TEMP temperature comparison

Eddie Wu

2023-07-27

Introduction

This .Rmd file is to show the results of water temperature comparison between the global model prediction and the tributary field observation for 11 Great Lake locations. (missing Saginaw data)

For all the graphs below, the black line is the mean weekly average temperature, the blue lines are the 95% confidence interval, the red dots are field observations by week.

```
library(ggplot2)
library(dplyr)
library(knitr)
```

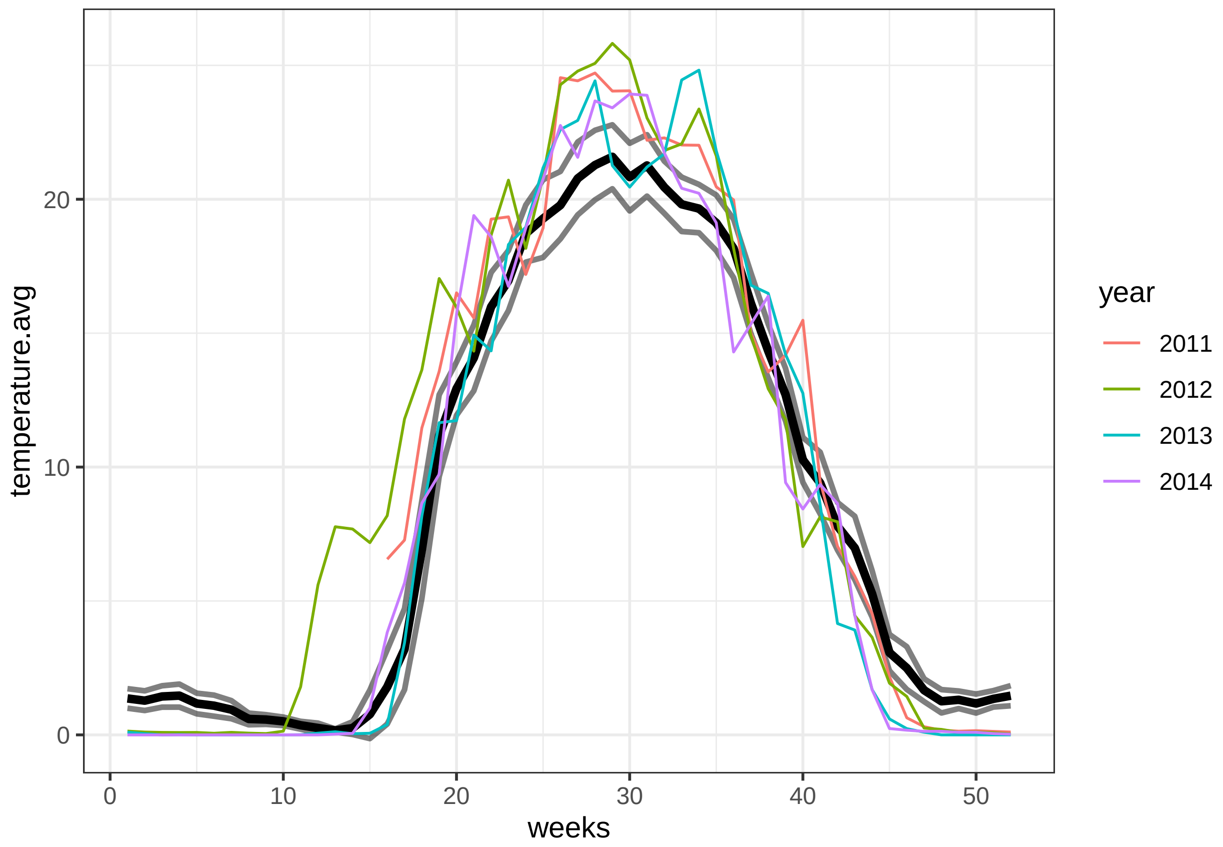
```
WT <- read.csv("water temperature clean/water_temperature_final_clean.csv",
               stringsAsFactors = TRUE)
WT$year <- as.factor(WT$year)
```

St. Louis River

```
# observed values
st.louis.field <- WT %>% filter(river == "stlouis")

# predicted values
st.louis.pred <- read.csv("water temperature clean/st_louis_model.csv")

## Plotting
ggplot(st.louis.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(st.louis.field, mapping = aes(x = weeks, y = temp, color = year))+
  scale_colour_hue()+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)
# Remove the observations from non-complete years
st.louis.field.c <- st.louis.field[st.louis.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(st.louis.field.c$year)
unique_year
```

```
## [1] 2012 2013 2014
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(st.louis.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - st.louis.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

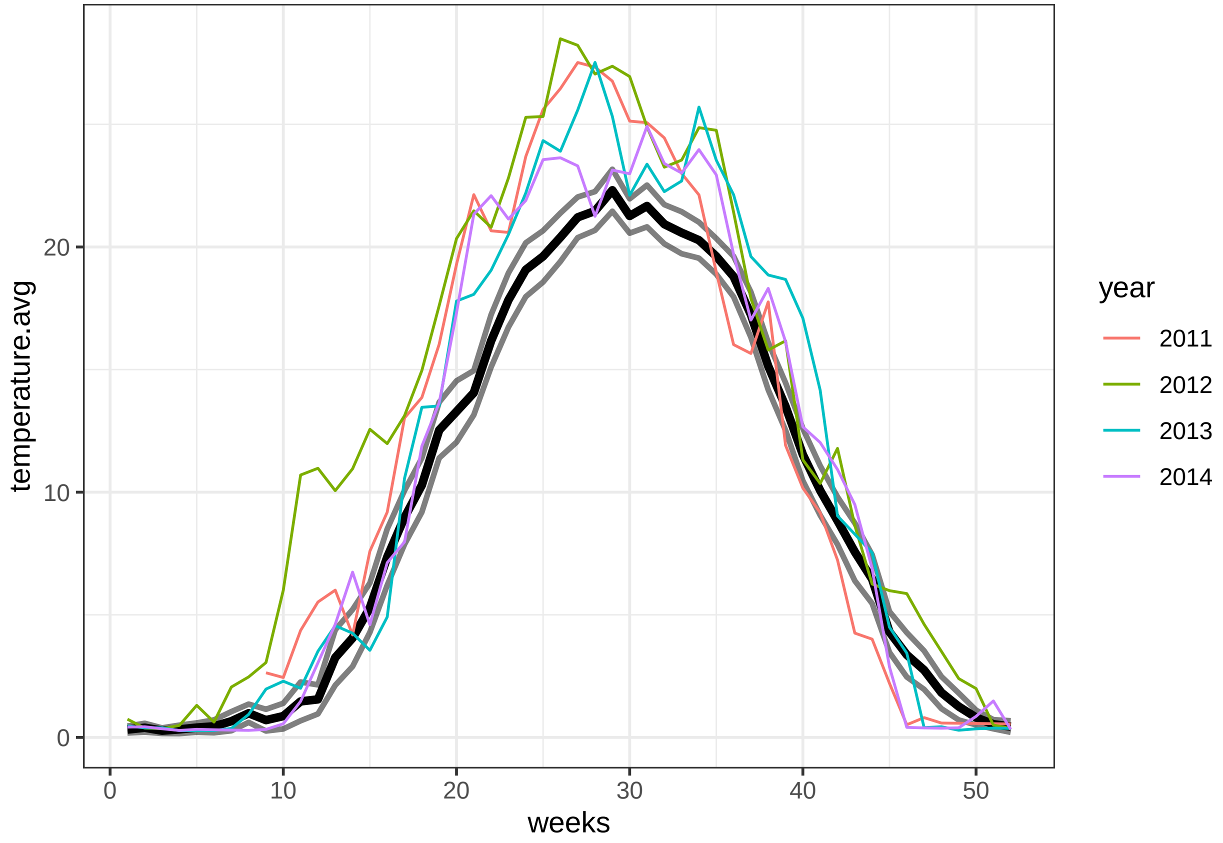
```
##     2012     2013     2014
## 3.333021 1.862993 1.898446
```

Fox River

```
# observed values
fox.field <- WT %>% filter(river == "fox")

# predicted values
fox.pred <- read.csv("water temperature clean/fox_model.csv")

## Plotting
ggplot(fox.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(fox.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)
# Remove the observations from non-complete years
fox.field.c <- fox.field[fox.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(fox.field.c$year)
unique_year
```

```
## [1] 2012 2013 2014
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```r
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(fox.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - fox.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

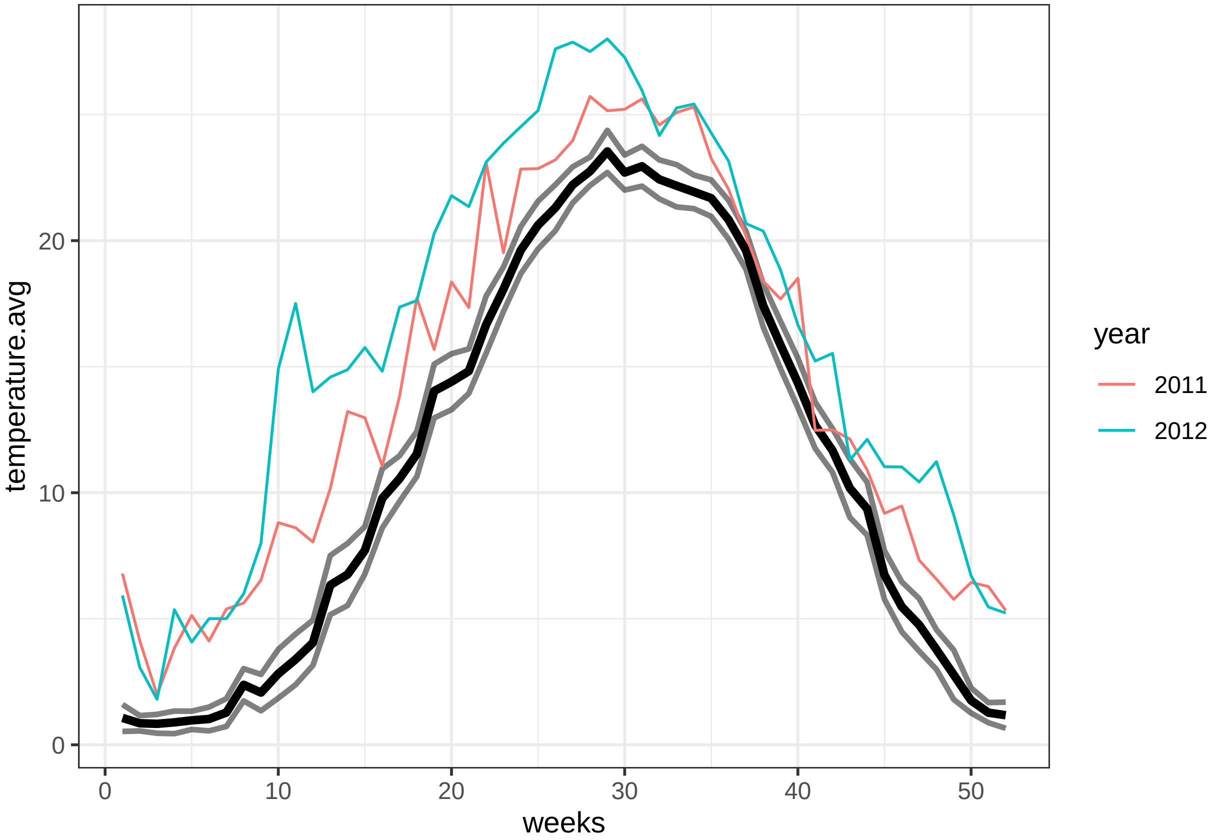
```
##     2012     2013     2014
## 4.352852 2.643798 2.293099
```

Portage-Burns Waterways

```r
# observed values
pb.field <- WT %>% filter(river == "portage")

# predicted values
pb.pred <- read.csv("water temperature clean/pb_model.csv")

## Plotting
ggplot(pb.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(pb.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)
#' All the years are complete for pb

# Get the unique levels of the "year" variable
unique_year <- unique(pb.field$year)
unique_year
```

```
## [1] 2011 2012
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(pb.field, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - pb.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

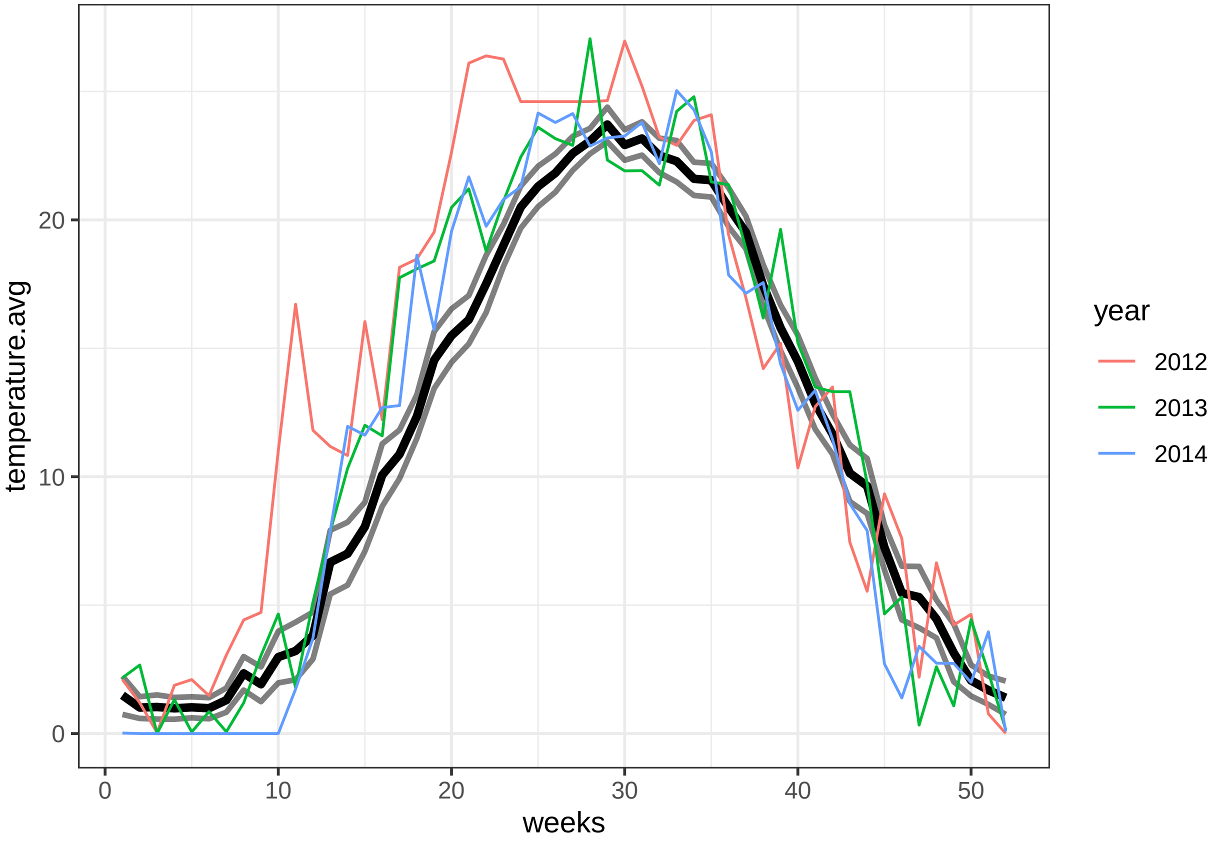
```
##     2011     2012
## 3.462018 5.574531
```

Vermilion River

```
# observed values
vermilion.field <- WT %>% filter(river == "vermilion")

# predicted values
vermilion.pred <- read.csv("water temperature clean/vermilion_model.csv")

## Plotting
ggplot(vermilion.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(vermilion.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)
#' All years complete for vermilion

# Get the unique levels of the "year" variable
unique_year <- unique(vermilion.field$year)
unique_year
```

```
## [1] 2012 2013 2014
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```r
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(vermilion.field, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - vermilion.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

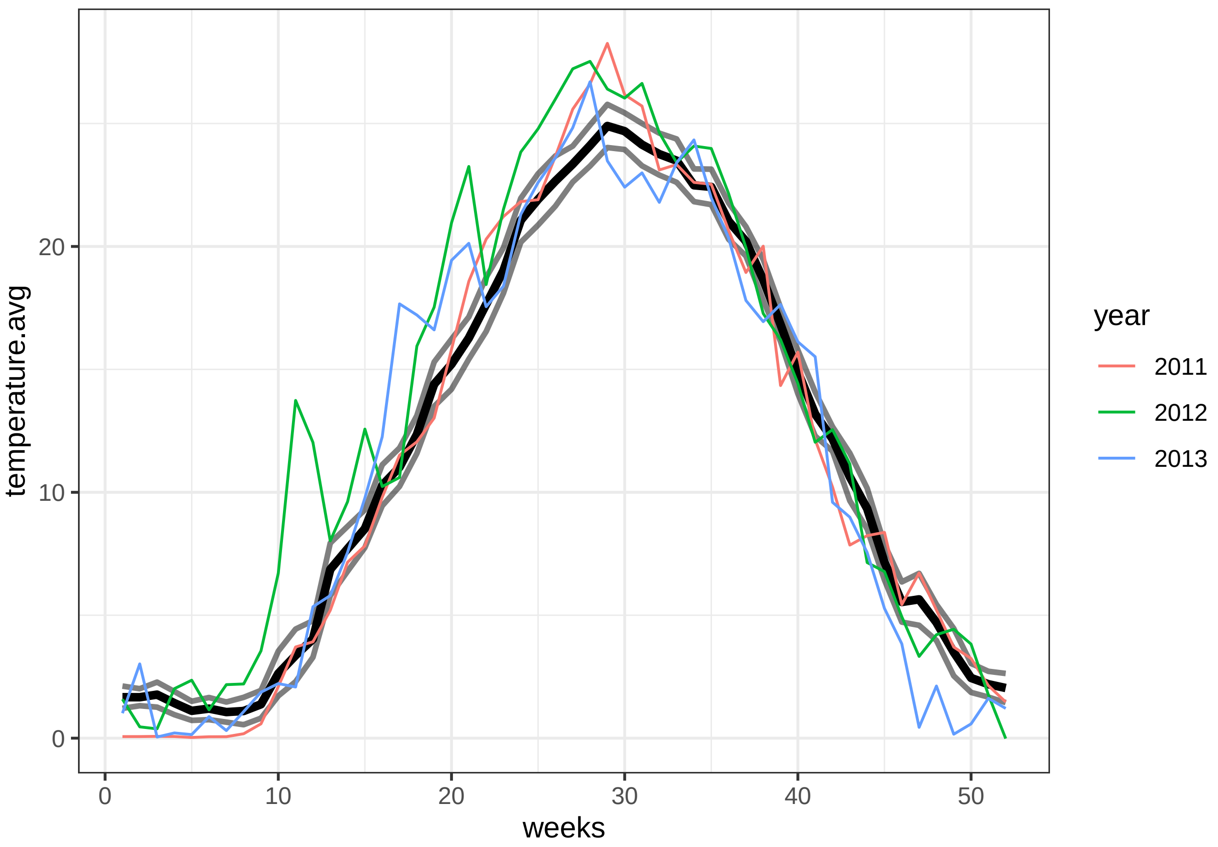
```
##     2012     2013     2014
## 4.392703 2.487492 2.323831
```

Genesee River

```r
# observed values
genesee.field <- WT %>% filter(river == "genesee")

# predicted values
genesee.pred <- read.csv("water temperature clean/genesee_model.csv")

## Plotting
ggplot(genesee.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(genesee.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)
#' All years complete for genesee

# Get the unique levels of the "year" variable
unique_year <- unique(genesee.field$year)
unique_year
```

```
## [1] 2011 2012 2013
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(genesee.field, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - genesee.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

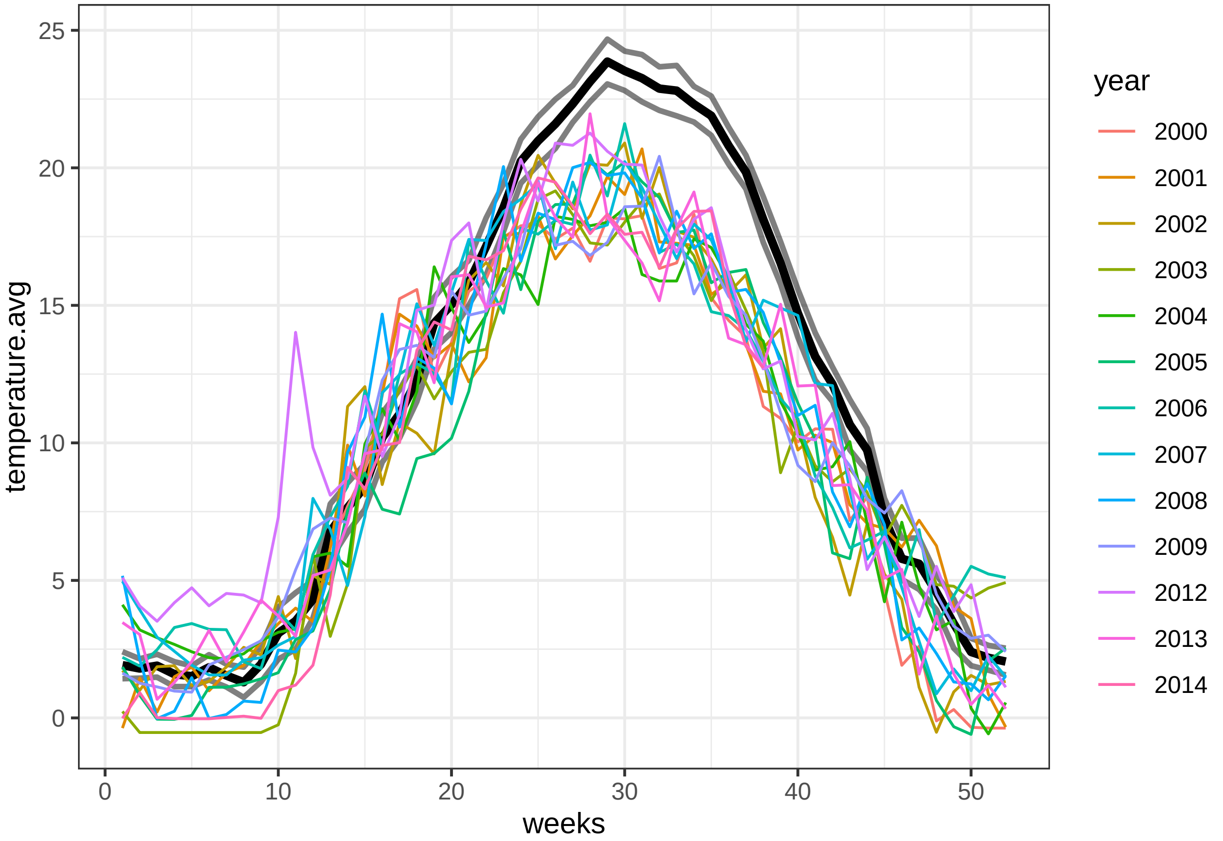
```
##     2011     2012     2013
## 1.359875 2.865459 2.129912
```

Big Creek River

```
# observed values
big.creek.field <- WT %>% filter(river == "big creek")

# predicted values
big.creek.pred <- read.csv("water temperature clean/bigcreek_model.csv")


## Plotting
ggplot(big.creek.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(big.creek.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)

# Remove the observations from non-complete years
big.creek.field.c <- big.creek.field[big.creek.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(big.creek.field.c$year)
unique_year
```

```
##  [1] 2001 2002 2003 2004 2005 2006 2007 2008 2009 2012 2013
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(big.creek.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - big.creek.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

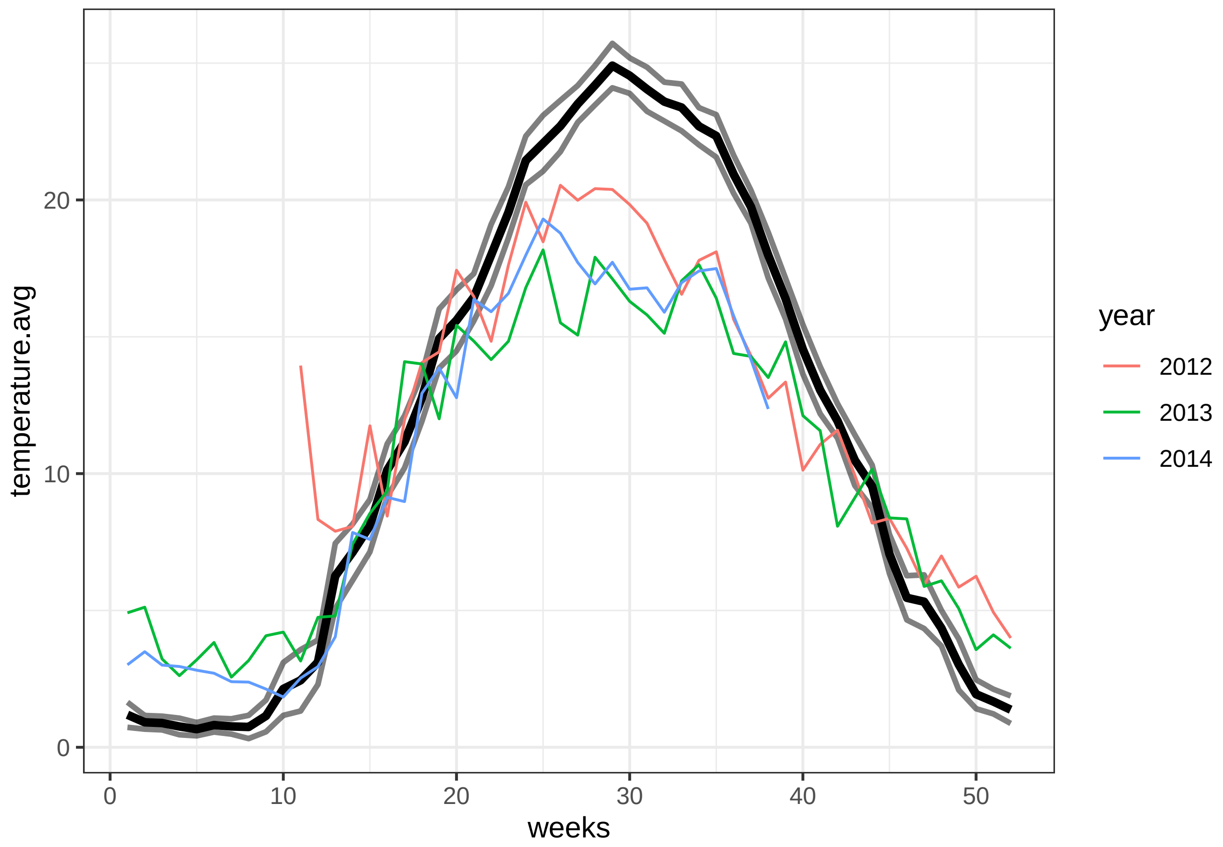
```
##     2001     2002     2003     2004     2005     2006     2007     2008
## 3.175168 3.074429 3.445788 3.357855 3.149056 3.277303 2.788316 2.852734
##     2009     2012     2013
## 3.155650 3.250036 3.183567
```

Big Otter River

```
# observed values
big.otter.field <- WT %>% filter(river == "big otter")

# predicted values
big.otter.pred <- read.csv("water temperature clean/bigotter_model.csv")


## Plotting
ggplot(big.otter.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(big.otter.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)

# Remove the observations from non-complete years
big.otter.field.c <- big.otter.field[big.otter.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(big.otter.field.c$year)
unique_year
```

```
## [1] 2013
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(big.otter.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - big.otter.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

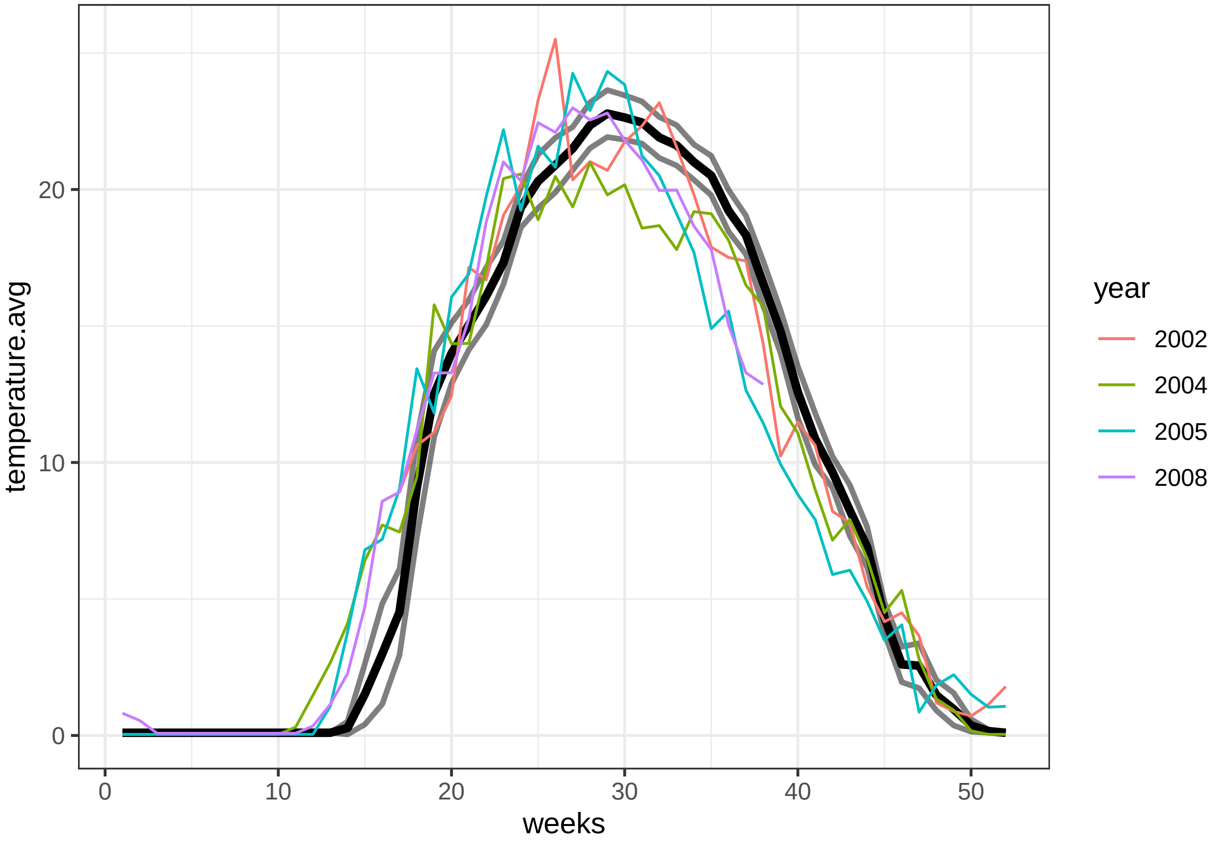
```
##     2013
## 4.072743
```

Still River

```
# observed values
still.field <- WT %>% filter(river == "still")

# predicted values
still.pred <- read.csv("water temperature clean/still_model.csv")


## Plotting
ggplot(still.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(still.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)

# Remove the observations from non-complete years
still.field.c <- still.field[still.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(still.field.c$year)
unique_year
```

```
## [1] 2004 2005
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(still.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - still.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

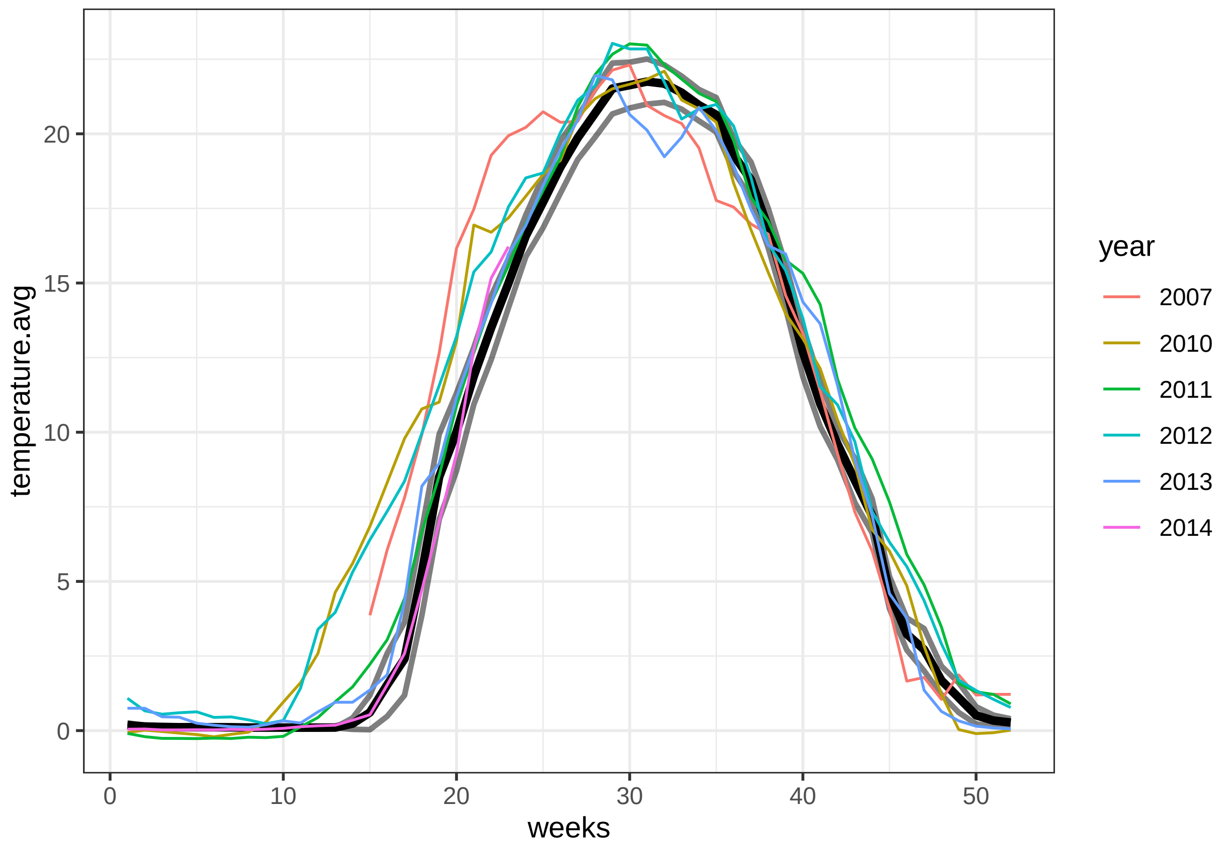
```
##     2004     2005
## 1.974017 2.633387
```

Mississagi River

```
# observed values
mississagi.field <- WT %>% filter(river == "mississagi")

# predicted values
mississagi.pred <- read.csv("water temperature clean/mississagi_model.csv")


## Plotting
ggplot(mississagi.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(mississagi.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```r
## Calculate the Root Mean Square Error (RMSE)

# Remove the observations from non-complete years
mississagi.field.c <- mississagi.field[mississagi.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(mississagi.field.c$year)
unique_year
```

```
## [1] 2010 2011 2012 2013
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```r
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(mississagi.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - mississagi.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

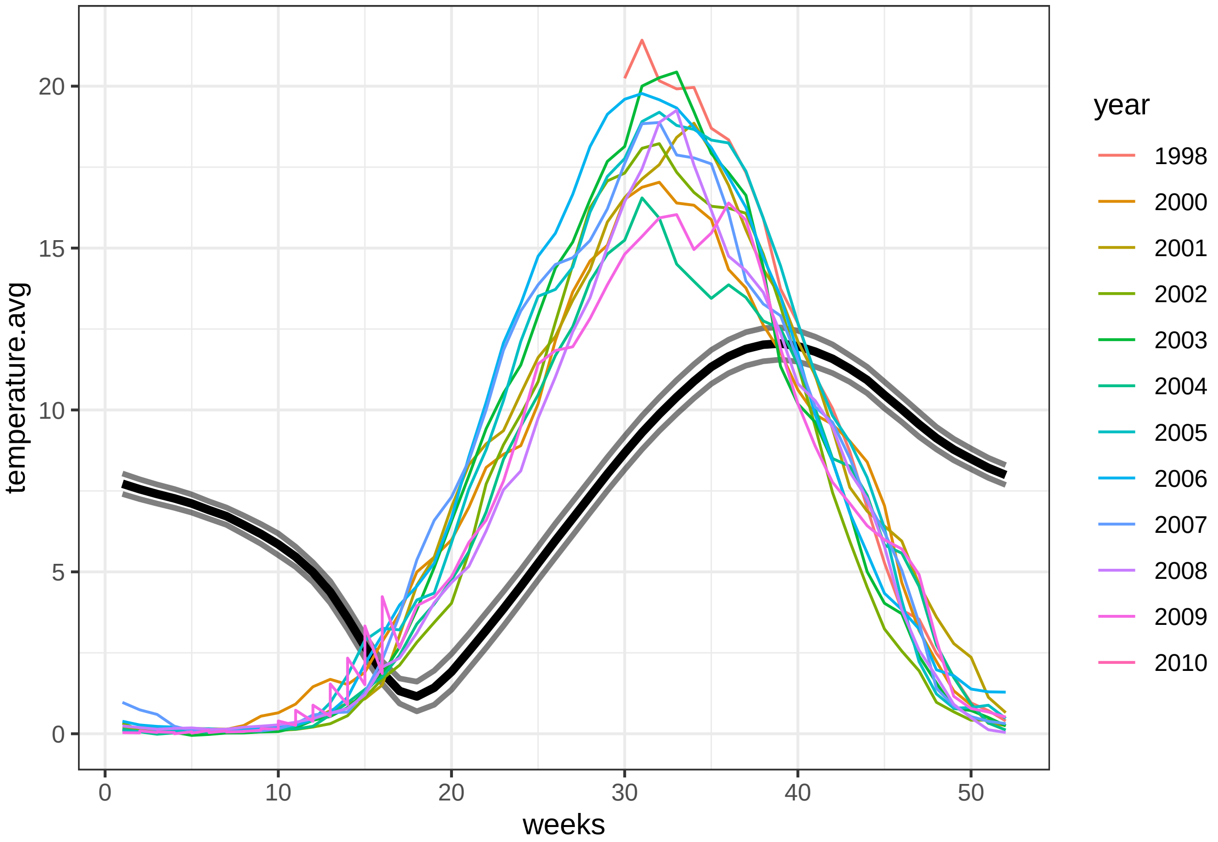
```
##     2010     2011     2012     2013
## 2.418733 1.261284 2.234349 1.030389
```

Nipigon River

```
# observed values
nipigon.field <- WT %>% filter(river == "nipigon")

# predicted values
nipigon.pred <- read.csv("water temperature clean/nipigon_model.csv")


## Plotting
ggplot(nipigon.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(nipigon.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)

# Remove the observations from non-complete years
nipigon.field.c <- nipigon.field[nipigon.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(nipigon.field.c$year)
unique_year
```

```
##  [1] 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```r
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(nipigon.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - nipigon.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

```
##     2000     2001     2002     2003     2004     2005     2006     2007
## 5.379168 5.644761 6.032138 6.541750 5.117281 6.243819 6.679698 6.172821
##     2008     2009
## 5.503978 5.206301
```

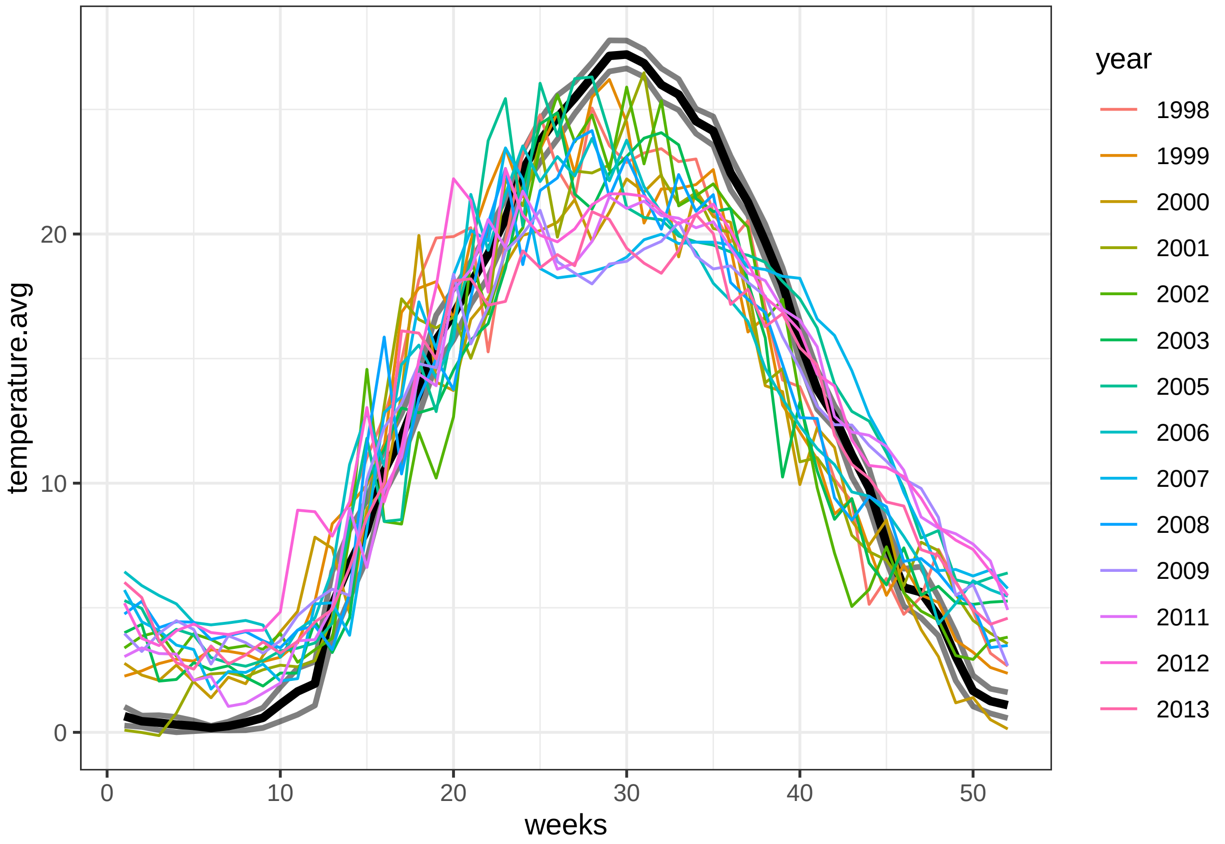
Humber River

```r
# observed values
humber.field <- WT %>% filter(river == "humber")
str(humber.field)
```

```
## 'data.frame':    715 obs. of  5 variables:
##  $ river   : Factor w/ 11 levels "big creek","big otter",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ year    : Factor w/ 17 levels "1998","1999",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ weeks   : int  14 15 16 17 18 19 20 21 22 23 ...
##  $ temp    : num  9.08 11.02 12.7 14.93 18.16 ...
##  $ complete: int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
# predicted values
humber.pred <- read.csv("water temperature clean/humber_model.csv")


## Plotting
ggplot(humber.pred, aes(x = weeks, y = temperature.avg))+
  geom_line(size = 1.5)+
  geom_line(aes(x = weeks, y = lower.CI), size = 1, alpha = 0.5)+
  geom_line(aes(x = weeks, y = upper.CI), size = 1, alpha = 0.5)+
  geom_line(humber.field, mapping = aes(x = weeks, y = temp, color = year))+
  theme_bw()
```



```
## Calculate the Root Mean Square Error (RMSE)

# Remove the observations from non-complete years
humber.field.c <- humber.field[humber.field$complete == 1,]

# Get the unique levels of the "year" variable
unique_year <- unique(humber.field.c$year)
unique_year
```

```
##  [1] 1999 2000 2001 2002 2003 2005 2006 2007 2008 2009 2011 2012 2013
## 17 Levels: 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ... 2014
```

```
# Create a vector list to store the results
rmse_list <- vector("numeric", 0)

# Loop through each level and calculate the RMSE
for (year.number in unique_year) {
  sub <- subset(humber.field.c, year == year.number) # create subsets
  value <- sqrt(mean((sub$temp - humber.pred$temperature.avg)^2))
  rmse_list[year.number] <- value
}

rmse_list
```

```
##     1999     2000     2001     2002     2003     2005     2006     2007
## 2.735022 3.273304 2.689886 2.912912 2.832167 3.305571 3.606752 3.957558
##     2008     2009     2011     2012     2013
## 3.120743 3.877193 3.339086 3.856890 3.723763
```