Thermocline Model

Nick Strayer

March 8, 2015

Define the model.

In this case we will be using a logistic curve to model the rise and fall of the thermocline

```r
thermoclineDist = function(t){
  maxThermDepth = 40
  if (t < (2190*2)) { #Winter + Spring
    n1 = .003
    n2 = -5
    x = t
    dist = (maxThermDepth)/(1 + exp(-(n2 + n1*x)))
  } else { #Summer+ fall
    n1 = .005
    n2 = -8
    x = t - (2190*2)
    dist = (-maxThermDepth)/(1 + exp(-(n2 + n1*x))) + maxThermDepth
  }
  return(dist)
}
```

Generate data

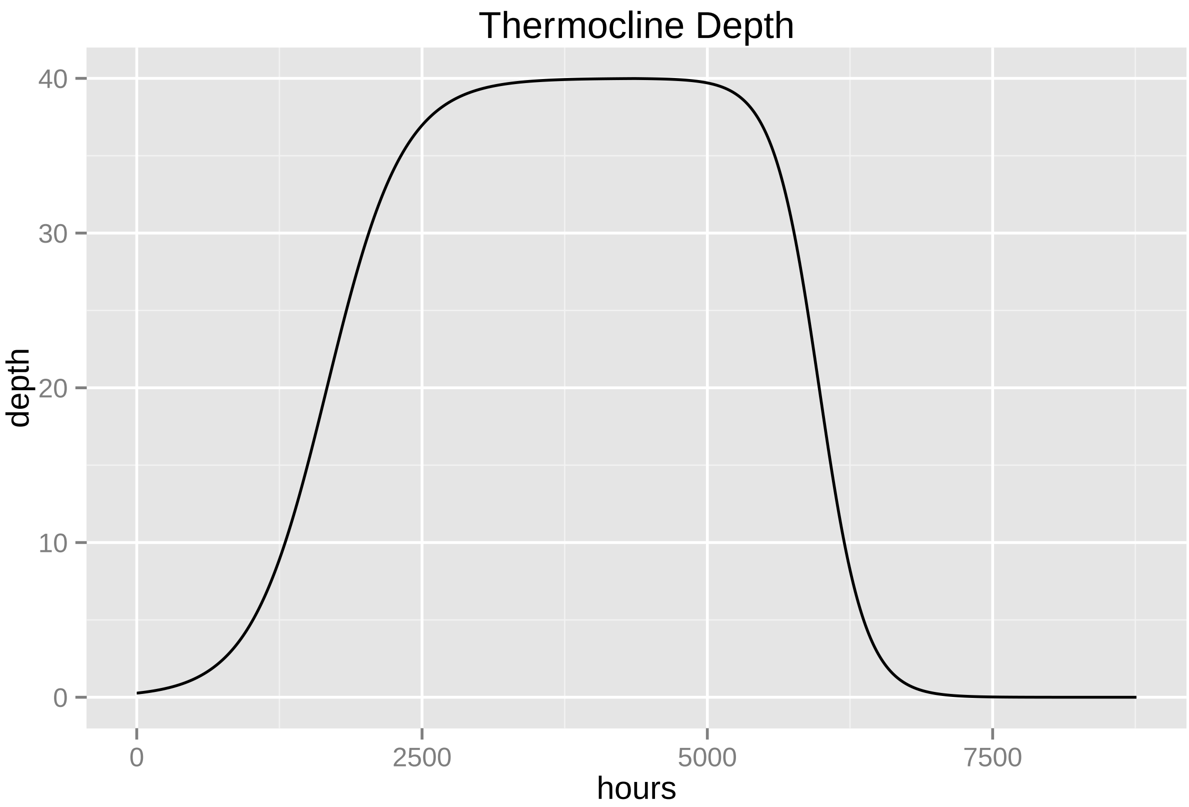
Now we initialize an hour list and run the thermocline function over it to get our data.

```r
hours  = 0:(365*24)
dist = NULL

for (t in hours){ dist = c(dist, thermoclineDist(t)) }
```

Plot and save

Now let's check out the results and save to a csv file.

```r
data = cbind(hours, dist) # Wrap the data.

thermocline = as.data.frame(data)
ggplot(thermocline, aes(x = hours, y = dist)) + geom_line() +
  labs(title = "Thermocline Depth", y = "depth")
```



```
ggsave(filename = "../paperMaterials/figures/pres_thermoclineDepth.pdf", width = 6, height = 2.5)
#write.csv(data, "/Users/Nick/mysisModeling/data/Depth_Thermocline_Hour.csv", row.names=FALSE)
```