Bivariate Analysis

William Norfolk

11/3/2019

This script runs bivariate/deeper analysis of the water quality dataset to produce additional/complex data visualizations. This script operates using the *processed\_data* file produced via running the *WQprocessing* script located inside the *code* folder and *processing\_code* subfolder loacted within this project.

Start by loading required libraries

library(readxl)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyverse)

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

## v ggplot2 3.2.1 v readr 1.3.1  
## v tibble 2.1.3 v purrr 0.3.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v ggplot2 3.2.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(forcats)  
library(ggthemes)  
library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

library(knitr)  
library(naniar)  
library(broom)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(zoo)

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(ggpubr)

## Loading required package: magrittr

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

Load the data and take a look at the variables.

WQ\_clean\_data <- readRDS("../../data/processed\_data/processeddata.rds")  
  
glimpse(WQ\_clean\_data)

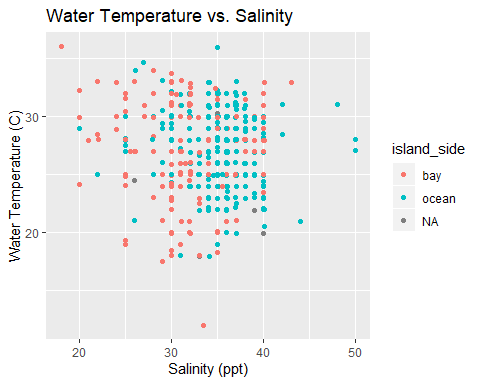
## Observations: 522  
## Variables: 15  
## $ Month <chr> "01", "02", "02", "02", "02", "02", "02", "02...  
## $ Day <chr> "08", "08", "08", "08", "09", "09", "09", "12...  
## $ Year <chr> "16", "16", "16", "16", "16", "16", "16", "16...  
## $ military\_time <dbl> 1415, 1515, 1550, 1555, 1001, 1015, 1022, 103...  
## $ location <chr> "Boat Ramp", "Grecian Dry Rocks", "Grecian Dr...  
## $ instructor\_name <chr> "Katy, Sarah, Driver", "Chelsea", "Katy, Tomm...  
## $ group\_name <chr> "NA", "McLean High School", "McLean High Scho...  
## $ ph <dbl> 8.0, 8.4, 8.2, 8.4, 8.0, 8.0, 8.0, 8.0, 8.4, ...  
## $ ammonia <dbl> 0.00, 0.00, 0.00, 0.00, 0.25, 0.00, 0.00, 0.0...  
## $ dissolved\_oxygen <dbl> 5.0, 4.0, 4.0, 6.0, 8.0, 4.0, 5.0, 6.0, 6.0, ...  
## $ water\_temp <dbl> NA, 23.5, 21.0, 36.0, 18.0, 18.0, 18.0, 18.3,...  
## $ salinity <dbl> 36, 40, 44, 35, 33, 30, 33, 35, 40, 30, 35, 2...  
## $ equipment <chr> "kit", "kit", "kit", "kit", "kit", "kit", "ki...  
## $ island\_side <chr> "ocean", "ocean", "ocean", "ocean", NA, "bay"...  
## $ site\_type <chr> "Seagrass/Mangrove", "Coral Reef", "Coral Ree...

## Bivariate Visualization of Water Quality Parameters

Lets start by taking a look at the five water quality parameters plotted against one another with bivariate visualizations. For this data we will use jitter plots.

temp\_and\_salt <- ggplot(WQ\_clean\_data, aes(x = salinity, y = water\_temp, color = island\_side)) + geom\_jitter() + xlab("Salinity (ppt)") + ylab("Water Temperature (C)") + ggtitle("Water Temperature vs. Salinity")  
  
temp\_and\_salt

## Warning: Removed 26 rows containing missing values (geom\_point).



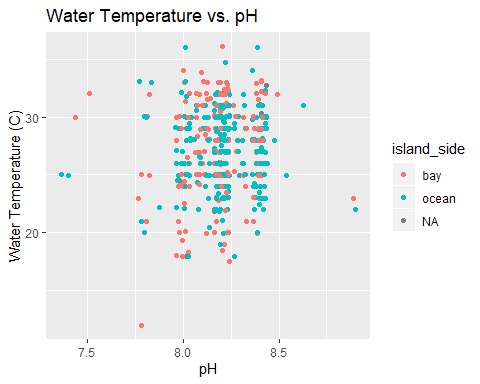
ggsave(filename = "../../results/Bivariate\_Figures/temp\_and\_salt.png", plot = temp\_and\_salt)

## Saving 5 x 4 in image

## Warning: Removed 26 rows containing missing values (geom\_point).

temp\_and\_ph <- ggplot(WQ\_clean\_data, aes(x = ph, y = water\_temp, color = island\_side)) + geom\_jitter() + xlab("pH") + ylab("Water Temperature (C)") + ggtitle("Water Temperature vs. pH")  
  
temp\_and\_ph

## Warning: Removed 17 rows containing missing values (geom\_point).



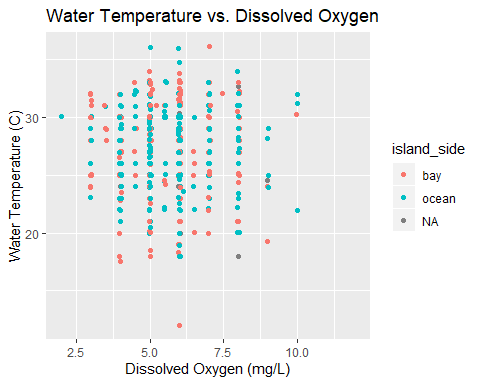
ggsave(filename = "../../results/Bivariate\_Figures/temp\_and\_ph.png", plot = temp\_and\_ph)

## Saving 5 x 4 in image

## Warning: Removed 17 rows containing missing values (geom\_point).

temp\_and\_do <- ggplot(WQ\_clean\_data, aes(x = dissolved\_oxygen, y = water\_temp, color = island\_side)) + geom\_jitter() + xlab("Dissolved Oxygen (mg/L)") + ylab("Water Temperature (C)") + ggtitle("Water Temperature vs. Dissolved Oxygen")  
  
temp\_and\_do

## Warning: Removed 16 rows containing missing values (geom\_point).



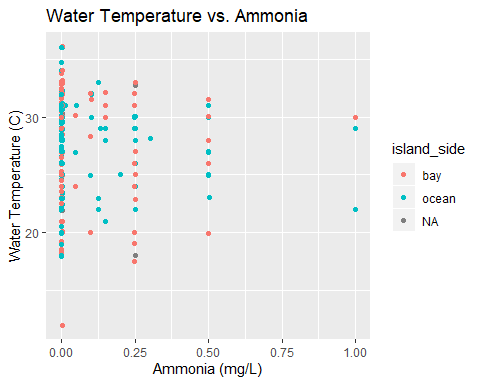
ggsave(filename = "../../results/Bivariate\_Figures/temp\_and\_do.png", plot = temp\_and\_do)

## Saving 5 x 4 in image

## Warning: Removed 16 rows containing missing values (geom\_point).

temp\_and\_ammonia <- ggplot(WQ\_clean\_data, aes(x = ammonia, y = water\_temp, color = island\_side)) + geom\_jitter() + xlab("Ammonia (mg/L)") + ylab("Water Temperature (C)") + ggtitle("Water Temperature vs. Ammonia")  
  
temp\_and\_ammonia

## Warning: Removed 20 rows containing missing values (geom\_point).



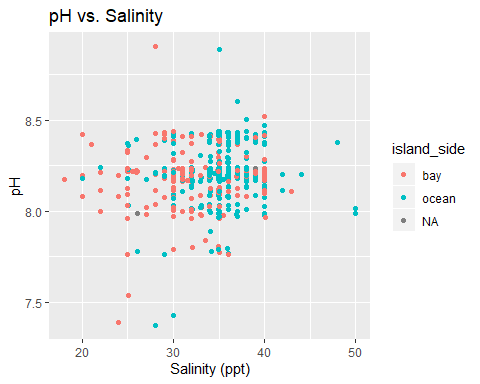
ggsave(filename = "../../results/Bivariate\_Figures/temp\_and\_ammonia.png", plot = temp\_and\_ammonia)

## Saving 5 x 4 in image

## Warning: Removed 20 rows containing missing values (geom\_point).

salt\_and\_ph <- ggplot(WQ\_clean\_data, aes(x = salinity, y = ph, color = island\_side)) + geom\_jitter() + xlab("Salinity (ppt)") + ylab("pH") + ggtitle("pH vs. Salinity")  
  
salt\_and\_ph

## Warning: Removed 20 rows containing missing values (geom\_point).



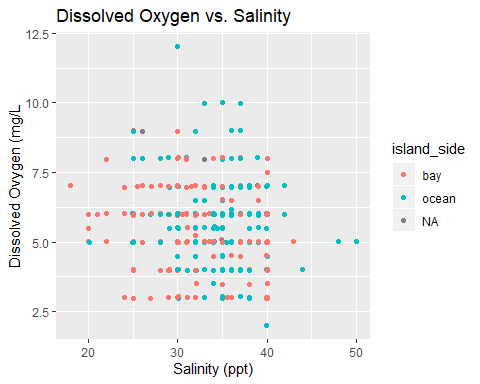
ggsave(filename = "../../results/Bivariate\_Figures/salt\_and\_ph.png", plot = salt\_and\_ph)

## Saving 5 x 4 in image

## Warning: Removed 20 rows containing missing values (geom\_point).

salt\_and\_do <- ggplot(WQ\_clean\_data, aes(x = salinity, y = dissolved\_oxygen, color = island\_side)) + geom\_jitter() + xlab("Salinity (ppt)") + ylab("Dissolved Ovygen (mg/L") + ggtitle("Dissolved Oxygen vs. Salinity")  
  
salt\_and\_do

## Warning: Removed 20 rows containing missing values (geom\_point).



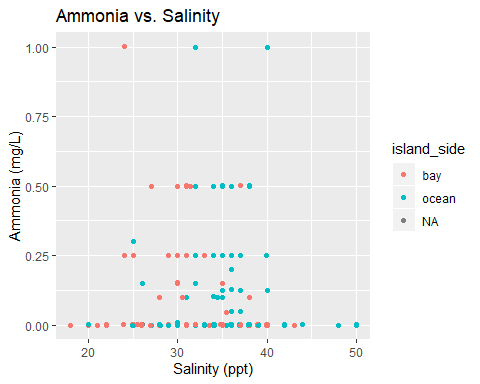
ggsave(filename = "../../results/Bivariate\_Figures/salt\_and\_do.png", plot = salt\_and\_do)

## Saving 5 x 4 in image

## Warning: Removed 20 rows containing missing values (geom\_point).

salt\_and\_ammonia <- ggplot(WQ\_clean\_data, aes(x = salinity, y = ammonia, color = island\_side)) + geom\_jitter() + xlab("Salinity (ppt)") + ylab( "Ammonia (mg/L)") + ggtitle("Ammonia vs. Salinity")  
  
salt\_and\_ammonia

## Warning: Removed 26 rows containing missing values (geom\_point).



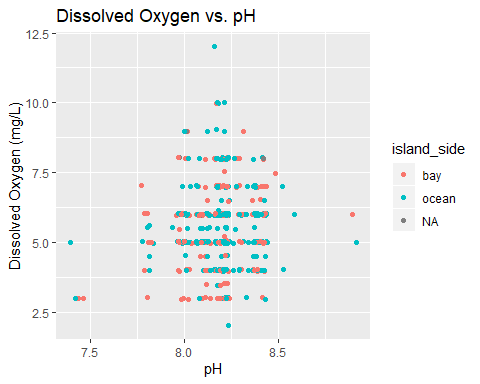
ggsave(filename = "../../results/Bivariate\_Figures/salt\_and\_ammonia.png", plot = salt\_and\_ammonia)

## Saving 5 x 4 in image

## Warning: Removed 26 rows containing missing values (geom\_point).

ph\_and\_do <- ggplot(WQ\_clean\_data, aes(x = ph, y = dissolved\_oxygen, color = island\_side)) + geom\_jitter() + xlab("pH") + ylab("Dissolved Oxygen (mg/L)") + ggtitle("Dissolved Oxygen vs. pH")  
  
ph\_and\_do

## Warning: Removed 9 rows containing missing values (geom\_point).



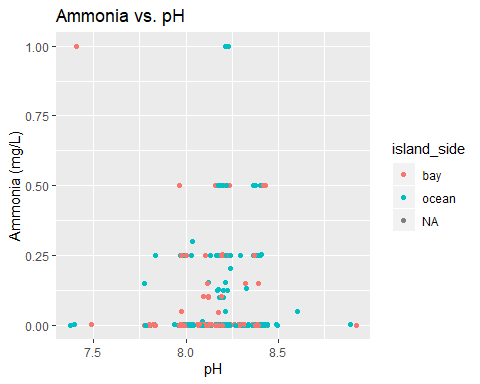
ggsave(filename = "../../results/Bivariate\_Figures/ph\_and\_do.png", plot = ph\_and\_do)

## Saving 5 x 4 in image

## Warning: Removed 9 rows containing missing values (geom\_point).

ph\_and\_ammonia <- ggplot(WQ\_clean\_data, aes(x = ph, y = ammonia, color = island\_side)) + geom\_jitter() + xlab("pH") + ylab("Ammonia (mg/L)") + ggtitle("Ammonia vs. pH")  
  
ph\_and\_ammonia

## Warning: Removed 14 rows containing missing values (geom\_point).



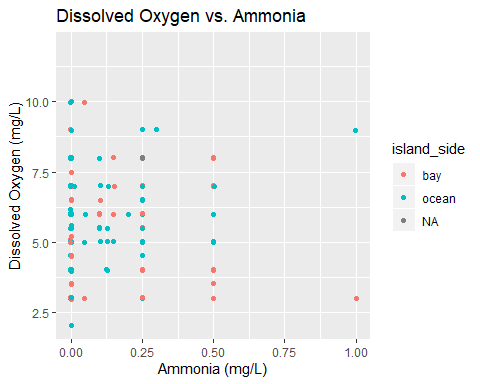
ggsave(filename = "../../results/Bivariate\_Figures/ph\_and\_ammonia.png", plot = ph\_and\_ammonia)

## Saving 5 x 4 in image

## Warning: Removed 14 rows containing missing values (geom\_point).

do\_and\_ammonia <- ggplot(WQ\_clean\_data, aes(x = ammonia, y = dissolved\_oxygen, color = island\_side)) + geom\_jitter() + xlab("Ammonia (mg/L)") + ylab("Dissolved Oxygen (mg/L)") + ggtitle("Dissolved Oxygen vs. Ammonia")  
  
do\_and\_ammonia

## Warning: Removed 14 rows containing missing values (geom\_point).



ggsave(filename = "../../results/Bivariate\_Figures/do\_and\_ammonia.png", plot = do\_and\_ammonia)

## Saving 5 x 4 in image

## Warning: Removed 14 rows containing missing values (geom\_point).

## In Depth Seasonality

It looks like these comparisons are not very informative. This is not suprising due to the geographic spread of the site locations. While ocean waters and bay waters are quite different interms of abiotic conditions, typically the difference is primairly apparent when viewing the range of abiotic conditions rather than individual correlations. These ranges are better visualized with the violin-jitter plots produced in the *Exploratory\_Data\_Analysis\_Location* script.

In our exploratory analysis we reviewed broad-scale seasonal changes (primairly using the average measure for each month). It would be worthwhile to delve deeper into the seasonal aspects of the data to see if there are any additional/stronger/weat=ker correlations we can observe. We will break down the information for each year and predictor to get a more in depth visualization of seasonality.

We will start with monthly data to view seasons.

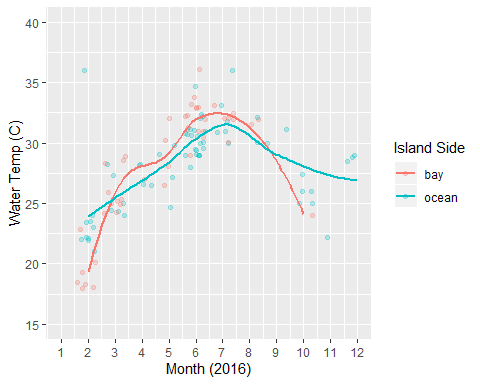
year\_16 <- subset(WQ\_clean\_data, Year == "16")  
year\_17 <- subset(WQ\_clean\_data, Year == "17")  
year\_18 <- subset(WQ\_clean\_data, Year == "18")  
year\_19 <- subset(WQ\_clean\_data, Year == "19")  
  
#To scale all figures to same axis  
year\_16$Month <- as.numeric(as.character(year\_16$Month))  
year\_17$Month <- as.numeric(as.character(year\_17$Month))  
year\_18$Month <- as.numeric(as.character(year\_18$Month))  
year\_19$Month <- as.numeric(as.character(year\_19$Month))  
  
#To get rid of clutter in the figures  
year\_16 <- year\_16[!is.na(year\_16$island\_side), ]  
year\_17 <- year\_17[!is.na(year\_17$island\_side), ]  
year\_18 <- year\_18[!is.na(year\_18$island\_side), ]  
year\_19 <- year\_19[!is.na(year\_19$island\_side), ]

month\_vs\_temp\_16 <- ggplot(year\_16, aes(x = Month, y = water\_temp, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2016)") + ylab("Water Temp (C)") + labs(color = "Island Side")  
  
  
month\_vs\_temp\_16

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 3 rows containing non-finite values (stat\_smooth).

## Warning: Removed 3 rows containing missing values (geom\_point).

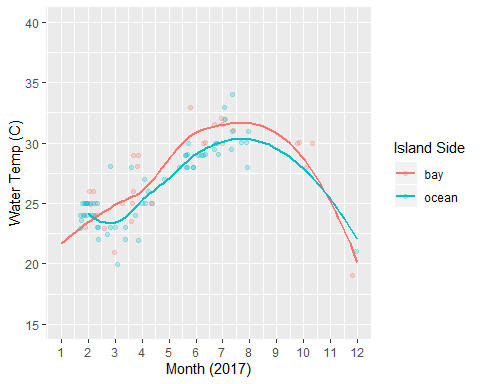


month\_vs\_temp\_17 <- ggplot(year\_17, aes(x = Month, y = water\_temp, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2017)") + ylab("Water Temp (C)") + labs(color = "Island Side")  
  
month\_vs\_temp\_17

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 5 rows containing non-finite values (stat\_smooth).

## Warning: Removed 9 rows containing missing values (geom\_point).

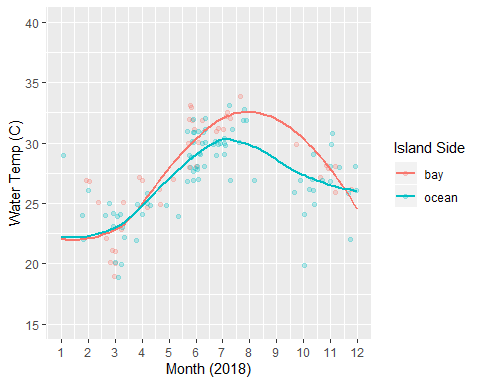


month\_vs\_temp\_18 <- ggplot(year\_18, aes(x = Month, y = water\_temp, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2018)") + ylab("Water Temp (C)") + labs(color = "Island Side")  
  
month\_vs\_temp\_18

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (stat\_smooth).

## Warning: Removed 8 rows containing missing values (geom\_point).

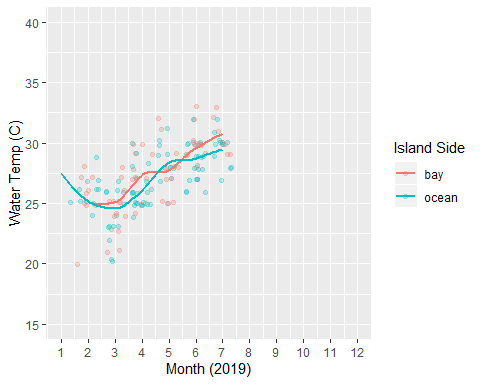


month\_vs\_temp\_19 <- ggplot(year\_19, aes(x = Month, y = water\_temp, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2019)") + ylab("Water Temp (C)") + labs(color = "Island Side")  
  
  
month\_vs\_temp\_19

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 4 rows containing non-finite values (stat\_smooth).

## Warning: Removed 4 rows containing missing values (geom\_point).



temp\_by\_year <- ggarrange(month\_vs\_temp\_16, month\_vs\_temp\_17, month\_vs\_temp\_18, month\_vs\_temp\_19, nrow = 2, ncol = 2, common.legend = TRUE, legend = "bottom")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 3 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 3 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 5 rows containing non-finite values (stat\_smooth).

## Warning: Removed 7 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

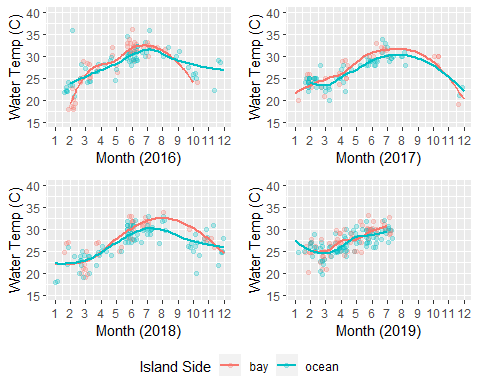
## Warning: Removed 1 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 7 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 4 rows containing non-finite values (stat\_smooth).

## Warning: Removed 4 rows containing missing values (geom\_point).

temp\_by\_year



ggsave(filename = "../../results/Bivariate\_Figures/temp\_by\_year.png", plot = temp\_by\_year)

## Saving 5 x 4 in image

From the looks of these figures it seems that temperature fluctuates in a manner that is to be expected (i.e. cold in the winter and warmer in the summer). It is also pretty consistent that the temperature range is greater for the bayside waters than the ocean side waters which is consistent with the previously generated violin plots from *Exploratory\_Data\_Analysis\_Seasonal*. Aside from basic seasonality there does not appear to be any additional trends or oddities in the temperature data.

month\_vs\_sal\_16 <- ggplot(year\_16, aes(x = Month, y = salinity, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2016)") + ylab("Salinity (ppt)") + labs(color = "Island Side")  
  
  
month\_vs\_sal\_16

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

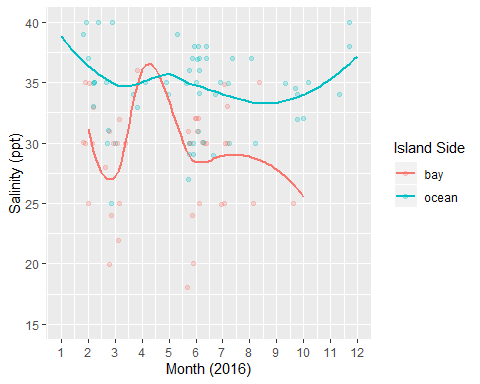
## Warning: Removed 8 rows containing non-finite values (stat\_smooth).

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : pseudoinverse used at 4

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : reciprocal condition number 3.4511e-017

## Warning: Removed 17 rows containing missing values (geom\_point).

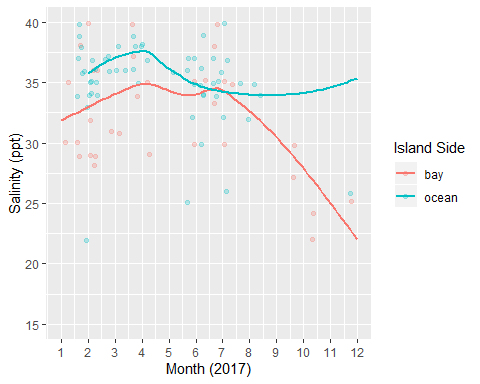


month\_vs\_sal\_17 <- ggplot(year\_17, aes(x = Month, y = salinity, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2017)") + ylab("Salinity (ppt)") + labs(color = "Island Side")  
  
month\_vs\_sal\_17

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 4 rows containing non-finite values (stat\_smooth).

## Warning: Removed 13 rows containing missing values (geom\_point).

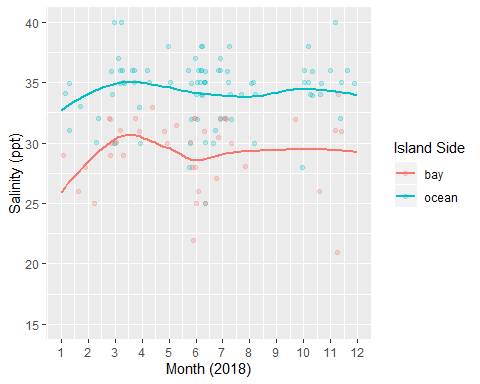


month\_vs\_sal\_18 <- ggplot(year\_18, aes(x = Month, y = salinity, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2018)") + ylab("Salinity (ppt)") + labs(color = "Island Side")  
  
month\_vs\_sal\_18

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 3 rows containing non-finite values (stat\_smooth).

## Warning: Removed 10 rows containing missing values (geom\_point).

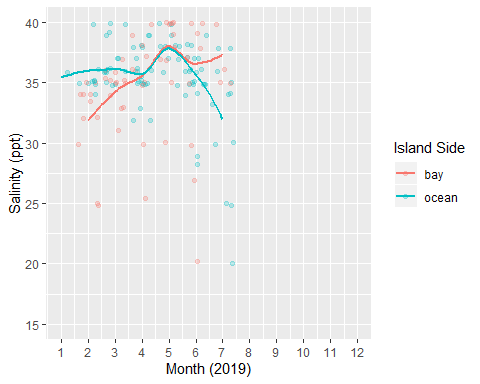


month\_vs\_sal\_19 <- ggplot(year\_19, aes(x = Month, y = salinity, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(15, 40)) + geom\_smooth(se = FALSE) + xlab("Month (2019)") + ylab("Salinity (ppt)") + labs(color = "Island Side")  
  
month\_vs\_sal\_19

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 7 rows containing non-finite values (stat\_smooth).

## Warning: Removed 15 rows containing missing values (geom\_point).



sal\_by\_year <- ggarrange(month\_vs\_sal\_16, month\_vs\_sal\_17, month\_vs\_sal\_18, month\_vs\_sal\_19, nrow = 2, ncol = 2, common.legend = TRUE, legend = "bottom")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 8 rows containing non-finite values (stat\_smooth).

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : pseudoinverse used at 4

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : reciprocal condition number 3.4511e-017

## Warning: Removed 13 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 8 rows containing non-finite values (stat\_smooth).

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : pseudoinverse used at 4

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : reciprocal condition number 3.4511e-017

## Warning: Removed 14 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 4 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 14 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 3 rows containing non-finite values (stat\_smooth).

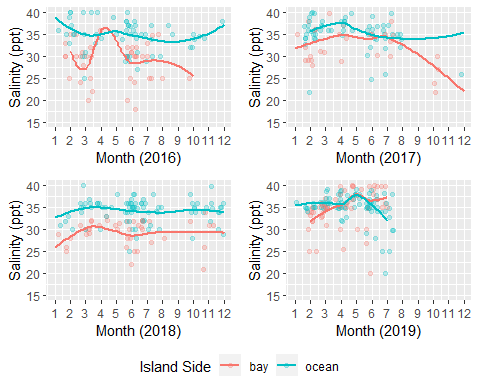
## Warning: Removed 8 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 7 rows containing non-finite values (stat\_smooth).

## Warning: Removed 20 rows containing missing values (geom\_point).

sal\_by\_year



ggsave(filename = "../../results/Bivariate\_Figures/sal\_by\_year.png", plot = sal\_by\_year)

## Saving 5 x 4 in image

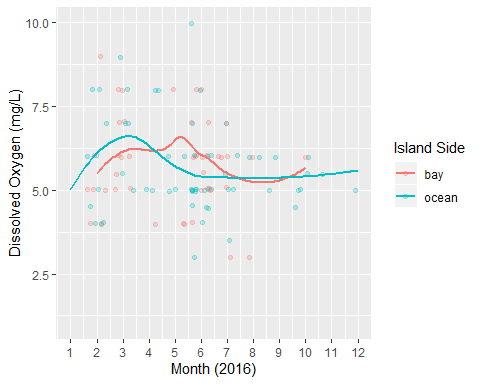
Salinity looks pretty consistent with the exploratory plots as well. Ocean side water are consistently higher in salinity than the bay side, which is expected since the bay is an estuary. Additionally, there is a wider range of salinity on the bay side, which is consistent with exploratory analysis. There is an odd jump in the salinity around April-May for 2016 and 2019. This may simply be a result of few observations taken during that time or it is possibly due to a very low rainfall year (salinitly tends to spike in the bay when rainfall is reduced). In 2017, there is a distinct salinity drop in bayside locations. The timing of this drop is correlated with the timing of Hurricane Irma landfall. We will explore more details of Irma impacts in the *Hurricane\_Irma\_Analysis* script.

month\_vs\_do\_16 <- ggplot(year\_16, aes(x = Month, y = dissolved\_oxygen, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(1, 10)) + geom\_smooth(se = FALSE) + xlab("Month (2016)") + ylab("Dissolved Oxygen (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_do\_16

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 7 rows containing missing values (geom\_point).

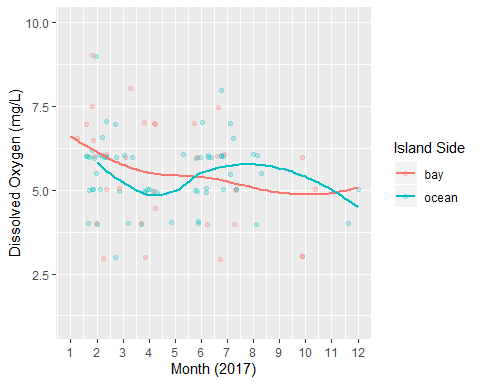


month\_vs\_do\_17 <- ggplot(year\_17, aes(x = Month, y = dissolved\_oxygen, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(1, 10)) + geom\_smooth(se = FALSE) + xlab("Month (2017)") + ylab("Dissolved Oxygen (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_do\_17

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 6 rows containing missing values (geom\_point).

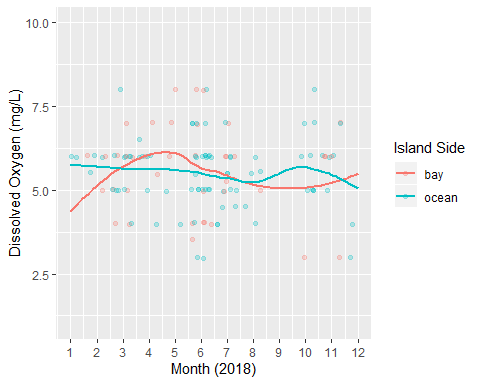


month\_vs\_do\_18 <- ggplot(year\_18, aes(x = Month, y = dissolved\_oxygen, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(1, 10)) + geom\_smooth(se = FALSE) + xlab("Month (2018)") + ylab("Dissolved Oxygen (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_do\_18

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (stat\_smooth).

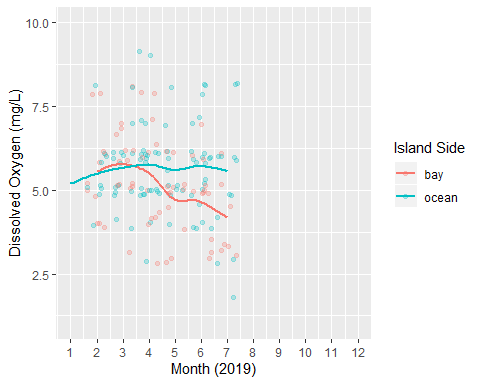
## Warning: Removed 10 rows containing missing values (geom\_point).



month\_vs\_do\_19 <- ggplot(year\_19, aes(x = Month, y = dissolved\_oxygen, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(1, 10)) + geom\_smooth(se = FALSE) + xlab("Month (2019)") + ylab("Dissolved Oxygen (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_do\_19

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing missing values (geom\_point).



do\_by\_year <- ggarrange(month\_vs\_do\_16, month\_vs\_do\_17, month\_vs\_do\_18, month\_vs\_do\_19, nrow = 2, ncol = 2, common.legend = TRUE, legend = "bottom")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 6 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 6 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 5 rows containing missing values (geom\_point).

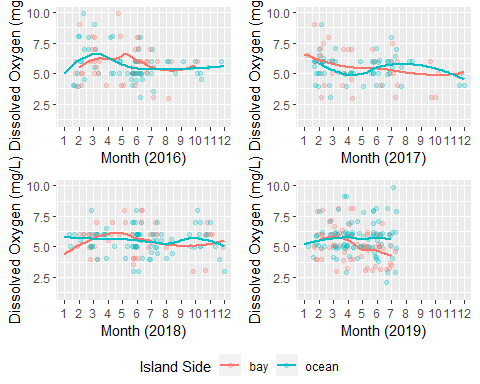
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (stat\_smooth).

## Warning: Removed 9 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

do\_by\_year



ggsave(filename = "../../results/Bivariate\_Figures/do\_by\_year.png", plot = do\_by\_year)

## Saving 5 x 4 in image

Dissolved oxygen does not appear to show any distict seasonal patterns for either the bay side or the ocean side. There is a somewhat notable drop in the dissolved oxygen levels towards the end of the 2019 data for bayside samples, however or dataset ends in July of that year so it is unclear if this is just a standard fluctuation or not. It would be advised to monitor that situation if the pattern continued.

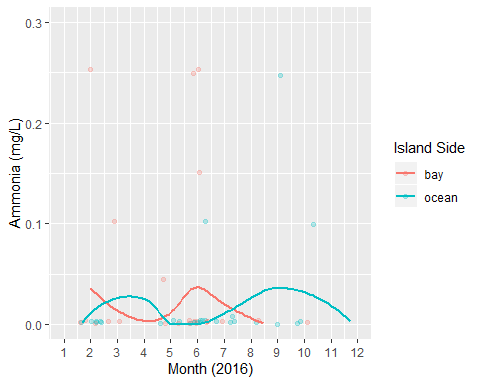
month\_vs\_amm\_16 <- ggplot(year\_16, aes(x = Month, y = ammonia, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(0, 0.3)) + geom\_smooth(se = FALSE) + xlab("Month (2016)") + ylab("Ammonia (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_amm\_16

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 7 rows containing non-finite values (stat\_smooth).

## Warning: Removed 62 rows containing missing values (geom\_point).

## Warning: Removed 29 rows containing missing values (geom\_smooth).



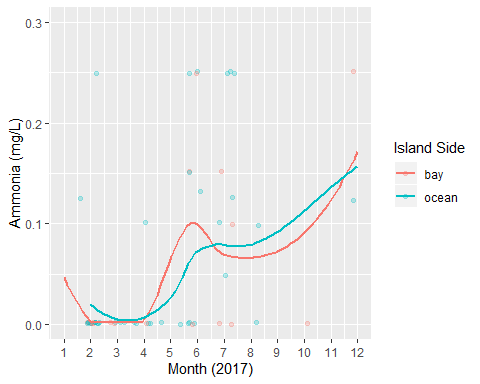
month\_vs\_amm\_17 <- ggplot(year\_17, aes(x = Month, y = ammonia, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(0, 0.3)) + geom\_smooth(se = FALSE) + xlab("Month (2017)") + ylab("Ammonia (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_amm\_17

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 8 rows containing non-finite values (stat\_smooth).

## Warning: Removed 47 rows containing missing values (geom\_point).

## Warning: Removed 13 rows containing missing values (geom\_smooth).



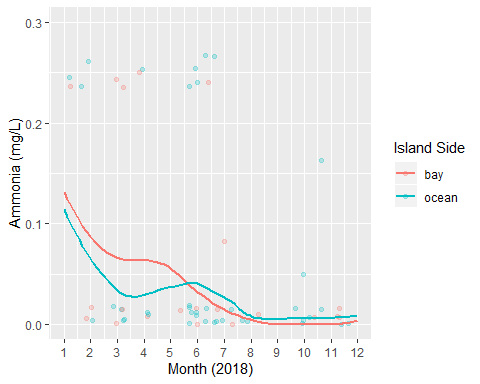
month\_vs\_amm\_18 <- ggplot(year\_18, aes(x = Month, y = ammonia, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(0, 0.3)) + geom\_smooth(se = FALSE) + xlab("Month (2018)") + ylab("Ammonia (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_amm\_18

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 11 rows containing non-finite values (stat\_smooth).

## Warning: Removed 73 rows containing missing values (geom\_point).

## Warning: Removed 18 rows containing missing values (geom\_smooth).



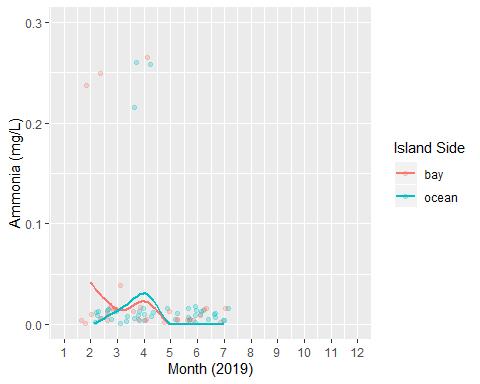
month\_vs\_amm\_19 <- ggplot(year\_19, aes(x = Month, y = ammonia, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(0, 0.3)) + geom\_smooth(se = FALSE) + xlab("Month (2019)") + ylab("Ammonia (mg/L)") + labs(color = "Island Side")  
  
month\_vs\_amm\_19

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 5 rows containing non-finite values (stat\_smooth).

## Warning: Removed 85 rows containing missing values (geom\_point).

## Warning: Removed 57 rows containing missing values (geom\_smooth).



amm\_by\_year <- ggarrange(month\_vs\_amm\_16, month\_vs\_amm\_17, month\_vs\_amm\_18, month\_vs\_amm\_19, nrow = 2, ncol = 2, common.legend = TRUE, legend = "bottom")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 7 rows containing non-finite values (stat\_smooth).

## Warning: Removed 62 rows containing missing values (geom\_point).

## Warning: Removed 29 rows containing missing values (geom\_smooth).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 7 rows containing non-finite values (stat\_smooth).

## Warning: Removed 54 rows containing missing values (geom\_point).

## Warning: Removed 29 rows containing missing values (geom\_smooth).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 8 rows containing non-finite values (stat\_smooth).

## Warning: Removed 48 rows containing missing values (geom\_point).

## Warning: Removed 13 rows containing missing values (geom\_smooth).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 11 rows containing non-finite values (stat\_smooth).

## Warning: Removed 67 rows containing missing values (geom\_point).

## Warning: Removed 18 rows containing missing values (geom\_smooth).

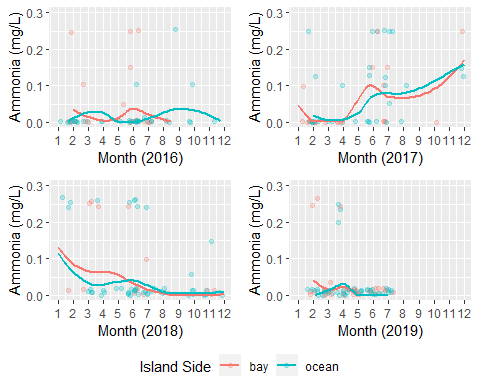
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 5 rows containing non-finite values (stat\_smooth).

## Warning: Removed 85 rows containing missing values (geom\_point).

## Warning: Removed 57 rows containing missing values (geom\_smooth).

amm\_by\_year



ggsave(filename = "../../results/Bivariate\_Figures/amm\_by\_year.png", plot = amm\_by\_year)

## Saving 5 x 4 in image

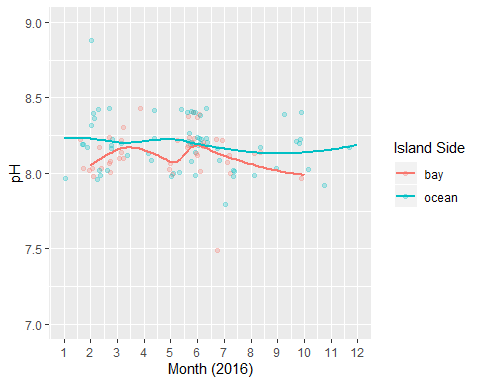
Ammonia appears quite interesting. Most of the observations are fairly consistent with very low to negligable ammonia levels in the water. There is a distinct increase in fall 2017 which is consistent with the timing of Hurricane Irma landfall.

month\_vs\_ph\_16 <- ggplot(year\_16, aes(x = Month, y = ph, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(7, 9)) + geom\_smooth(se = FALSE) + xlab("Month (2016)") + ylab("pH") + labs(color = "Island Side")  
  
month\_vs\_ph\_16

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

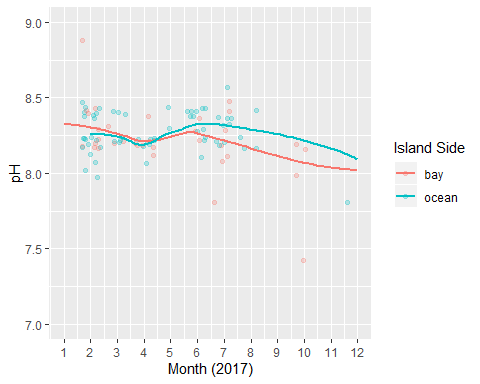
## Warning: Removed 4 rows containing missing values (geom\_point).



month\_vs\_ph\_17 <- ggplot(year\_17, aes(x = Month, y = ph, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(7, 9)) + geom\_smooth(se = FALSE) + xlab("Month (2017)") + ylab("pH") + labs(color = "Island Side")  
  
month\_vs\_ph\_17

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 6 rows containing missing values (geom\_point).

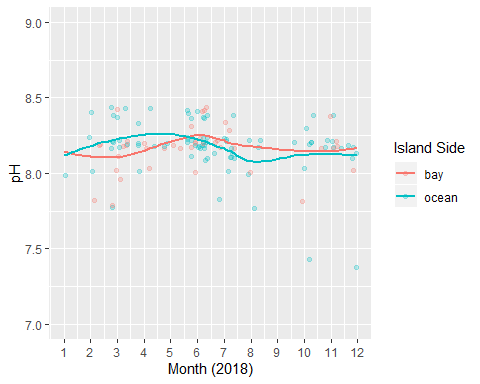


month\_vs\_ph\_18 <- ggplot(year\_18, aes(x = Month, y = ph, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(7, 9)) + geom\_smooth(se = FALSE) + xlab("Month (2018)") + ylab("pH") + labs(color = "Island Side")  
  
month\_vs\_ph\_18

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 8 rows containing missing values (geom\_point).

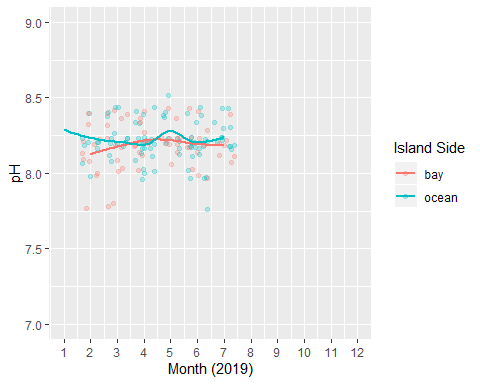


month\_vs\_ph\_19 <- ggplot(year\_19, aes(x = Month, y = ph, color = island\_side)) + geom\_jitter(alpha = 0.25) + scale\_x\_continuous(limits=c(1, 12), breaks = 1:12) + scale\_y\_continuous(limits=c(7, 9)) + geom\_smooth(se = FALSE) + xlab("Month (2019)") + ylab("pH") + labs(color = "Island Side")  
  
month\_vs\_ph\_19

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 3 rows containing missing values (geom\_point).



ph\_by\_year <- ggarrange(month\_vs\_ph\_16, month\_vs\_ph\_17, month\_vs\_ph\_18, month\_vs\_ph\_19, nrow = 2, ncol = 2, common.legend = TRUE, legend = "bottom")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 5 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 4 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 6 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

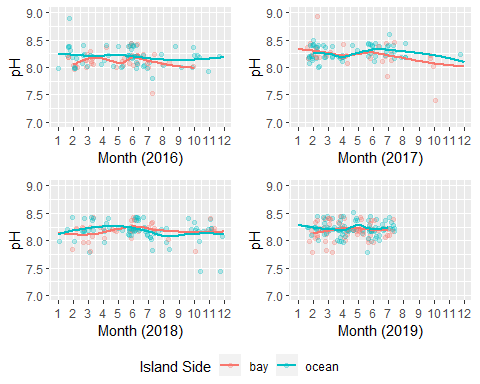
## Warning: Removed 12 rows containing missing values (geom\_point).

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 3 rows containing missing values (geom\_point).

ph\_by\_year



ggsave(filename = "../../results/Bivariate\_Figures/ph\_by\_year.png", plot = ph\_by\_year)

## Saving 5 x 4 in image

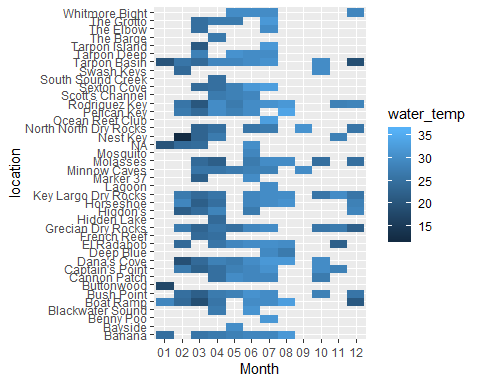
pH visualization does not show any distinct patterns across various seasons of measurement. This is consistent with expected results due to the fact that both the ocean and bay are strongly influenced by the alkaline limestone bedrock of the Keys.

## Seasonality With Raster Plots

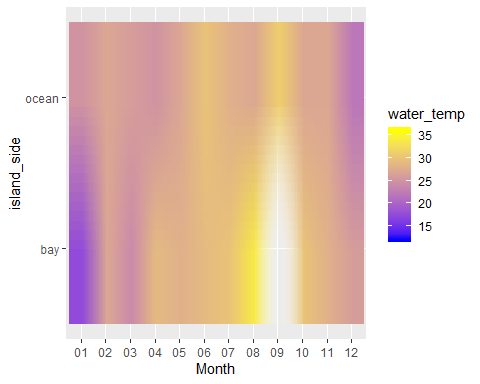
Next lets visualize the data using raster plots to see if we can see any differences in the correlations or additional patterns. To avoid clutter we will drop NAs from the individual predictors since they will not provide any information in the raster.

temp\_drop\_na <- WQ\_clean\_data[!is.na(WQ\_clean\_data$water\_temp), ]  
sal\_drop\_na <- WQ\_clean\_data[!is.na(WQ\_clean\_data$salinity), ]  
do\_drop\_na <- WQ\_clean\_data[!is.na(WQ\_clean\_data$dissolved\_oxygen), ]  
ph\_drop\_na <- WQ\_clean\_data[!is.na(WQ\_clean\_data$ph), ]  
amm\_drop\_na <- WQ\_clean\_data[!is.na(WQ\_clean\_data$ammonia), ]

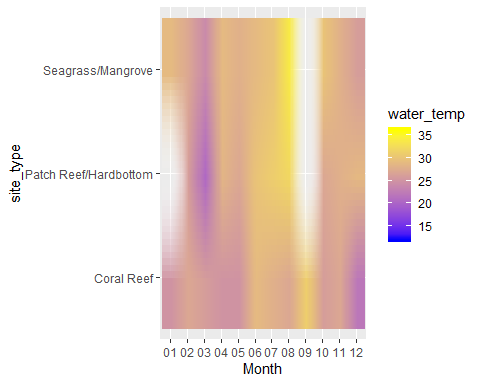
temp\_drop\_na\_1 <- temp\_drop\_na[!is.na(temp\_drop\_na$island\_side), ]  
temp\_drop\_na\_2 <- temp\_drop\_na\_1[!is.na(temp\_drop\_na\_1$site\_type), ]  
  
raster\_by\_site\_temp <- ggplot(temp\_drop\_na, aes(x = Month, y = location, fill = water\_temp)) + geom\_raster()  
  
raster\_by\_site\_temp



raster\_by\_island\_side\_temp <- ggplot(temp\_drop\_na\_1, aes(x = Month, y = island\_side, fill = water\_temp)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_island\_side\_temp



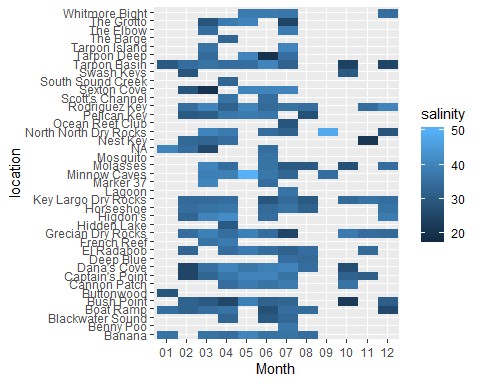
raster\_by\_site\_type\_temp <- ggplot(temp\_drop\_na\_2, aes(x = Month, y = site\_type, fill = water\_temp)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_site\_type\_temp



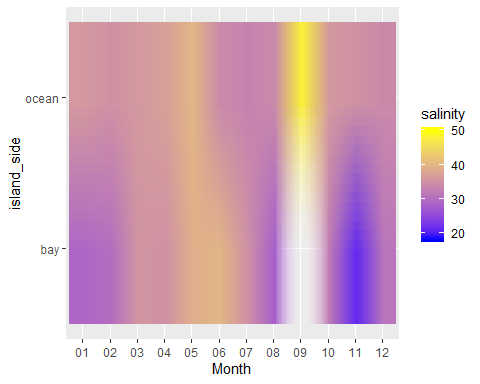
ggsave(filename = "../../results/Bivariate\_Figures/raster\_by\_site\_type\_temp.png", plot = raster\_by\_site\_type\_temp)

## Saving 5 x 4 in image

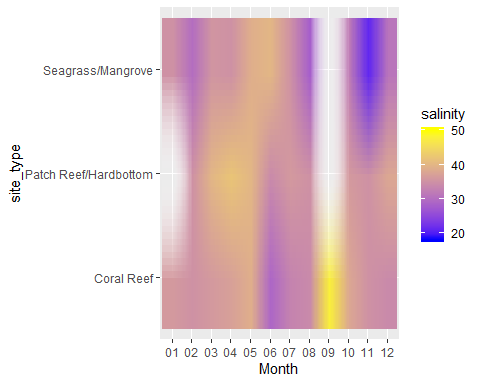
sal\_drop\_na\_1 <- sal\_drop\_na[!is.na(sal\_drop\_na$island\_side), ]  
sal\_drop\_na\_2 <- sal\_drop\_na\_1[!is.na(sal\_drop\_na\_1$site\_type), ]  
  
raster\_by\_site\_sal <- ggplot(sal\_drop\_na, aes(x = Month, y = location, fill = salinity)) + geom\_raster()  
  
raster\_by\_site\_sal



raster\_by\_island\_side\_sal <- ggplot(sal\_drop\_na\_1, aes(x = Month, y = island\_side, fill = salinity)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_island\_side\_sal



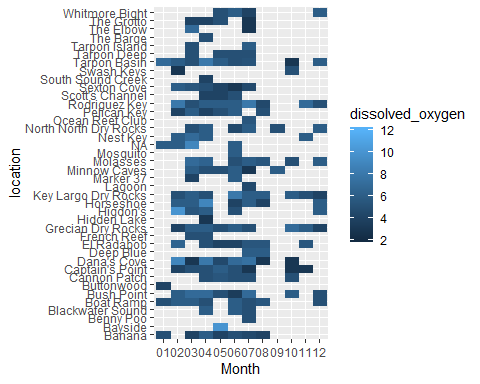
raster\_by\_site\_type\_sal <- ggplot(sal\_drop\_na\_2, aes(x = Month, y = site\_type, fill = salinity)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_site\_type\_sal



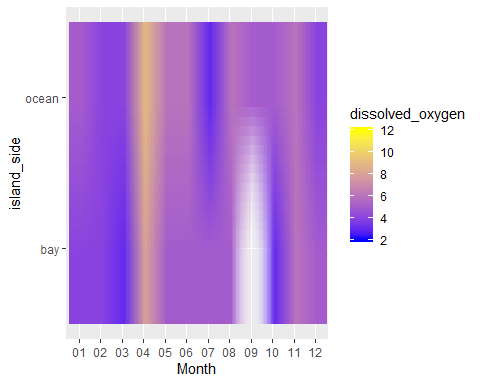
ggsave(filename = "../../results/Bivariate\_Figures/raster\_by\_site\_type\_sal.png", plot = raster\_by\_site\_type\_sal)

## Saving 5 x 4 in image

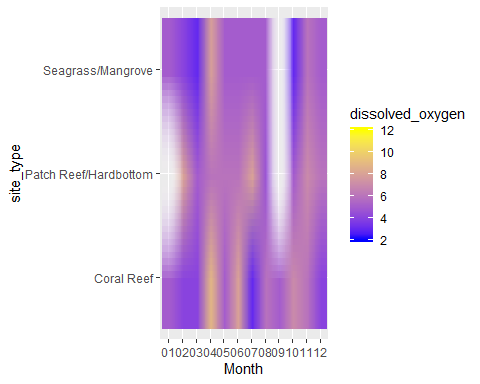
do\_drop\_na\_1 <- do\_drop\_na[!is.na(do\_drop\_na$island\_side), ]  
do\_drop\_na\_2 <- do\_drop\_na\_1[!is.na(do\_drop\_na\_1$site\_type), ]  
  
raster\_by\_site\_do <- ggplot(do\_drop\_na, aes(x = Month, y = location, fill = dissolved\_oxygen)) + geom\_raster()  
  
raster\_by\_site\_do



raster\_by\_island\_side\_do <- ggplot(do\_drop\_na\_1, aes(x = Month, y = island\_side, fill = dissolved\_oxygen)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_island\_side\_do



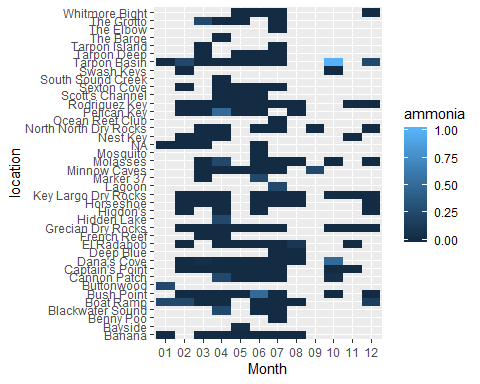
raster\_by\_site\_type\_do <- ggplot(do\_drop\_na\_2, aes(x = Month, y = site\_type, fill = dissolved\_oxygen)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_site\_type\_do



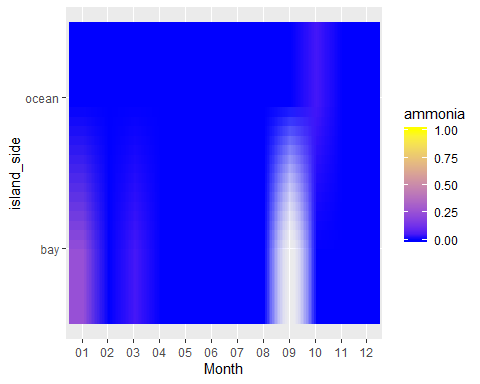
ggsave(filename = "../../results/Bivariate\_Figures/raster\_by\_site\_type\_do.png", plot = raster\_by\_site\_type\_do)

## Saving 5 x 4 in image

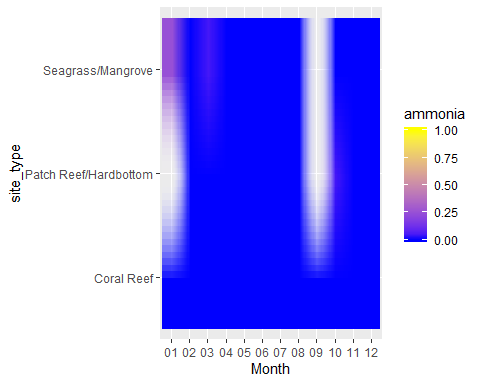
amm\_drop\_na\_1 <- amm\_drop\_na[!is.na(amm\_drop\_na$island\_side), ]  
amm\_drop\_na\_2 <- amm\_drop\_na\_1[!is.na(amm\_drop\_na\_1$site\_type), ]  
  
raster\_by\_site\_amm <- ggplot(amm\_drop\_na, aes(x = Month, y = location, fill = ammonia)) + geom\_raster()  
  
raster\_by\_site\_amm



raster\_by\_island\_side\_amm <- ggplot(amm\_drop\_na\_1, aes(x = Month, y = island\_side, fill = ammonia)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_island\_side\_amm



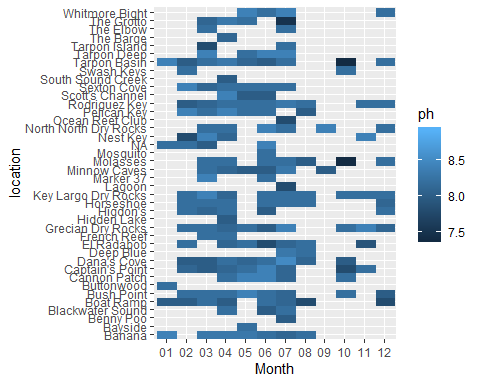
raster\_by\_site\_type\_amm <- ggplot(amm\_drop\_na\_2, aes(x = Month, y = site\_type, fill = ammonia)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_site\_type\_amm



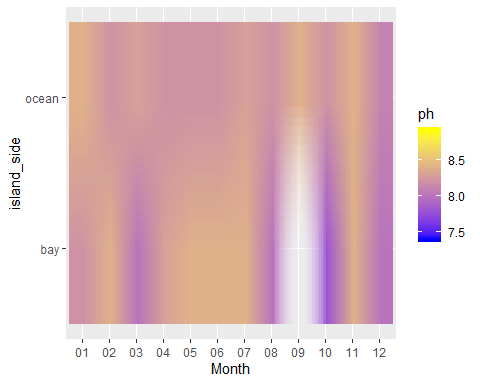
ggsave(filename = "../../results/Bivariate\_Figures/raster\_by\_site\_type\_amm.png", plot = raster\_by\_site\_type\_amm)

## Saving 5 x 4 in image

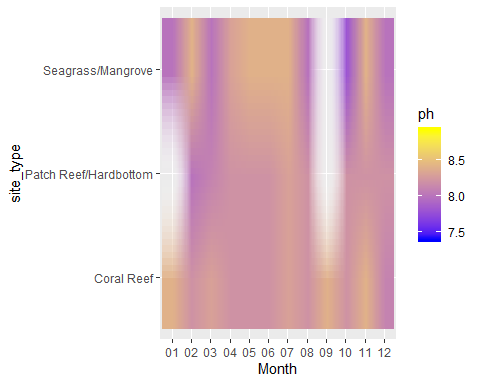
ph\_drop\_na\_1 <- ph\_drop\_na[!is.na(ph\_drop\_na$island\_side), ]  
ph\_drop\_na\_2 <- ph\_drop\_na\_1[!is.na(ph\_drop\_na\_1$site\_type), ]  
  
raster\_by\_site\_ph <- ggplot(ph\_drop\_na, aes(x = Month, y = location, fill = ph)) + geom\_raster()  
  
raster\_by\_site\_ph



raster\_by\_island\_side\_ph <- ggplot(ph\_drop\_na\_1, aes(x = Month, y = island\_side, fill = ph)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_island\_side\_ph



raster\_by\_site\_type\_ph <- ggplot(ph\_drop\_na\_2, aes(x = Month, y = site\_type, fill = ph)) + geom\_raster(interpolate = TRUE) + scale\_fill\_gradientn(colors = c("blue", "yellow"))  
  
raster\_by\_site\_type\_ph



ggsave(filename = "../../results/Bivariate\_Figures/raster\_by\_site\_type\_ph.png", plot = raster\_by\_site\_type\_ph)

## Saving 5 x 4 in image

Unsuprisingly, the raster visualizations show results consistent with the individual jitter-smooth plots above. Most of the distinct patterns are easily visualized using the raster images, however the rasters strongly emphasize the missing data values form our dataset more than any of the actual predictor information. There is a large piece of missing data in August (primairly from Seagrass/Mangrove sites) is is due to the fact that the sampling facility is typically closed for the majority of this month, and the few trips that are operated at this time tend to be reef trips since the weather is typically very good. Additionally, there is a data gap for Patch Reef/Hardbottom locations in January. Is likely due to the fact that winter winds typicaly make patch/hardbottom site more difficult to access safely at this time.

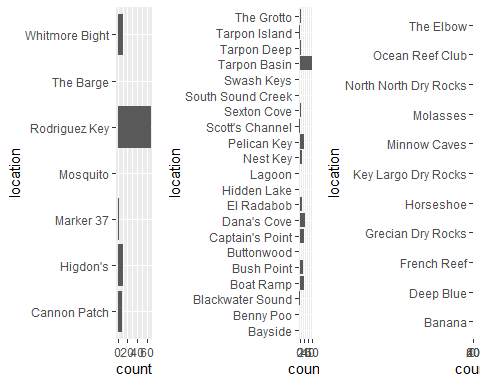
Though visualy appealing, the line graphs produced above show data patterns that are more consistent with the actual measures of interest thus rasters will be relegated to supplemental information.

## Most Frequent Site Visited By Type

Lastly in our bivariate analysis we will zoom into three specific locations that have the most recorded observations of Seagrass/Mangrove, Patch Reef/Hardbottom, and Coral Reef site types to visualize specific seasonal impacts on these locations.

First we must determine the most abundant locations of each type and subset the data.

patch\_hb\_subset <- subset(WQ\_clean\_data, site\_type == "Patch Reef/Hardbottom")  
sg\_mg\_subset <- subset(WQ\_clean\_data, site\_type == "Seagrass/Mangrove")  
coral\_reef\_subset <- subset(WQ\_clean\_data, site\_type == "Coral Reef")  
  
patch\_plot <- ggplot(patch\_hb\_subset, aes(x = location)) + geom\_bar() + coord\_flip()  
sg\_mg\_plot <- ggplot(sg\_mg\_subset, aes(x = location)) + geom\_bar() + coord\_flip()  
coral\_plot <- ggplot(coral\_reef\_subset, aes(x = location)) + geom\_bar() + coord\_flip()  
  
grid.arrange(patch\_plot, sg\_mg\_plot, coral\_plot, nrow = 1)



So it looks like our most numerous observation for Patch Reef/Hardbottom is Rodriguez Key, Seagrass/Mangrove is Tarpon Basin, and Coral Reef is Grecian Dry Rocks. This is unsuprising as all of these sites are in close proximity to the marina and are typically better sheltered on windy days, thus permitting more frequent access.

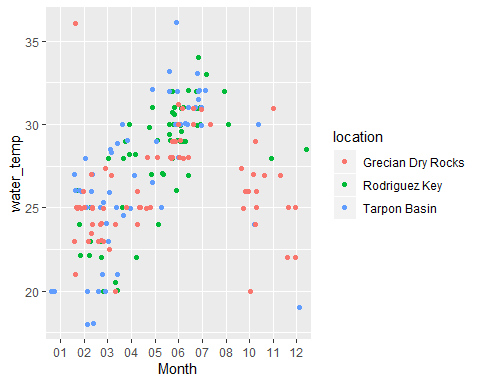
Let’s subset these locations and visualize.

roddy\_subset <- subset(WQ\_clean\_data, location == "Rodriguez Key")  
tarpon\_subset <- subset(WQ\_clean\_data, location == "Tarpon Basin")  
grecian\_subset <- subset(WQ\_clean\_data, location == "Grecian Dry Rocks")  
  
group\_subset <- rbind(roddy\_subset, tarpon\_subset, grecian\_subset)

We will start by grouping them all togeather with each of the five predictors and then compare seasonal fluctuations across the entire sampling timeframe.

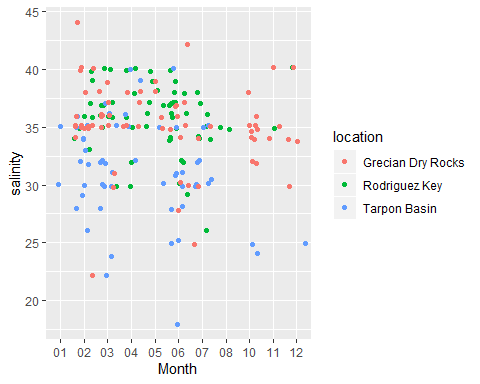
freq\_site\_temp <- ggplot(group\_subset, aes(x = Month, y = water\_temp, color = location)) + geom\_jitter()  
  
freq\_site\_temp

## Warning: Removed 3 rows containing missing values (geom\_point).



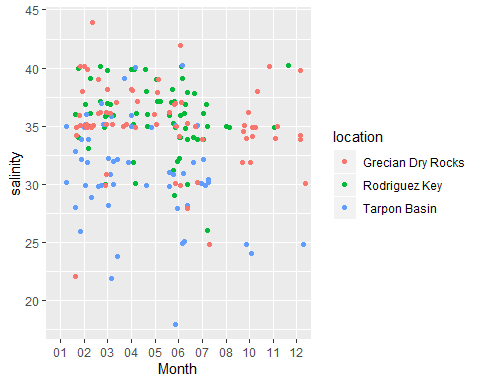
freq\_site\_sal <- ggplot(group\_subset, aes(x = Month, y = salinity, color = location)) + geom\_jitter()  
  
freq\_site\_sal

## Warning: Removed 7 rows containing missing values (geom\_point).



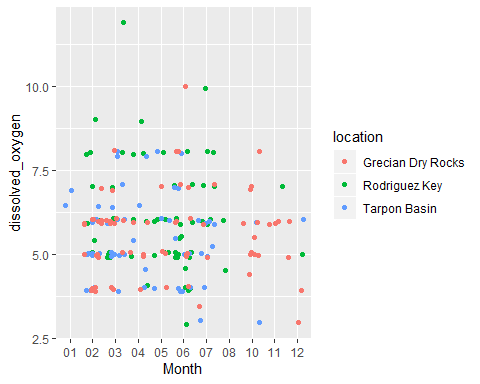
freq\_site\_amm <- ggplot(group\_subset, aes(x = Month, y = ammonia, color = location)) + geom\_jitter()  
  
freq\_site\_sal

## Warning: Removed 7 rows containing missing values (geom\_point).



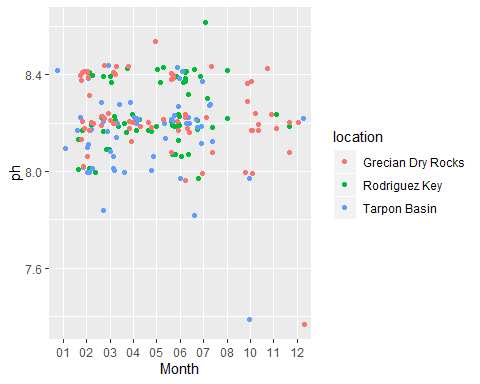
freq\_site\_do <- ggplot(group\_subset, aes(x = Month, y = dissolved\_oxygen, color = location)) + geom\_jitter()  
  
freq\_site\_do

## Warning: Removed 3 rows containing missing values (geom\_point).



freq\_site\_ph <- ggplot(group\_subset, aes(x = Month, y = ph, color = location)) + geom\_jitter()  
  
freq\_site\_ph

## Warning: Removed 5 rows containing missing values (geom\_point).

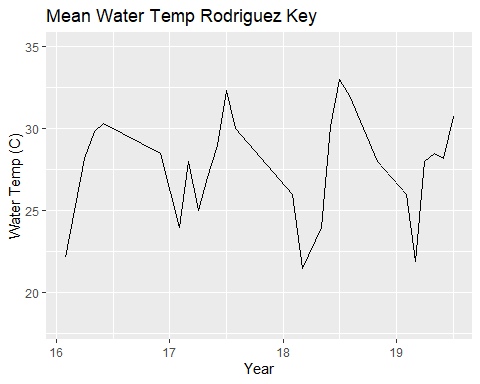


These plots are very consistent with the plots of the full data set.

Lastly lets take the seasonal means for each parameter and apply them to our three sites of interest. We will generate line graphs to show specific fluctuation between sites.

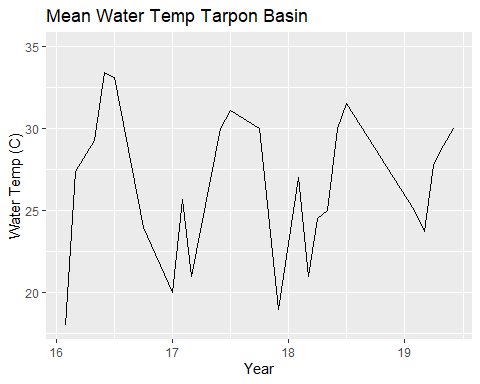
mean\_temp\_roddy <- aggregate(water\_temp ~ Month + Year, roddy\_subset, mean)   
  
mean\_temp\_roddy$Date <- as.yearmon(paste(mean\_temp\_roddy$Month, mean\_temp\_roddy$Year, sep = "."), format = "%m.%Y")  
  
mean\_temp\_plot\_roddy <- mean\_temp\_roddy %>% ggplot(aes(x = Date, y = water\_temp)) + geom\_line() + xlab("Year") + ylab("Water Temp (C)") + ggtitle("Mean Water Temp Rodriguez Key") + scale\_y\_continuous(limits=c(18, 35))  
  
mean\_temp\_plot\_roddy

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



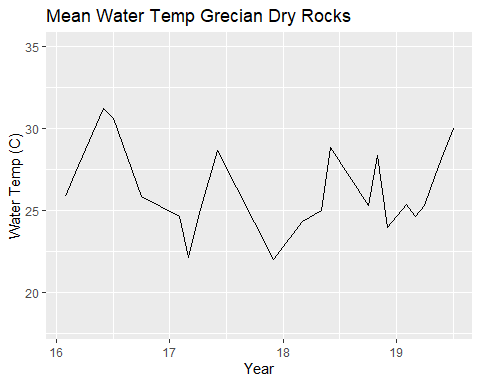
mean\_temp\_tarpon <- aggregate(water\_temp ~ Month + Year, tarpon\_subset, mean)   
  
mean\_temp\_tarpon$Date <- as.yearmon(paste(mean\_temp\_tarpon$Month, mean\_temp\_tarpon$Year, sep = "."), format = "%m.%Y")  
  
mean\_temp\_plot\_tarpon <- mean\_temp\_tarpon %>% ggplot(aes(x = Date, y = water\_temp)) + geom\_line() + xlab("Year") + ylab("Water Temp (C)") + ggtitle("Mean Water Temp Tarpon Basin") + scale\_y\_continuous(limits=c(18, 35))  
  
mean\_temp\_plot\_tarpon

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



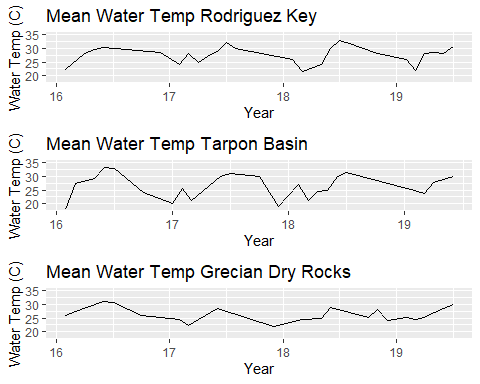
mean\_temp\_grecian <- aggregate(water\_temp ~ Month + Year, grecian\_subset, mean)   
  
mean\_temp\_grecian$Date <- as.yearmon(paste(mean\_temp\_grecian$Month, mean\_temp\_grecian$Year, sep = "."), format = "%m.%Y")  
  
mean\_temp\_plot\_grecian <- mean\_temp\_grecian %>% ggplot(aes(x = Date, y = water\_temp)) + geom\_line() + xlab("Year") + ylab("Water Temp (C)") + ggtitle("Mean Water Temp Grecian Dry Rocks") + scale\_y\_continuous(limits=c(18, 35))  
  
mean\_temp\_plot\_grecian

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_temp <- grid.arrange(mean\_temp\_plot\_roddy, mean\_temp\_plot\_tarpon, mean\_temp\_plot\_grecian, nrow = 3)

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_temp

## TableGrob (3 x 1) "arrange": 3 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (2-2,1-1) arrange gtable[layout]  
## 3 3 (3-3,1-1) arrange gtable[layout]

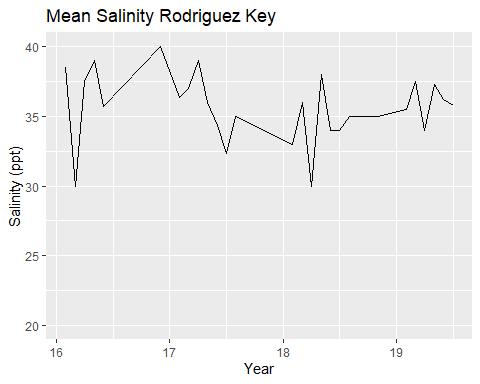
ggsave(filename = "../../results/Bivariate\_Figures/rep\_sites\_temp.png", plot = rep\_sites\_temp)

## Saving 5 x 4 in image

Temperature looks pretty seasonally consistent withe the other plots previously generated.

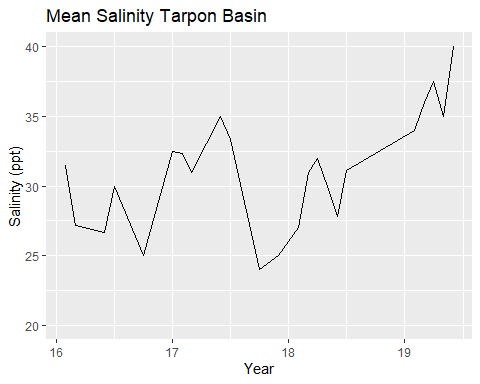
mean\_sal\_roddy <- aggregate(salinity ~ Month + Year, roddy\_subset, mean)   
  
mean\_sal\_roddy$Date <- as.yearmon(paste(mean\_sal\_roddy$Month, mean\_sal\_roddy$Year, sep = "."), format = "%m.%Y")  
  
mean\_sal\_plot\_roddy <- mean\_sal\_roddy %>% ggplot(aes(x = Date, y = salinity)) + geom\_line() + xlab("Year") + ylab("Salinity (ppt)") + ggtitle("Mean Salinity Rodriguez Key") + scale\_y\_continuous(limits=c(20, 40))  
  
mean\_sal\_plot\_roddy

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



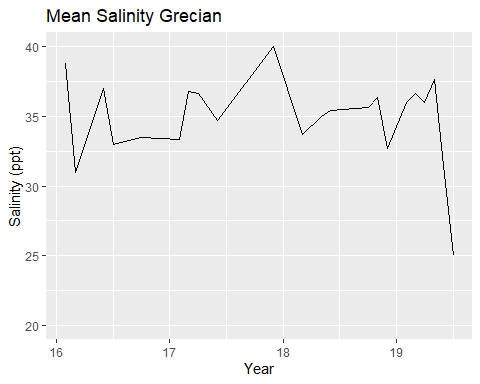
mean\_sal\_tarpon <- aggregate(salinity ~ Month + Year, tarpon\_subset, mean)   
  
mean\_sal\_tarpon$Date <- as.yearmon(paste(mean\_sal\_tarpon$Month, mean\_sal\_tarpon$Year, sep = "."), format = "%m.%Y")  
  
mean\_sal\_plot\_tarpon <- mean\_sal\_tarpon %>% ggplot(aes(x = Date, y = salinity)) + geom\_line() + xlab("Year") + ylab("Salinity (ppt)") + ggtitle("Mean Salinity Tarpon Basin") + scale\_y\_continuous(limits=c(20, 40))  
  
mean\_sal\_plot\_tarpon

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



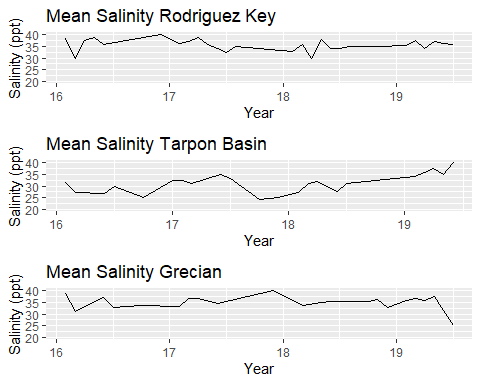
mean\_sal\_grecian <- aggregate(salinity ~ Month + Year, grecian\_subset, mean)   
  
mean\_sal\_grecian$Date <- as.yearmon(paste(mean\_sal\_grecian$Month, mean\_sal\_grecian$Year, sep = "."), format = "%m.%Y")  
  
mean\_sal\_plot\_grecian <- mean\_sal\_grecian %>% ggplot(aes(x = Date, y = salinity)) + geom\_line() + xlab("Year") + ylab("Salinity (ppt)") + ggtitle("Mean Salinity Grecian") + scale\_y\_continuous(limits=c(20, 40))  
  
mean\_sal\_plot\_grecian

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_sal <- grid.arrange(mean\_sal\_plot\_roddy, mean\_sal\_plot\_tarpon, mean\_sal\_plot\_grecian, nrow = 3)

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_sal

## TableGrob (3 x 1) "arrange": 3 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (2-2,1-1) arrange gtable[layout]  
## 3 3 (3-3,1-1) arrange gtable[layout]

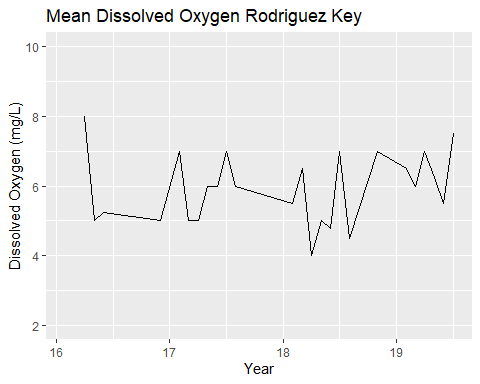
ggsave(filename = "../../results/Bivariate\_Figures/rep\_sites\_sal.png", plot = rep\_sites\_sal)

## Saving 5 x 4 in image

Salinity shows the distinct between ocean and bayside sites as expected. Additionall we can see a dip in the salinity of Tarpon Basin sharply near the Irma timeframe, this suggests that the bay may have been more impacted by the storm. This makes sense as the bay is shallower. There is also an interesting decline in the salinity of Grecian Dry Rocks at the end of 2019, I am unsure as to what may be the cause of this decline.

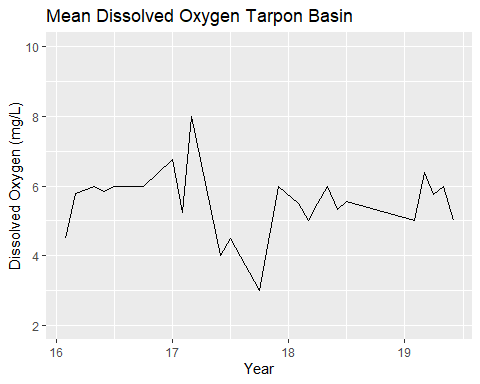
mean\_do\_roddy <- aggregate(dissolved\_oxygen ~ Month + Year, roddy\_subset, mean)   
  
mean\_do\_roddy$Date <- as.yearmon(paste(mean\_do\_roddy$Month, mean\_do\_roddy$Year, sep = "."), format = "%m.%Y")  
  
mean\_do\_plot\_roddy <- mean\_do\_roddy %>% ggplot(aes(x = Date, y = dissolved\_oxygen)) + geom\_line() + xlab("Year") + ylab("Dissolved Oxygen (mg/L)") + ggtitle("Mean Dissolved Oxygen Rodriguez Key") + scale\_y\_continuous(limits=c(2, 10))  
  
mean\_do\_plot\_roddy

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



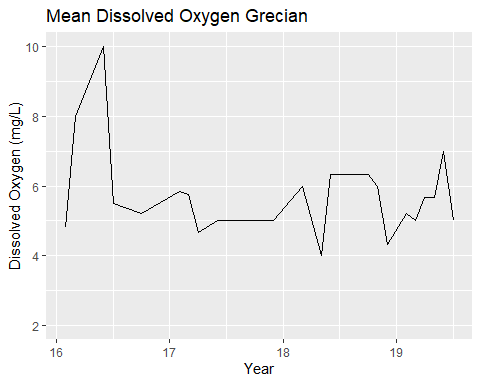
mean\_do\_tarpon <- aggregate(dissolved\_oxygen ~ Month + Year, tarpon\_subset, mean)   
  
mean\_do\_tarpon$Date <- as.yearmon(paste(mean\_do\_tarpon$Month, mean\_do\_tarpon$Year, sep = "."), format = "%m.%Y")  
  
mean\_do\_plot\_tarpon <- mean\_do\_tarpon %>% ggplot(aes(x = Date, y = dissolved\_oxygen)) + geom\_line() + xlab("Year") + ylab("Dissolved Oxygen (mg/L)") + ggtitle("Mean Dissolved Oxygen Tarpon Basin") + scale\_y\_continuous(limits=c(2, 10))  
  
mean\_do\_plot\_tarpon

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



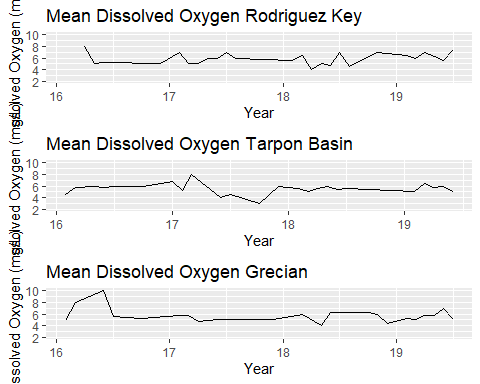
mean\_do\_grecian <- aggregate(dissolved\_oxygen ~ Month + Year, grecian\_subset, mean)   
  
mean\_do\_grecian$Date <- as.yearmon(paste(mean\_do\_grecian$Month, mean\_do\_grecian$Year, sep = "."), format = "%m.%Y")  
  
mean\_do\_plot\_grecian <- mean\_do\_grecian %>% ggplot(aes(x = Date, y = dissolved\_oxygen)) + geom\_line() + xlab("Year") + ylab("Dissolved Oxygen (mg/L)") + ggtitle("Mean Dissolved Oxygen Grecian") + scale\_y\_continuous(limits=c(2, 10))  
  
mean\_do\_plot\_grecian

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_do <- grid.arrange(mean\_do\_plot\_roddy, mean\_do\_plot\_tarpon, mean\_do\_plot\_grecian, nrow = 3)

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_do

## TableGrob (3 x 1) "arrange": 3 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (2-2,1-1) arrange gtable[layout]  
## 3 3 (3-3,1-1) arrange gtable[layout]

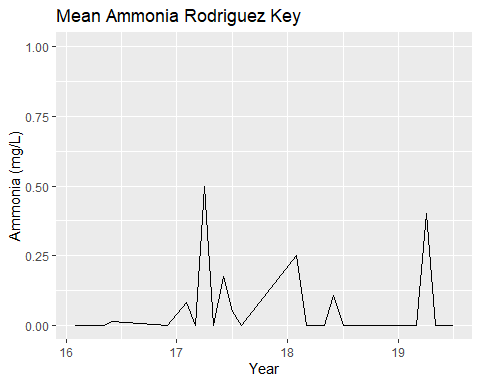
ggsave(filename = "../../results/Bivariate\_Figures/rep\_sites\_do.png", plot = rep\_sites\_do)

## Saving 5 x 4 in image

Dissolved oxygen is rather interesting. The general patterns of the graphs are similar to the expected variation between sites, however there is a notable decrease in Tarpon Basin around the Irma timeline. This also suggests the bay was strongly impacted by the storm. There is also an interesting spike in DO in early 2016 for Grecian Dry Rocks. I am unsure as to the cause of this.

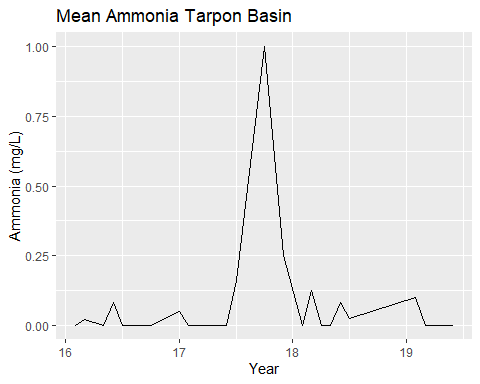
mean\_amm\_roddy <- aggregate(ammonia ~ Month + Year, roddy\_subset, mean)   
  
mean\_amm\_roddy$Date <- as.yearmon(paste(mean\_amm\_roddy$Month, mean\_amm\_roddy$Year, sep = "."), format = "%m.%Y")  
  
mean\_amm\_plot\_roddy <- mean\_amm\_roddy %>% ggplot(aes(x = Date, y = ammonia)) + geom\_line() + xlab("Year") + ylab("Ammonia (mg/L)") + ggtitle("Mean Ammonia Rodriguez Key") + scale\_y\_continuous(limits=c(0, 1))  
  
mean\_amm\_plot\_roddy

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



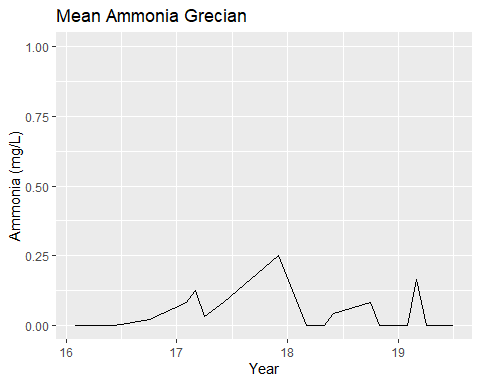
mean\_amm\_tarpon <- aggregate(ammonia ~ Month + Year, tarpon\_subset, mean)   
  
mean\_amm\_tarpon$Date <- as.yearmon(paste(mean\_amm\_tarpon$Month, mean\_amm\_tarpon$Year, sep = "."), format = "%m.%Y")  
  
mean\_amm\_plot\_tarpon <- mean\_amm\_tarpon %>% ggplot(aes(x = Date, y = ammonia)) + geom\_line() + xlab("Year") + ylab("Ammonia (mg/L)") + ggtitle("Mean Ammonia Tarpon Basin") + scale\_y\_continuous(limits=c(0, 1))  
  
mean\_amm\_plot\_tarpon

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



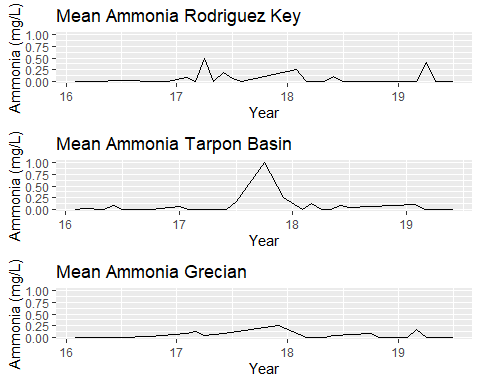
mean\_amm\_grecian <- aggregate(ammonia ~ Month + Year, grecian\_subset, mean)   
  
mean\_amm\_grecian$Date <- as.yearmon(paste(mean\_amm\_grecian$Month, mean\_amm\_grecian$Year, sep = "."), format = "%m.%Y")  
  
mean\_amm\_plot\_grecian <- mean\_amm\_grecian %>% ggplot(aes(x = Date, y = ammonia)) + geom\_line() + xlab("Year") + ylab("Ammonia (mg/L)") + ggtitle("Mean Ammonia Grecian") + scale\_y\_continuous(limits=c(0, 1))  
  
mean\_amm\_plot\_grecian

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_amm <- grid.arrange(mean\_amm\_plot\_roddy, mean\_amm\_plot\_tarpon, mean\_amm\_plot\_grecian, nrow = 3)

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_amm

## TableGrob (3 x 1) "arrange": 3 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (2-2,1-1) arrange gtable[layout]  
## 3 3 (3-3,1-1) arrange gtable[layout]

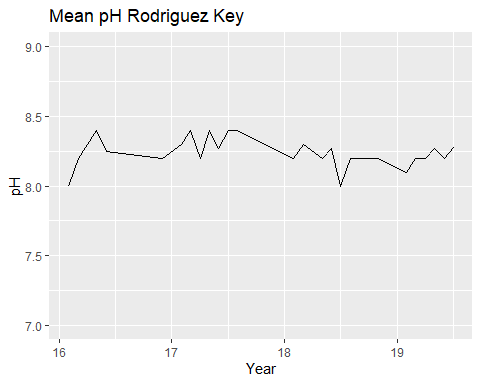
ggsave(filename = "../../results/Bivariate\_Figures/rep\_sites\_amm.png", plot = rep\_sites\_amm)

## Saving 5 x 4 in image

Ammonia is similar to other visuals created above, relatively flat across the board with intermittant spikes and a substantial spike around Irma.

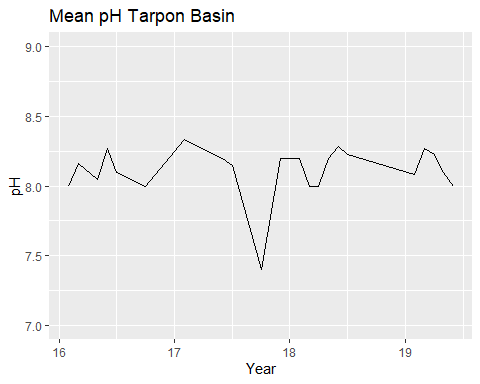
mean\_ph\_roddy <- aggregate(ph ~ Month + Year, roddy\_subset, mean)   
  
mean\_ph\_roddy$Date <- as.yearmon(paste(mean\_ph\_roddy$Month, mean\_ph\_roddy$Year, sep = "."), format = "%m.%Y")  
  
mean\_ph\_plot\_roddy <- mean\_ph\_roddy %>% ggplot(aes(x = Date, y = ph)) + geom\_line() + xlab("Year") + ylab("pH") + ggtitle("Mean pH Rodriguez Key") + scale\_y\_continuous(limits=c(7, 9))  
  
mean\_ph\_plot\_roddy

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



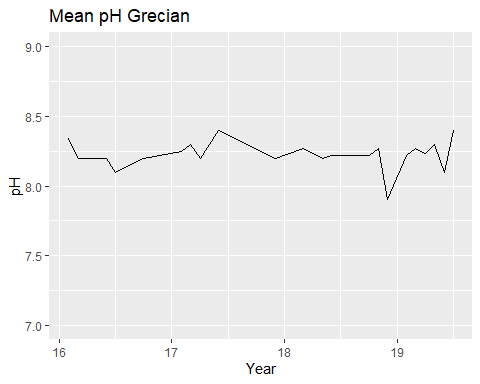
mean\_ph\_tarpon <- aggregate(ph ~ Month + Year, tarpon\_subset, mean)   
  
mean\_ph\_tarpon$Date <- as.yearmon(paste(mean\_ph\_tarpon$Month, mean\_ph\_tarpon$Year, sep = "."), format = "%m.%Y")  
  
mean\_ph\_plot\_tarpon <- mean\_ph\_tarpon %>% ggplot(aes(x = Date, y = ph)) + geom\_line() + xlab("Year") + ylab("pH") + ggtitle("Mean pH Tarpon Basin") + scale\_y\_continuous(limits=c(7, 9))  
  
mean\_ph\_plot\_tarpon

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



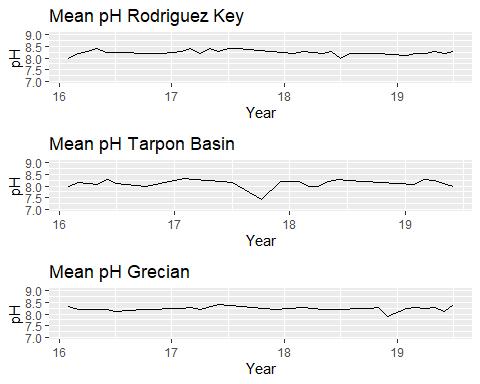
mean\_ph\_grecian <- aggregate(ph ~ Month + Year, grecian\_subset, mean)   
  
mean\_ph\_grecian$Date <- as.yearmon(paste(mean\_ph\_grecian$Month, mean\_ph\_grecian$Year, sep = "."), format = "%m.%Y")  
  
mean\_ph\_plot\_grecian <- mean\_ph\_grecian %>% ggplot(aes(x = Date, y = ph)) + geom\_line() + xlab("Year") + ylab("pH") + ggtitle("Mean pH Grecian") + scale\_y\_continuous(limits=c(7, 9))  
  
mean\_ph\_plot\_grecian

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_ph <- grid.arrange(mean\_ph\_plot\_roddy, mean\_ph\_plot\_tarpon, mean\_ph\_plot\_grecian, nrow = 3)

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.



rep\_sites\_ph

## TableGrob (3 x 1) "arrange": 3 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (2-2,1-1) arrange gtable[layout]  
## 3 3 (3-3,1-1) arrange gtable[layout]

ggsave(filename = "../../results/Bivariate\_Figures/rep\_sites\_ph.png", plot = rep\_sites\_ph)

## Saving 5 x 4 in image

pH is also quite interesting. Similar to dissolved oxygen, there is a notable drop in pH around hurricane Irma land fall that is not present in the group data. This again suggests the bay was strongly affected by this storm.

## Conclusions

Overall, the there seems to be little seasonal fluctuation in ammonia, dissolved oxygen and pH measurements. This is not suprising due to the fact that these parameters are largely independent of one another (barring some unique event). Water temperature shows standard seasonal fluctuation with warmer and cooler months, and bays side waters exhibit a wider ranger of temperatures. Salinity is reasonable unique in terms of observations. Oceanside water have distinctly greater salinity for the vast majority of the year and have a very narrow range. Bayside waters typically have a reduced salinity, however large spikes upward can be associated with the dry season.

Despite these overall trends, there is a notable variation in some of the parameters consistent with the timing of Hurricane Irma landfall (September 2017). Ammonia most noteably spkies drastically at this time. This is likely due to hurricane-induced nutrint influx. Salinity drops substantially from the normal levels, likely due to storm water influx. Water temperature shows a notable decrease, but it is likely this can be attributed to standard seasonal fluctuation. Both pH and dissolved oxygen do not show any unusual trends with the group data, but show notable changes when aplied to Tarpon Basin measures. Hurricane Irma warrents further investigation and will be analyzed in the *hurricane\_irma\_analysis* script. It should be noted that post Hirricane Irma observations are limited in number due to the inability to collect samples directly following the storm.