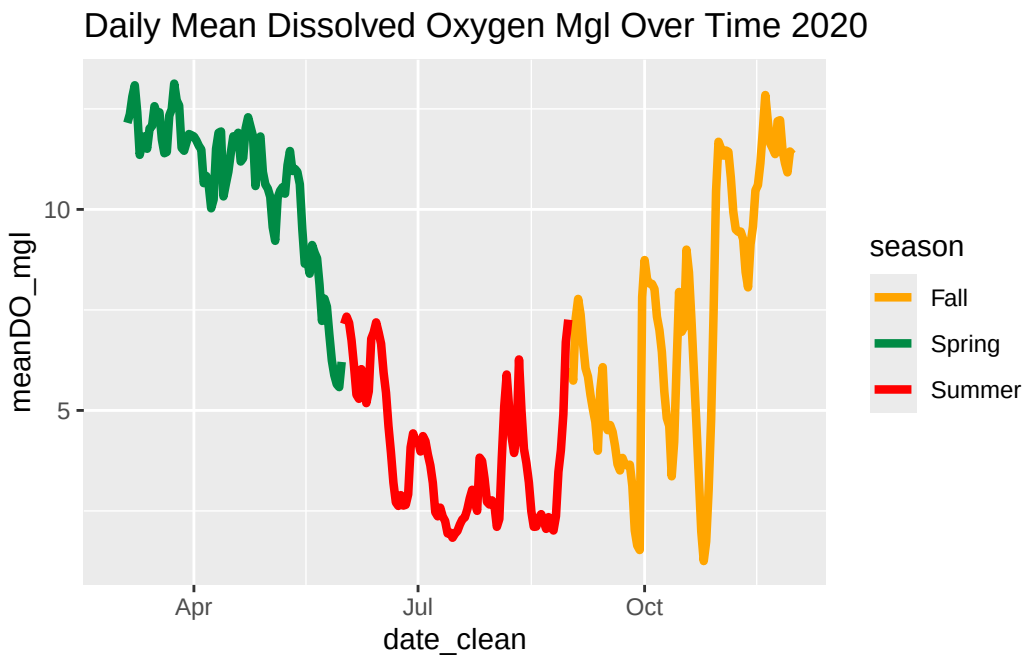

```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanDO_Pct, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Dissolved Oxygen Pct Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```



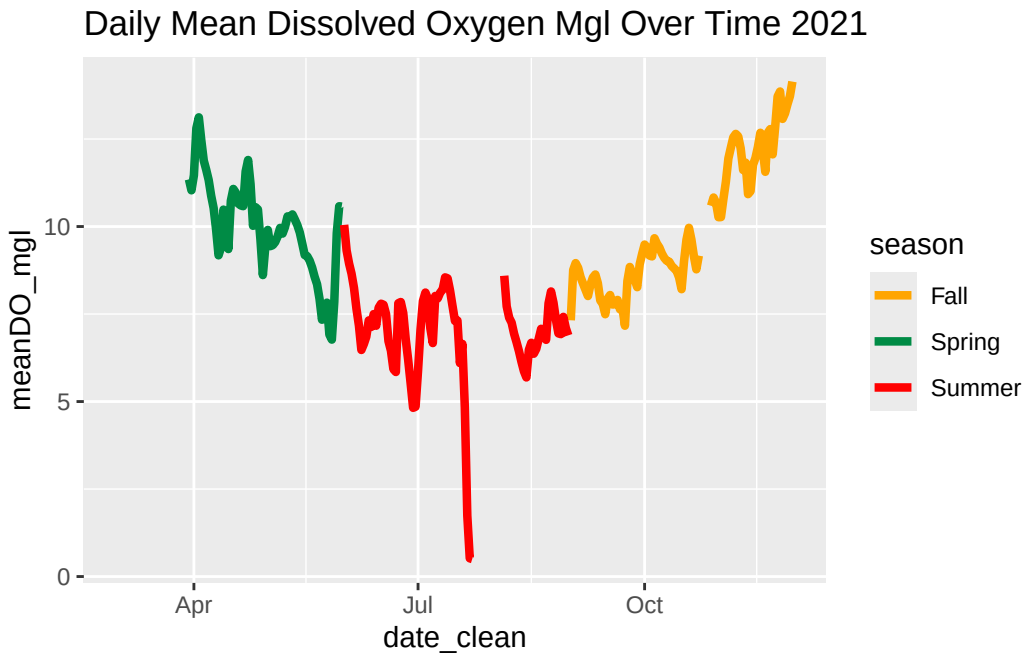

```
ggplot(daily_2020, aes(x = date_clean, y = meanDO_mgl, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Dissolved Oxygen Mgl Over Time 2020")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).
```




```
#copy for 2021
ggplot(daily_2021, aes(x = date_clean, y = meanDO_mgl, color = season))+
  geom_line(size = 1.5)+
  labs(title = "Daily Mean Dissolved Oxygen Mgl Over Time 2021")+
  scale_color_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"))
```

```
Warning: Removed 29 rows containing missing values or values outside the scale range
(`geom_line()`).
```

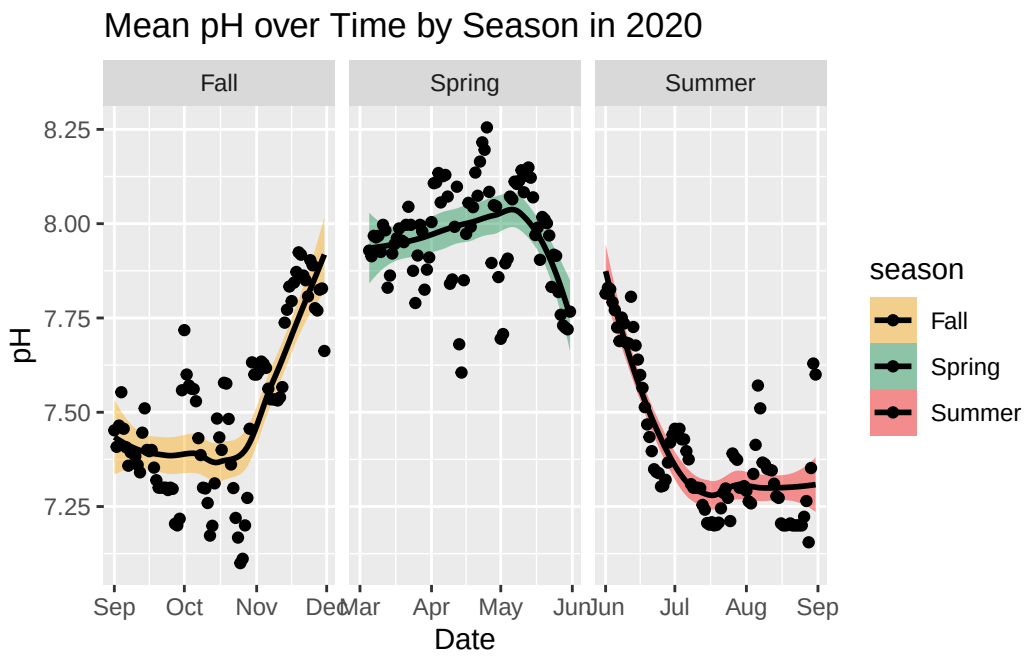



```
ggplot(data = daily_2020, aes(x = date_clean, y = meanpH, fill = season))+
  facet_wrap(~season, scales = "free_x")+
  geom_smooth(method = "loess", color = "black")+ geom_point()+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Mean pH over Time by Season in 2020",
       x = "Date",
       y = "pH")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



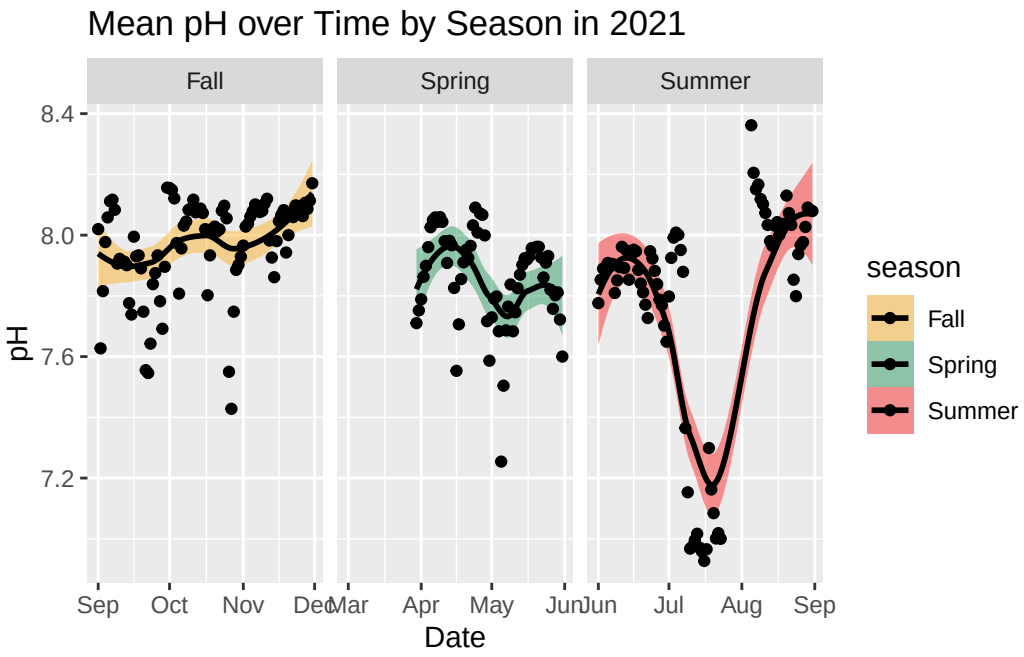
This plot visualizes the daily average pH over time, colored by season, in order to observe seasonal trends in the pH levels. Looking at pH over time, we can see that major shifts of pH by season tend to occur around November, May, and June. The points in this plot paired with the lines are helpful as the lines can capture overall changes in pH during the season, while the points show that there is actually a lot of variation. Though just by looking at the points, it does appear that pH level certainly has a pattern in its peaks and valleys- this is particularly seen in Summer.

```
#copy for 2021
ggplot(data = daily_2021, aes(x = date_clean, y = meanpH, fill = season))+
  facet_wrap(~season, scales = "free_x")+
  geom_smooth(method = "loess", color = "black")+ geom_point()+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Mean pH over Time by Season in 2021",
       x = "Date",
       y = "pH")
```

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

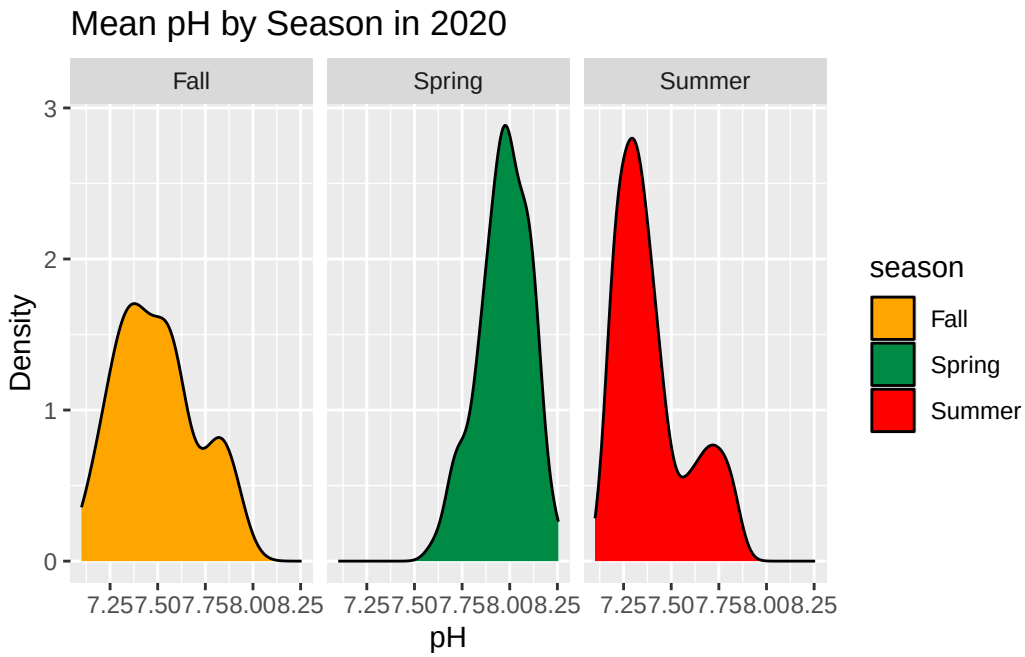
```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```r
ggplot(data = daily_2020, aes(x = meanpH, fill =season)) +
  geom_density()+
  facet_wrap(~season)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Mean pH by Season in 2020",
       x = "pH",
       y = "Density")
```

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_density()`).
```

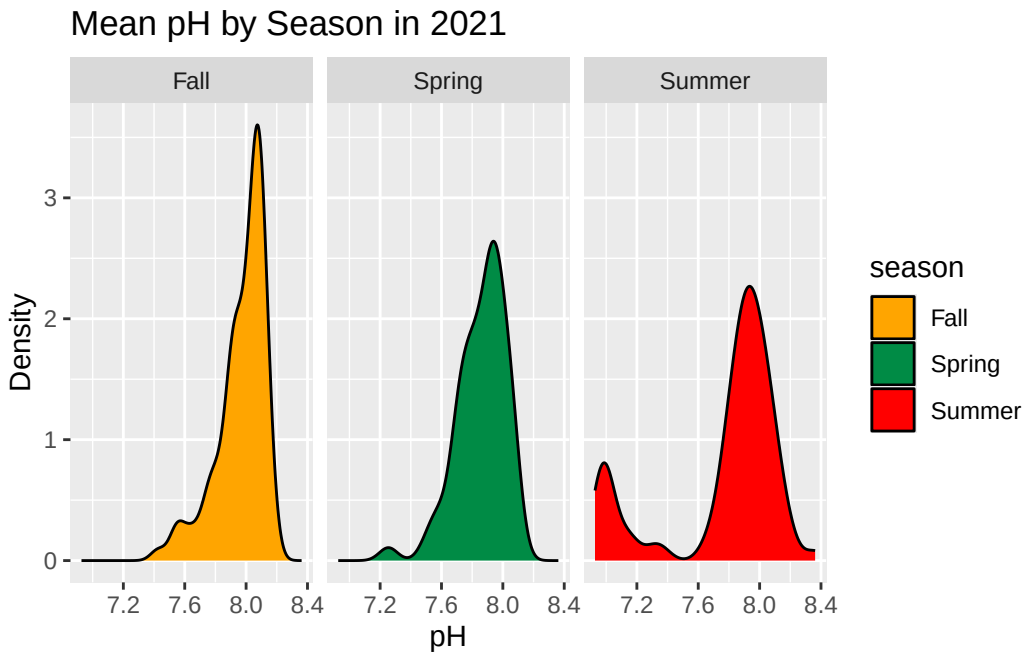


This plot shows the distribution of daily average pH levels for each season to compare pH levels. Using a density plot rather than a scatterplot, we can see more general patterns and the shape of each variable. Based on the plot, we can see smoothed daily averages of pH. With this plot, we lose the spikes and drops in variability, but can more clearly see overall seasonal pH levels. We particularly can more clearly see the ranges where Summer and Spring peak. These findings suggest that pH can significantly fluctuate throughout the day based on other variables. Therefore, we ought to look at other variables which may link to changes in pH.Next: Salinity and temperature, Temperature v. pH, Salinity v. pH.

```
#copy for 2021
ggplot(data = daily_2021, aes(x = meanpH, fill =season)) +
  geom_density()+
  facet_wrap(~season)+
  scale_fill_manual(values = c(
  "Fall"   = "orange",
  "Spring" = "springgreen4",
  "Summer" = "red"
))+
  labs(title = "Mean pH by Season in 2021",
       x = "pH",
       y = "Density")
```

```
Warning: Removed 41 rows containing non-finite outside the scale range
```

(`stat_density()`).



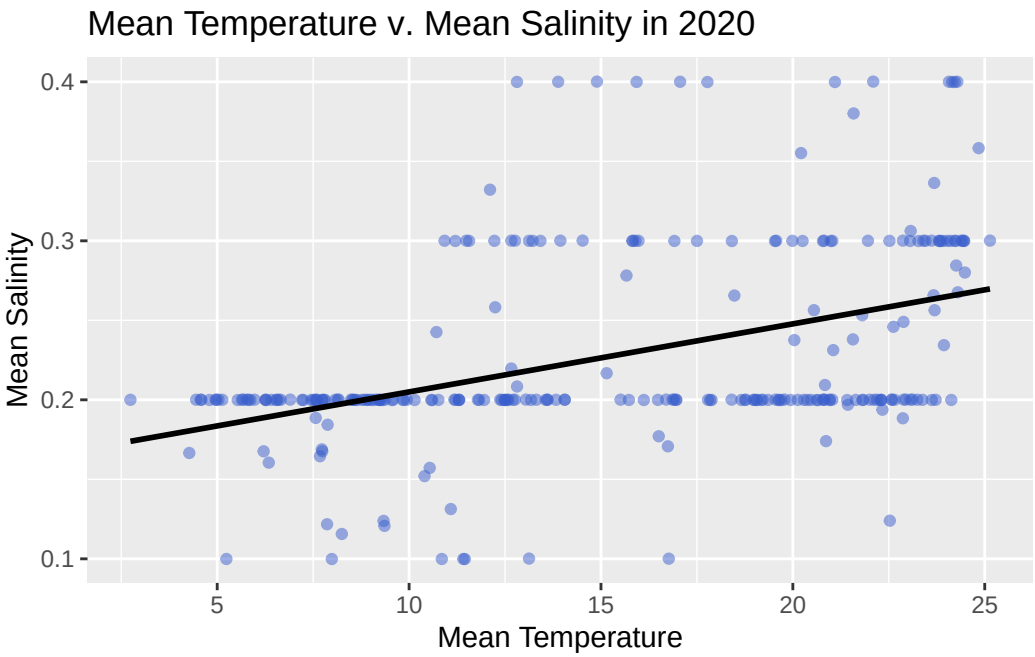
Correlation Plots

```
ggplot(daily_2020, aes(x = meanTemp, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Temperature v. Mean Salinity in 2020",
       x = "Mean Temperature",
       y = "Mean Salinity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



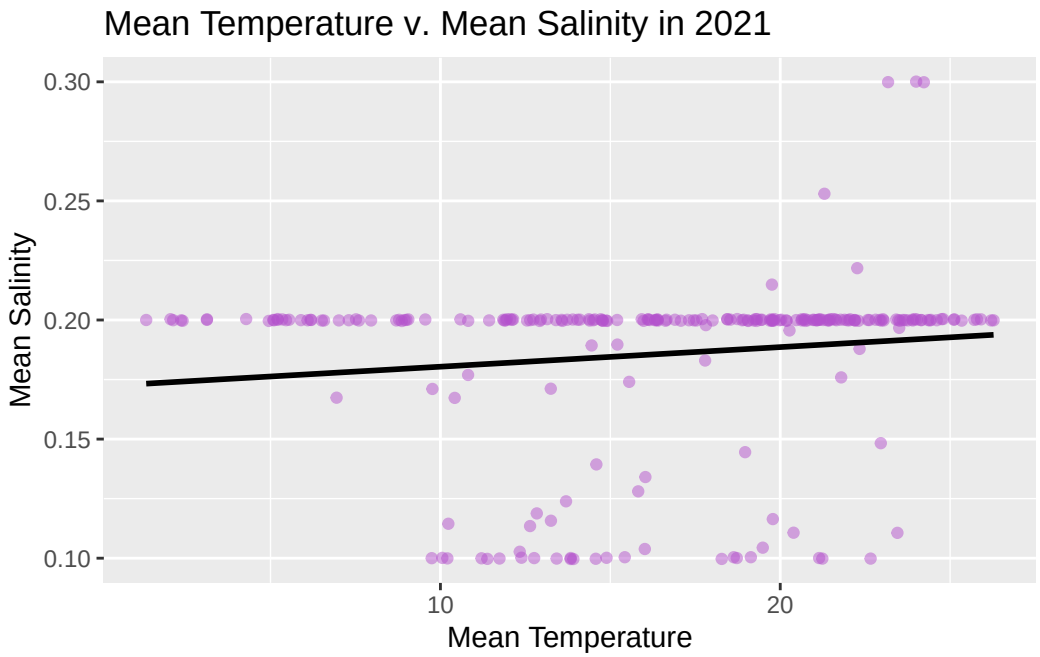
Using this code, we can test the relationship between temperature and salinity. Use transparency in case of overlap Using jitter plot instead of scatterplot to spread out the points more. This plot is not very visually appealling because salinity has such a small and limited range. Might be helpful to change this. But the trendline does show a positive relationship between temperature and salinity.

```
#copy for 2021
ggplot(daily_2021, aes(x = meanTemp, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Temperature v. Mean Salinity in 2021",
       x = "Mean Temperature",
       y = "Mean Salinity")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```

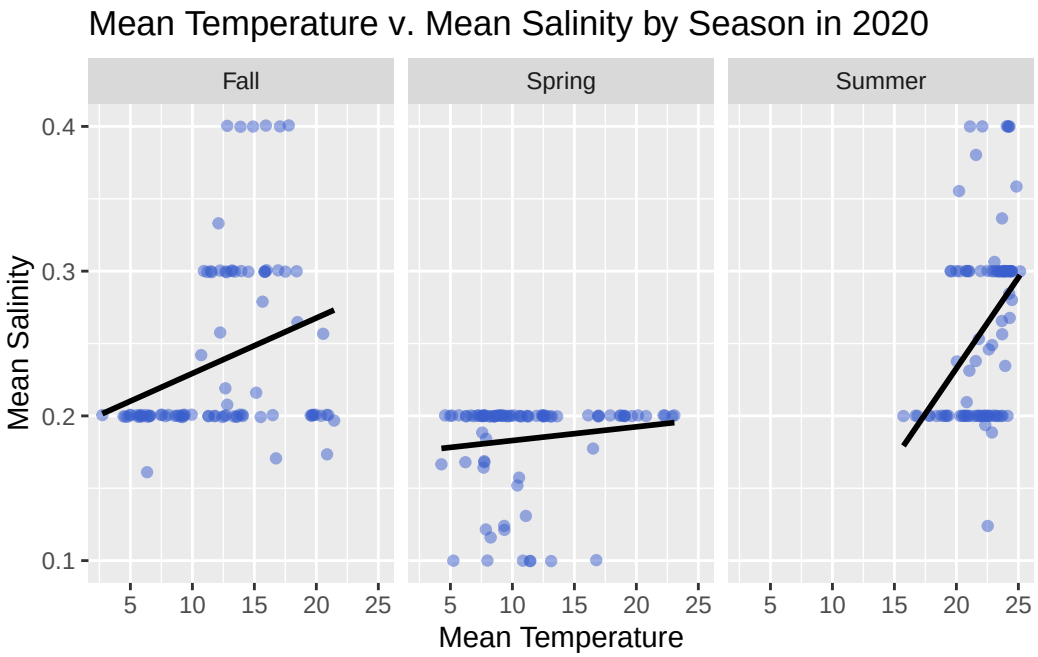



```
ggplot(daily_2020, aes(x = meanTemp, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  facet_wrap(~season)+
  labs(title = "Mean Temperature v. Mean Salinity by Season in 2020",
       x = "Mean Temperature",
       y = "Mean Salinity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



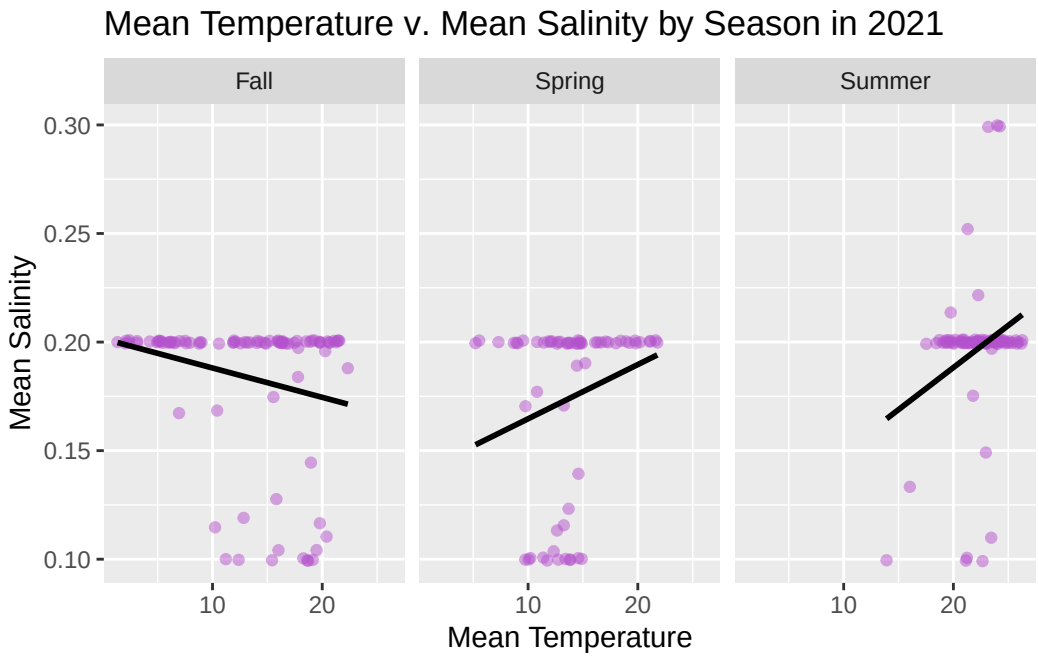
Same plot as before except separating by season. Looking at it seasonally, we still see positive trend lines for Spring, Summer, and Fall- though it is most prominently positive for Summer.

```
#copy for 2021
ggplot(daily_2021, aes(x = meanTemp, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  facet_wrap(~season)+
  labs(title = "Mean Temperature v. Mean Salinity by Season in 2021",
       x = "Mean Temperature",
       y = "Mean Salinity")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```

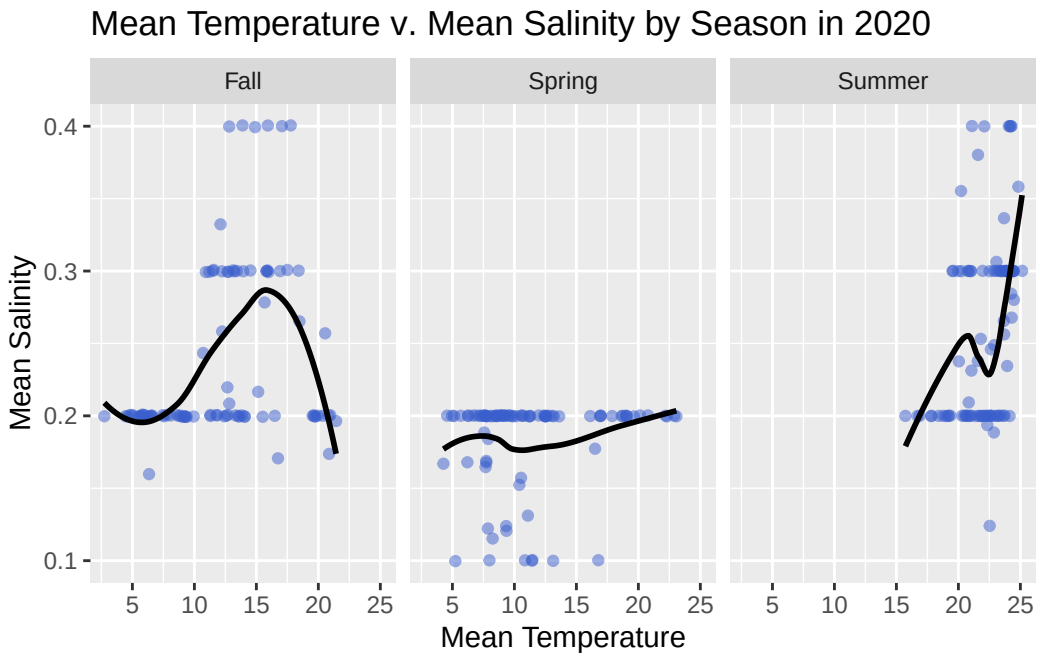



```
ggplot(daily_2020, aes(x = meanTemp, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "loess", se = FALSE, color = "black")+
  facet_wrap(~season)+
  labs(title = "Mean Temperature v. Mean Salinity by Season in 2020",
       x = "Mean Temperature",
       y = "Mean Salinity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



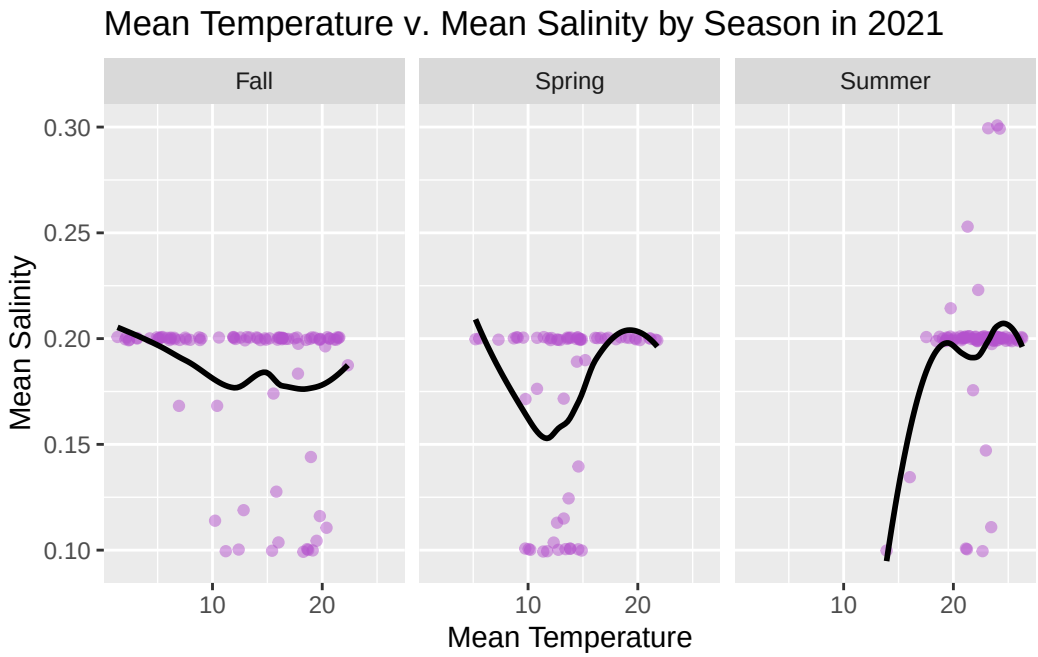

```
#Using method = "loess" instead of "lm"
```

```
#copy for 2021
ggplot(daily_2021, aes(x = meanTemp, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "loess", se = FALSE, color = "black")+
  facet_wrap(~season)+
  labs(title = "Mean Temperature v. Mean Salinity by Season in 2021",
       x = "Mean Temperature",
       y = "Mean Salinity")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

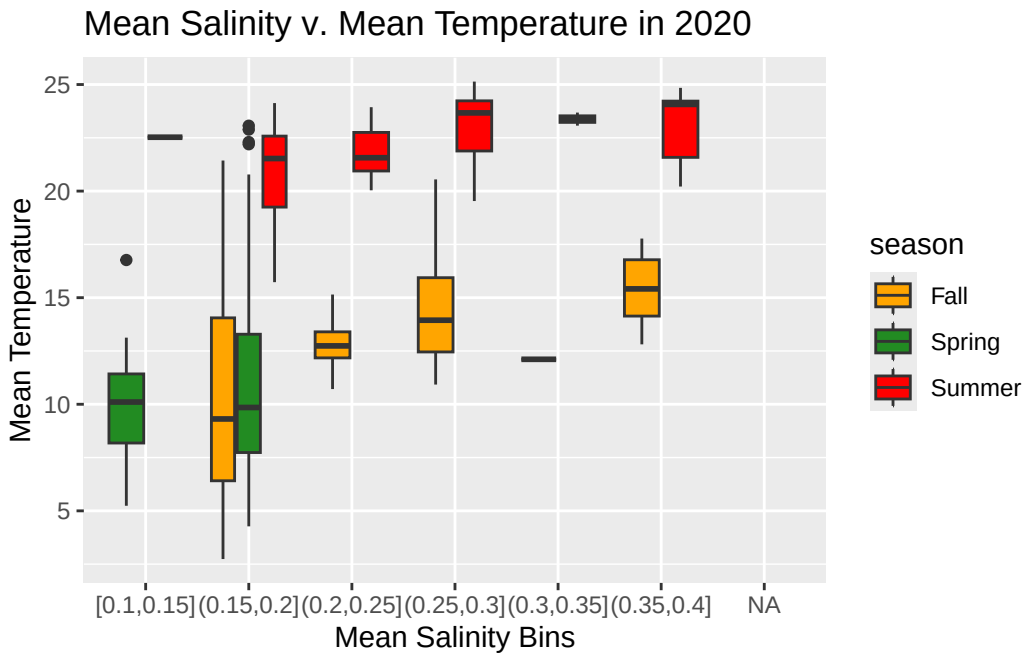
```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
#Using method = "loess" instead of "lm"
```

```
daily_2020 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05),
                      include.lowest = TRUE))%>%
           ggplot(aes(x = SalBin, y = meanTemp, fill = season)) +
  geom_boxplot() +
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red")
  labs(title = "Mean Salinity v. Mean Temperature in 2020",
       x = "Mean Salinity Bins",
       y = "Mean Temperature")
```

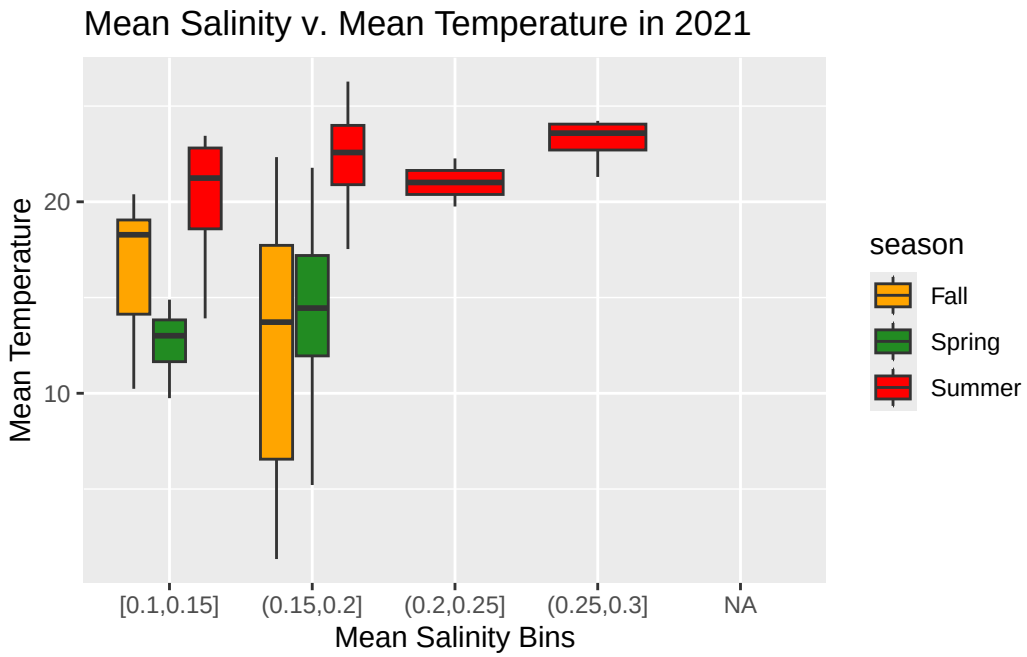
```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



Here, we can mutate the mean salinity variable to create SalBin in order to make breaks between means. Using boxplot here though instead of scatter/jitter to better visualize salinity. This plot allows us to view mean temperature in relation to categories of salinity level from lowest to highest by season. Include.lowest gets rid of NAs.

```
#copy for 2021
daily_2021 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05),
                      include.lowest = TRUE))%>%
           ggplot(aes(x = SalBin, y = meanTemp, fill = season)) +
  geom_boxplot() +
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red")
  labs(title = "Mean Salinity v. Mean Temperature in 2021",
       x = "Mean Salinity Bins",
       y = "Mean Temperature")
```

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

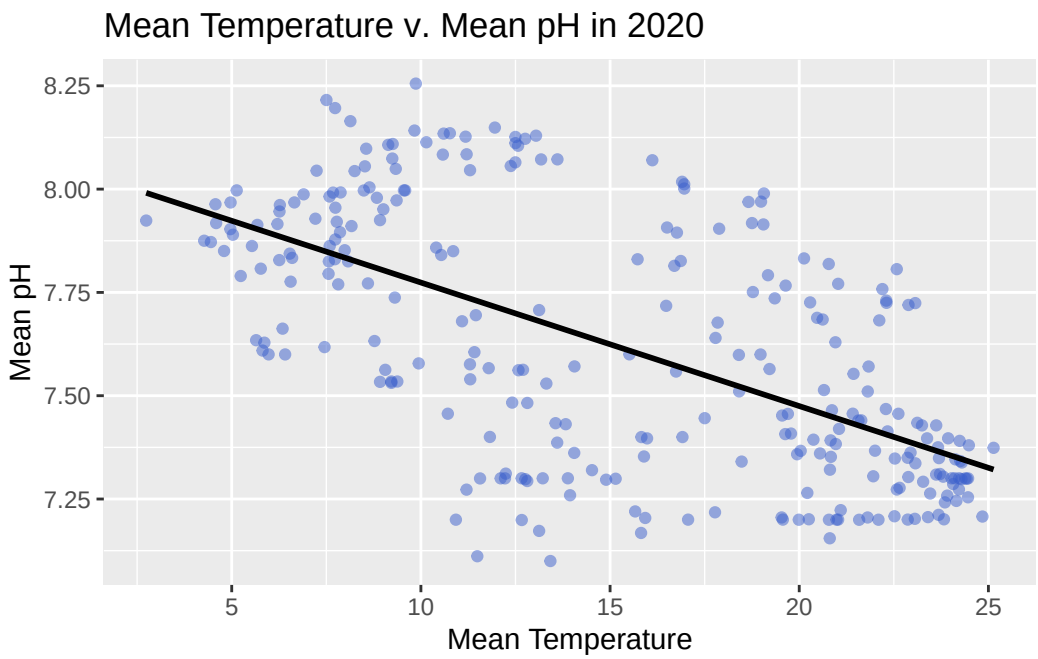



```
ggplot(daily_2020, aes(x = meanTemp, y = meanpH))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Temperature v. Mean pH in 2020",
       x = "Mean Temperature",
       y = "Mean pH")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



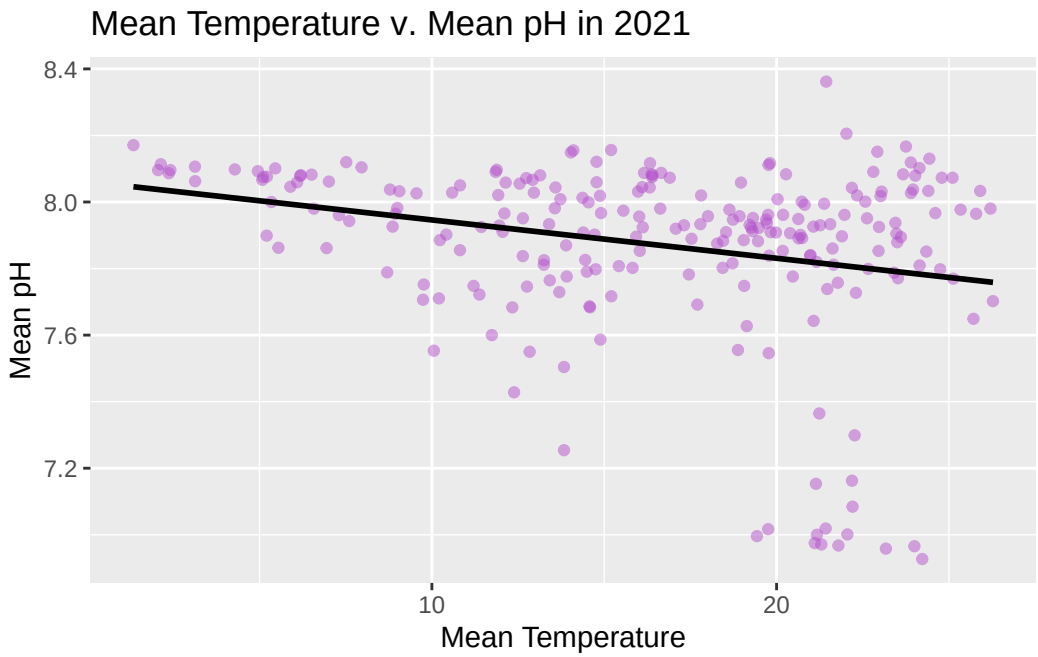
pH is also a bit limited in its range, but we do see more spreads of values. Based on the plot, temperature and pH have a strong negative relationship based on the trend line.

```
#copy for 2021
ggplot(daily_2021, aes(x = meanTemp, y = meanpH))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Temperature v. Mean pH in 2021",
       x = "Mean Temperature",
       y = "Mean pH")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```

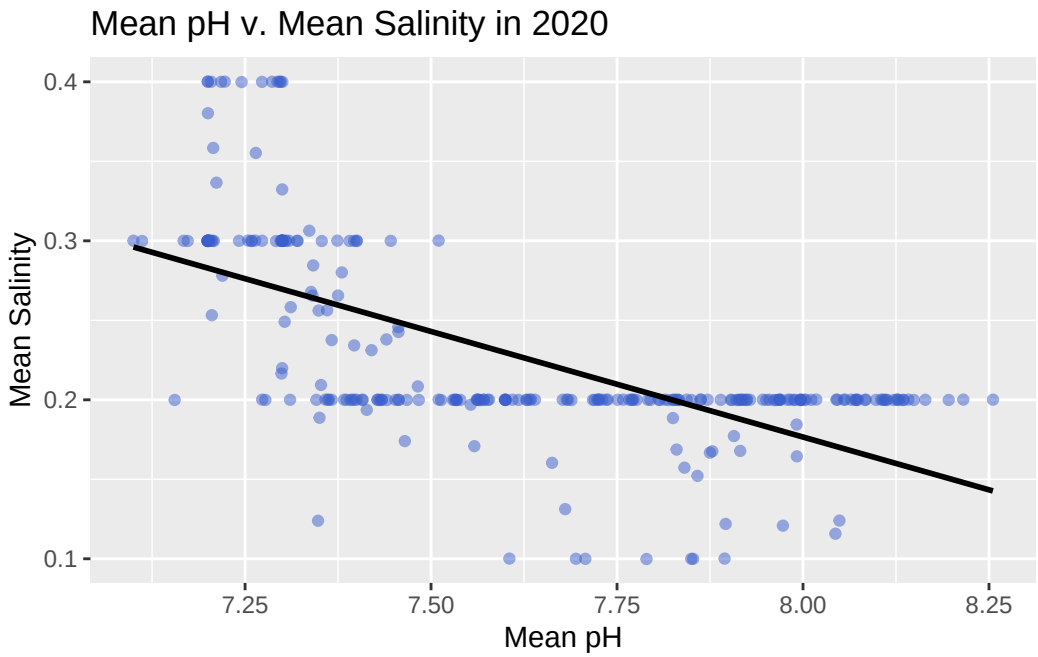



```
ggplot(daily_2020, aes(x = meanpH, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean pH v. Mean Salinity in 2020",
       x = "Mean pH",
       y = "Mean Salinity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



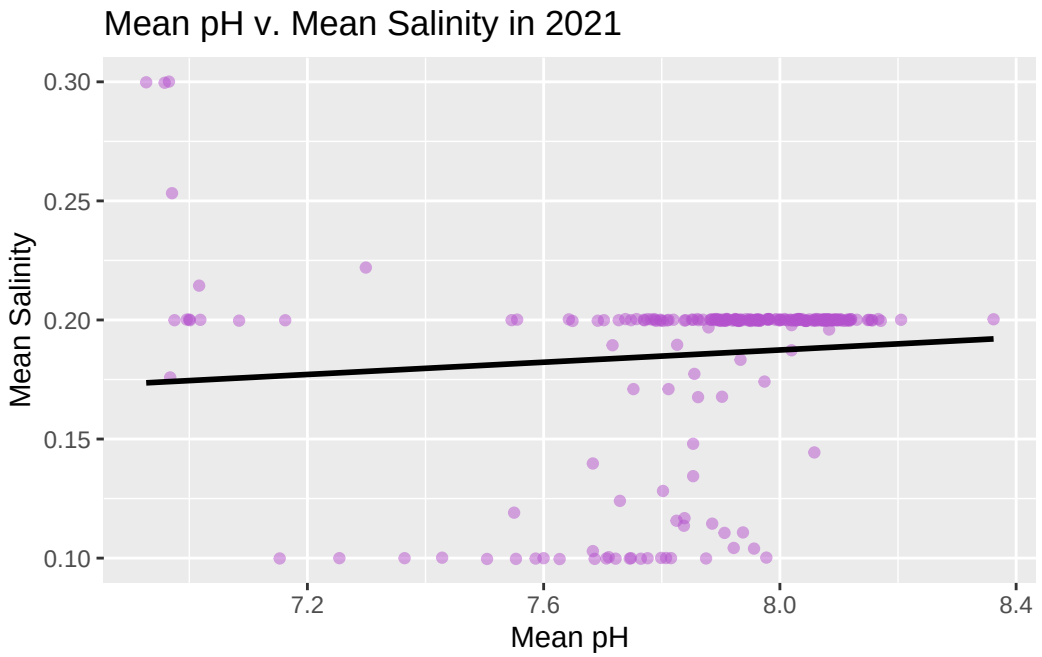
Using this code, we can explore the how salinity changes across pH levels. This plot also isnt very visually appealing due to the nature of the variables, but we can still see a distinct negative trend. Next, we can mutate meanSal in order to create bins of the variable to see if this allows us to be understand the relationship between pH and Salinity,

```
#copy for 2021
ggplot(daily_2021, aes(x = meanpH, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean pH v. Mean Salinity in 2021",
       x = "Mean pH",
       y = "Mean Salinity")
```

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

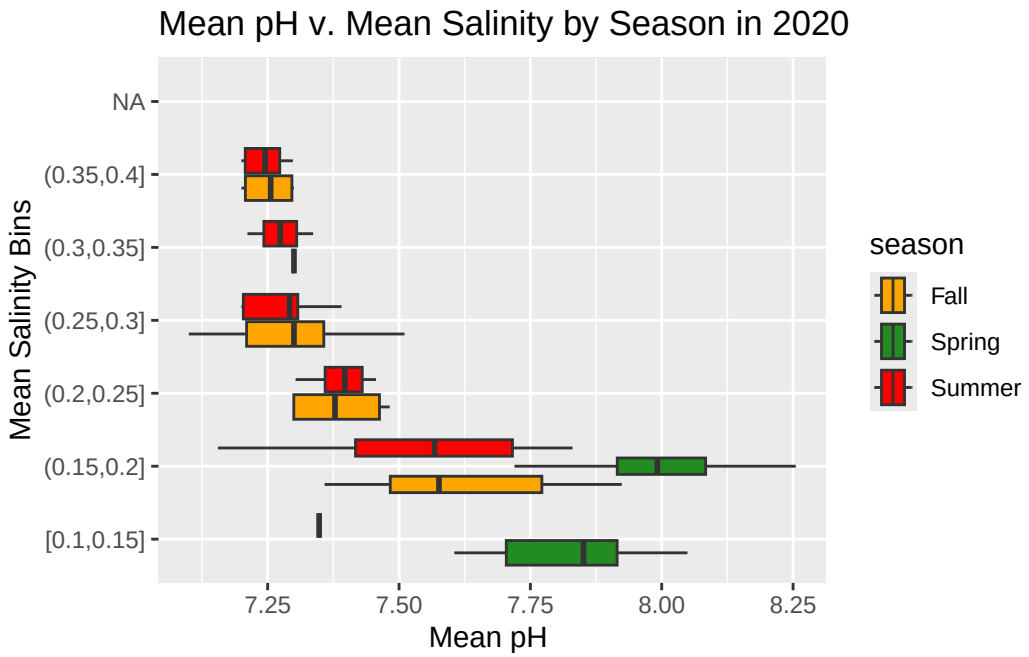
```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
daily_2020 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05), include.lowest = TRUE)) %>%
ggplot(aes(x = meanpH, y = SalBin, fill = season))+
  geom_boxplot() +
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red"))
  labs(title = "Mean pH v. Mean Salinity by Season in 2020",
       x = "Mean pH ",
       y = "Mean Salinity Bins")
```

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

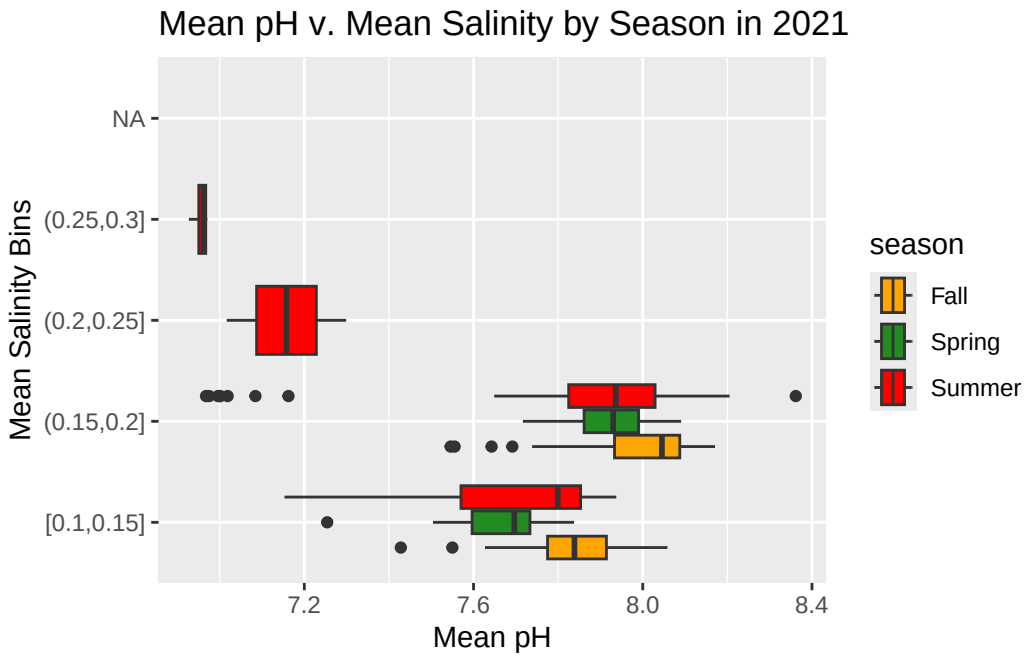


This plot shows SalBin against meanTemp by season. This plot is certainly interesting- looks like there are pronounced seasonal differences in the distribution of pH by Salinity level.

Next, we can try to use conductivity rather than salinity because conductivity is a raw variable- perhaps this will fix our issue in the jitter plot.

```
#copy for 2021
daily_2021 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05), include.lowest = TRUE)) %>%
ggplot(aes(x = meanpH, y = SalBin, fill = season))+
  geom_boxplot() +
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red")
  labs(title = "Mean pH v. Mean Salinity by Season in 2021",
       x = "Mean pH",
       y = "Mean Salinity Bins")
```

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

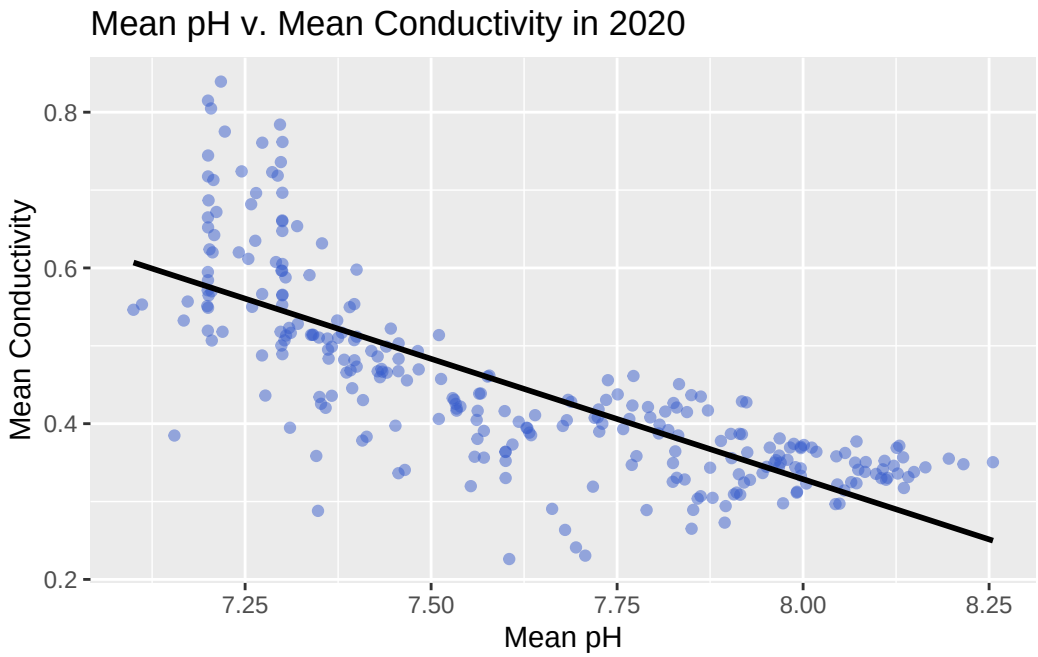



```
ggplot(daily_2020, aes(x = meanpH, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean pH v. Mean Conductivity in 2020",
       x = "Mean pH",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```

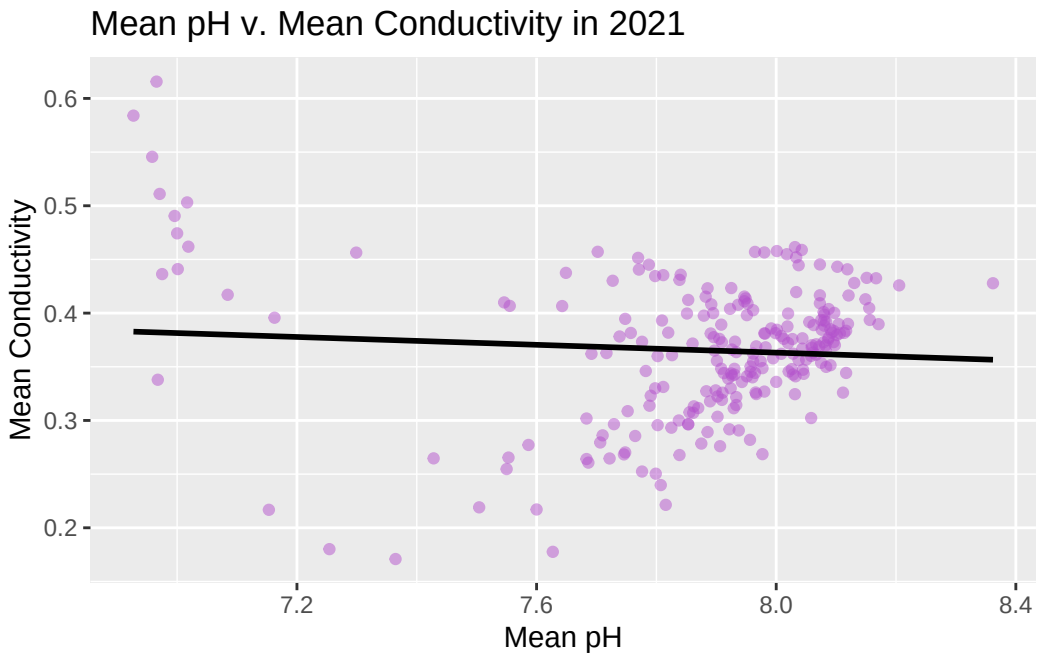



```
#copy for 2021
ggplot(daily_2021, aes(x = meanpH, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean pH v. Mean Conductivity in 2021",
       x = "Mean pH",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```



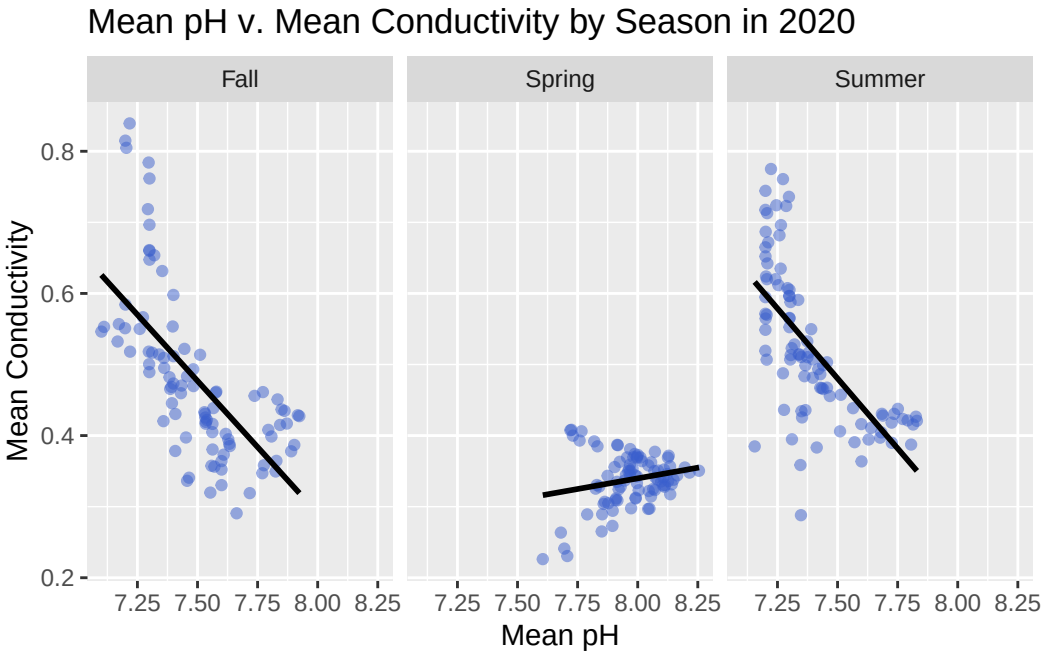
Same plot except using conductivity instead of salinity. Certainly a negative relationship, but looks like some outliers especially near higher average conductivity levels.

```
ggplot(daily_2020, aes(x = meanpH, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  facet_wrap(~season)+
  labs(title = "Mean pH v. Mean Conductivity by Season in 2020",
       x = "Mean pH",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



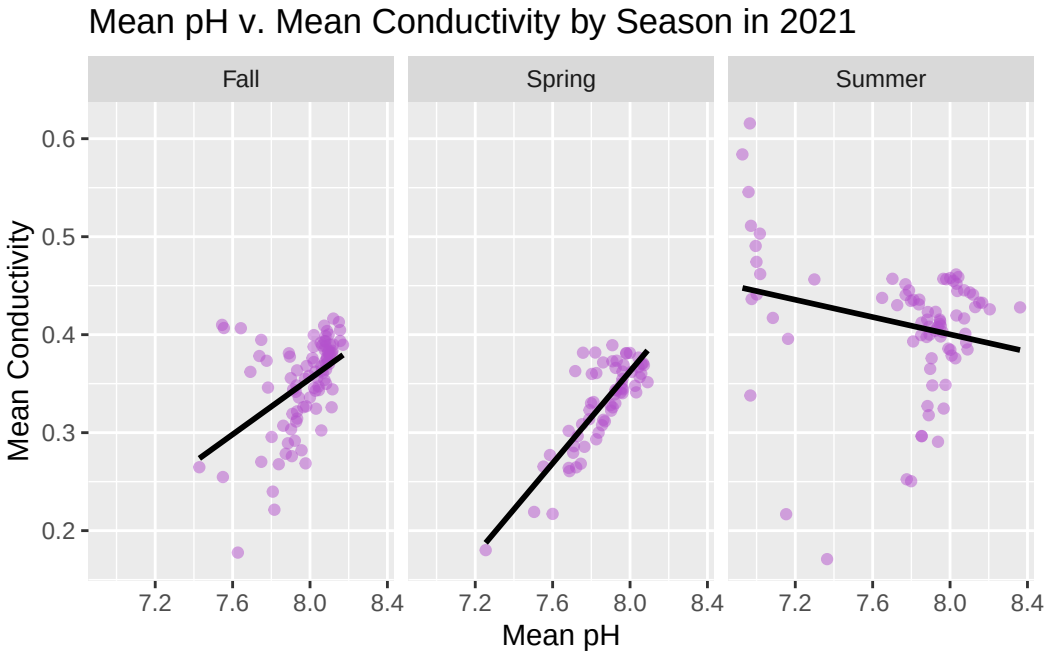
Separating by season to see if outliers may be explained by seasonal patterns. Based on plot, outliers at the high average conductivity levels lie in Fall and Summer.

```
#copy for 2021
ggplot(daily_2021, aes(x = meanpH, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  facet_wrap(~season)+
  labs(title = "Mean pH v. Mean Conductivity by Season in 2021",
       x = "Mean pH",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```

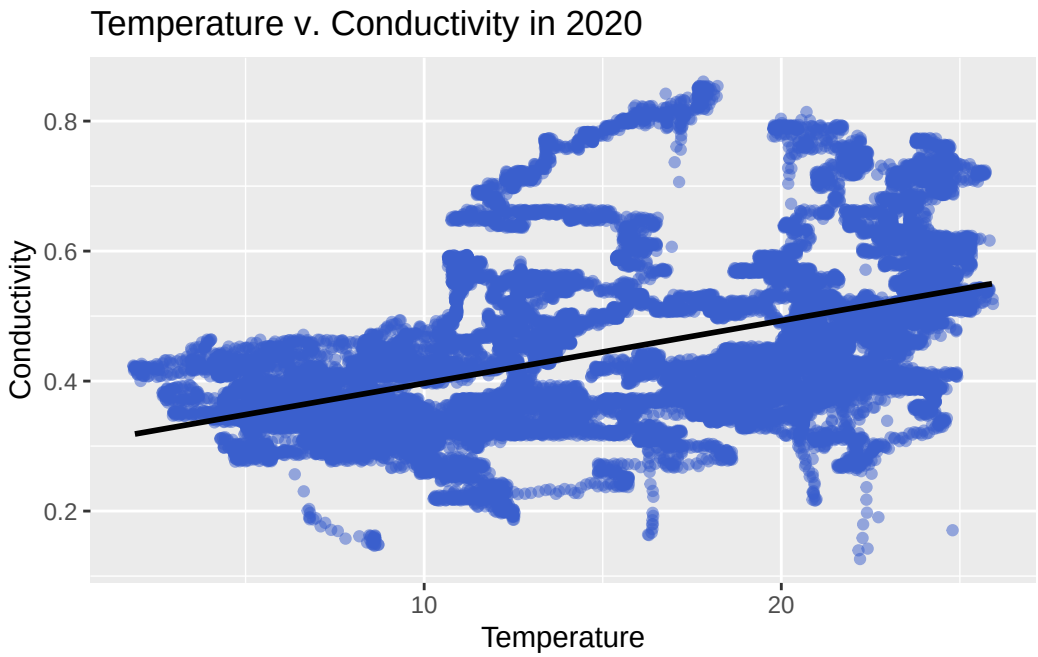



```
ggplot(no_winter_2020, aes(x = Temp, y = SpCond))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Temperature v. Conductivity in 2020",
       x = "Temperature",
       y = "Conductivity")
```

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

```
Warning: Removed 441 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 441 rows containing missing values or values outside the scale range
(`geom_point()`).
```



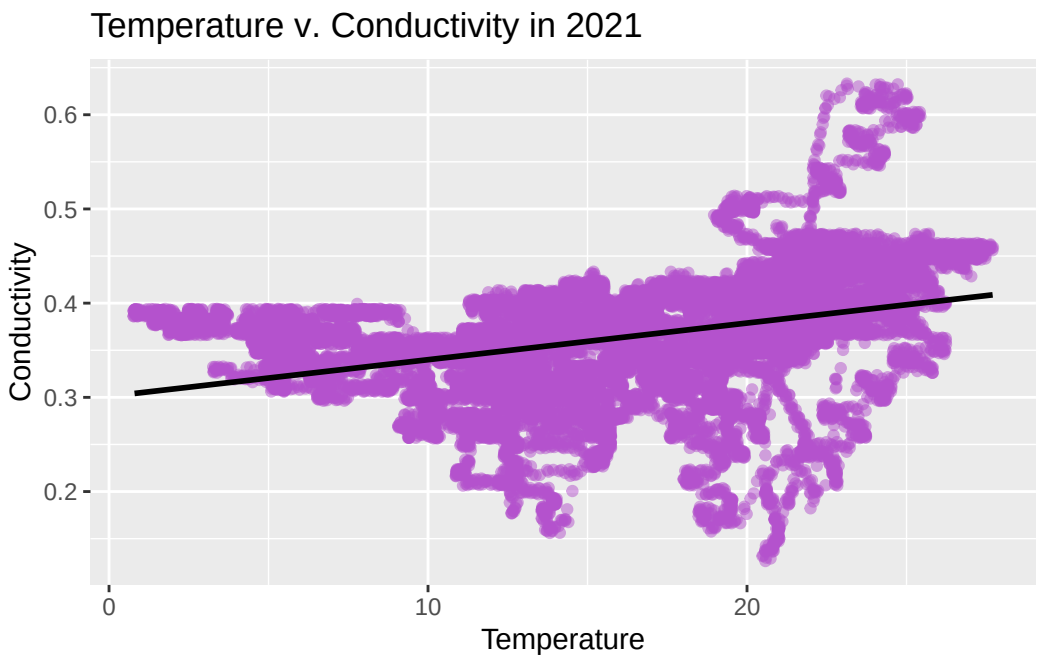
Temperature against conductivity- using conductivity instead of salinity here is especially important because salinity is calculated using temperature. This plot is not very visually appealing- should use averages to try to randomize.

```
#copy for 2021
ggplot(no_winter_2021, aes(x = Temp, y = SpCond))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Temperature v. Conductivity in 2021",
       x = "Temperature",
       y = "Conductivity")
```

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

```
Warning: Removed 4119 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4119 rows containing missing values or values outside the scale range
(`geom_point()`).
```

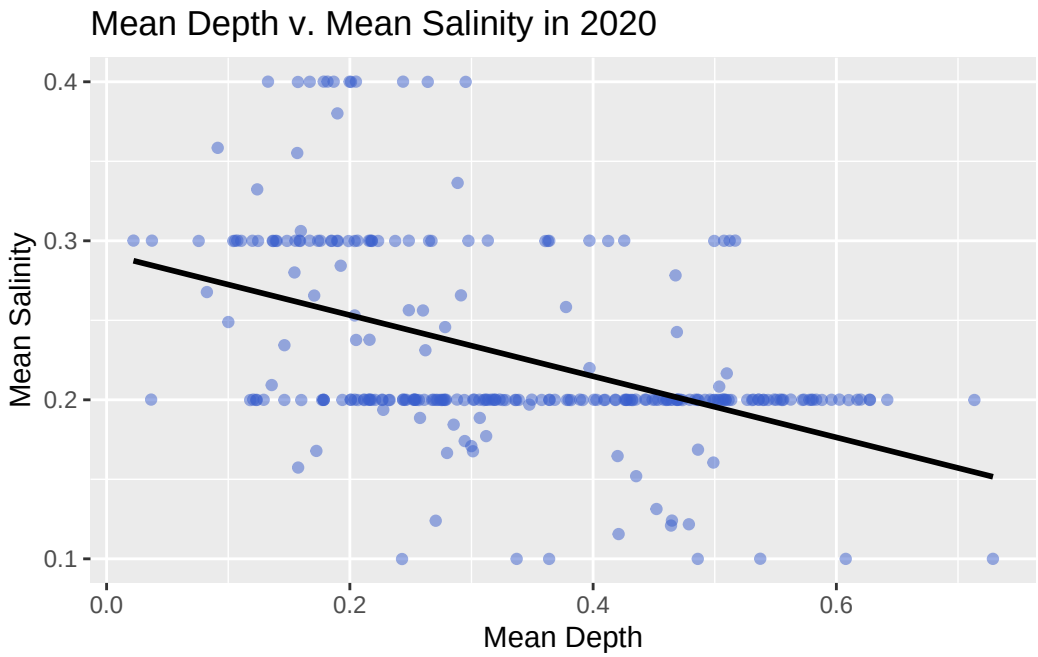



```
ggplot(daily_2020, aes(x = meanDepth, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Depth v. Mean Salinity in 2020",
       x = "Mean Depth",
       y = "Mean Salinity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



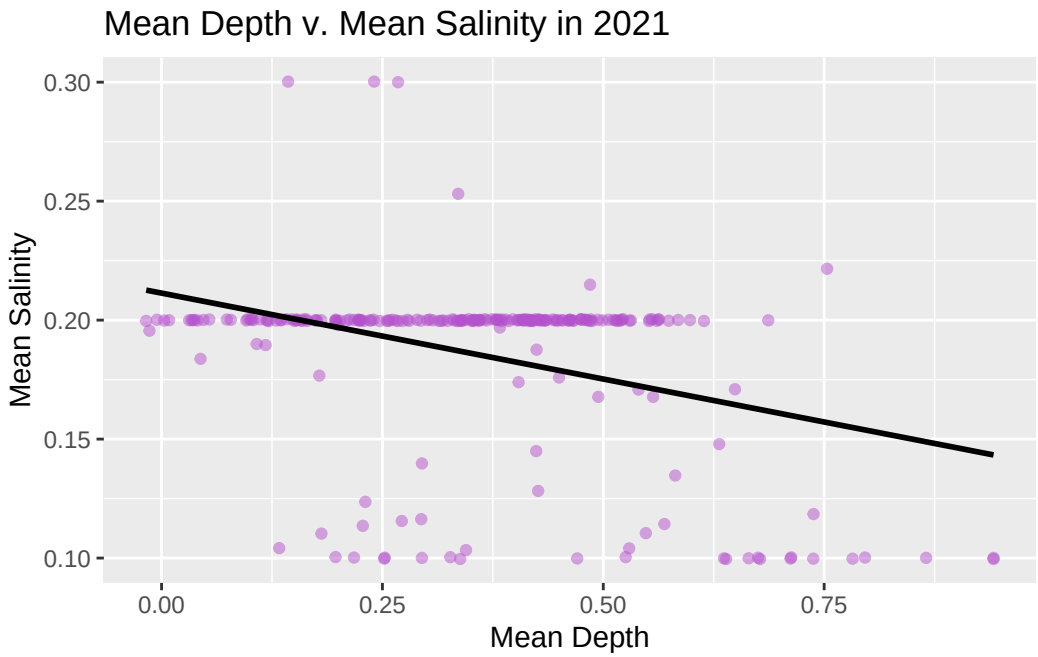
Mean depth against mean salinity plot. This plot is not visually appealing, despite using means to spread out points. We should try using a different plot to visualize these variables.

```
#copy for 2021
ggplot(daily_2021, aes(x = meanDepth, y = meanSal))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Depth v. Mean Salinity in 2021",
       x = "Mean Depth",
       y = "Mean Salinity")
```

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

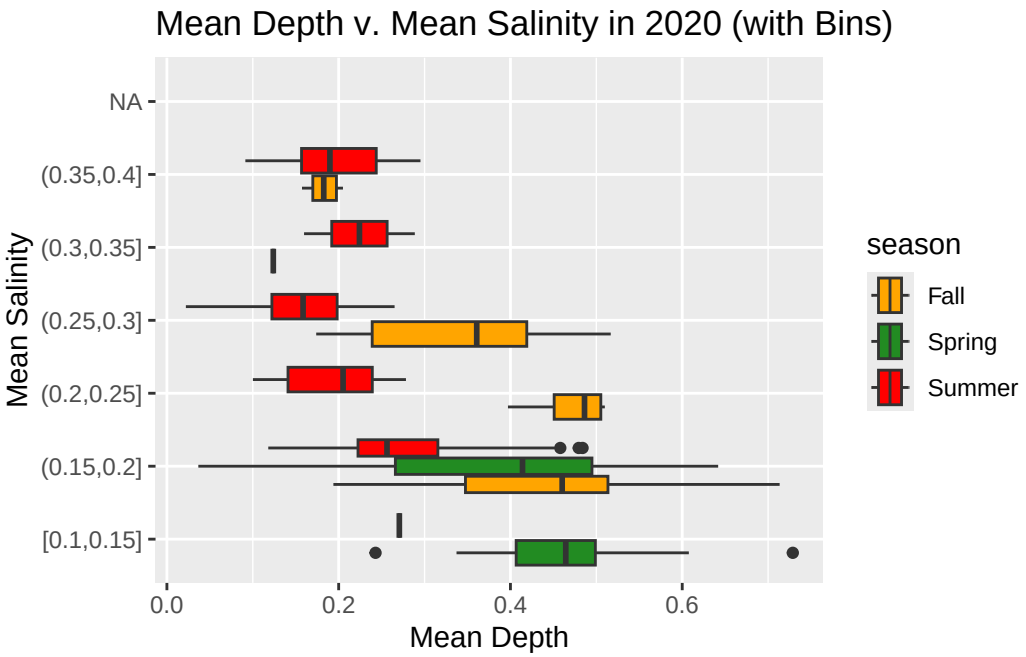
```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
daily_2020 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05), include.lowest = TRUE)) %>%
  ggplot(aes(x = meanDepth, y = SalBin, fill = season))+
  geom_boxplot()+
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red"))
  labs(title = "Mean Depth v. Mean Salinity in 2020 (with Bins)",
       x = "Mean Depth",
       y = "Mean Salinity")
```

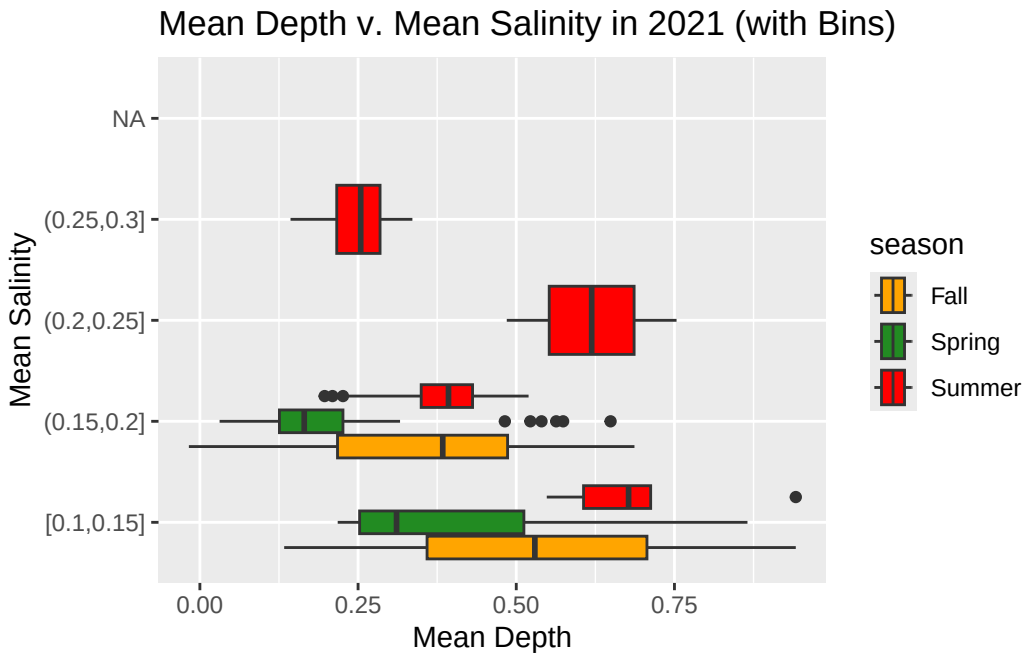
```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



This plot adds bins for the meanSal variable to look against meanDepth by season so now we can see seasonal depth distribution across levels of salinity.

```
#copy for 2021
daily_2021 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05), include.lowest = TRUE)) %>%
  ggplot(aes(x = meanDepth, y = SalBin, fill = season))+
  geom_boxplot()+
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red"))+
  labs(title = "Mean Depth v. Mean Salinity in 2021 (with Bins)",
       x = "Mean Depth",
       y = "Mean Salinity")
```

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

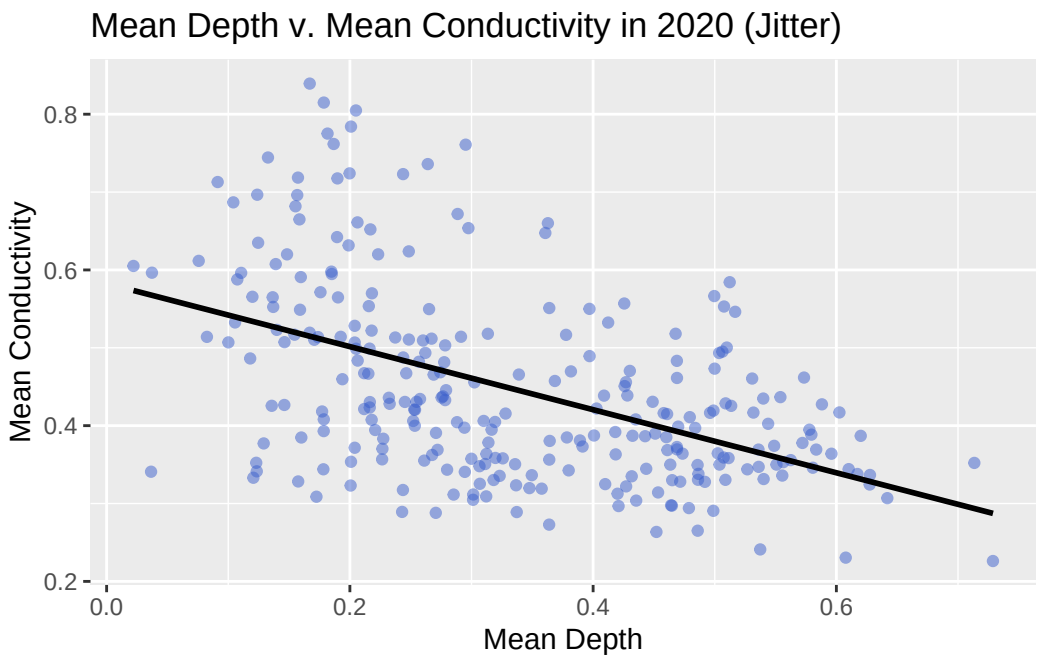



```
ggplot(daily_2020, aes(x = meanDepth, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Depth v. Mean Conductivity in 2020 (Jitter)",
       x = "Mean Depth",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```



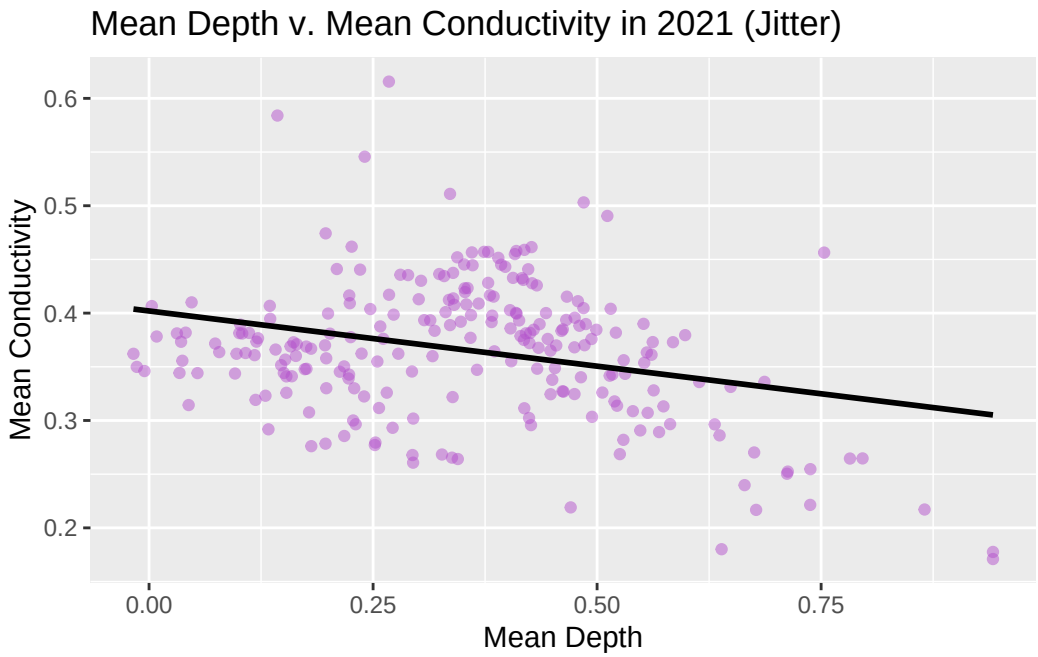
Same plot except using conductivity instead of salinity. Since conductivity is directly measured, we have much more variability, but we still see a similar relationship between the two.

```
#copy for 2021
ggplot(daily_2021, aes(x = meanDepth, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Depth v. Mean Conductivity in 2021 (Jitter)",
       x = "Mean Depth",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```

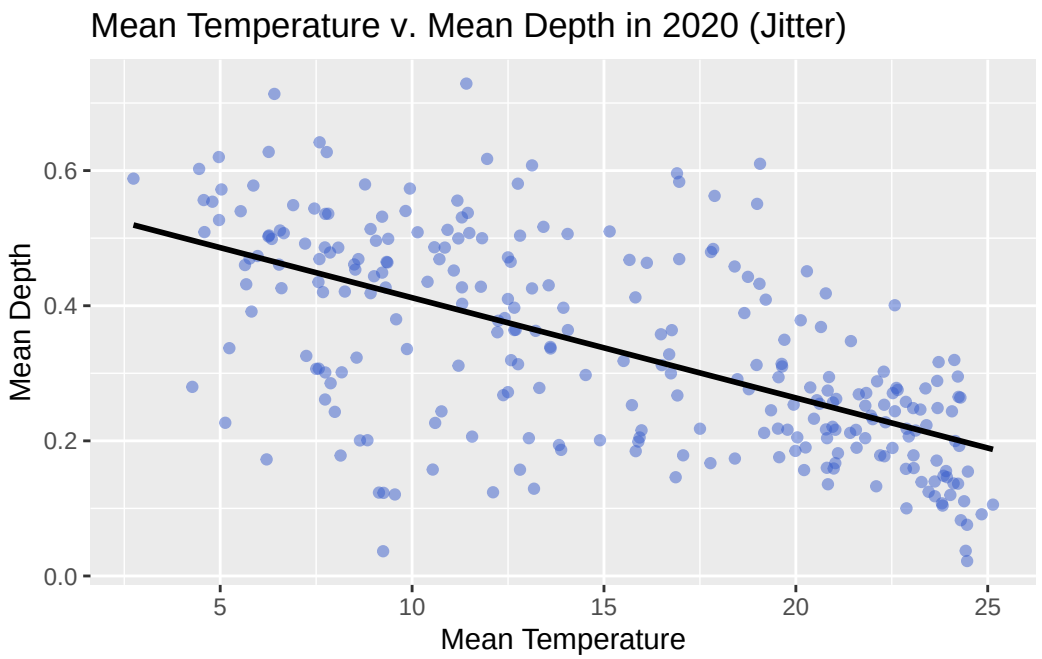



```
ggplot(daily_2020, aes(x = meanTemp, y = meanDepth))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Temperature v. Mean Depth in 2020 (Jitter)",
       x = "Mean Temperature",
       y = "Mean Depth")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```

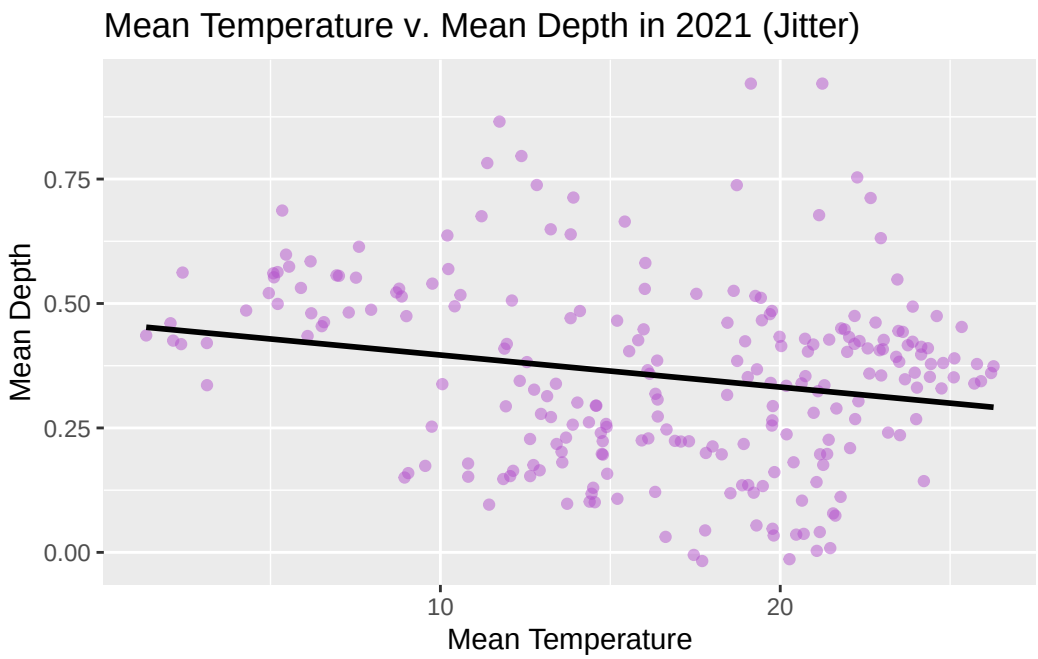



```
#copy for 2021
ggplot(daily_2021, aes(x = meanTemp, y = meanDepth))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Temperature v. Mean Depth in 2021 (Jitter)",
       x = "Mean Temperature",
       y = "Mean Depth")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```



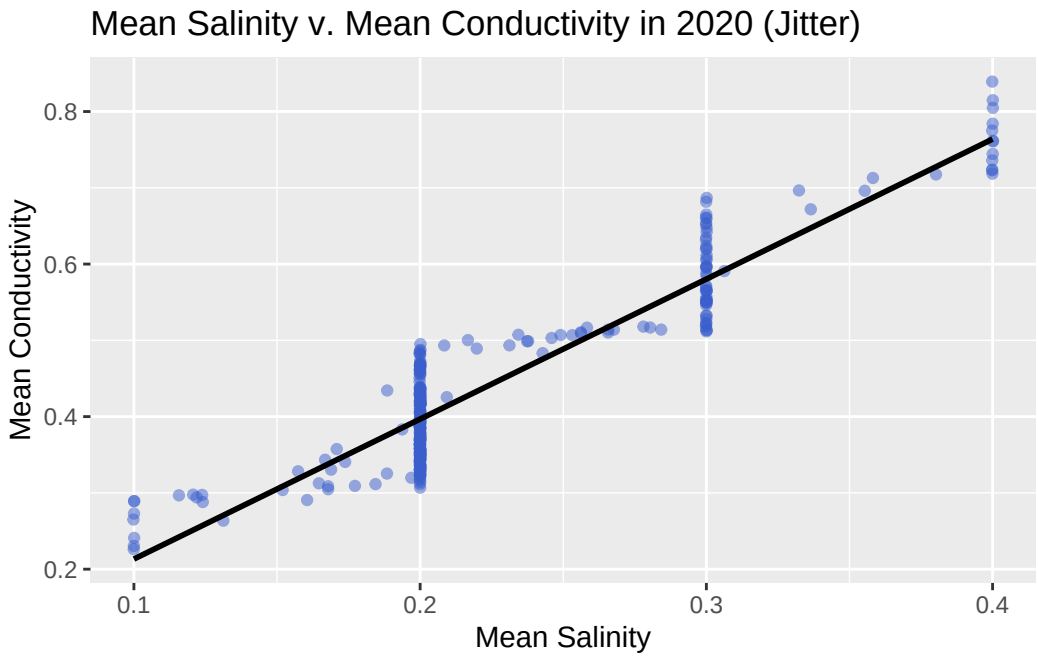
Mean Depth against mean temperature

```
ggplot(daily_2020, aes(x = meanSal, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "royalblue3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Salinity v. Mean Conductivity in 2020 (Jitter)",
       x = "Mean Salinity",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).
```

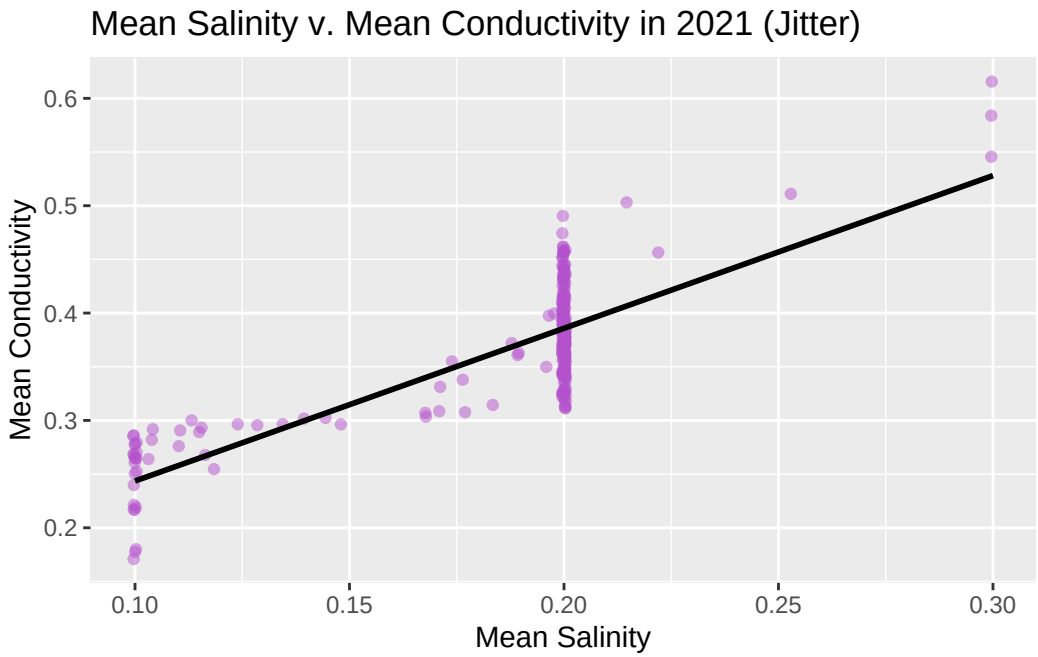



```
#copy for 2021
ggplot(daily_2021, aes(x = meanSal, y = meanSpCond))+
  geom_jitter(alpha = 0.5, color = "mediumorchid3") +
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(title = "Mean Salinity v. Mean Conductivity in 2021 (Jitter)",
       x = "Mean Salinity",
       y = "Mean Conductivity")
```

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

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

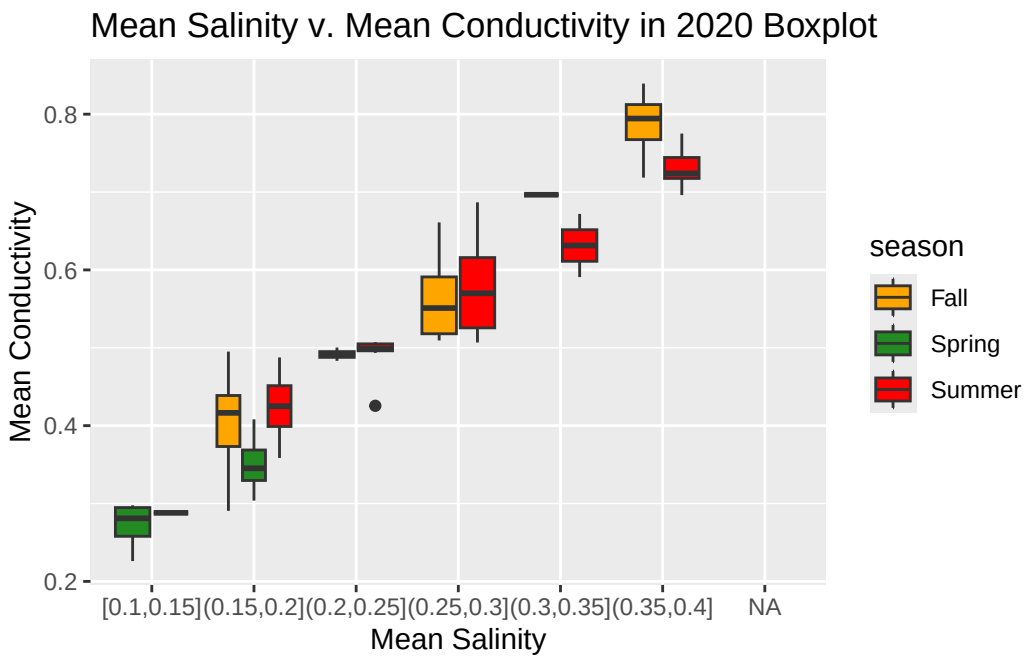
```
Warning: Removed 41 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Checking relation between salinity and conductivity to see if they are consistent. Looks like there is certainly consistency between the two.

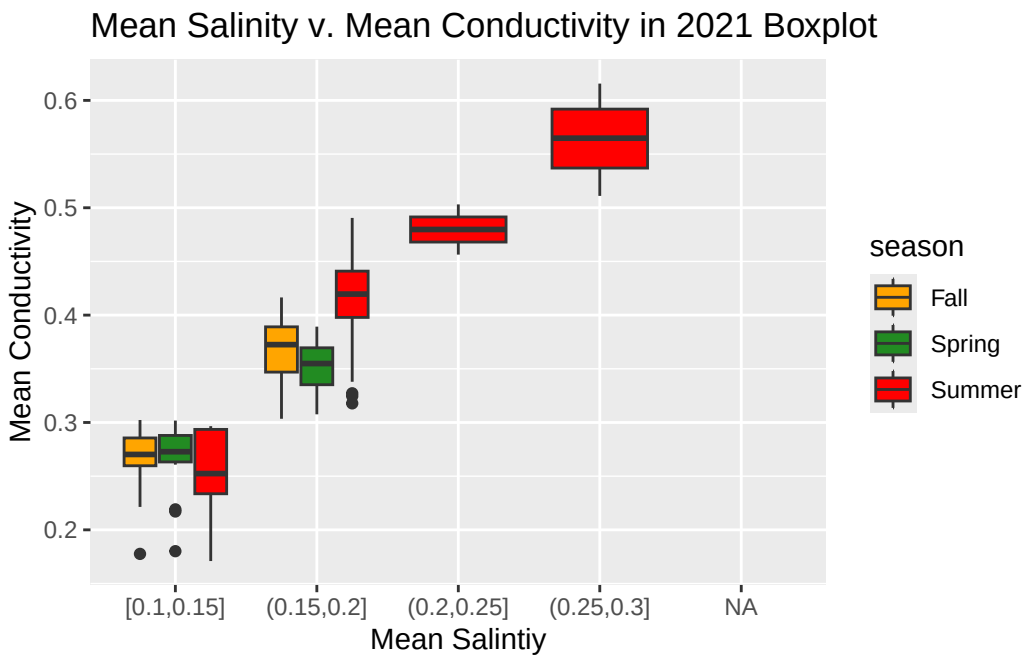
```
daily_2020 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05),
                      include.lowest = TRUE))%>%
ggplot(aes(x = SalBin, y = meanSpCond, fill = season)) +
  geom_boxplot() +
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red"
  labs(title = "Mean Salinity v. Mean Conductivity in 2020 Boxplot",
       x = "Mean Salinity",
       y = "Mean Conductivity")
```

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```




```
#copy for 2021
daily_2021 %>%
  mutate(SalBin = cut(meanSal, breaks = seq(0.1, 0.4, by = 0.05),
                      include.lowest = TRUE))%>%
ggplot(aes(x = SalBin, y = meanSpCond, fill = season)) +
  geom_boxplot() +
  scale_fill_manual(values = c("Fall" = "orange", "Spring" = "forestgreen", "Summer" = "red")
  labs(title = "Mean Salinity v. Mean Conductivity in 2021 Boxplot",
       x = "Mean Salintiy",
       y = "Mean Conductivity")
```

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



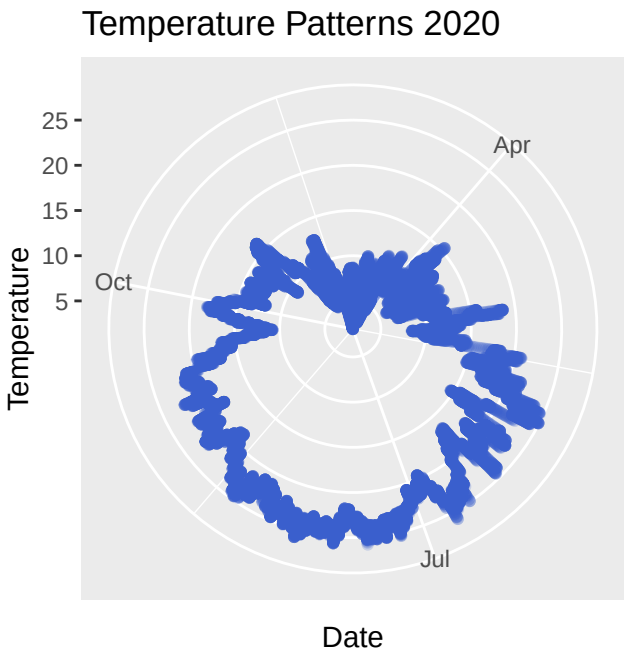
Same information as prior plot (conductivity against salinity) except using bins for salinity. We can definitely still see the consistency of the relationship. We should try to use bins for other plots using meanSal to visualize the variable better.

Explore these same ones using method = "loess"

Other Plots- Polar plots

```
ggplot(no_winter_2020, aes(x = date_clean, y = Temp)) +
  geom_point(alpha = 0.2, color = "royalblue3") +
  coord_polar()+
  labs(
    title = "Temperature Patterns 2020",
    x = "Date",
    y = "Temperature")
```

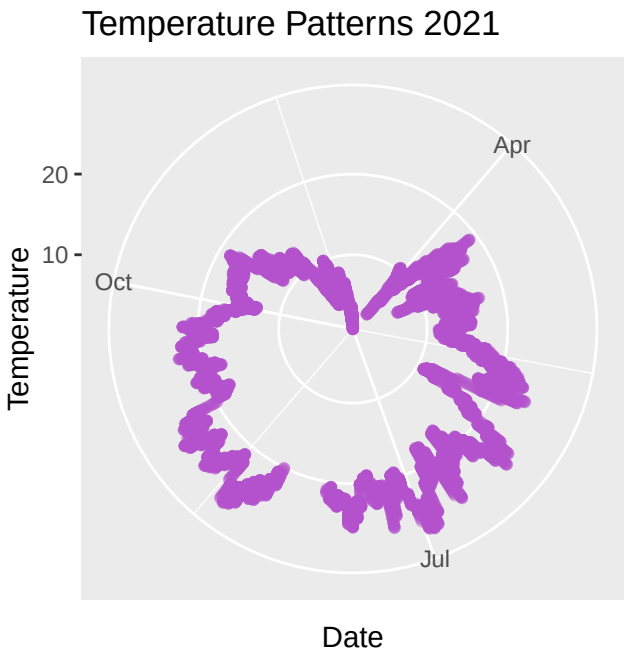
```
Warning: Removed 441 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
#error getting color to show up
```

```
#copy for 2021
ggplot(no_winter_2021, aes(x = date_clean, y = Temp)) +
  geom_point(alpha = 0.2, color= "mediumorchid3") +
  coord_polar()+
  labs(
    title = "Temperature Patterns 2021",
    x = "Date",
    y = "Temperature")
```

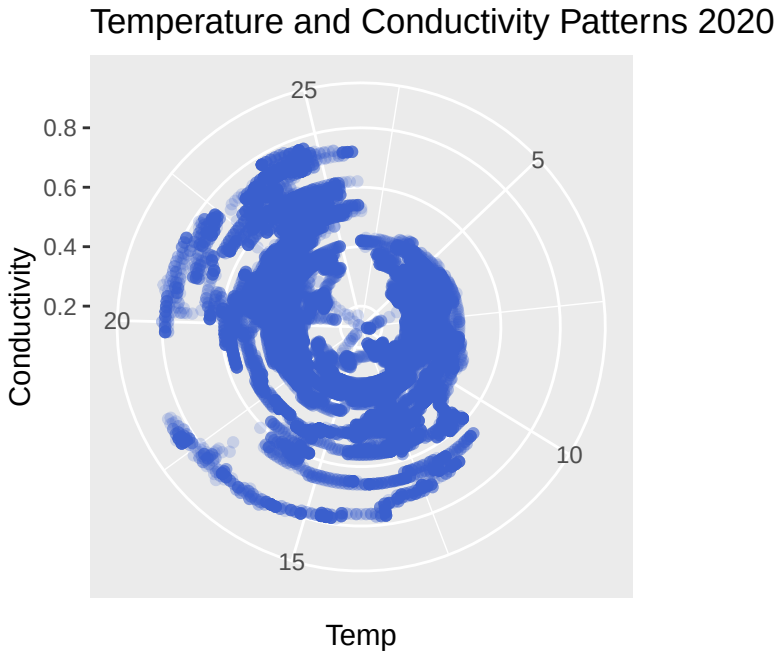
```
Warning: Removed 4119 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Next, lets try using Temperature and Conductivity Variables in a new plot since the jitterplot turned out very messy earlier.

```
ggplot(no_winter_2020, aes(x = Temp, y = SpCond)) +
  geom_point(alpha = 0.2, color = "royalblue3") +
  coord_polar()+
  labs(
    title = "Temperature and Conductivity Patterns 2020",
    x = "Temp",
    y = "Conductivity")
```

```
Warning: Removed 441 rows containing missing values or values outside the scale range
(`geom_point()`).
```



This plot still looks insane!

Combine Datasets into New Dataset with Both Years

```
no_winter_year_2020 <- no_winter_2020 %>% mutate(year = 2020)
no_winter_year_2021 <- no_winter_2021 %>% mutate(year = 2021)
combined_datasets <- bind_rows(no_winter_year_2020, no_winter_year_2021)
```

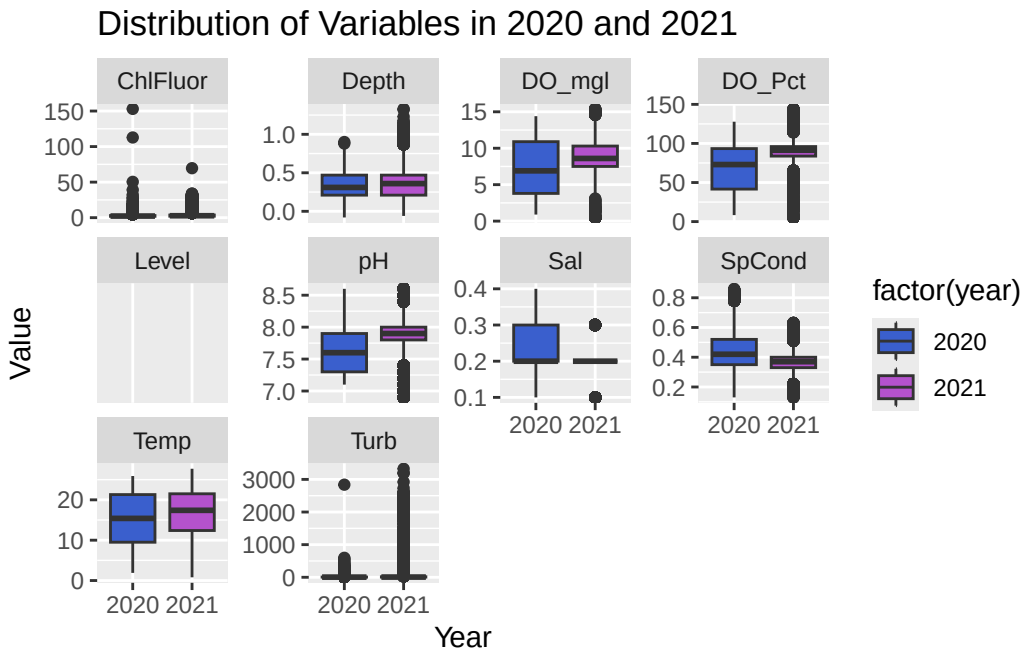
```
New names:
New names:
* `...31` -> `...32`
```

Combined Dataset, observing Distribution of Variables by Season 2020 and 2021

```
combined_datasets %>%
  pivot_longer(
    cols = c(Temp, Sal, Turb, ChlFluor, Level, Depth, pH, SpCond, DO_Pct, DO_mgl),
    names_to = "variable",
    values_to = "value") %>%
  ggplot(aes(x = factor(year), y = value, fill = factor(year)))+
  geom_boxplot()+
  facet_wrap(~variable, scales = "free_y")+
```

```
  labs(y = NULL)+
  labs(title = "Distribution of Variables in 2020 and 2021",
       x = "Year",
       y = "Value")+
  scale_fill_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

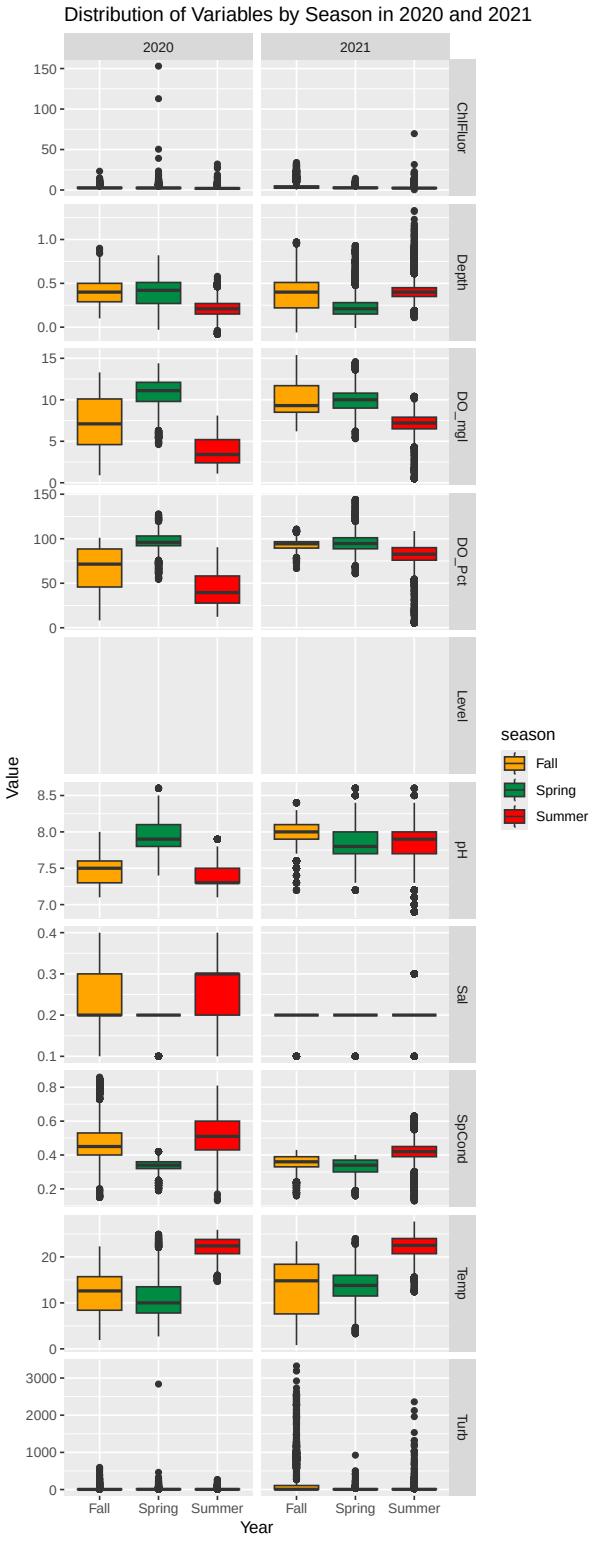
```
Warning: Removed 94762 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```




```
#temperature facet b yyear combined dataset over time (secind plot from beginning)
combined_datasets %>%
  pivot_longer(
    cols = c(Temp, Sal, Turb, ChlFluor, Level, Depth, pH, SpCond, DO_Pct, DO_mgl),
    names_to = "variable",
    values_to = "value") %>%
  ggplot(aes(x = season, y = value, fill = season))+
  geom_boxplot()+
  facet_grid(variable ~ year, scales = "free_y")+
  labs(title = "Distribution of Variables by Season in 2020 and 2021",
       x = "Year",
       y = "Value")+
```

```
  scale_fill_manual(values = c("Spring" = "springgreen4",
                               "Summer" = "red",
                               "Fall" = "orange"))
```

```
Warning: Removed 94762 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```


Distribution of Variables by Season in 2020 and 2021
2020
2021
ChlFluor
Depth
DO_mgl
DO_Pct
Level
pH
Sal
SpCond
Temp
Turb
Value
Year
Fall
Spring
Summer
season

Combined Dataset Exploration

These plots visualize average daily measurements of each variable by month looking at both 2020 and 2021. I figured that using geom_smooth with loess would provide an aesthetic and flexible trendline to look at measurements yearly and monthly.

```
#figure measurements edited in order to better fit plot
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = Temp, color = fa
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "Mean Temperature by Month in 2020 and 2021",
       x = "Date",
       y = "Temp")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

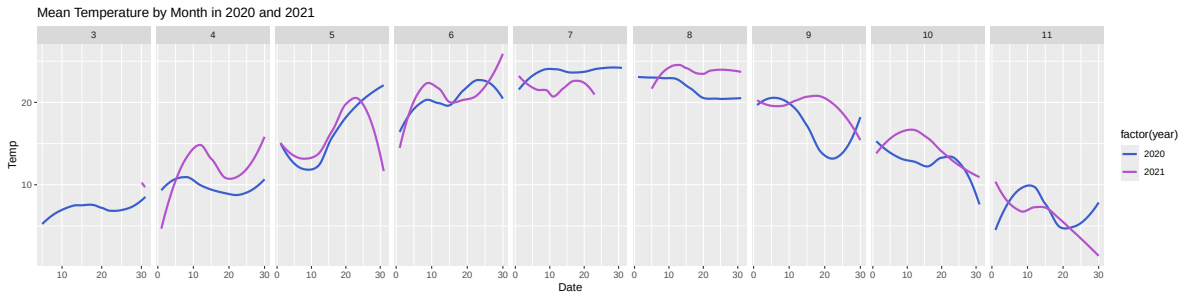
```
Warning: Removed 4560 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005
```

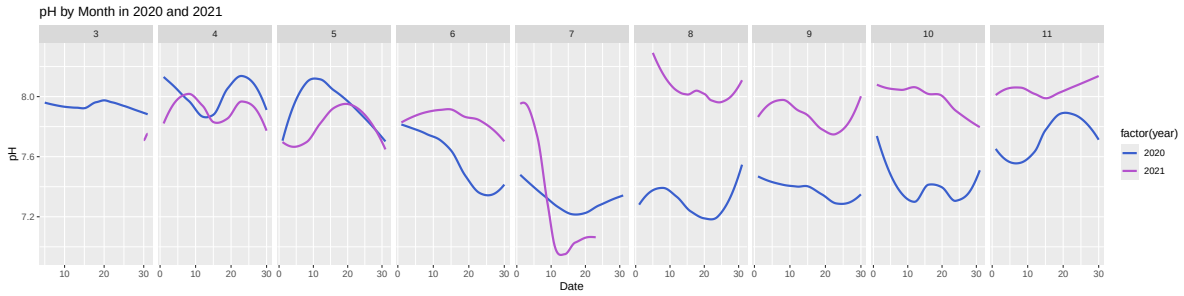
```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```



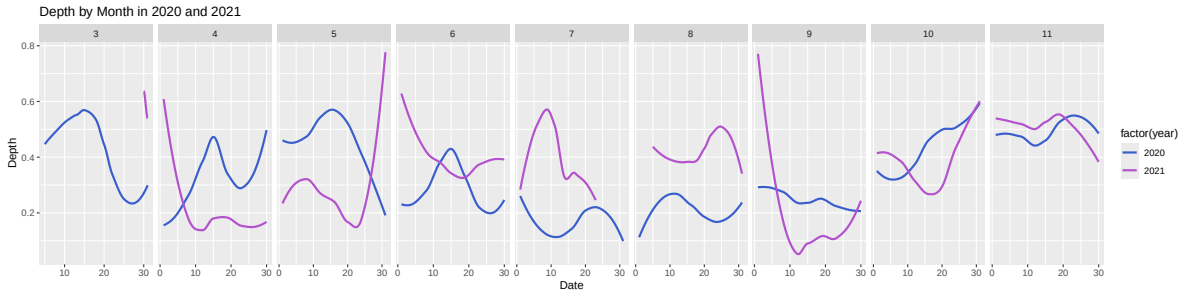
```
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = pH, color = facto
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "pH by Month in 2020 and 2021",
       x = "Date",
       y = "pH")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

Warning: Removed 4560 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```




```
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = Depth, color = fa
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "Depth by Month in 2020 and 2021",
       x = "Date",
       y = "Depth")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

Warning: Removed 4560 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```




```
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = SpCond, color = i
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "Sp Conductivity by Month in 2020 and 2021",
       x = "Date",
       y = "Sp Conductivity")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

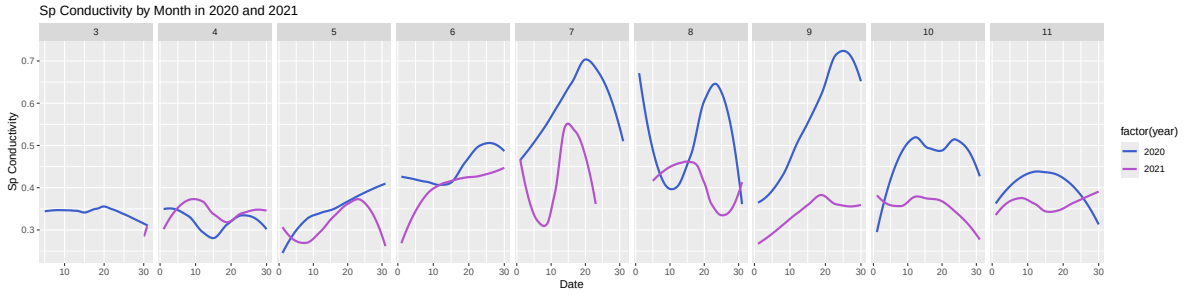
```
`geom_smooth()` using formula = 'y ~ x'

Warning: Removed 4560 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```




```
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = Turb, color = fa
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "Turbidity by Month in 2020 and 2021",
       x = "Date",
       y = "Turbidity")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

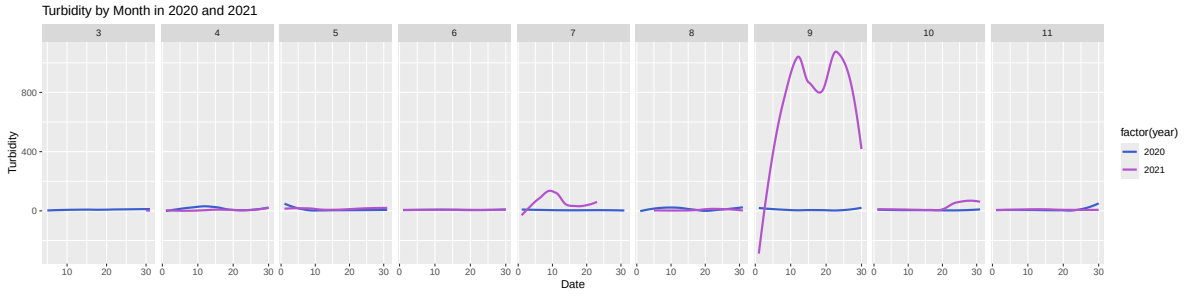
```
Warning: Removed 4560 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```




```
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = DO_mgl, color = 
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "Dissolved Oxygen Mgl by Month in 2020 and 2021",
       x = "Date",
       y = "DO mgl")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

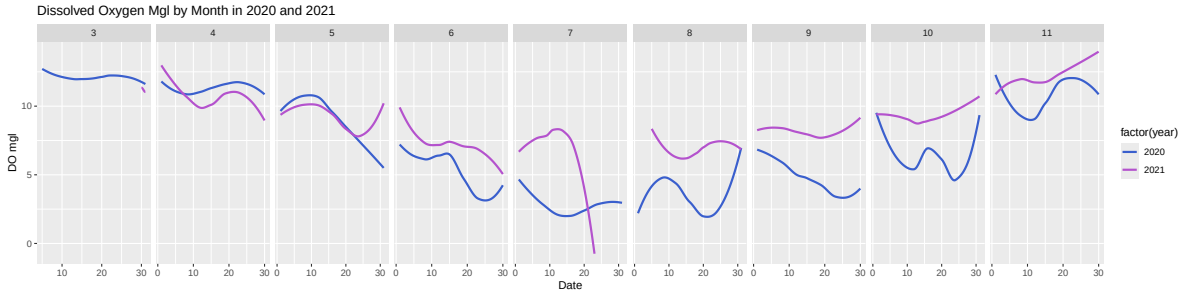
```
Warning: Removed 5021 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005
```

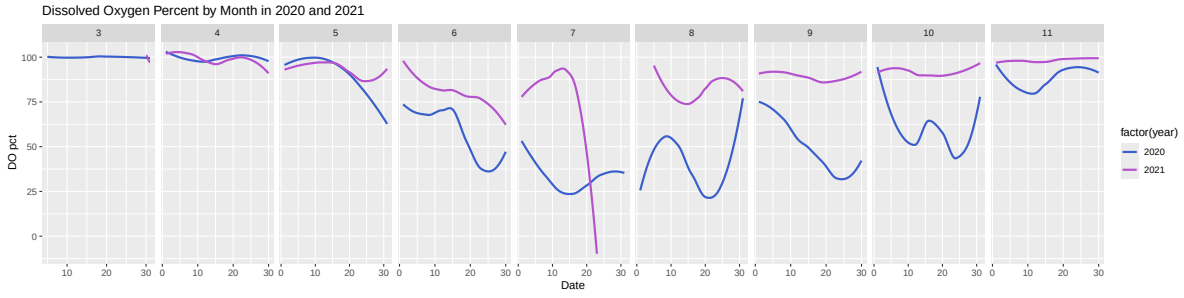
```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```




```
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = DO_Pct, color = f
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "Dissolved Oxygen Percent by Month in 2020 and 2021",
       x = "Date",
       y = "DO pct")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

Warning: Removed 5021 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```



```r
ggplot(combined_datasets, aes(x = as.integer(format(date_clean, "%d")), y = ChlFluor, color =
  geom_smooth(method = "loess", se = FALSE)+
  facet_wrap(~ month, scales = "free_x", nrow = 1) +
  labs(title = "ChlFluor by Month in 2020 and 2021",
       x = "Date",
       y = "ChlFluor")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

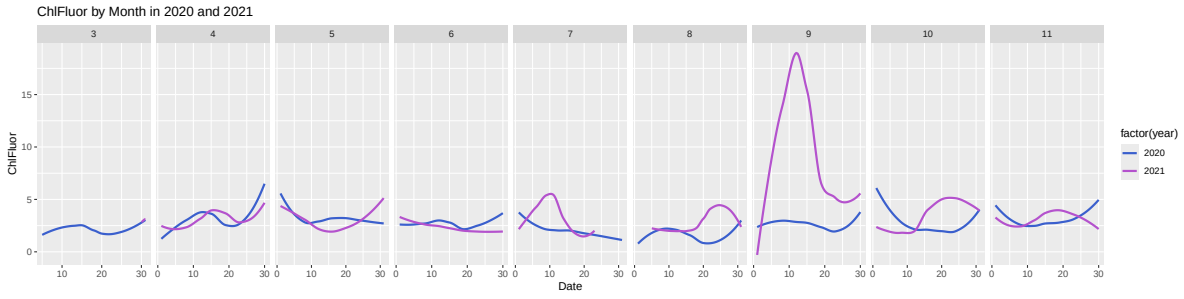
```
Warning: Removed 4560 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 29.995
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.005
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 1.01
```



Daily Average Combined Dataset

```
daily_combined_datasets <- combined_datasets %>%
  group_by(year, month, date_clean, season)%>%
  summarise(
    meanpH = mean(pH, na.rm = TRUE),
    meanSal = mean(Sal, na.rm = TRUE),
    meanTemp = mean(Temp, na.rm = TRUE),
    meanDepth = mean(Depth, na.rm = TRUE),
    meanSpCond = mean(SpCond, na.rm = TRUE),
    meanTurb = mean(Turb, na.rm = TRUE),
    meanChlFluor = mean(ChlFluor, na.rm = TRUE),
    meanDO_Pct = mean(DO_Pct, na.rm = TRUE),
    meanDO_mgl = mean(DO_mgl, na.rm = TRUE),
    .groups = "drop"
)
```

```
head(daily_combined_datasets)
```

```
# A tibble: 6 x 13
   year month date_clean season meanpH meanSal meanTemp meanDepth meanSpCond
  <dbl> <dbl> <date>     <chr>   <dbl>   <dbl>    <dbl>     <dbl>      <dbl>
1  2020     3 2020-03-01 Spring NaN      NaN     NaN      NaN         NaN
2  2020     3 2020-03-02 Spring NaN      NaN     NaN      NaN         NaN
3  2020     3 2020-03-03 Spring NaN      NaN     NaN      NaN         NaN
4  2020     3 2020-03-04 Spring NaN      NaN     NaN      NaN         NaN
5  2020     3 2020-03-05 Spring   7.93     0.2     7.21     0.492       0.328
6  2020     3 2020-03-06 Spring   7.91     0.2     5.68     0.432       0.335
# i 4 more variables: meanTurb <dbl>, meanChlFluor <dbl>, meanDO_Pct <dbl>,
#   meanDO_mgl <dbl>
```

Format date as factor to share x axis

```
daily_combined_datasets$doy <-
  as.numeric(format(daily_combined_datasets$date_clean, "%j"))
```

Correlation Plots Combined Years

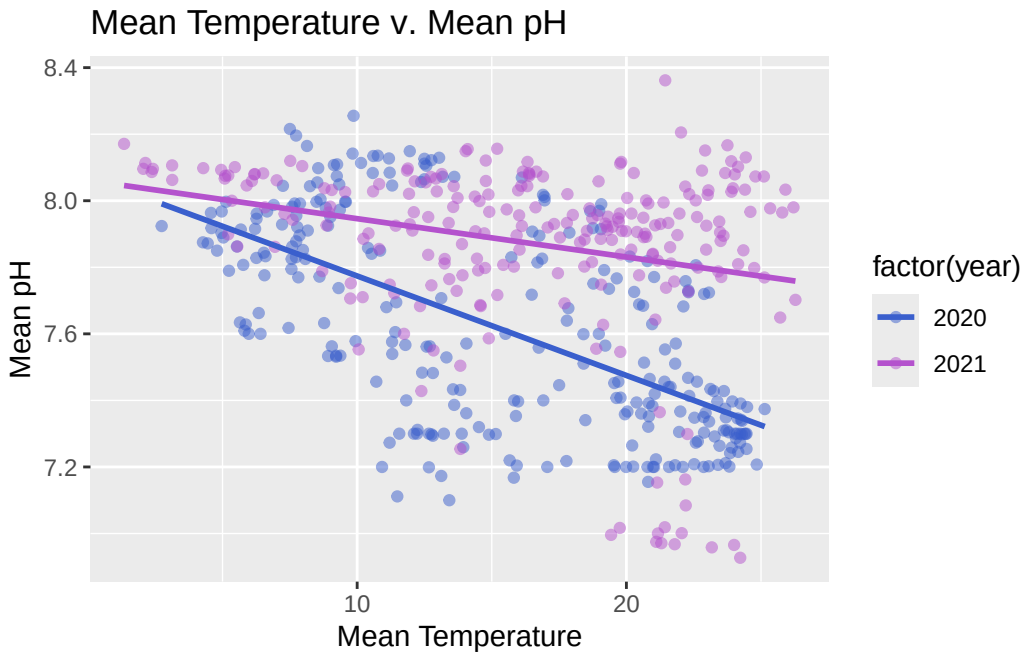
```
ggplot(daily_combined_datasets, aes(x = meanTemp, y = meanpH, color = factor(year), group = 
  geom_jitter(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)+
  labs(title = "Mean Temperature v. Mean pH",
```

```
       x = "Mean Temperature",
       y = "Mean pH")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

```
Warning: Removed 45 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

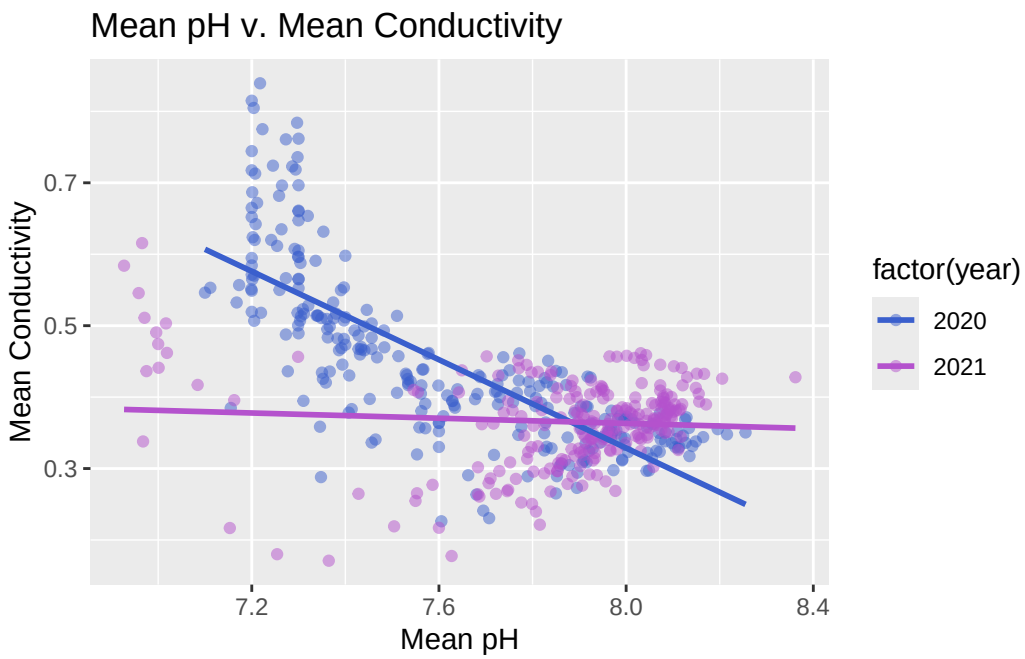
```
Warning: Removed 45 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
ggplot(daily_combined_datasets, aes(x = meanpH, y = meanSpCond, color = factor(year), group =
  geom_jitter(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)+
  labs(title = "Mean pH v. Mean Conductivity",
       x = "Mean pH",
       y = "Mean Conductivity")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

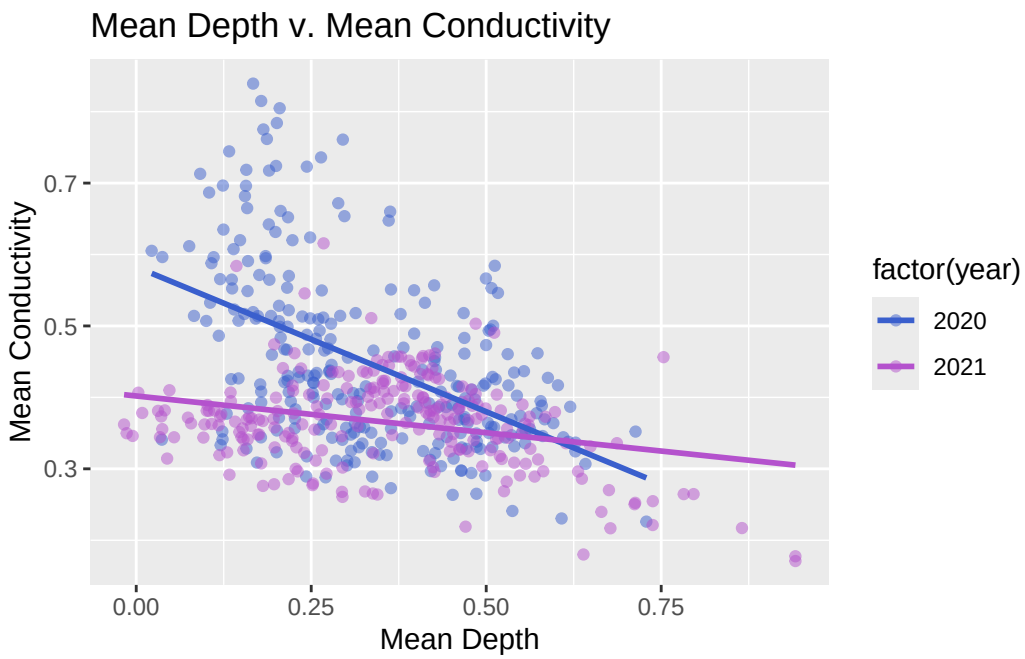
```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 45 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 45 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
ggplot(daily_combined_datasets, aes(x = meanDepth, y = meanSpCond, color = factor(year), grou
  geom_jitter(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)+
  labs(title = "Mean Depth v. Mean Conductivity",
       x = "Mean Depth",
       y = "Mean Conductivity")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

Warning: Removed 45 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 45 rows containing missing values or values outside the scale range
(`geom_point()`).
```

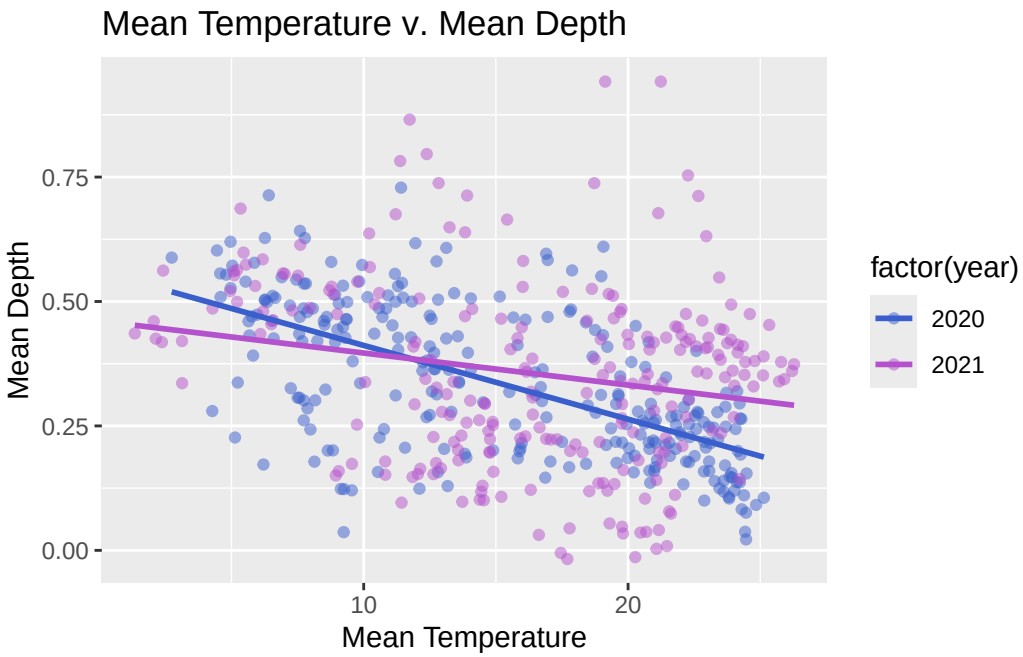



```
ggplot(daily_combined_datasets, aes(x = meanTemp, y = meanDepth, color = factor(year), group
  geom_jitter(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)+
  labs(title = "Mean Temperature v. Mean Depth",
       x = "Mean Temperature",
       y = "Mean Depth")+
  scale_color_manual(values = c("2020" = "royalblue3", "2021" = "mediumorchid3"))
```

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

```
Warning: Removed 45 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 45 rows containing missing values or values outside the scale range
(`geom_point()`).
```


Mean Temperature v. Mean Depth
Mean Depth
0.75
0.50
0.25
0.00
10
20
Mean Temperature
factor(year)
2020
2021