# Envr_Violin_Plots

2024-10-08

#Download packages

```r
library(dplyr) #Used for working with data frames
```

```
##
## 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
```

```r
library(lubridate) #Used for time-date conversions
```

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

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

```r
library(readr) #Used to read the CSV file
library(ggplot2) #plot with ggplot
library(cowplot) #arrange ggplots
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```r
library(ggpmisc)
```

```
## Loading required package: ggpp
## Registered S3 methods overwritten by 'ggpp':
##   method                  from
##   heightDetails.titleGrob ggplot2
##   widthDetails.titleGrob  ggplot2

##
## Attaching package: 'ggpp'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

# Set working directory

```r
#set working directory to Rmd file location
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

```r
#Read in salinity files
TX_sal <- read.csv("../../data/envr_of_origin/full_sal/TX_sal_full.csv")
#remove extra empty columns
TX_sal <- subset(TX_sal, select = c(site_name, datetime, salinity))

LA_sal <- read.csv("../../data/envr_of_origin/full_sal/LA_sal_full.csv")

FL_sal <- read.csv("../../data/envr_of_origin/full_sal/FL_sal_full.csv")

LOLA_sal <- read.csv("../../data/envr_of_origin/full_sal/LOLA_sal_full.csv")

DEBY_sal <- read.csv("../../data/envr_of_origin/full_sal/DEBY_sal_full.csv")

VA_sal <- read.csv("../../data/envr_of_origin/full_sal/VA_sal_full.csv")
#remove uncorrected salinity column and rename corrected_salinity to salinity
VA_sal <- subset(VA_sal, select = -c(salinity))
colnames(VA_sal) <- c("site_name", "datetime", "salinity")

NH_sal <- read.csv("../../data/envr_of_origin/full_sal/NH_sal_full.csv")
#remove extra empty columns
NH_sal <- subset(NH_sal, select = c(site_name, datetime, salinity))

ME_sal <- read.csv("../../data/envr_of_origin/full_sal/ME_sal_full.csv")
```

```r
#convert all date times to POSIXct
TX_sal$datetime <- as.POSIXct(TX_sal$datetime, "%m/%d/%y %H:%M", tz = "")

LA_sal$datetime <- as.POSIXct(LA_sal$datetime, "%Y-%m-%d %H:%M:%S", tz = "")

FL_sal$datetime <- as.POSIXct(FL_sal$datetime, "%m/%d/%y %H:%M", tz = "")

LOLA_sal$datetime <- as.POSIXct(LOLA_sal$datetime, "%Y-%m-%d %H:%M:%S", tz = "")
```

```r
DEBY_sal$datetime <- as.POSIXct(DEBY_sal$datetime, "%Y-%m-%d %H:%M:%S", tz = "")

VA_sal$datetime <- as.POSIXct(VA_sal$datetime, "%Y-%m-%d %H:%M:%S", tz = "")

NH_sal$datetime <- as.POSIXct(NH_sal$datetime,"%m/%d/%y %H:%M", tz = "")

ME_sal$datetime <- as.POSIXct(ME_sal$datetime, "%m/%d/%y %H:%M", tz = "")

#ME only has 2 data points in October 2022 and one in July 2022, so exclude those points
#ME_sal <- ME_sal[-c(26312, 26313, 26314), ]
```

```r
#make all salinity variables numeric
TX_sal$salinity <- as.numeric(TX_sal$salinity)
LA_sal$salinity <- as.numeric(LA_sal$salinity)
FL_sal$salinity <- as.numeric(FL_sal$salinity)
LOLA_sal$salinity <- as.numeric(LOLA_sal$salinity)
DEBY_sal$salinity <- as.numeric(DEBY_sal$salinity)
VA_sal$salinity <- as.numeric(VA_sal$salinity)
NH_sal$salinity <- as.numeric(NH_sal$salinity)
ME_sal$salinity <- as.numeric(ME_sal$salinity)

#merge data frames
full_sal <- rbind(TX_sal, LA_sal,  FL_sal, LOLA_sal, DEBY_sal, VA_sal, NH_sal, ME_sal)

#remove NAs
na_full_sal <- is.na(full_sal) # store our NAs in a variable
summary(na_full_sal) # we have 553 NAs in datetime and 192380 NAs in salinity that are stored as "TRUE"
```

```
##  site_name        datetime        salinity
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:2695811   FALSE:2695258   FALSE:2503431
##                  TRUE :553       TRUE :192380
```

```r
full_sal <- na.omit(full_sal) #remove NAs using na.omit

#filter out salinity values below 0 and above 40
filtered_sal <- full_sal %>%
    filter(between(salinity, 0, 40))
```

#Temperature

```r
#Read in temp files
TX_temp <- read.csv("../../data/envr_of_origin/full_temp/TX_temp_full.csv")

LA_temp <- read.csv("../../data/envr_of_origin/full_temp/LA_temp_full.csv")

FL_temp <- read.csv("../../data/envr_of_origin/full_temp/FL_temp_full.csv")

LOLA_temp <- read.csv("../../data/envr_of_origin/full_temp/LOLA_temp_full.csv")

DEBY_temp <- read.csv("../../data/envr_of_origin/full_temp/DEBY_temp_full.csv")
```

```r
VA_temp <- read.csv("../../data/envr_of_origin/full_temp/VA_temp_full.csv")
#remove uncorrected salinity column and rename corrected_salinity to salinity
VA_temp <- subset(VA_temp, select = -c(temp))
colnames(VA_temp) <- c("site_name", "datetime", "temp")

NH_temp <- read.csv("../../data/envr_of_origin/full_temp/NH_temp_full.csv")
#remove extra empty columns
NH_temp <- subset(NH_temp, select = c(site_name, datetime, temp))

ME_temp <- read.csv("../../data/envr_of_origin/full_temp/ME_temp_full.csv")
```

```r
#convert all date times to POSIXct
TX_temp$datetime <- as.POSIXct(TX_temp$datetime, "%m/%d/%y %H:%M", tz = "")

LA_temp$datetime <- as.POSIXct(LA_temp$datetime, "%Y-%m-%d %H:%M:%S", tz = "")

FL_temp$datetime <- as.POSIXct(FL_temp$datetime, "%m/%d/%y %H:%M", tz = "")

LOLA_temp$datetime <- as.POSIXct(LOLA_temp$datetime, "%Y-%m-%d %H:%M:%S", tz = "")

DEBY_temp$datetime <- as.POSIXct(DEBY_temp$datetime, "%Y-%m-%d %H:%M:%S", tz = "")

VA_temp$datetime <- as.POSIXct(VA_temp$datetime, "%Y-%m-%d %H:%M", tz = "")

NH_temp$datetime <- as.POSIXct(NH_temp$datetime, "%m/%d/%y %H:%M", tz = "")

ME_temp$datetime <- as.POSIXct(ME_temp$datetime, "%m/%d/%y %H:%M", tz = "")
```

```r
#make all temp variables numeric
TX_temp$temp <- as.numeric(TX_temp$temp)
LA_temp$temp <- as.numeric(LA_temp$temp)
FL_temp$temp <- as.numeric(FL_temp$temp)
LOLA_temp$temp <- as.numeric(LOLA_temp$temp)
DEBY_temp$temp <- as.numeric(DEBY_temp$temp)
VA_temp$temp <- as.numeric(VA_temp$temp)
NH_temp$temp <- as.numeric(NH_temp$temp)
ME_temp$temp <- as.numeric(ME_temp$temp)

#remove extra columns from NH
NH_temp <- NH_temp[, c(1,2,3)]

#merge data frames
full_temp <- rbind(DEBY_temp, FL_temp, VA_temp, LA_temp, LOLA_temp, ME_temp, NH_temp, TX_temp)

#remove NAs
na_full_temp <- is.na(full_temp) # store our NAs in a variable
summary(na_full_temp) # we have 3820 NAs in datetime and 180627 NAs in temp that are stored as "TRUE"
```

```
##  site_name        datetime          temp
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:2704145   FALSE:2703590   FALSE:2523519
##                  TRUE :555       TRUE :180626
```

```
full_temp <- na.omit(full_temp) #remove NAs using na.omit

#filter out temps below 0 and above 40
filtered_temp <- full_temp %>%
    filter(between(temp, 0, 40))
```

Violin plots with the raw data cause R to crash due to large data size. Therefore, I will make the violin plots using monthly temp/salinity averages rather than raw data.

```
#average monthly temperatures
monthly_temp <- filtered_temp %>%
    mutate(year = year(datetime), month = month(datetime)) %>%
    group_by(site_name, year, month) %>%
    summarise(mean_temp = mean(temp))
```

```
## `summarise()` has grouped output by 'site_name', 'year'. You can override using
## the `.groups` argument.
```

```
#make sure all 8 sites are still present
monthly_temp_sites <- list(unique(monthly_temp$site_name))

monthly_temp_sites
```

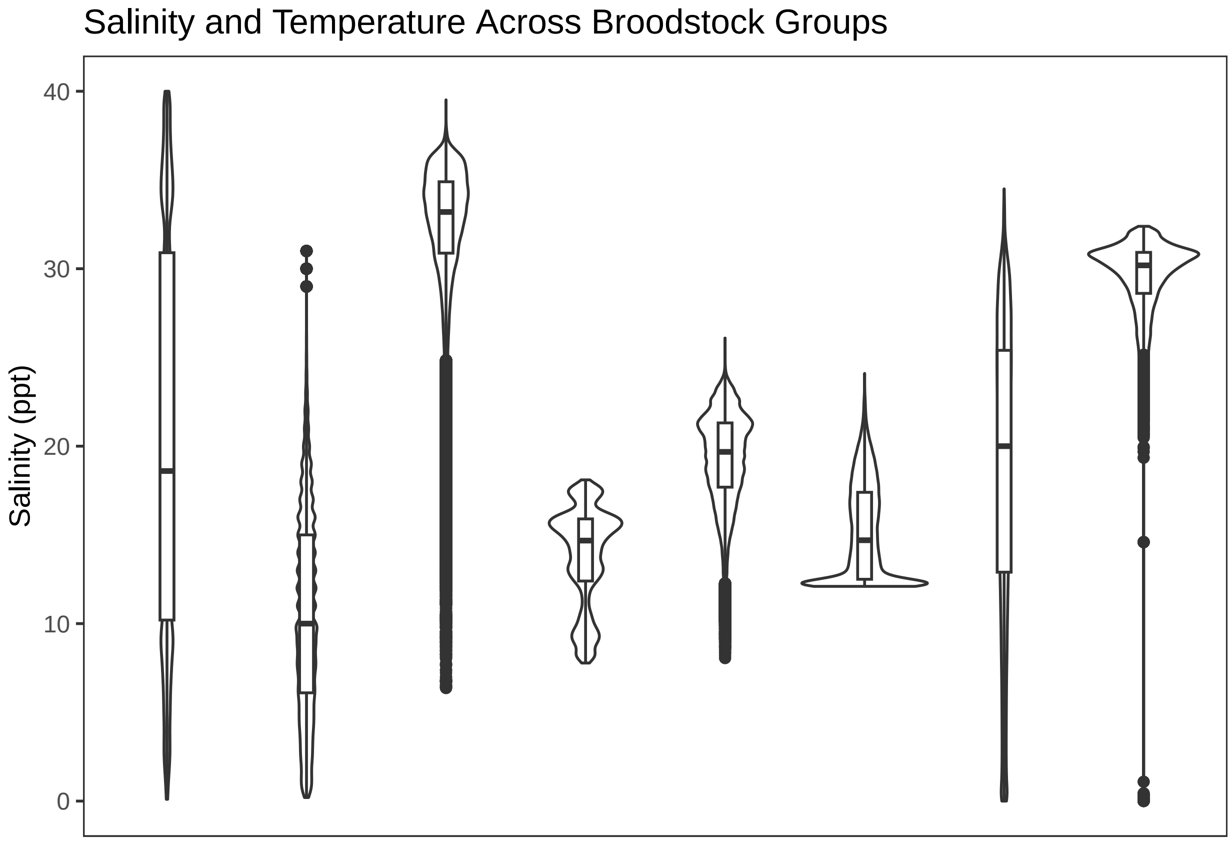
```
## [[1]]
## [1] "DEBY" "FL"   "LA"   "LOLA" "ME"   "NH"   "TX"   "VA"
```

```
#write to csv for future analyses
write.csv(monthly_temp, "/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/data/envr_raw
```

## Violin Plots

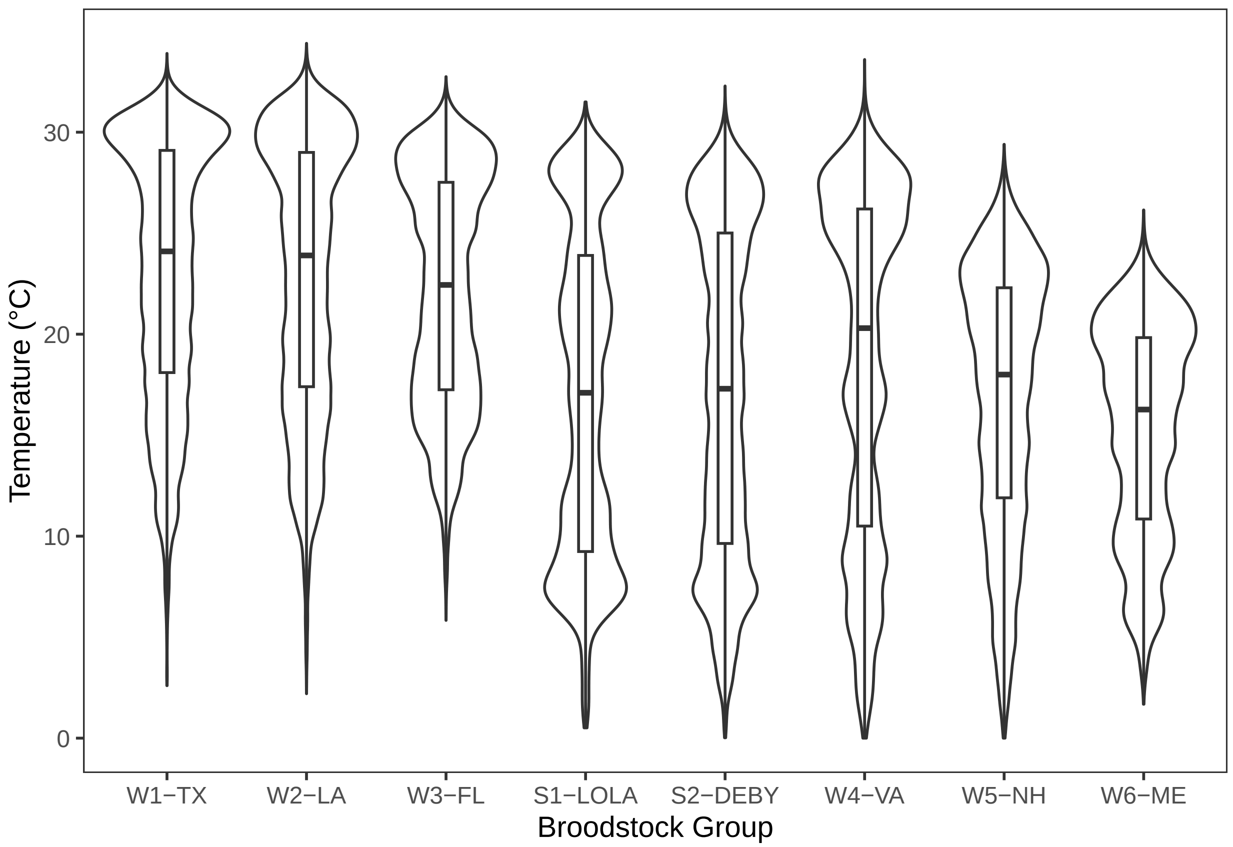
```
sal_violin <- ggplot(filtered_sal, aes(x = site_name, y = salinity)) +
  geom_violin()+
  geom_boxplot(width = .1) +
  ggtitle("Salinity and Temperature Across Broodstock Groups")+
  ylab("Salinity (ppt)")+
  scale_x_discrete(name = "Broodstock Group", limits = c("TX","LA","FL", "LOLA", "DEBY", "VA", "NH", "MI
  theme_bw()+
  theme(axis.title.x = element_blank(), axis.ticks.x = element_blank(), axis.text.x = element_blank(), p

sal_violin
```



```
temp_violin <- ggplot(filtered_temp, aes(x = site_name, y = temp)) +
  geom_violin()+
  geom_boxplot(width = .1) +
  xlab("Broodstock Group")+
  ylab("Temperature (°C)")+
  scale_x_discrete(name = "Broodstock Group", limits = c("TX","LA","FL", "LOLA", "DEBY", "VA", "NH", "M
  theme_bw()+
  theme(plot.title = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_bla

temp_violin
```



```
#merge plots
violin_plots <- plot_grid(sal_violin, temp_violin, ncol = 1, align = "v")

#save
ggsave(violin_plots,
       filename = "violinplots.png",
       device = "png",
       path = "../../figures/envr_of_origin")
```

```
## Saving 6.5 x 4.5 in image
```