DA Submission

Axel Amzallag, Joshua Hug & Tim Kalnins

4/3/2021

All the data in this file should run as long as the folders are kept intact, and that the directory that this file is located in is the working directory.

Our current progress is still mostly focused on exploring the data more statistically, but we did manage to fit a model right at the end of the most recent week. However, it is doubtful that this will be our final model. Additionally, we have found a quality typhoon data set that we can use for comparing the typhoon have combined our data set with a new data set. In addition, we have found 2 papers that describe methods for fitting mean fields on ARGOs data. We've also fit a model that takes empirical thermocline data as its input and outputs a continuous, Gaussian model for the depth of the initial thermocline dip. Using a continuous model like this is logical because the depth at which the initial dip occurs doesn't include jumps and so behaves like a smooth function.

Creating Measurements for Thermocline Depth

In order to fit the model that we wanted to attempt, we had to create a new variable for the thermocline depth, which we did by consulting an empirical study by Paul C. Fiedler in 2010 (https://aslopubs.onlinelibrary.wiley.com/doi/pdfdirect/10.4319/lom.2010.8.313). The difficulty in this measurement comes from the fact that the temperature decreases slowly during the "mixed layer" layer at the top of the ocean (called the epipelagic zone), and then begins dropping more steeply as some point between 30-200 meters below the surface. For example, choosing the point with steepest decrease leads to choosing a depth in the center of the thermocline instead of at the top of it. In fact, oceanographers are still debating the merits of different methods to calculate the depth of the thermocline.

Fiedler's study looks at 5 different empirical methods to try and locate the depth at which the temperature of ocean water begins its steep drop from its values at the "mixed layer", near the surface (called the epipelagic or sunlight zone). Out of these 5 methods, when tested on data where thermocline location was known, the one that was most accurate was what he termed the "variable representative isotherm method".

Variable Representative Isotherm Method

The variable representative isotherm method uses the formula $\mathrm{TT} = 0.25\,[\mathrm{T(MLD)} - \mathrm{T(400m)}]$ to approximate the temperature at which the thermocline begins to drop. TT is the temperature of the thermocline, T(MLD) is the temperature at the base of the mixed layer – defined by Fiedler as $\mathrm{T(MLD)} = \mathrm{SST} - 0.8$ – and T(400m) is the temperature at a depth of 400 meters. SST stand for Sea Surface Temperature. In the temperature range used in the study, 12 - 30 degrees surface temperature, this method was the most accurate, with a mean error of only a few meters. The Root Mean Squared Error (RMSE) was 10 meters, while the other methods had RMSE values around 20 meters on the same data.

Weaknesses of Variable Representative Isotherm: The paper mentions that this method is not tested in water with surface temperatures lower than 12 degrees Celsius, and in our empirical analysis the method gave poor results in this region. Also, looking at ocean water in climate zones considered polar or temperate gives inaccurate measurements. Therefore, the measurement should only be used in the tropical and sub-tropical climate zones (roughly latitude 0 to 40 degrees North/South of Equator). Since typhoons typically begin in these latitudes, this measurement of thermocline depth should not cause issues for use in our project.

Below is the function used to get thermocline depth, as well as surface temperature:

```
##
## Useful buoys -- Deep enough to use the variable representative isotherm
## formula (> 400 meters) and should also have surface temperature values.
##
## For the variable isotherm formula, the equation is as follows:
##
##  TT = T(MLD) - 0.25[T(MLD) - T(400m)]
##
## TT -- Thermocline Temperature
## T(MLD) -- Temperature at the base of the mixed layer
## T(400m) -- Temperature at 400 meters
##
## For T(MLD), we use surface temperature minus 0.8. Surface temperature is
## calculated as the mean of the temperature from 10 to 30 meters deep.
##
## See Fiedler 2010, pg. 319, for more details. Link below:
## https://aslopubs.onlinelibrary.wiley.com/doi/pdfdirect/10.4319/lom.2010.8.313
##
getThermDepth <- function(all_buoys_no_date){

  deepBuoys <- all_buoys_no_date %>%
    filter(PRES_ADJUSTED > 400 & TEMP_ADJUSTED < 35) %>%
                 group_by(PLATFORM_NUMBER , LONGITUDE , LATITUDE) %>%
                 summarise( TEMP_DEEP = max(TEMP_ADJUSTED ) )

  surfaceBuoys <- all_buoys_no_date %>%
    filter(PRES_ADJUSTED < 30 & PRES_ADJUSTED > 10) %>%
                  group_by(PLATFORM_NUMBER , LONGITUDE , LATITUDE) %>%
                  summarise( SURFACE_TEMP = mean(TEMP_ADJUSTED) )

  usefulBuoys <- inner_join( deepBuoys , surfaceBuoys )

  usefulBuoys$TEMP_MLD <- usefulBuoys$SURFACE_TEMP - 0.8

  usefulBuoys$TEMP_THERMOCLINE <- usefulBuoys$TEMP_MLD - 0.25 *
    ( usefulBuoys$TEMP_MLD - usefulBuoys$TEMP_DEEP )

  usefulBuoys$THERMOCLINE_DEPTH <- NA



  ## This for() loop adds the THERMOCLINE_DEPTH to each entry using the value
  ## immediately below the thermocline level.
  ##
  ## ~ 1-2% of the data is quite strange here: These data points tend to be at
  ## high latitudes with low surface temperatures.
  ##
  ## Also, this code is slow, even though it runs.
  for(i in 1:nrow(usefulBuoys)){
    single_working_buoy <- all_buoys_no_date %>%
                           filter( PLATFORM_NUMBER ==
                                     usefulBuoys$PLATFORM_NUMBER[i] &
```

```
                                   LONGITUDE == usefulBuoys$LONGITUDE[i] &
                                   LATITUDE == usefulBuoys$LATITUDE[i] )

    dist_to_thermocline <- abs( single_working_buoy$TEMP_ADJUSTED -
                                  usefulBuoys$TEMP_THERMOCLINE[i] )
    temp_sorting_value <- sort( dist_to_thermocline )[7]

    rows_near_therm <- single_working_buoy[ dist_to_thermocline <
                                              temp_sorting_value , ]

    usefulBuoys$THERMOCLINE_DEPTH[i] <- min( rows_near_therm$PRES_ADJUSTED )

  }

return(usefulBuoys)
}
```

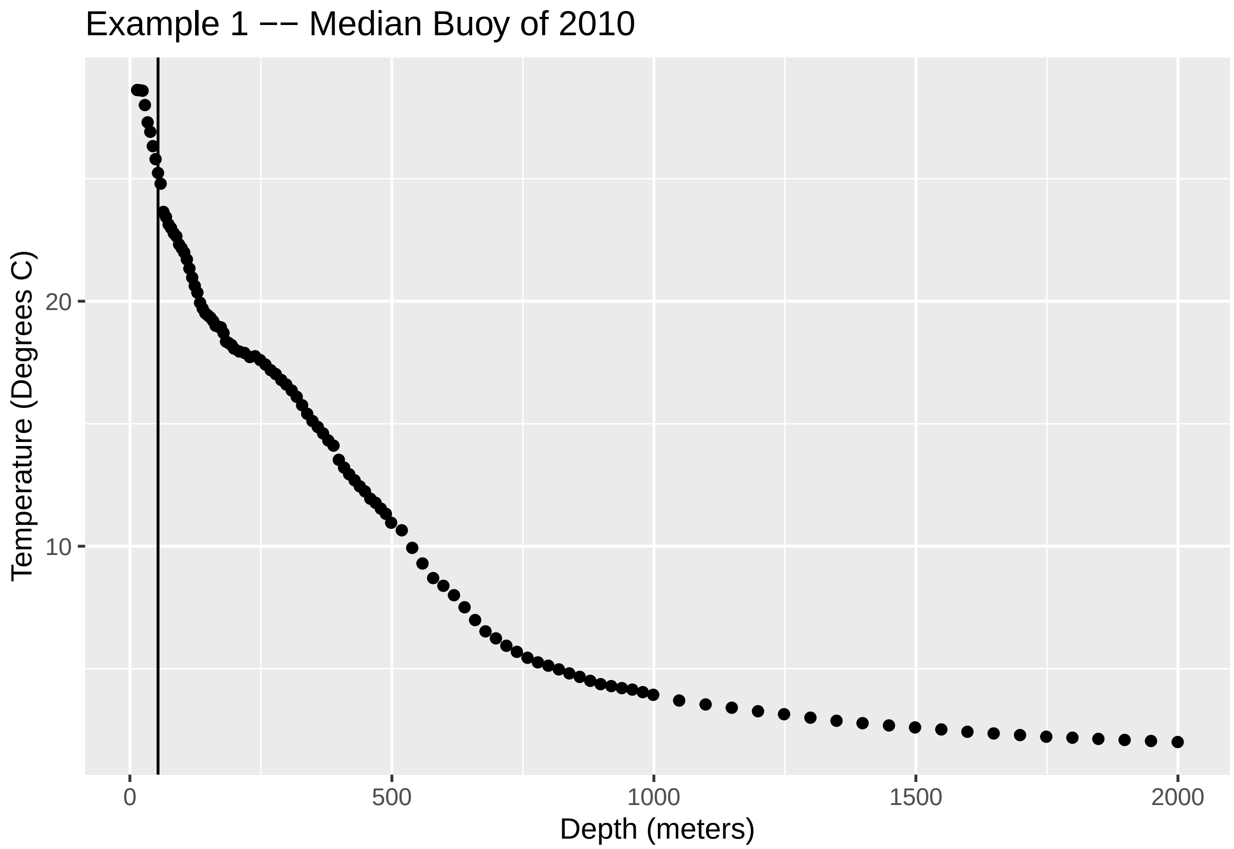
Next, on the data from August 2010, we show a few examples of the thermocline depth measurement using the variable isotherm method. **The vertical lines indicate the depth at which the empirical measurement estimates the temperature dip associated with the beginning of the thermocline**. The graphs are on the following pages.

- Example 1 is a buoy with a typical thermocline. In fact, this was the buoy with the median thermocline depth from August 2010. This buoy is within the latitude boundaries that are best for the variable isotherm method. You can see that the vertical line does a good job of measuring a clear change in the pattern of measurement of the temperature. This demarcation of the surface layer and the thermocline is quite accurate. For these types of points, the vertical line is typically at the 60 meter mark or less.
- Example 2 is a buoy that is outside the climate zone recommended for use of the variable isotherm method. You can see that the vertical line is in a terrible location, and in fact this buoy has no thermocline dip really. This is because this buoy is at 57 degrees of latitude, further north than the variable representative isotherm method should be used, and has a surface temperature of around 3 degrees Celsius, which is a lower value than the variable representative isotherm method should be used for. Buoys like this should not be used in our final analysis.
- Example 3 is a buoy that has a steadily decreasing surface temperature, then has a steep thermocline dip at a deeper point in the ocean. This is the type of buoy that we really want to be identifying correctly using our method. The variable isotherm method is effective at locating the depth at which the steep dip begins. Here, the vertical line is almost exactly at the location where the thermocline begins.

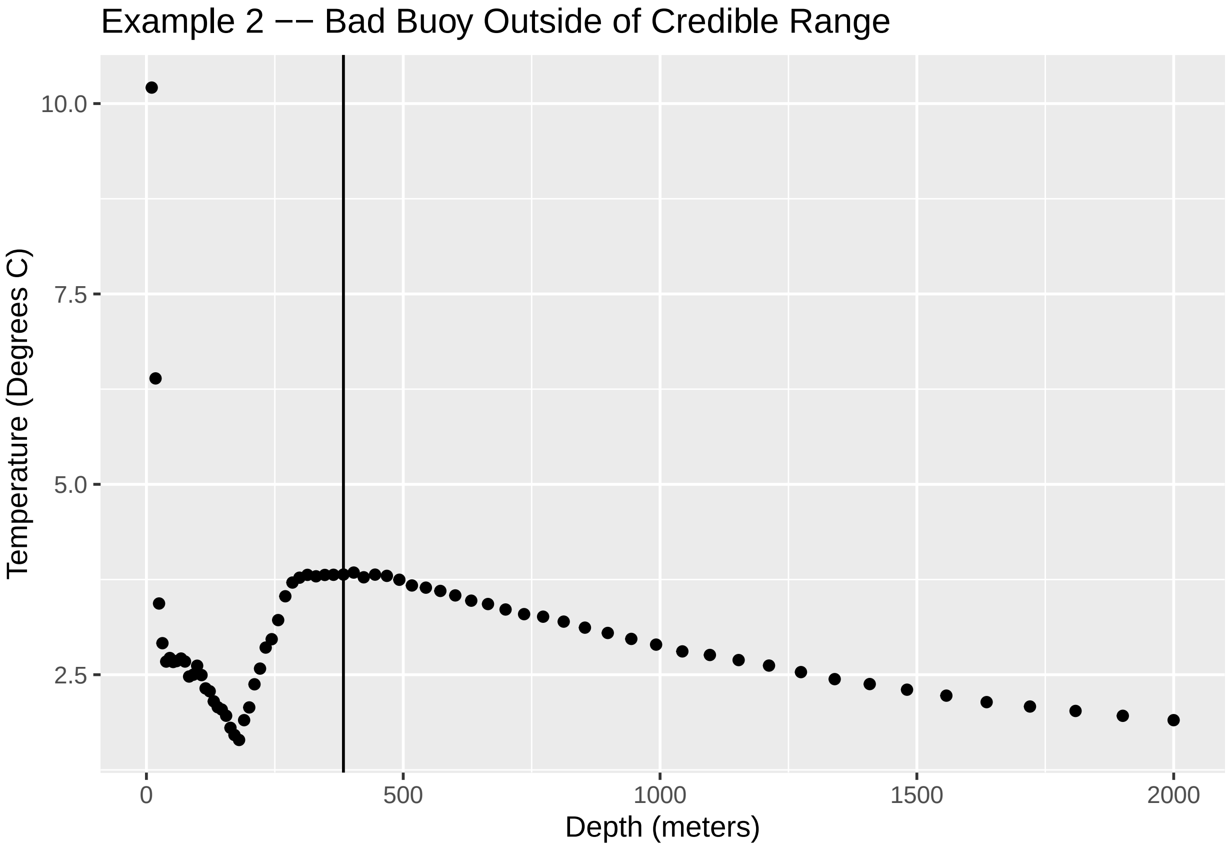
```
# read in all the files
aug2010_full <-read.csv("./august/2010_08_Covariates_Pacific.csv")

# remove NA
aug2010_full <- na.omit(aug2010_full)

# select columns we need
aug2010_subset <- aug2010_full %>% dplyr::select(PLATFORM_NUMBER,
                                                 LONGITUDE, LATITUDE,
                                                 TEMP_ADJUSTED, PRES_ADJUSTED )

aug2010_subset %<>% dplyr::filter(PRES_ADJUSTED > 10)

# get thermocline depth vector and attach it
```

```r
thermdepth <- getThermDepth(aug2010_subset)
```

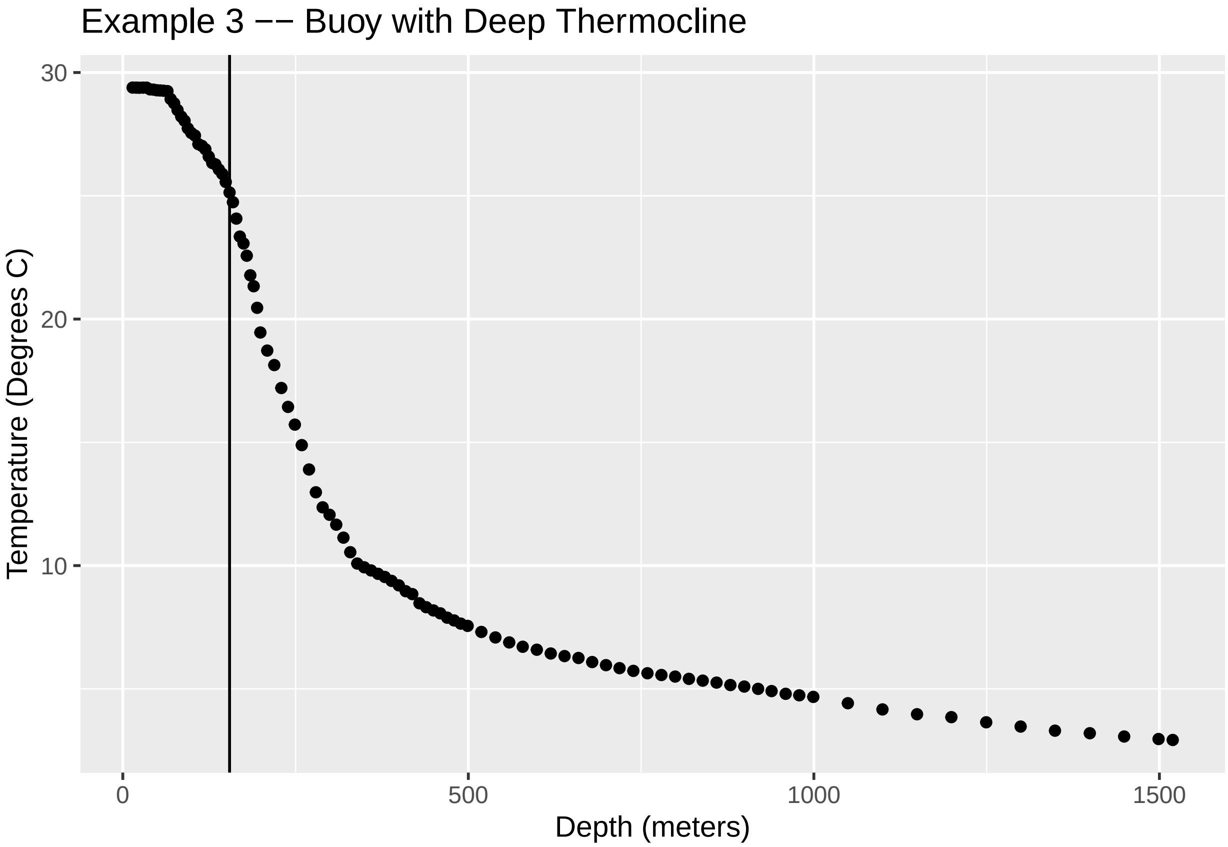
```r
## Example 1 -- Median Buoy of August 2010 ##
test_buoy <- aug2010_subset %>% filter( PLATFORM_NUMBER == 5901537 &
                                        LONGITUDE == 153.830 &
                                        LATITUDE == 26.253 )

test_buoy %>% ggplot( aes( x=PRES_ADJUSTED , y=TEMP_ADJUSTED ) ) +
              geom_point()+ geom_vline(xintercept = 53.7) +
              xlab("Depth (meters)") + ylab("Temperature (Degrees C)") +
              ggtitle("Example 1 -- Median Buoy of 2010")
```



```r
## Example 2 -- Bad Buoy Outside of Credible Range ##
test_buoy <- aug2010_subset %>% filter( PLATFORM_NUMBER == 4900806 &
                                        LONGITUDE == 177.278 &
                                        LATITUDE == 57.794 )

test_buoy %>% ggplot( aes( x=PRES_ADJUSTED , y=TEMP_ADJUSTED ) ) +
              geom_point()+ geom_vline(xintercept = 383.600) +
              xlab("Depth (meters)") + ylab("Temperature (Degrees C)") +
              ggtitle("Example 2 -- Bad Buoy Outside of Credible Range")
```



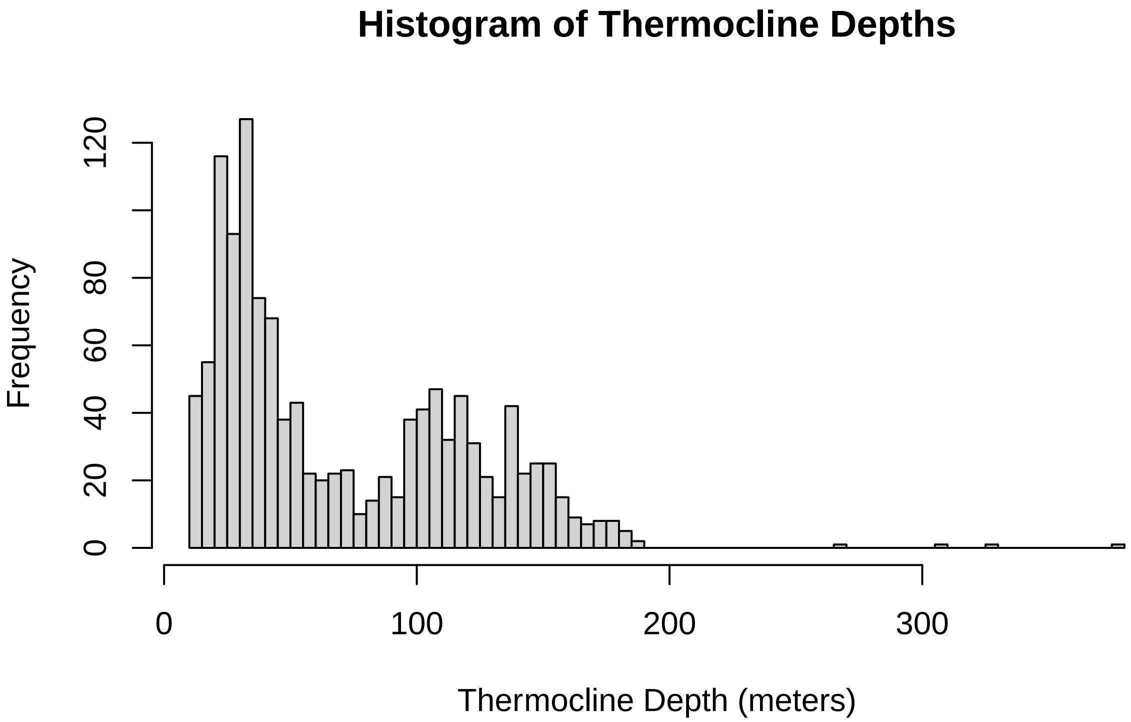
```
## Example 3 -- Buoy with Deep Thermocline ##
test_buoy <- aug2010_subset %>% filter( PLATFORM_NUMBER == 5901922 &
                                        LONGITUDE == 151.980 &
                                        LATITUDE == 12.277 )

test_buoy %>% ggplot( aes( x=PRES_ADJUSTED , y=TEMP_ADJUSTED ) ) +
              geom_point()+ geom_vline(xintercept = 154.5) +
              xlab("Depth (meters)") + ylab("Temperature (Degrees C)") +
              ggtitle("Example 3 -- Buoy with Deep Thermocline")
```

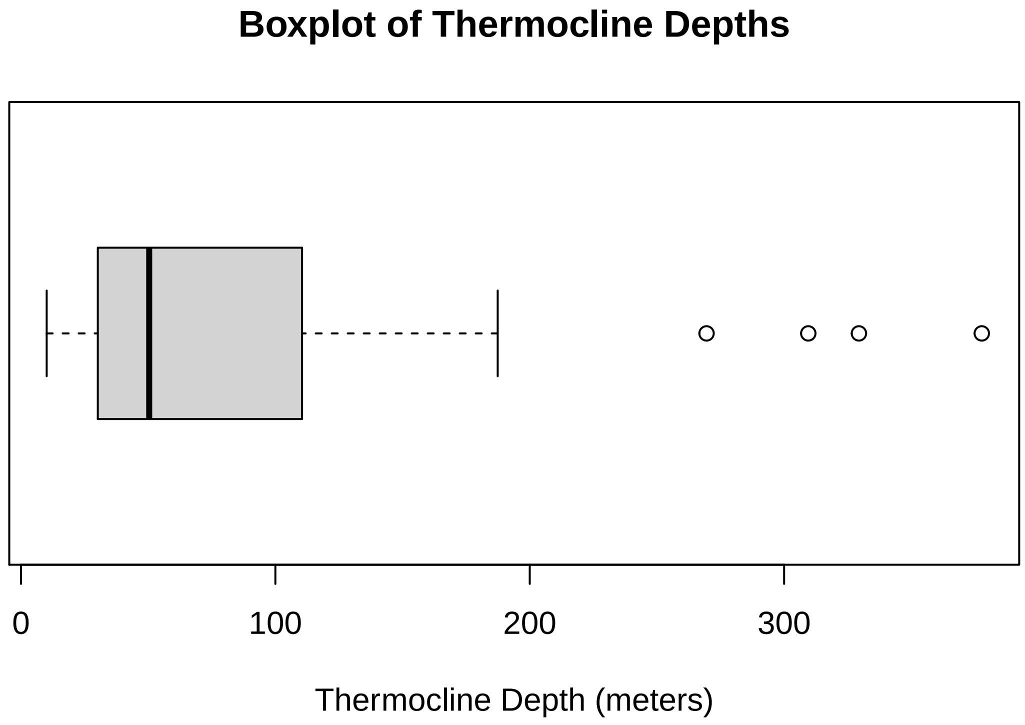


In order to get a better look at the thermocline depth data as a whole, we can look as some summary statistics, a histogram, and a boxplot for it. What jumps out at us the most is that the mean is much higher than the median, signifying that there could be outliers in the data. The histogram also has this pattern, with a few values much larger than the rest. It is highly likely that these values are not real thermoclines because they are taken at points where the latitude is not conducive to the variable representative isotherm calculation. The buoy from example 2 above, for instance, is one of the values on the tail. The boxplot also identifies these points as outliers. If we subset the data to points where our thermocline calculation is valid, we should be able to clean up most, if not all, of these outliers:

```
## Summary Statistics and Histogram of Thermocline Depths
sum_stats <- stat.desc(thermdepth$THERMOCLINE_DEPTH)
sum_stats[4:13]
```

```
##          min          max        range          sum       median         mean
##    10.100000   377.700012   367.600012 88993.017966    50.400000    71.308508
##      SE.mean CI.mean.0.95          var      std.dev
##     1.388836     2.724712  2407.222339    49.063452
```



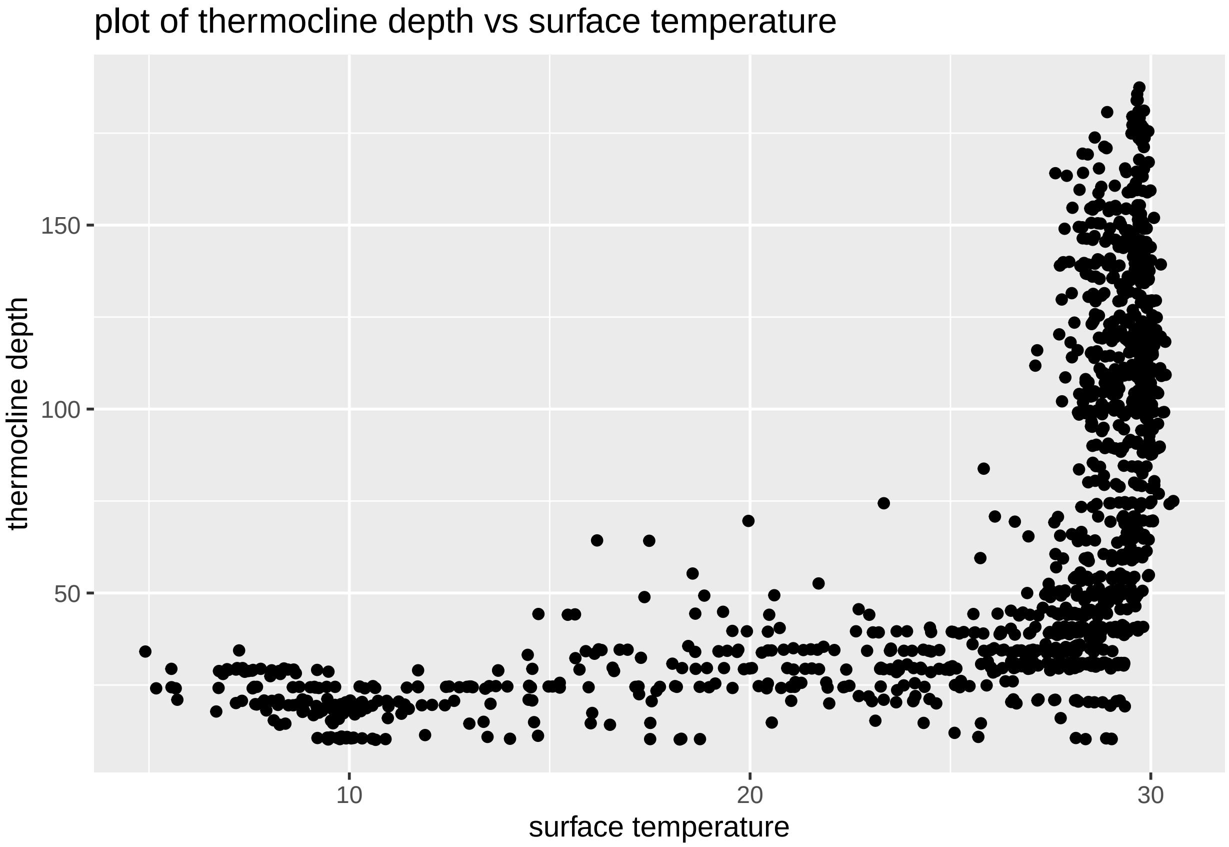


Further Data Wrangling and Observation Criterions

A problem that we are still struggling with is determining the quality of a given observation. While the ARGOs data does contain quality control labeling, from our experience, the accuracy of such labeling leaves much to be desired. It is not too uncommon to find partially incomplete (in the sense that there is an uneven number of readings per obersavtion, e.g. uneven temperature and salinity readings for instance), large gaps within a given observation (as the float ascends, in some rare cases, readings cease to be recorded at certain pressure levels), or simply bizarre geospatial readings (two specitic types of float, the APEX and Navis, have faulty gps units that lead to erroneous latitude and longitude readings). As such, we need to develop a criterion to filter out these aforementioned observations. Thankfully one of the mean field papers lays out a rough outline for observation filtering, which we will implement and apply to our data. While not directly related to the topic of model fitting, our plans for further data cleaning should aid in future model fitting and refinement.

Correlation and basic relationship between thermocline and surface temperature

```
thermdepth %>% filter(THERMOCLINE_DEPTH < 250) %>%
  ggplot(aes(x = SURFACE_TEMP, y = THERMOCLINE_DEPTH))+
  geom_point()+
  labs(x="surface temperature", y= "thermocline depth",
       title= "plot of thermocline depth vs surface temperature")
```



```
cor.test(thermdepth$SURFACE_TEMP,thermdepth$THERMOCLINE_DEPTH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  thermdepth$SURFACE_TEMP and thermdepth$THERMOCLINE_DEPTH
## t = 20.372, df = 1246, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4570450 0.5403603
## sample estimates:
```

```
##       cor
## 0.4998581
```

There is some correlation however the relationship looks far from linear, we can include SURFACE_TEMP as a predictor in our current model for thermoclines. We will see later that this actually does not seem to improve our predictions when entering this variable in the model.

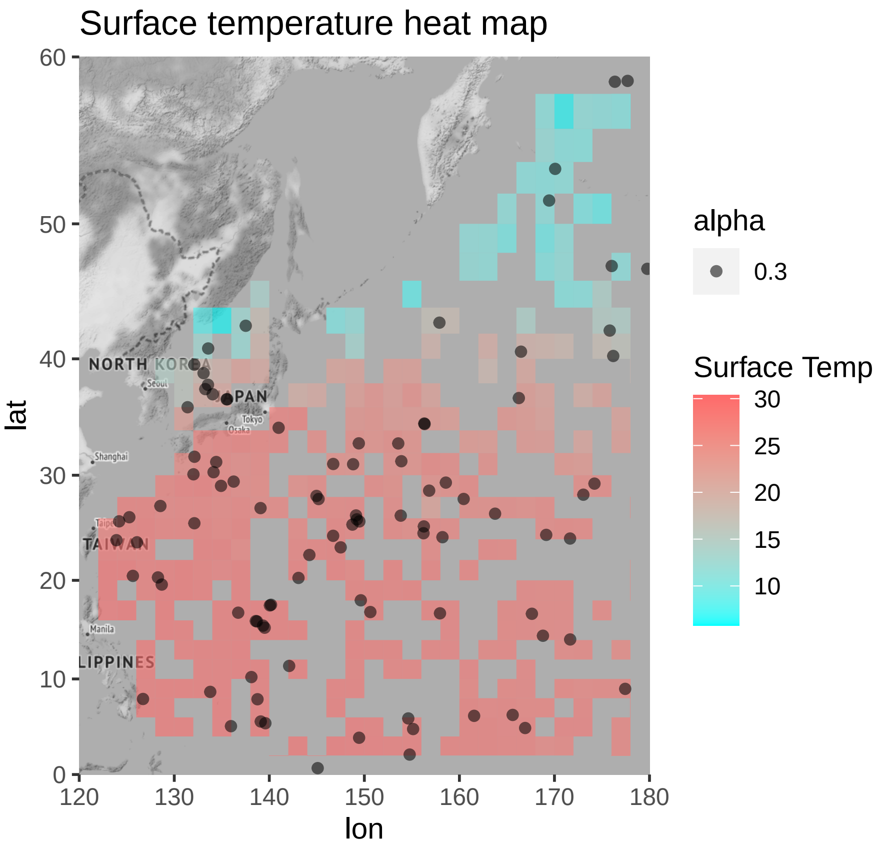
Heat Map

```
world <- get_map(location=c(left = 120, right = 180, bottom = 0, top = 60),
                 source="osm", color="bw", crop=TRUE)
```

```
## Source : http://tile.stamen.com/terrain/4/13/4.png

## Source : http://tile.stamen.com/terrain/4/14/4.png

## Source : http://tile.stamen.com/terrain/4/15/4.png

## Source : http://tile.stamen.com/terrain/4/16/4.png

## Not Found (HTTP 404). Failed to aquire tile /terrain/4/16/4.png.
## Source : http://tile.stamen.com/terrain/4/13/5.png
## Source : http://tile.stamen.com/terrain/4/14/5.png
## Source : http://tile.stamen.com/terrain/4/15/5.png
## Source : http://tile.stamen.com/terrain/4/16/5.png
## Not Found (HTTP 404). Failed to aquire tile /terrain/4/16/5.png.
## Source : http://tile.stamen.com/terrain/4/13/6.png
## Source : http://tile.stamen.com/terrain/4/14/6.png
## Source : http://tile.stamen.com/terrain/4/15/6.png
## Source : http://tile.stamen.com/terrain/4/16/6.png
## Not Found (HTTP 404). Failed to aquire tile /terrain/4/16/6.png.
## Source : http://tile.stamen.com/terrain/4/13/7.png
## Source : http://tile.stamen.com/terrain/4/14/7.png
## Source : http://tile.stamen.com/terrain/4/15/7.png
## Source : http://tile.stamen.com/terrain/4/16/7.png
## Not Found (HTTP 404). Failed to aquire tile /terrain/4/16/7.png.
## Source : http://tile.stamen.com/terrain/4/13/8.png
## Source : http://tile.stamen.com/terrain/4/14/8.png
## Source : http://tile.stamen.com/terrain/4/15/8.png
## Source : http://tile.stamen.com/terrain/4/16/8.png
## Not Found (HTTP 404). Failed to aquire tile /terrain/4/16/8.png.
```

```
heatmapdat <- thermdepth %>% ungroup() %>%  select(LONGITUDE,
                                                   LATITUDE,
                                                   SURFACE_TEMP,
                                                   THERMOCLINE_DEPTH)



ggmap(world) +
    stat_summary_2d(data = heatmapdat, aes(x = LONGITUDE, y = LATITUDE,
        z = SURFACE_TEMP), fun = mean, alpha = 0.6, bins = 30) +
    scale_fill_continuous(name = "Surface Temp", low = "cyan1",
                          high = "indianred1")+
  geom_point(data= heatmapdat[sample(1:nrow(heatmapdat),100,
                                     replace =FALSE),],
             aes(x = LONGITUDE, y = LATITUDE, alpha = 0.3)) +
  labs(title = "Surface temperature heat map")
```

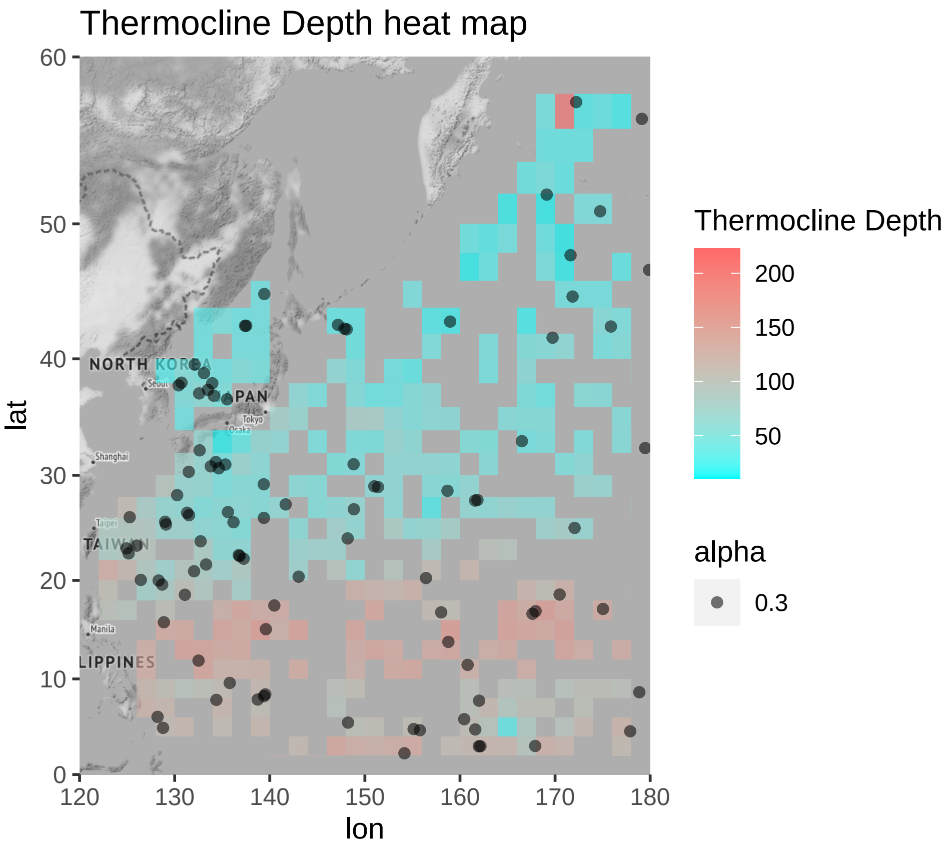
```
## Warning: Removed 27 rows containing non-finite values (stat_summary2d).

## Warning: Removed 3 rows containing missing values (geom_point).
```



```
ggmap(world) +
    stat_summary_2d(data = heatmapdat, aes(x = LONGITUDE, y = LATITUDE,
        z = THERMOCLINE_DEPTH), fun = mean, alpha = 0.6, bins = 30) +
    scale_fill_continuous(name = "Thermocline Depth", low = "cyan1",
                          high = "indianred1")+
  geom_point(data= heatmapdat[sample(1:nrow(heatmapdat),100,replace =FALSE),],
             aes(x = LONGITUDE, y = LATITUDE, alpha = 0.3)) +
  labs(title = "Thermocline Depth heat map")
```

```
## Warning: Removed 27 rows containing non-finite values (stat_summary2d).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



Heat map with typhoon data

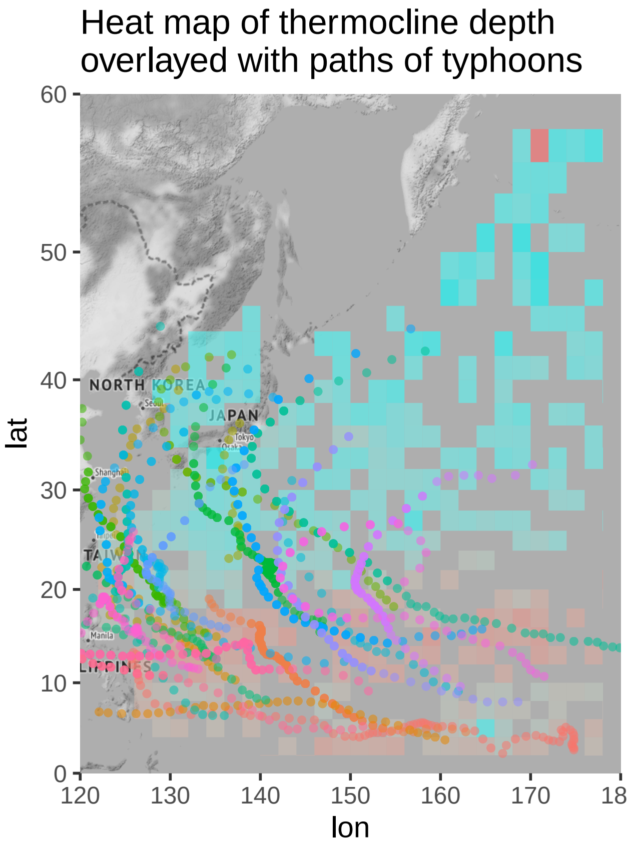
```
# import typhoon data


typh <- readr::read_csv("./typhoon_data/bwp_2019/2019_typh_data.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   yr_cyc_num = col_character(),
##   date_time = col_number(),
##   lat = col_character(),
##   long = col_character(),
##   vmax = col_number(),
##   mslp = col_character(),
##   typh_grade = col_character()
## )
```

```
typh$yr_cyc_num <- gsub(pattern = ",",replacement = "",
                        x = typh$yr_cyc_num)
typh$mslp <- as.numeric(gsub(pattern = ",",replacement = "",
                             x = typh$mslp))

# NEED TO REPLACE with NEGATIVE SIGNS FOR W/S
```

```
typh$lat <- as.numeric(gsub(pattern = "(N,|S,)",replacement = "",
                            x = typh$lat))/10
typh$long <- as.integer(gsub(pattern = "(E,|W,)",replacement = "",
                             x = typh$long))/10


# plot the heat map with the typhoon paths over


ggmap(world) +
    stat_summary_2d(data = heatmapdat, aes(x = LONGITUDE, y = LATITUDE,
        z = THERMOCLINE_DEPTH), fun = mean, alpha = 0.6, bins = 30) +
    scale_fill_continuous(name = "Surface Temp", low = "cyan1",
                          high = "indianred1")+
  geom_point(data= typh,aes(x = long, y = lat, alpha = 0.3,colour=yr_cyc_num,
                            stroke = 0))+theme(legend.position="none") +
  labs(title= "Heat map of thermocline depth\noverlayed with paths of typhoons")
```

```
## Warning: Removed 27 rows containing non-finite values (stat_summary2d).

## Warning: Removed 250 rows containing missing values (geom_point).
```



From here it appears that typhoons do appear to begin in areas with higher thermocline depth.

Models for thermocline depth

Since we only have finite buoy data, and not nearly at every longitude and latitude, we initially model the thermocline depth that can be predicted from a latitude and longitude value. We use a generalized additive model for this. We fit two models here the first is a gam that simply combines all the data from any given month, and so we fit a model for each month. The second is a multilevel model where we model the levels as months where the data was collected therefore we only fit one model and are able to pool our data together. This method can also be used to generally model the thermocline, rather than simply predicting the thermocline depth.

The first will fit a specific model August Thermocline Depth.

```r
# get file names
files <- fs::dir_ls(path = "./august/")[1:3]

# read in all the files

multipleaug <-
  do.call(rbind,
          lapply(files, read.csv))

# remove NA
fulldata <- na.omit(multipleaug)

# select columns we need
colsneed <- fulldata %>% dplyr::select(PLATFORM_NUMBER,
                                          LONGITUDE,
                                          LATITUDE,
                                          TEMP_ADJUSTED,
                                          PRES_ADJUSTED )

colsneed %<>% dplyr::filter(PRES_ADJUSTED > 15)

# get thermocline depth vector and attach it

thermdepthaug <- getThermDepth(colsneed)

# first we fit a model with just latitude and longitude

fitThermDepth1 <- bam(formula = THERMOCLINE_DEPTH ~ s(LONGITUDE,LATITUDE),
                      data = thermdepthaug)

summary(fitThermDepth1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## THERMOCLINE_DEPTH ~ s(LONGITUDE, LATITUDE)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.0736     0.3579   195.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df     F p-value
## s(LONGITUDE,LATITUDE) 28.37  28.97 421.9  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.709   Deviance explained = 71.1%
## fREML =  23345  Scale est. = 641.33    n = 5007
```

```
# next we fit a model with latitude longitude and surface temperature

fitThermDepth2 <- bam(formula = THERMOCLINE_DEPTH ~ s(LONGITUDE,LATITUDE)+
                        s(SURFACE_TEMP, bs = "cs"), data = thermdepthaug)

summary(fitThermDepth2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## THERMOCLINE_DEPTH ~ s(LONGITUDE, LATITUDE) + s(SURFACE_TEMP,
##     bs = "cs")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   70.074      0.338   207.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                          edf Ref.df      F p-value
## s(LONGITUDE,LATITUDE) 28.431  28.98 154.23  <2e-16 ***
## s(SURFACE_TEMP)        6.154   9.00  66.68  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.741   Deviance explained = 74.3%
## fREML =  23077  Scale est. = 572.12    n = 5007
```

```
# check the mean absolute error from predicting from a different year month
# of august

testdata <- read.csv(file = "./august/2013_08_Covariates_Pacific.csv")

# remove NA
testdata <- na.omit(testdata)

# select columns we need
testdata <- testdata %>% dplyr::select(PLATFORM_NUMBER,
                                          LONGITUDE,
                                          LATITUDE,
                                          TEMP_ADJUSTED,
                                          PRES_ADJUSTED )

testdata %<>% dplyr::filter(PRES_ADJUSTED > 15)


# compare the root mean squared error on new data

testdatatherm <- getThermDepth(testdata)



predict1 <- predict.bam(fitThermDepth1,newdata = testdatatherm)
predict2 <- predict.bam(fitThermDepth2, newdata =testdatatherm)

sqrt(mean((predict1 - testdatatherm$THERMOCLINE_DEPTH)^2))
```

```
## [1] 20.06622
```

```
sqrt(mean((predict2 - testdatatherm$THERMOCLINE_DEPTH)^2))
```

```
## [1] 19.77932
```

The second will be a model of multiple months combined together in a multilevel sense.

```
# import data from multiple months of same year for instance.

datjul <- read.csv(file ="./mixmonths/2010_07_Covariates_Pacific.csv")
dataug <-  read.csv(file ="./mixmonths/2010_08_Covariates_Pacific.csv")
datsep <-  read.csv(file ="./mixmonths/2010_09_Covariates_Pacific.csv")


# label with the Month

datjul <- cbind(datjul,month = "jul")
dataug <- cbind(dataug, month = "aug")
datsep <- cbind(datsep, month = "sep")

dattog <- rbind(datjul,dataug,datsep)

dattog$month <- as.factor(dattog$month)

dattog <- na.omit(dattog)

thermdepthmlm <- getThermDepth(dattog)
```

```
## `summarise()` has grouped output by 'PLATFORM_NUMBER', 'LONGITUDE'. You can override using the `.grou
## `summarise()` has grouped output by 'PLATFORM_NUMBER', 'LONGITUDE'. You can override using the `.grou

## Joining, by = c("PLATFORM_NUMBER", "LONGITUDE", "LATITUDE")
```

```
monthsonly <- dattog %>% select(PLATFORM_NUMBER,LONGITUDE, LATITUDE, month)

thermdepthmlmwm <- distinct(inner_join(thermdepthmlm,monthsonly, by =
                                c("PLATFORM_NUMBER","LONGITUDE","LATITUDE")))
```

```
fitThermDepthmlm1 <- bam(formula = THERMOCLINE_DEPTH ~ s(LONGITUDE,LATITUDE)+
                          s(LATITUDE, month, bs = "re")+s(month, bs='re'),
                         data = thermdepthmlmwm)

summary(fitThermDepthmlm1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## THERMOCLINE_DEPTH ~ s(LONGITUDE, LATITUDE) + s(LATITUDE, month,
##     bs = "re") + s(month, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    72.04       4.67   15.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                          edf Ref.df      F  p-value
## s(LONGITUDE,LATITUDE) 28.685  28.99  149.5  < 2e-16 ***
## s(LATITUDE,month)      1.931   2.00 4132.2  < 2e-16 ***
## s(month)               1.865   2.00 3016.2 2.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.768   Deviance explained =   77%
## fREML =  17494  Scale est. = 545.13    n = 3812
```

```
fitThermDepthmlm2 <- bam(formula = THERMOCLINE_DEPTH ~ s(LONGITUDE,LATITUDE)+
                          s(LATITUDE, month, bs = "re")+s(month, bs='re')+
                        s(SURFACE_TEMP, month, bs = "re"),
                        data = thermdepthmlmwm)

summary(fitThermDepthmlm2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## THERMOCLINE_DEPTH ~ s(LONGITUDE, LATITUDE) + s(LATITUDE, month,
##     bs = "re") + s(month, bs = "re") + s(SURFACE_TEMP, month,
##     bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   73.596      3.571   20.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
```

```
##                              edf Ref.df      F  p-value
## s(LONGITUDE,LATITUDE) 2.869e+01  28.99  157.0  < 2e-16 ***
## s(LATITUDE,month)     1.938e+00   2.00 1333.8  < 2e-16 ***
## s(month)              3.096e-04   2.00    0.0    0.848
## s(SURFACE_TEMP,month) 2.036e+00   3.00  599.5 7.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.769   Deviance explained = 77.1%
## fREML =  17492  Scale est. = 544.57    n = 3812
```

```
# test set results

testDataThermMlm <- cbind(testdatatherm,month = "aug")
testDataThermMlm$month <- as.factor(testDataThermMlm$month)


predictM1 <- predict.bam(fitThermDepthmlm1,newdata = testDataThermMlm)
predictM2 <- predict.bam(fitThermDepthmlm2, newdata =testDataThermMlm)

sqrt(mean((predictM1 - testDataThermMlm$THERMOCLINE_DEPTH)^2))
```

```
## [1] 20.4772
```

```
sqrt(mean((predictM2 - testDataThermMlm$THERMOCLINE_DEPTH)^2))
```

```
## [1] 20.47066
```

The multilevel models appear to perform better, and the models with surface temperature also appear to perform marginally better, since they are the simpler model we would most likely remove using the temperature as a potential predictor. The root mean squared error is around 20 which may be concerning we have a lot more data and potentially more predictors to add to improve the model.

Second Model to related to typhoons

The second model takes in output from the first model (the thermocline depth) at any specified latitude and longitude and demonstrates the relationship between this and typhoon occurrence in an area. To do this we plan on taking an average of predicted thermocline depth in some region and then modeling that with respect to the number of typhoons that occured in that region.