Daily_measurements

PutnamLab

7/19/2022

This script plots daily measurements from the CBLS Wet lab for the Putnam lab. Run this script daily after daily measurements are taken and inputted into the datasheet.

Script updated 11/14/23 by ZD. Script updated 11/16/23 by JA - added upper and lower threshold lines on plots.

Load packages

Load data

```
daily <- read.csv("Data/2023_Daily_measurements_tracking.csv")
head(daily)
```

```
##        Date Time Observer.Initials      Tank_ID Temperature_C pH_mv Salinity_psu
## 1 20230101                              BT 1/2          26.24 -59.4        34.37
## 2 20230102                              BT 1/2          26.32 -61.3        34.44
## 3 20230105                          header_tank         26.23 -51.5        34.48
## 4 20230105                              BT 1/2          26.27 -52.3        34.45
## 5 20230105                              BT 3/4          26.26 -52.6        34.41
## 6 20230106                          header_tank         26.26 -51.9        34.58
##   Orion_Temp tris.date Probe.Set notes
## 1         NA  20221019    Probe1
## 2         NA  20221019    Probe1
## 3         NA  20221019    Probe1
## 4         NA  20221019    Probe1
## 5         NA  20221019    Probe1
## 6         NA  20221019    Probe1
```

```
tail(daily) # check to make sure data from today is there
```

```
##          Date  Time Observer.Initials      Tank_ID Temperature_C pH_mv
## 721 20231224 12:05                HP       BT 1/2          26.58 -57.6
## 722 20231224 12:05                HP       BT 3/4          26.60 -57.7
## 723 20231224 12:05                HP  header_tank         26.55 -57.7
## 724 20231226 15:25                HP       BT 1/2          26.73 -62.6
## 725 20231226 15:25                HP       BT 3/4          26.74 -62.8
## 726 20231226 15:25                HP  header_tank         26.70 -61.6
##     Salinity_psu Orion_Temp tris.date Probe.Set notes
## 721        34.63         NA  20231220    Probe1
## 722        34.65         NA  20231220    Probe1
## 723        34.68         NA  20231220    Probe1
## 724        34.61         NA  20231220    Probe1
```

```
## 725          34.61        NA  20231220     Probe1
## 726          34.67        NA  20231220     Probe1
```

Set dates as characters (needs to be done for merging with the tris calibration file)

```
daily$Date <- as.character(daily$Date)
daily$tris.date <- as.character(daily$tris.date)
```

Probe Set 1 (for the header, all corals, and for the non-quarantine tank) e.g., BT 3/4 and header tank and salt mixing tank

```
daily.probe1 <- daily %>%
  filter(Probe.Set == "Probe1")

range(na.omit(daily.probe1$Temperature_C))
```

```
## [1] 22.220 26.906
```

```
range(na.omit(daily.probe1$pH_mv))
```

```
## [1] -69.6 -38.0
```

```
range(na.omit(daily.probe1$Salinity_psu))
```

```
## [1] 33.40 41.36
```

```
daily.probe2 <- daily %>%
  filter(Probe.Set == "Probe2")

range(na.omit(daily.probe2$Temperature_C))
```

```
## [1] 22.25 26.48
```

```
range(na.omit(daily.probe2$pH_mv))
```

```
## [1] -76.0 -43.7
```

```
range(na.omit(daily.probe2$Salinity_psu))
```

```
## [1] 34.23 36.13
```

Calculate total pH from Probe Set 1

Calculate the calibration curve from the Tris calibration and calculate pH on the total scale from pH.mV.

```
pHcalib<-read_csv("Data/Tris_Calibration.csv")
```

```
## Rows: 240 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): tris.date, mVTris, TTris
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pHcalib$tris.date<-as.character(pHcalib$tris.date)

pHSlope.probe1 <- pHcalib %>%
  group_by(tris.date) %>%
  nest() %>%
```

```
  mutate(fitpH = map(data, ~ lm(mVTris ~ TTris, data = .x))) %>%
  mutate(tidy_fit = map(fitpH, broom::tidy)) %>%
  unnest(tidy_fit) %>%
  select(tris.date, term, estimate) %>%
  pivot_wider(names_from = term, values_from = estimate) %>%
  left_join(daily.probe1, ., by = "tris.date") %>%
  mutate(mVTris = Temperature_C * TTris + `(Intercept)`)

range(pHSlope.probe1$Temperature_C)
```

```
## [1] NA NA
```

```
pHSlope.probe1 <- pHSlope.probe1 %>% filter(!is.na(Temperature_C))
range(pHSlope.probe1$Temperature_C)
```

```
## [1] 22.220 26.906
```

```
range(pHSlope.probe1$pH_mv)
```

```
## [1] NA NA
```

```
pHSlope.probe1 <- pHSlope.probe1 %>% filter(!is.na(pH_mv))
range(pHSlope.probe1$pH_mv)
```

```
## [1] -69.6 -38.0
```

```
range(pHSlope.probe1$Salinity_psu)
```

```
## [1] 33.40 40.51
```

```
pHSlope.probe1 <- pHSlope.probe1 %>% filter(!is.na(Salinity_psu))
range(pHSlope.probe1$Salinity_psu)
```

```
## [1] 33.40 40.51
```

```
range(pHSlope.probe1$pH_mv)
```

```
## [1] -69.6 -38.0
```

```
pHSlope.probe1 <- pHSlope.probe1 %>% filter(!is.na(pH_mv))
range(pHSlope.probe1$pH_mv)
```

```
## [1] -69.6 -38.0
```

```
range(pHSlope.probe1$mVTris)
```

```
## [1] -63.89333 -55.28032
```

```
pHSlope.probe1 <- pHSlope.probe1 %>% filter(!is.na(mVTris))
range(pHSlope.probe1$mVTris)
```

```
## [1] -63.89333 -55.28032
```

```
pHSlope.probe1 <- pHSlope.probe1 %>%
  mutate(pH.total = seacarb::pH(Ex = pH_mv, Etris = mVTris, S=Salinity_psu, T=Temperature_C))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `pH.total = seacarb::pH(Ex = pH_mv, Etris = mVTris, S =
##   Salinity_psu, T = Temperature_C)`.
## Caused by warning in `tris()`:
## ! S, T, and/or b is outside the range of validity for the TRIS buffer pH formulation by DelValls and
```

Calculate total pH from Probe Set 2

Probe Set 2 (for the header, all corals, and for the non-quarantine tank) e.g., BT 1/2 while HI corals were islotaed and quarantine tank

Calculate the calibration curve from the Tris calibration and calculate pH on the total scale from pH.mV.

```
pHcalib2<-read_csv("Data/Tris_Calibration_BT1_2.csv")
```

```
## Rows: 47 Columns: 3
## -- Column specification ---------------------------------------------------
## Delimiter: ","
## dbl (3): tris.date, mVTris, TTris
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pHcalib2$tris.date<-as.character(pHcalib2$tris.date)
```

```
pHSlope.probe2 <- pHcalib2 %>%
  group_by(tris.date) %>%
  nest() %>%
  mutate(fitpH = map(data, ~ lm(mVTris ~ TTris, data = .x))) %>%
  mutate(tidy_fit = map(fitpH, broom::tidy)) %>%
  unnest(tidy_fit) %>%
  select(tris.date, term, estimate) %>%
  pivot_wider(names_from = term, values_from = estimate) %>%
  left_join(daily.probe2, ., by = "tris.date") %>%
  mutate(mVTris = Temperature_C * TTris + `(Intercept)`)
```

```
range(pHSlope.probe2$Temperature_C)
```

```
## [1] NA NA
```

```
pHSlope.probe2 <- pHSlope.probe2 %>% filter(!is.na(Temperature_C))
range(pHSlope.probe2$Temperature_C)
```

```
## [1] 22.25 26.48
```

```
range(pHSlope.probe2$pH_mv)
```

```
## [1] -76.0 -43.7
```

```
pHSlope.probe2 <- pHSlope.probe2 %>% filter(!is.na(pH_mv))
range(pHSlope.probe2$pH_mv)
```

```
## [1] -76.0 -43.7
```

```
range(pHSlope.probe2$Salinity_psu)
```

```
## [1] 34.23 36.05
```

```
pHSlope.probe2 <- pHSlope.probe2 %>% filter(!is.na(Salinity_psu))
range(pHSlope.probe2$Salinity_psu)
```

```
## [1] 34.23 36.05
```

```
range(pHSlope.probe2$pH_mv)
```

```
## [1] -76.0 -43.7
```

```
pHSlope.probe2 <- pHSlope.probe2 %>% filter(!is.na(pH_mv))
range(pHSlope.probe2$pH_mv)
```

```
## [1] -76.0 -43.7
range(pHSlope.probe2$mVTris)
```

```
## [1] NA NA
pHSlope.probe2 <- pHSlope.probe2 %>% filter(!is.na(mVTris))
range(pHSlope.probe2$mVTris)
```

```
## [1] -70.93191 -66.57079
pHSlope.probe2 <- pHSlope.probe2 %>%
  mutate(pH.total = seacarb::pH(Ex = pH_mv, Etris = mVTris, S=Salinity_psu, T=Temperature_C))
```

Join Probe 1 and Probe 2 Sets

```
pHSlope <- rbind(pHSlope.probe1, pHSlope.probe2)
```

Convert date to ymd for plotting

```
pHSlope <- pHSlope %>%
  filter(!Tank_ID == "quarantine_tank")

pHSlope$Date <- ymd(pHSlope$Date) # convert 8 digit date into datetime format

pHSlope <- pHSlope %>% relocate("pH.total", .after = Salinity_psu) %>%
  relocate(pH_mv, .after = pH.total)
```

Change to long format

Change data format to long format

```
pHSlope.long <-pHSlope %>% pivot_longer(cols=Temperature_C:pH.total,
  names_to = "metric",
  values_to = "value")
```

Filter by relevant dates if needed

Plot

Make a list of dataframes, each containing a horizontal line that will correspond to the upper and lower
threshold of each parameter (temperature, salinity, pH total)

```
hlines_data <- list(
  data.frame(yintercept = 25.5, metric = "Temperature_C"), # lower threshold for temperature in C°
  data.frame(yintercept = 27, metric = "Temperature_C"), # upper threshold for temperature in C°
  data.frame(yintercept = 34.5, metric = "Salinity_psu"), # lower threshold for salinity in psu
  data.frame(yintercept = 35.5, metric = "Salinity_psu"), # upper threshold for salinity in psu
  data.frame(yintercept = 7.9, metric = "pH.total"), # lower threshold for total pH
  data.frame(yintercept = 8.1, metric = "pH.total") # upper threshold for total pH
    )
```
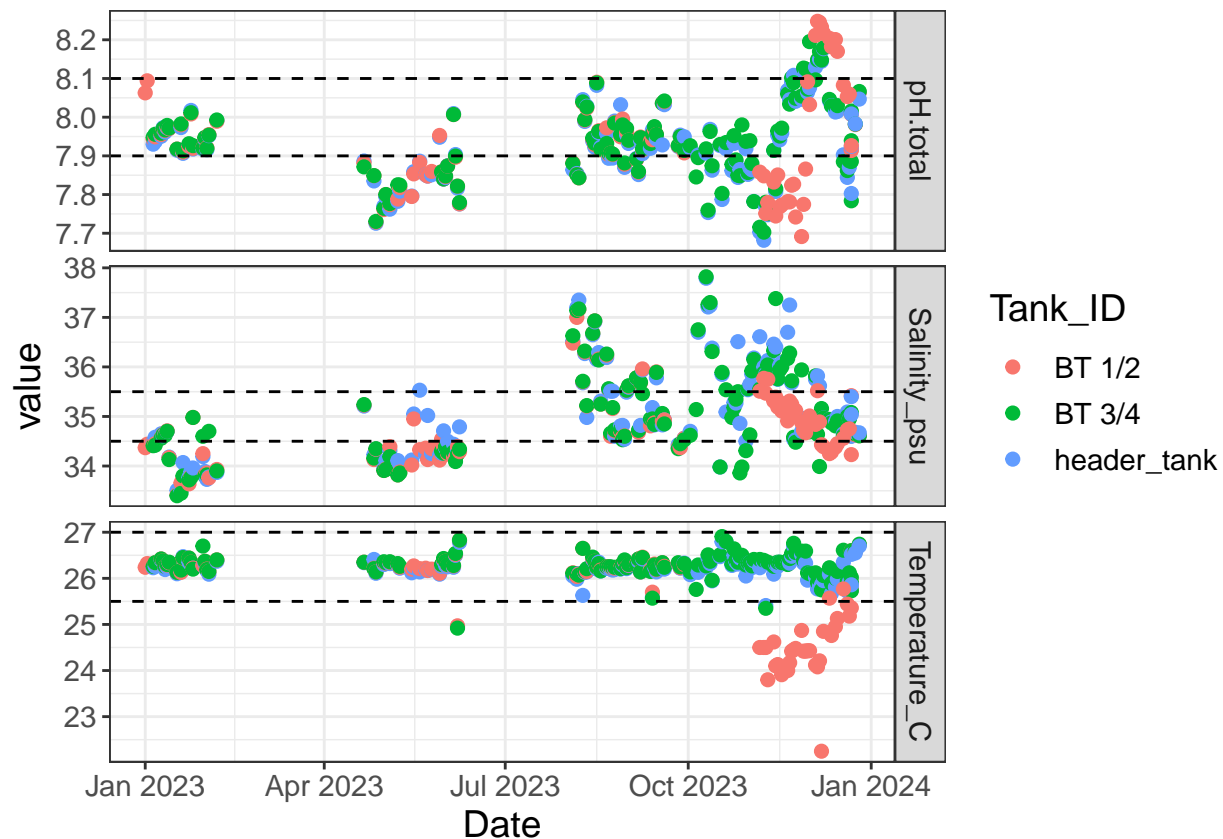
Plot all dates

```
daily_tank<-pHSlope.long %>%
  ggplot(aes(x=Date, y=value, colour=Tank_ID))+
  geom_point(size=2)+
  xlab("Date")+
  facet_grid(metric ~ ., scales = "free")+
  geom_hline(data = hlines_data[[1]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[2]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[3]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[4]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[5]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[6]], aes(yintercept = yintercept), linetype = "dashed") +
  theme_bw() +
  theme(text = element_text(size = 14)); daily_tank
```



```
# Save plot
ggsave("Output/Daily_Measurements.pdf", daily_tank, width = 20, height = 15, units = c("in"))
```

It isn't super informative to look at multiple years in one plot.

Filter and plot by the past month

```
# Filter to the past 30 days
daily_month <- pHSlope.long %>%
  filter(Date >= Sys.Date() - 30)  # Adjust 30 to the number of days you want to consider

# Plot data from the past month
daily_month_plot <-daily_month %>%
  ggplot(aes(x=Date, y=value, colour=Tank_ID))+
```

6

```
  geom_point(size=2)+
  xlab("Date")+
  facet_grid(metric ~ ., scales = "free")+
  geom_hline(data = hlines_data[[1]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[2]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[3]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[4]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[5]], aes(yintercept = yintercept), linetype = "dashed") +
  geom_hline(data = hlines_data[[6]], aes(yintercept = yintercept), linetype = "dashed") +
  theme_bw() +
  theme(text = element_text(size = 14))
daily_month_plot
```
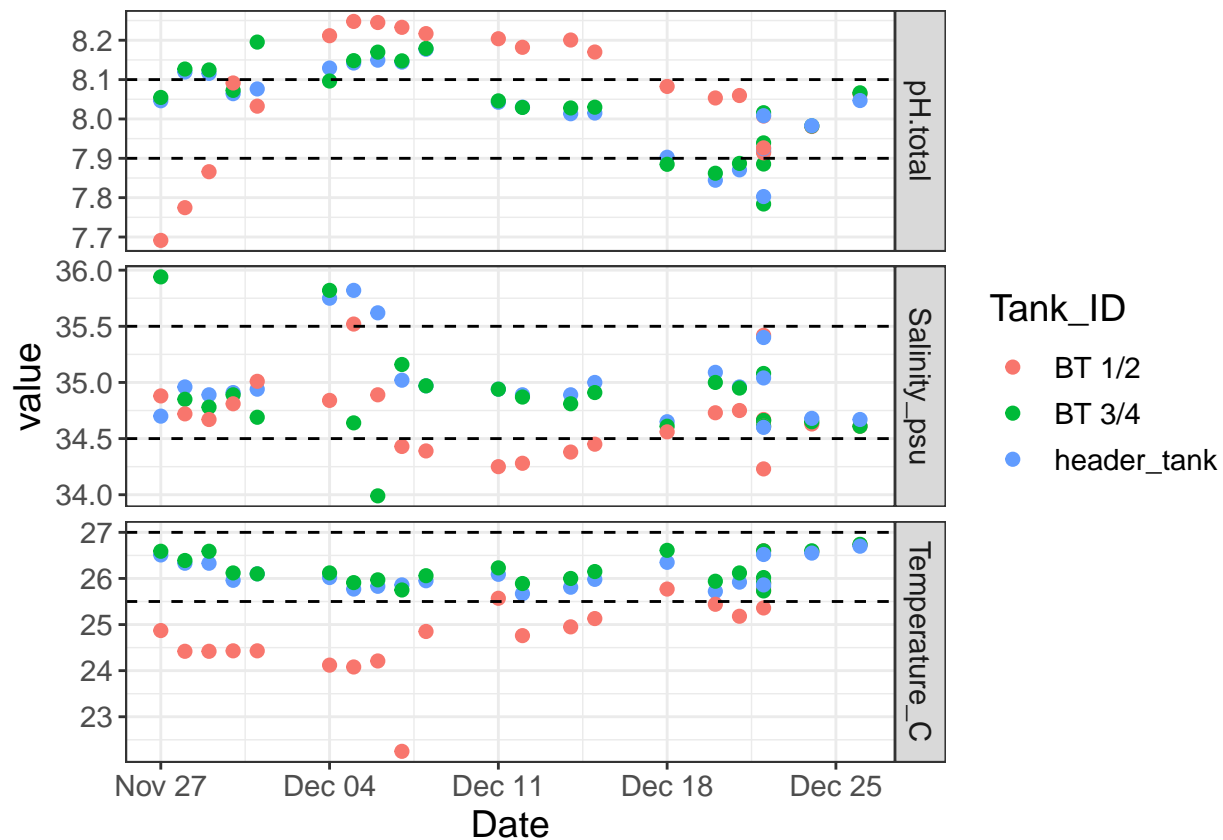


```
# Save plot
ggsave("Output/Daily_Measurements_Past_Month.pdf", daily_month_plot, width = 20, height = 15, units = c
```

Summarize daily measurements over the past month (this will use the `daily` df from above instead of the the `daily.long` df)

```
daily_month <- pHSlope %>%
  filter(Date >= Sys.Date() - 30)  # Adjust 30 to the number of days you want to consider

summary<-daily_month%>%
  group_by(Tank_ID)%>%
  #select(!tank)%>%
  select(Temperature_C:pH.total) %>%
  summarise(across(everything(), list(mean = mean, sd = sd), na.rm = TRUE)); summary
```

```
## Adding missing grouping variables: `Tank_ID`

## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), list(mean = mean, sd = sd), na.rm =
##   TRUE)`.
## i In group 1: `Tank_ID = "BT 1/2"`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))

## # A tibble: 3 x 7
##   Tank_ID  Temperature_C_mean Temperature_C_sd Salinity_psu_mean Salinity_psu_sd
##   <chr>                 <dbl>            <dbl>             <dbl>           <dbl>
## 1 BT 1/2                 25.0            0.998              34.7           0.328
## 2 BT 3/4                 26.2            0.305              34.9           0.395
## 3 header_~               26.1            0.305              35.0           0.340
## # i 2 more variables: pH.total_mean <dbl>, pH.total_sd <dbl>
```