LOLA_raw_env

2024-04-03

```
setwd("/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/src/NM")
```

Load required packages.

```
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
```

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

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

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

```
library("readr") #Used to read the CSV file
library("ggplot2")
```

Note the date of data download and source. All available data should be used for each site regardless of year. Note from the CSV file how often the site was sampled, and if there are replicates in the data. Also describe if the sampling occurred at only low tide, only high tide, or continuously.

```
#Data was downloaded on 05/10/2024
#Source - Dr. William "Willy" Reay of VIMS shared the file with Jess Small via email. Jess emailed the
#The site was sampled continuously every 15 min from May 2023 to April 2024
```

```
#Create text strings with metadata information that we want to include in the final data frame.
download_date <- ("05-10-2024")
source_description <- ("William Reay, VIMS")
site_name <- ("LOLA")
collection_type <- ("continuous")
```

Use the file path name in your working directory or desktop, see example below. Or, import data set through the "Files" window in R studio. Store the file in a variable with the "raw_ID_Site" format. If salinity and temperature data are in separate files, read in both and store them with "_sal" or "_temp" in the variable names.

```
#The files we will be working with are from the Coan River, VA, which is for the LOLA selection line. T

#set working directory to files location
setwd("/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/data/envr_raw_data/LOLA_Coan")

#merge files into one
raw_LOLA_env <- read.csv("/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/data/envr_ra

#The data include separate columns for date and time, which I want to merge into one.
raw_LOLA_env$datetime <- paste(raw_LOLA_env$Date, raw_LOLA_env$EST.Time)

#Remove Date and EST.Time columns, as well as depth column.
raw_LOLA_env <- subset(raw_LOLA_env, select = -c(Date, EST.Time, Depth))

#Reorder columns so datetime is first
raw_LOLA_env <- raw_LOLA_env[, c(3,1,2)]


# View how the data are stored. Note the variable names and the format and units that the data are stor
summary(raw_LOLA_env)
```

```
##    datetime              Temp           Salinity
##  Length:34352       Min.   : 0.513   Min.   : 7.78
##  Class :character   1st Qu.: 9.240   1st Qu.:12.40
##  Mode  :character   Median :17.100   Median :14.68
##                     Mean   :16.904   Mean   :13.84
##                     3rd Qu.:23.900   3rd Qu.:15.90
##                     Max.   :31.500   Max.   :18.10
##                     NA's   :264      NA's   :270
```

```
#rename columns. "datetime" = date and time of data collection, "temp" = water temperature in degrees C

colnames(raw_LOLA_env) <- c("datetime", "temp", "salinity")
```

Start with the date and time of collection. We will use the lubridate package to standardize all values into the date-time format called POSIXct. This format stores the date and time in number of seconds since a past point (1/1/1970). This makes comparisons easy and helps to standardizes values.

```
#Convert to POSIXct format. Tell R what the current date/time format is so it knows how to convert it.
raw_LOLA_env$datetime <- as.POSIXct(raw_LOLA_env$datetime, "%m/%d/%y %H:%M:%S", tz = "")



#Print the new data frame and examine to make sure the new datetime column is in the correct format.
head(raw_LOLA_env)
```

```
##              datetime temp salinity
## 1 2023-05-01 00:00:00 17.8     14.6
## 2 2023-05-01 00:15:00 17.7     14.8
## 3 2023-05-01 00:30:00 17.7     14.8
## 4 2023-05-01 00:45:00 17.9     14.5
## 5 2023-05-01 01:00:00 17.9     14.6
## 6 2023-05-01 01:15:00 17.8     14.6
```

Set up data frames for violin plots

```
#add site name and create new data frame with full envr data set
LOLA_env_full <- raw_LOLA_env %>%
  mutate(site_name, site_name = "LOLA")

#reorder columns with site_name first
LOLA_env_full <- LOLA_env_full[, c(4, 1, 2, 3)]

#separate out salinity and temp data into new frames for other analyses
LOLA_temp_full <- LOLA_env_full[, c(1,2,3)]

LOLA_sal_full <- LOLA_env_full[, c(1,2,4)]

#save LOLA  as csvs for future analyses
write.csv(LOLA_temp_full, "/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/data/envr_

write.csv(LOLA_sal_full, "/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/data/envr_ra
```

#Standardize column and variable names. We will use "lat" for latitude in degrees, and "lon" for longitude in degrees.

```
#Store variables that we will include in the final data frame
lat <- 37.9803
lon <- -76.4619
#Lat and long data are from this site(https://www.vims.edu/cbnerr/monitoring/water_quality/). Data came
firstyear <- 2023
finalyear <- 2024
```

Filter any of the variables that have data points outside of normal range. We will use 0-40 as the accepted range for salinity (ppt) and temperature (C) values. Note, in the summer, salinity values can sometimes exceed 40. Check to see if there are values above 40. In this case, adjust the range or notify someone that the site has particularly high salinity values.

```
#Filter the data between the values of 0 and 40 for both salinity and temperature.
filtered_LOLA_sal <- raw_LOLA_env %>%
    filter(between(salinity, 0, 40))

filtered_LOLA_env <- filtered_LOLA_sal %>%
    filter(between(temp, 0, 40))

# Sanity check - print the ranges to ensure values are filtered properly. We can see that the ranges fo
print(summary(filtered_LOLA_env$salinity))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.78   12.40   14.68   13.84   15.90   18.10
```

```
print(summary(filtered_LOLA_env$temp))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.513   9.238  17.100  16.903  23.900  31.500
```

```
#Store our data into a variable name with just the site name.
LOLA_env <- filtered_LOLA_env

# we have NAs in the our data frame in the datetime column - need to remove these
count.nas_LOLA_env <- is.na(LOLA_env$datetime) # store our NAs in a variable
summary(count.nas_LOLA_env) # we have 4 NAs that are stored as "TRUE" in our count.nas
```

```
##    Mode   FALSE    TRUE
## logical   34078       4
```

```
nrow(LOLA_env) # figure out how many rows we have in the original df: 34082
```

```
## [1] 34082
```

```
which(count.nas_LOLA_env == TRUE) # find the number of NA rows that we need to remove: 4
```

```
## [1] 29884 29885 29886 29887
```

```
LOLA_env <- na.omit(LOLA_env) #remove NAs using na.omit
nrow(LOLA_env) #there are 34078 rows in the new data frame. 34082-34078 = 4, meaning we removed the rig
```

```
## [1] 34078
```

```r
# check for NAs in our temperature column
count.nas_LOLA_temp <- is.na(LOLA_env$temp) # store our NAs in temp in a variable
summary(count.nas_LOLA_temp) # we have 0 NAs
```

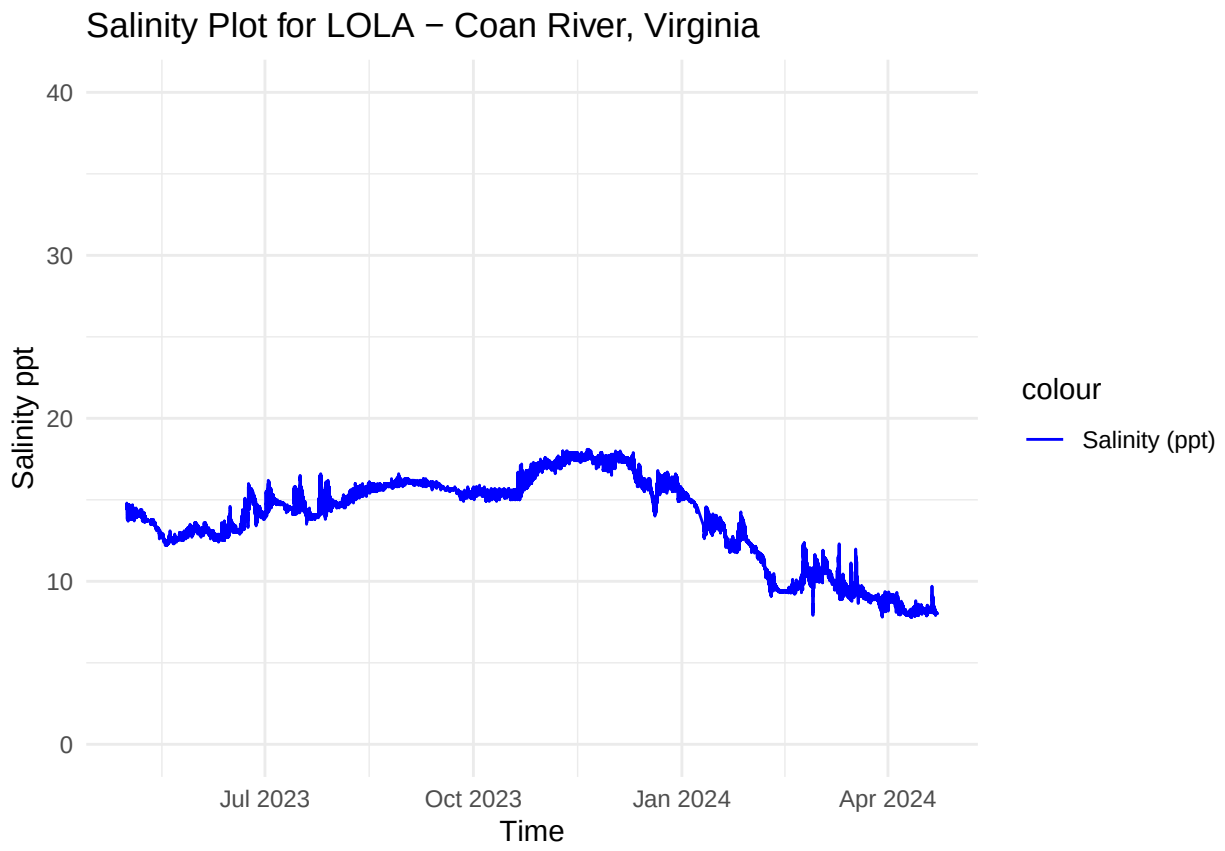
```
##    Mode   FALSE
## logical   34078
```

```r
#check for NAs in our salinity column
count.nas_LOLA_sal <- is.na(LOLA_env$salinity) #store our NAs in salinity in a variable
summary(count.nas_LOLA_sal) #we have no NAs
```

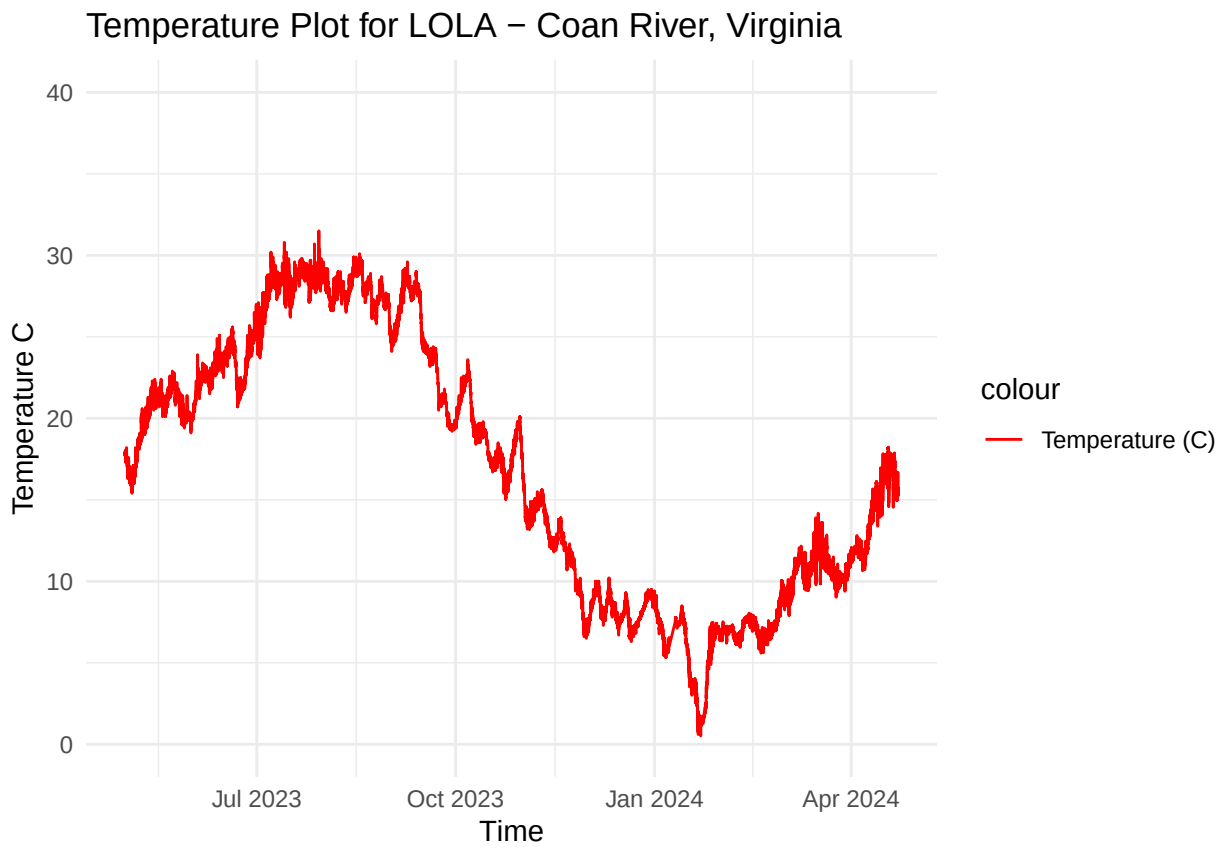
```
##    Mode   FALSE
## logical   34078
```

Visualize the salinity, temperature, and date ranges over time. This can help us see if there are any anomalies or gaps in the data and make sure the filtering was done correctly. Sanity check - do the temperature and salinity ranges look appropriate for the geography of the site (ex. near full ocean salinity for coastal sites, lower salinity for estuaries or near rivers)?

```r
salplot <- ggplot(LOLA_env, aes(x = datetime)) +
    geom_line(aes(y = salinity, color = "Salinity (ppt)")) +
    ylim(0,40) +
    labs(x = "Time", y = "Salinity ppt", title = "Salinity Plot for LOLA - Coan River, Virginia") +
    scale_color_manual(values = c("Salinity (ppt)" = "blue")) +
    theme_minimal()


salplot
```




```
tempplot <- ggplot(LOLA_env, aes(x = datetime)) +
    geom_line(aes(y = temp, color = "Temperature (C)")) +
    ylim(0, 40) +
    labs(x = "Time", y = "Temperature C", title = "Temperature Plot for LOLA - Coan River, Virginia") +
    scale_color_manual(values = c( "Temperature (C)" = "red")) +
    theme_minimal()

tempplot
```



We need to calculate the mean, maximum, and minimum values for salinity and temperature per month and year. First make two data frames to contain each of the annual and monthly averages.

```
#Calculate the mean, maximum, and minimum values for salinity and temperature for each month.
LOLA_envrmonth_sal <- LOLA_env %>%
    mutate(year = year(datetime), month = month(datetime)) %>%
    group_by(year, month) %>%
    summarise(
      min_salinity = min(salinity),
      max_salinity = max(salinity),
      mean_salinity = mean(salinity),
      length_salinity = length(salinity))
```

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

```
LOLA_envrmonth_temp <- LOLA_env %>%
    mutate(year = year(datetime), month = month(datetime)) %>%
    group_by(year, month) %>%
    summarise(
      min_temp = min(temp),
      max_temp = max(temp),
```

```
      mean_temp = mean(temp),
      length_temp = length(temp))
```

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

```
print(LOLA_envrmonth_sal)
```

```
## # A tibble: 12 x 6
## # Groups:   year [2]
##     year month min_salinity max_salinity mean_salinity length_salinity
##    <dbl> <dbl>        <dbl>        <dbl>         <dbl>           <int>
##  1  2023     5         12.2         14.8          13.3            2973
##  2  2023     6         12.4         16            13.6            2873
##  3  2023     7         13.5         16.6          14.7            2975
##  4  2023     8         14.5         16.6          15.6            2975
##  5  2023     9         14.9         16.3          15.8            2878
##  6  2023    10         14.9         17.5          15.8            2976
##  7  2023    11         16.5         18.1          17.4            2880
##  8  2023    12         14           18            16.4            2976
##  9  2024     1         11.8         15.6          13.5            2722
## 10  2024     2          7.91        12.4          10.3            2784
## 11  2024     3          7.8         12.3           9.58           2971
## 12  2024     4          7.78         9.7           8.36           2095
```

```
print(LOLA_envrmonth_temp)
```

```
## # A tibble: 12 x 6
## # Groups:   year [2]
##     year month min_temp max_temp mean_temp length_temp
##    <dbl> <dbl>    <dbl>    <dbl>     <dbl>       <int>
##  1  2023     5   15.4      22.9      19.9         2973
##  2  2023     6   19.8      27        23.0         2873
##  3  2023     7   23.7      31.5      28.2         2975
##  4  2023     8   24.8      30.1      27.8         2975
##  5  2023     9   19.2      29.6      24.5         2878
##  6  2023    10   15        23.6      18.9         2976
##  7  2023    11    6.5      16.9      12.4         2880
##  8  2023    12    6.3      10.2       8.32        2976
##  9  2024     1    0.513     9.14      5.58        2722
## 10  2024     2    5.59     10.1       7.17        2784
## 11  2024     3    7.91     14.2      10.7         2971
## 12  2024     4   10.6      18.2      14.3         2095
```

```
#Calculate the mean, maximum, and minimum values for salinity and temperature for each year.
LOLA_envryear_sal <- LOLA_env %>%
    mutate(year = year(datetime)) %>%
    group_by(year) %>%
    summarise(
      min_salinity = min(salinity),
      max_salinity = max(salinity),
```

```
      mean_salinity = mean(salinity))

LOLA_envryear_temp <- LOLA_env %>%
    mutate(year = year(datetime)) %>%
    group_by(year) %>%
    summarise(
      min_temp = min(temp),
      max_temp = max(temp),
      mean_temp = mean(temp))

print(LOLA_envryear_sal)
```

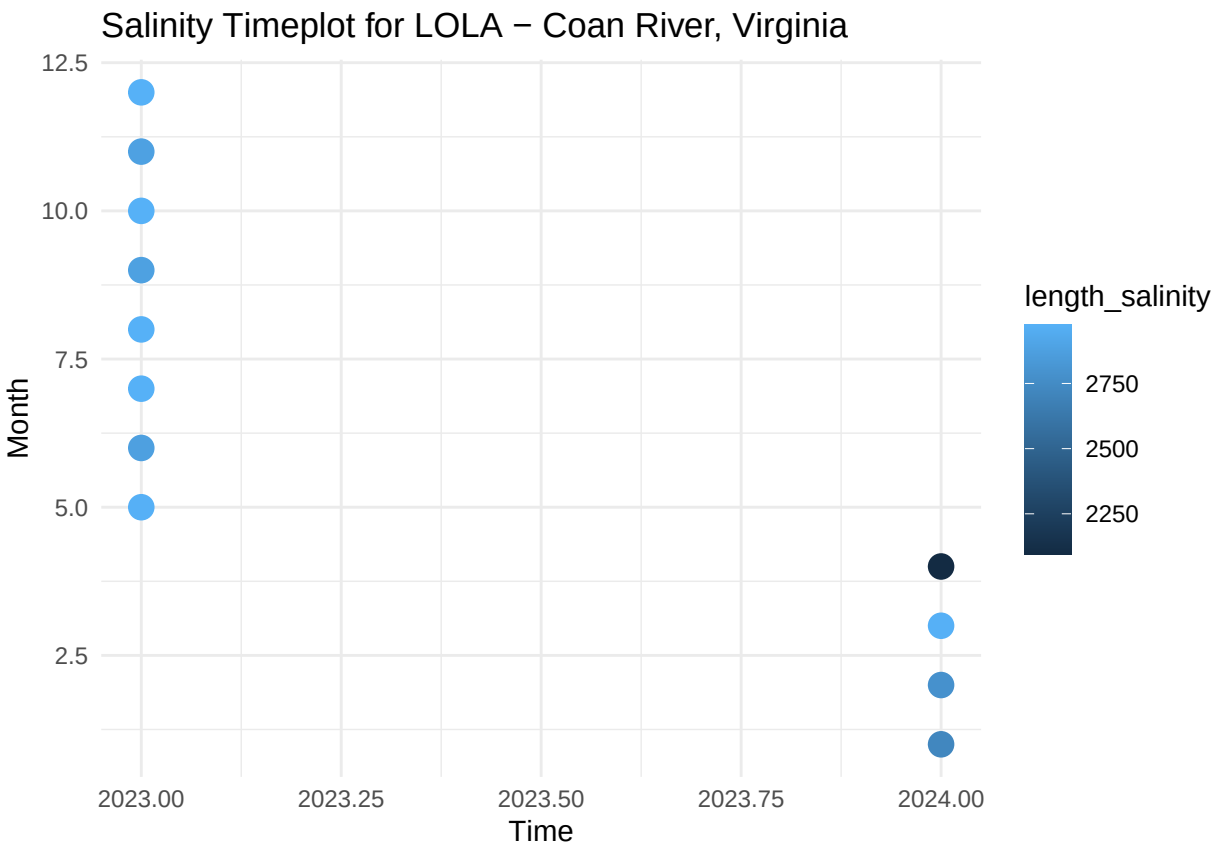
```
## # A tibble: 2 x 4
##    year min_salinity max_salinity mean_salinity
##   <dbl>        <dbl>        <dbl>         <dbl>
## 1  2023        12.2          18.1          15.3
## 2  2024         7.78         15.6          10.5
```

```
print(LOLA_envryear_temp)
```

```
## # A tibble: 2 x 4
##    year min_temp max_temp mean_temp
##   <dbl>    <dbl>    <dbl>     <dbl>
## 1  2023    6.3       31.5     20.4
## 2  2024    0.513     18.2      9.18
```

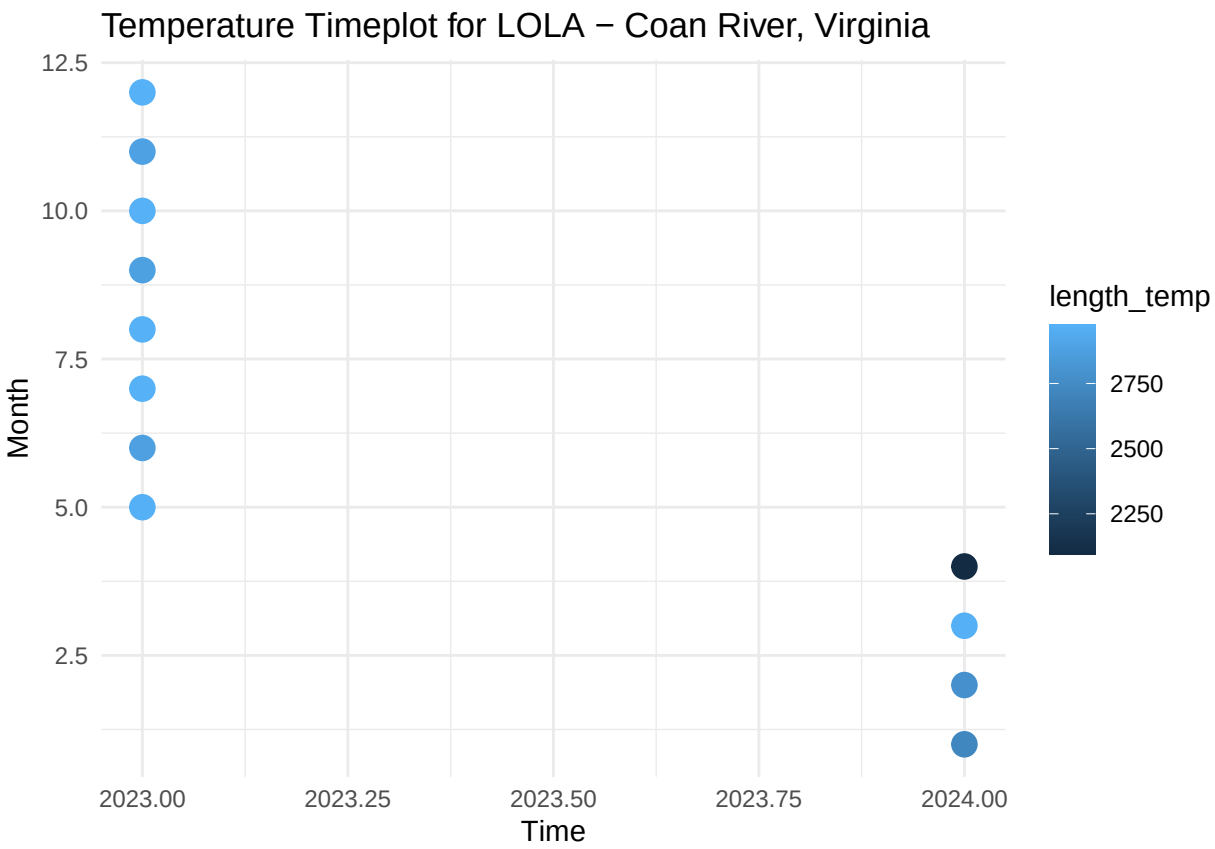
Plot the months and years of data collection to check if there are any collection gaps in the data.

```
timeplot_sal_LOLA <- ggplot(LOLA_envrmonth_sal, aes(x = year)) +
    geom_point(aes(y = month, color = length_salinity), size = 4) +
    labs(x = "Time", y = "Month", title = "Salinity Timeplot for LOLA - Coan River, Virginia") +
    ylim(1,12) +
    theme_minimal()

timeplot_sal_LOLA
```



Plot the months and years of data collection to check if there are any collection gaps in the data.

```
timeplot_temp_LOLA <- ggplot(LOLA_envrmonth_temp, aes(x = year)) +
    geom_point(aes(y = month, color = length_temp), size = 4) +
    labs(x = "Time", y = "Month", title = "Temperature Timeplot for LOLA - Coan River, Virginia") +
    ylim(1,12) +
    theme_minimal()

timeplot_temp_LOLA
```



We can now calculate a list of variables that we will have collected for all sites. This will allow us to compare sites easily. We will calculate the number of observations from each site, the mean annual, maximum annual, and minimum annual value for all variables.

Our list of variables includes:

- Mean_Annual_Temperature_C: average of all available data
- Mean_max_temperature_C: average of maximums for each year
- Mean_min_temperature_C: average of minimums for each year
- Temperature_st_dev: standard deviation of all available data
- Temperature_n: total number of data points
- Temperature_years: number of years in data set
- Mean_Annual_Salinity_ppt: average of all available data
- Mean_min_Salinity_ppt: average of minimums for each year
- Mean_max_Salinity_ppt: average of maximums for each year
- Salinity_st_dev: standard deviation of all available data
- Salinity_n: total number of data points
- Salinity_years: number of years in data set

```r
#Calculate temperature variables.
#Calculate temperature variables.
Mean_Annual_Temperature_C <- mean(LOLA_env$temp)
Mean_max_temperature_C <- mean(LOLA_envryear_temp$max_temp)
Mean_min_temperature_C <- mean(LOLA_envryear_temp$min_temp)
Temperature_st_dev <- sd(LOLA_env$temp)
Temperature_n <- nrow(LOLA_env)
Temperature_years <- nrow(LOLA_envryear_temp)

#Create a data frame to store the temperature results
LOLA_temp <- cbind(site_name, download_date, source_description, lat, lon, firstyear, finalyear, Mean_A
print(LOLA_temp)
```

```
##      site_name download_date source_description   lat       lon
## [1,] "LOLA"    "05-10-2024"  "William Reay, VIMS" "37.9803" "-76.4619"
##      firstyear finalyear Mean_Annual_Temperature_C Mean_max_temperature_C
## [1,] "2023"    "2024"    "16.9039561007101"        "24.8645"
##      Mean_min_temperature_C Temperature_st_dev Temperature_n Temperature_years
## [1,] "3.4065"               "7.96143220649764" "34078"       "2"
##      collection_type
## [1,] "continuous"
```

```r
# Write to a unique new CSV file
write.csv(LOLA_temp, "/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/data/envr_raw_d
```

```r
#Calculate the salinity variables
Mean_Annual_Salinity_ppt <- mean(LOLA_env$salinity)
Mean_max_Salinity_ppt <- mean(LOLA_envryear_sal$max_salinity)
Mean_min_Salinity_ppt <- mean(LOLA_envryear_sal$min_salinity)
Salinity_st_dev <- sd(LOLA_env$salinity)
Salinity_n <- nrow(LOLA_env)
Salinity_years <- nrow(LOLA_envryear_sal)


#Create a data frame to store the temperature results
LOLA_salinity <- cbind(site_name, download_date, source_description, lat, lon, firstyear, finalyear, Mea
print(LOLA_salinity)
```

```
##      site_name download_date source_description   lat       lon
## [1,] "LOLA"    "05-10-2024"  "William Reay, VIMS" "37.9803" "-76.4619"
##      firstyear finalyear Mean_Annual_Salinity_ppt Mean_max_Salinity_ppt
## [1,] "2023"    "2024"    "13.837441457832"        "16.875"
##      Mean_min_Salinity_ppt Salinity_st_dev    Salinity_n Salinity_years
## [1,] "9.99"                "2.75705619961467" "34078"    "2"
##      collection_type
## [1,] "continuous"
```

```r
# Write to the combined file with all sites
# Write to a unique new CSV file
write.csv(LOLA_salinity, "/Users/nicolemongillo/Desktop/GitHub/MVP_Chesapeake_VIMS_hatchery/data/envr_r
```