Mendota_EpiVolume

Lauren A. Knose, ORISE-EPA

2023-08-25

The purpose of this program is to identify the thermocline depth from temperature profiles and calculate the volume change of the epilimnion for mass balance.

Step 1. Load dependent packages and data:

a. loading dependent packages…

```
library(ggplot2) #needed for plots
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(dplyr) #needed for reformatting data
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## 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(rLakeAnalyzer) #needed to find thermocline depth from temperature profile
```

```
## Warning: package 'rLakeAnalyzer' was built under R version 4.2.3
```

```
library(reshape2) #needed to reshape data frame from wide to long or long to wide
```

```
## Warning: package 'reshape2' was built under R version 4.2.3
```

```
library(zoo) #needed to linearly interpolate missing values
```

```
## Warning: package 'zoo' was built under R version 4.2.3
```

```
##
## Attaching package: 'zoo'
```

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

Packages loaded.

b. loading dependent data…

```
watertemp<-read.csv(file="Original_data/ntl29_v12_temp.csv") #data from temp string
```

Data loaded.

Step 2. Define and filter the data:

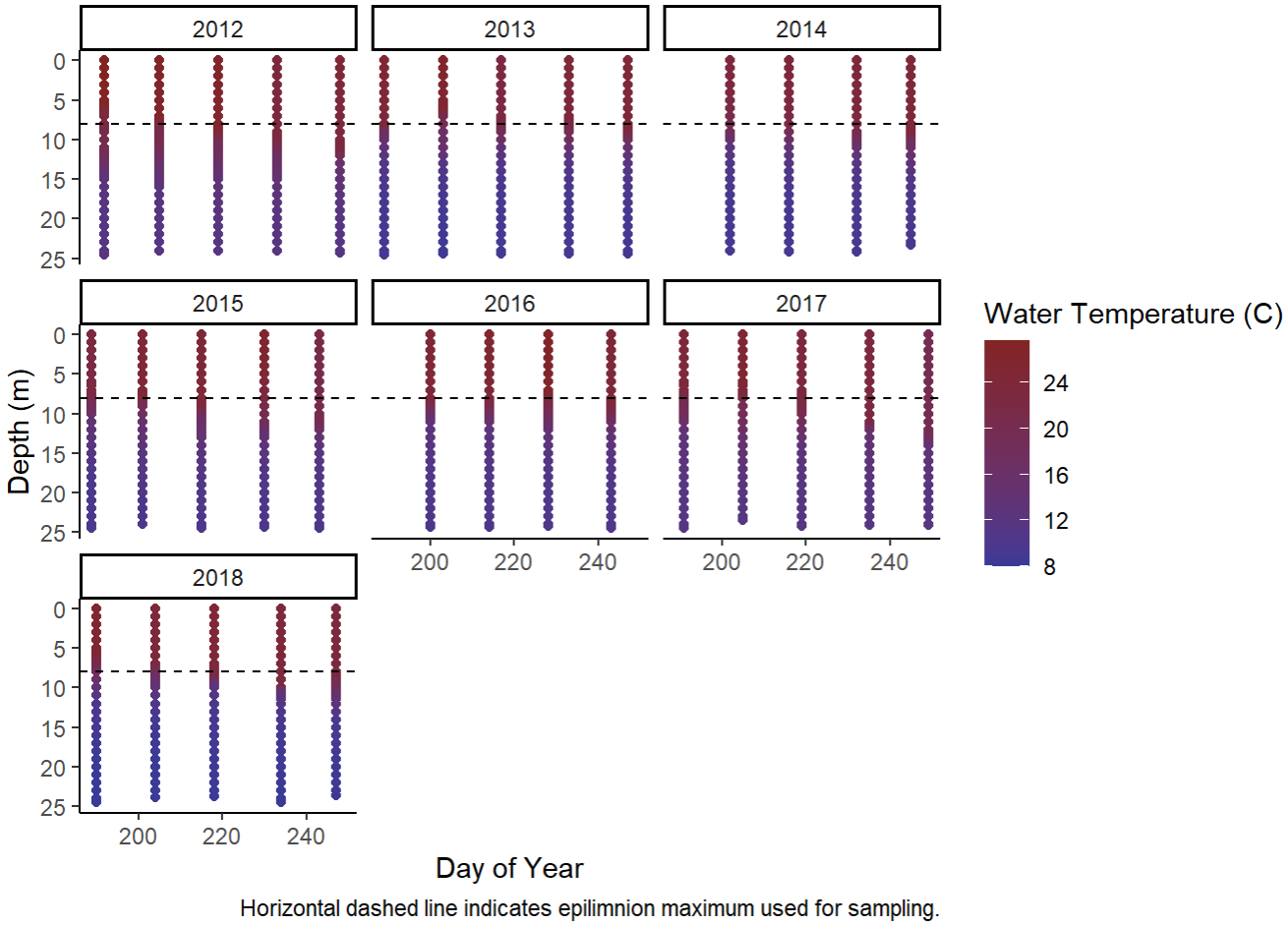
defining and filtering data…

```
watertemp2<- watertemp %>% #make new data frame
  mutate(sampledate=as.Date(sampledate, format="%m/%d/%Y",#specify format of date
                            origin="1899-12-30"), #tells R the origin for dates
         Depth_m=depth, depth=NULL, #rename column with units and remove duplicate
         watertemp_C=wtemp, wtemp=NULL,#rename column with units and remove duplicate
         daynum=as.integer(format(sampledate, format="%j")), #add in day number
         Yr=as.integer(format(sampledate, "%Y"))) %>% #add in year
  filter(lakeid=="ME") %>% #filter for Lake Mendota data only
  filter(rep==1) %>% #filter for rep 1 (no analytical reps)
  filter(!is.na(watertemp_C)) %>% #remove NAs
  filter(Yr>=2012 & Yr <=2018) %>% #filter for the study years
  filter(daynum>= 189 & daynum <=261) %>% #filter for CyanoHAB season +- 6 days
  select(sampledate, Depth_m, watertemp_C, daynum, Yr) #select vars needed
```

Water temperature data defined and filtered.

Step 3. View the data available:

```
epi_depth_m <- 8 #specify the bottom of the epilimnion depth sampled
ggplot(data=watertemp2, #create limno plot with depth inverse on y-axis
       aes(x=daynum, #sampled date as x-axis
           y=Depth_m,
           color=watertemp_C)) + #and data points colored by temperature
  geom_point() + #plot points
  facet_wrap(~Yr) + #make separate plot for each year
  scale_y_reverse() + #inverses the y-axis from zero to max(depth) for limno plot
  scale_color_gradient(high=scales::muted("red"), #red = hot temp
                       low=scales::muted("blue"), #blue = cold temp
                       name="Water Temperature (C)") + #name of legend
  geom_hline(yintercept=epi_depth_m, lty="dashed") + #add line for bottom of epi
  theme_classic() +
  labs(x="Day of Year", y="Depth (m)",
  caption="Horizontal dashed line indicates epilimnion maximum used for sampling.")
```



Temperature data plotted. Sampling determined to be bimonthly (n = 5 per year). Data frame is in long format.

Step 4. Calculate the thermocline depth for each observation:

Note, rLakeAnalyzer package is required here. Data frame needs to be in wide format with each depth a separate column (add prefix= “wtr_”) to match package function dependencies.

a. reshaping the data frame for package…

```
wtr<- watertemp2 %>%
  select(sampledate, Depth_m, watertemp_C) #only include needed columns
wtr<- dcast(wtr, sampledate ~ Depth_m) # reshape data frame from long to wide
```

```
## Using watertemp_C as value column: use value.var to override.
```

```
colnames(wtr)<- paste("wtr", colnames(wtr), sep="_") #add wtr_ before depth to
# fit package format dependencies
wtr<- wtr %>%
  mutate(datetime=wtr_sampledate, wtr_sampledate=NULL) #rename column to fit
# package format dependencies
```

Data frame reshaped and ready for input into the package.

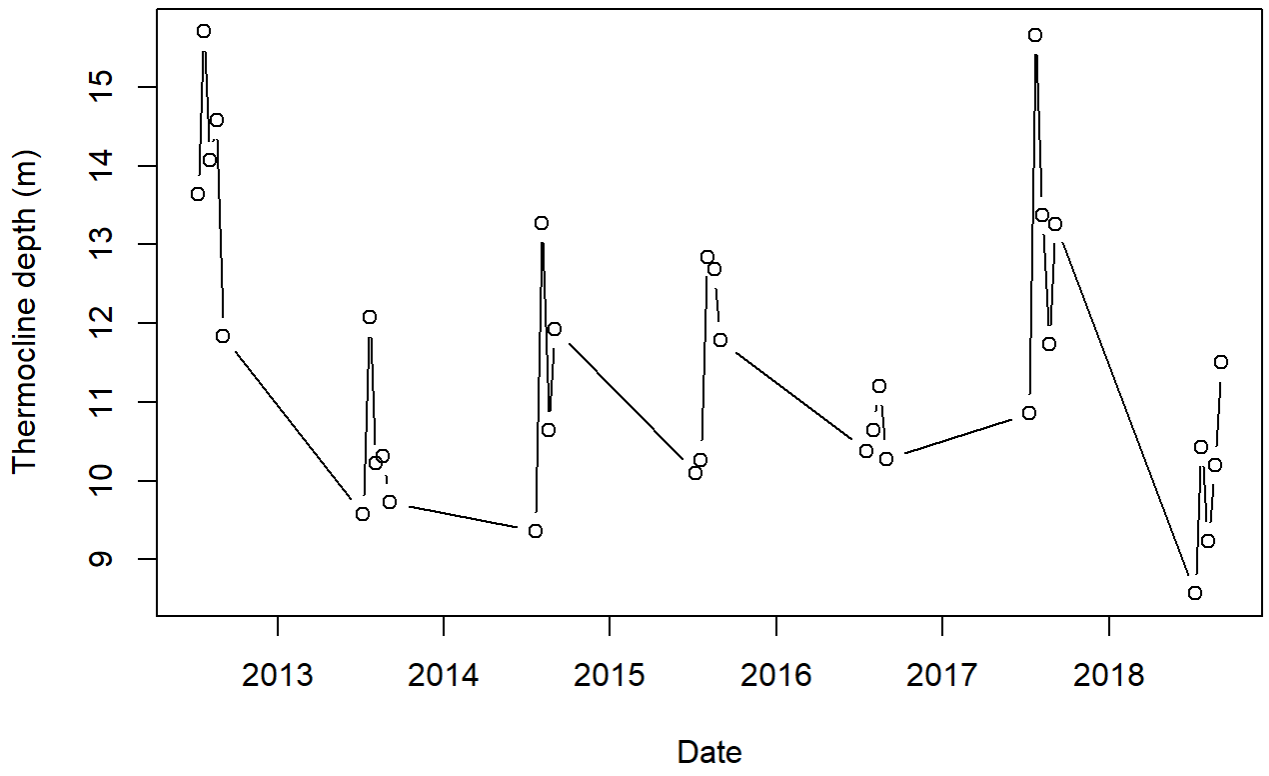
b. using ts.thermo.depth function to calculate thermocline depth…

```
t.d<- ts.thermo.depth(wtr, na.rm=TRUE) #calculates thermocline depth as time series
```

Thermocline depth calculated.

c. plotting the thermocline depth by day during CyanoHAB season…

```
plot(x=t.d$datetime, y=t.d$thermo.depth, type="b",
     xlab="Date",
     ylab="Thermocline depth (m)")
```



Thermocline depth calculated and plotted for each date.

Step 5. Calculate the change in volume of epilimnion from change in thermocline

depth:

a. separating data by year, then assigning thermocline depth at t and t-1…

```
t.d2<- t.d %>% #make new table
  mutate(Yr=as.numeric(format(datetime, "%Y"))) %>% #add in year
  group_by(Yr) %>% #for each year of day
  mutate(Thermo_depth_m=thermo.depth, thermo.depth=NULL, #add units and remove duplicate column
         Thermo_depth_m_k1=lag(Thermo_depth_m, n=1)) #thermocline depth at t-1
```

b. reassigning first thermocline depth at t-1 each year (NAs) as thermocline depth at t…

```
t.d2$Thermo_depth_m_k1[is.na(t.d2$Thermo_depth_m_k1)] <- t.d2$Thermo_depth_m[is.na(t.d2$Thermo_d
epth_m_k1)]#replace NAs in t-1 with the t value
```

c. calculating change in thermocline depth between t and t-1 and change in epilimnion volume…

```
A_strat= 25850000 # basin surface area of stratified region in m^2
Vol_epi<- t.d2 %>%
  mutate(Chg_Thermodepth_m = (Thermo_depth_m - Thermo_depth_m_k1),#calculate daily change in dep
th
         Vol_epi_m3 = Chg_Thermodepth_m*A_strat,#calculate daily change in volume
         sampledate=datetime, datetime=NULL, #rename column, remove duplicate
         daynum=as.numeric(format(sampledate, "%j")) #add in daynum
         )  %>%
  select(sampledate, Yr, daynum, Vol_epi_m3, Thermo_depth_m)
```

Change in epilimnion volume calculated between observations.

Step 6. Bin observations into same day number each year:

a. defining expected observation frequencies based on sampling regime…

```
Yr<- seq(from=2012, #define year start of study period
             to=2018, #define year end of study period
             by=1) #creates a sequence every 1 year
Yr #prints the expected years of data
```

```
## [1] 2012 2013 2014 2015 2016 2017 2018
```

```
daynum_bins<- seq(from=196, #define date start of CyanoHAB season
                  to=255, #define date end of CyanoHAB season
                  by=14) #creates a sequence every 14 days
daynum_bins #prints the expected observation dates of data
```

```
## [1] 196 210 224 238 252
```

Defined expected observation frequency.

b. binning observed dates into expected dates…

```
Vol_epi2<- Vol_epi %>%
mutate(daynum_bin=ifelse(daynum<(daynum_bins[2]-7), #if observation is <= bin 2 min
       daynum_bins[1], #yes returns bin 1
       ifelse(daynum>=(daynum_bins[2]-7)&daynum<(daynum_bins[3]-7),
              #if observation is > bin 2 min and <= bin 3 min
              daynum_bins[2], #yes returns bin 2
              ifelse(daynum>=(daynum_bins[3]-7)&daynum<(daynum_bins[4]-7),
                     #if observation is > bin 3 min and <= bin 4 min
                     daynum_bins[3], #yes returns bin 3
                     ifelse(daynum>=(daynum_bins[4]-7)&daynum<(daynum_bins[5]-7),
                            #if observation is > bin 4 min and <= bin 5 min
                            daynum_bins[4], #yes returns bin 4
                            daynum_bins[5])#no returns bin 5
                     )
              )
       )
)
Vol_epi2<- Vol_epi2 %>%
  filter(!duplicated(cbind(Yr, daynum_bin), fromLast = TRUE)) #remove duplicate
```

Observed dates assigned into closest expected dates.

Step 7. Interpolate missing observations:

a. creating new data frame with expected values…

```
Vol_epi3<- data.frame(# makes new data frame
                      daynum_bin=rep(daynum_bins, times=7), #creates field with
                      #expected observations (5/yr for 7 years)
                      Yr=rep(Yr, each= 5)) #creates field with expected yr
```

Table created with expected observation frequency.

b. merging the two data frames…

```
Vol_epi3<- merge(Vol_epi3, Vol_epi2, #two data frames to join
                 by=c("Yr", "daynum_bin"), #join by year and day number
                 all.x=TRUE) #keep the predicted observation days
```

Data frames merged.

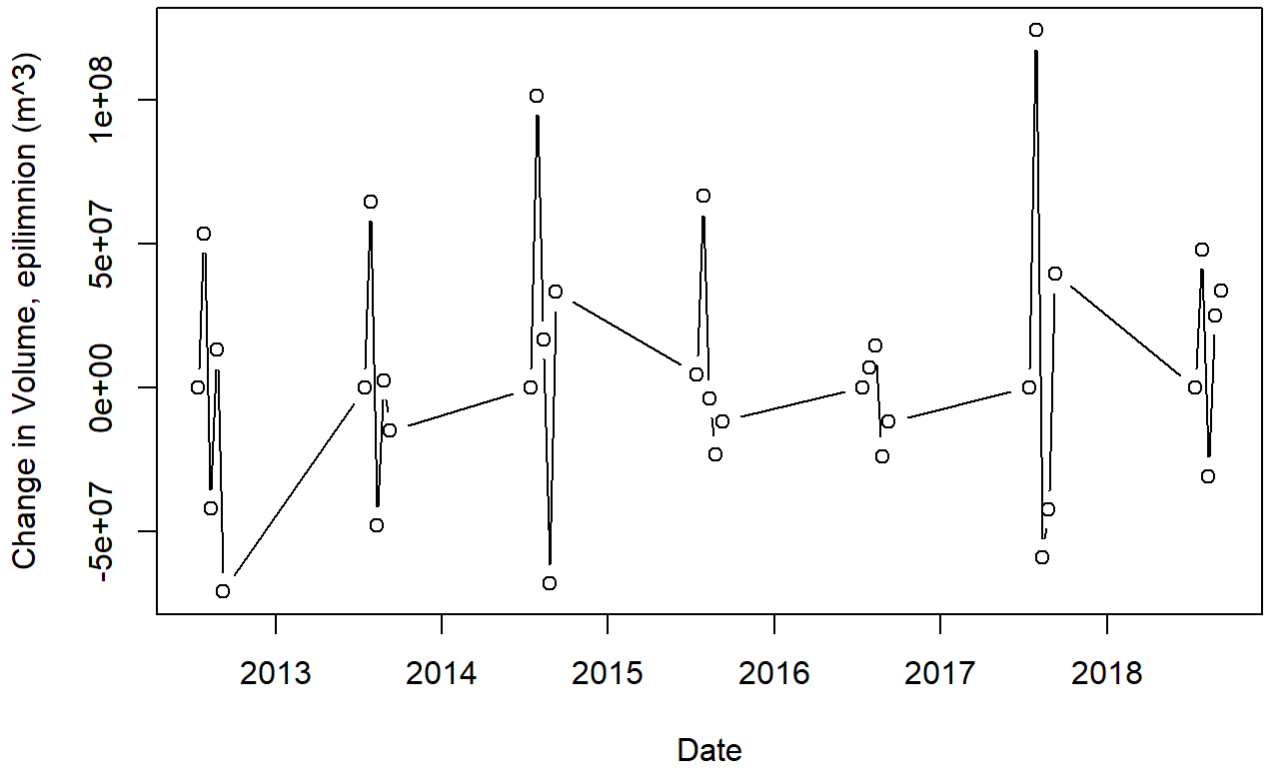
c. interpolating sample date from day and year…

```
Vol_epi_int<- Vol_epi3 %>%
  group_by(Yr) %>% #within each year
  mutate(YrDay=paste(Yr, daynum_bin, sep="/"), #combined day of year and year
         date=as.Date(YrDay, format="%Y/%j", #convert  year, day to date
                      oorigin="1899-12-30")) %>%
  ungroup()
```

Missing sample dates interpolated.

d. interpolating change in epi volume between observations…

```
Vol_epi_int$Vol_epi_m3<- na.approx(Vol_epi_int$Vol_epi_m3) #interpolates NAs
plot(x=Vol_epi_int$date, y=Vol_epi_int$Vol_epi_m3, type="b", #plots new data
     xlab="Date",
     ylab="Change in Volume, epilimnion (m^3)")
```



Missing data interpolated.

Step 8. Save data as new data file.

```
Vol_epi_int <- Vol_epi_int %>% select(date, Vol_epi_m3, Thermo_depth_m)
write.csv(Vol_epi_int, file="Cleaned_data/Mendota_EpiVolume.csv")
```

Data saved as new file in Cleaned_data folder as Mendota_EpiVolume.csv”