# R code for "Modelling mosquito population in Alachua County"

## Webscraping temperature

```
> getWeatherForDate("GNV", "2014-04-14")
> ts <- as.data.frame(
+   getDailyMinMaxTemp(station_id = "GNV",
+                      start_date = "2008-03-01",
+                      end_date = "2013-12-01")
+ )
> ts$date <- as.Date(substr(ts$TimeMin, 1, 10),
+                      format="%Y-%m-%d")
> write.csv(ts, "E:/workingdirectory/mosquito/rainAndTemp/a_tsTemp2008-2013.csv")
```

## Webscraping precipitation

```
> #THE FOLLOWING SCRIPT TAKES RAINFALL DATA FROM WUNDERGROUND
> #THE NAs SHOW UP BECAUSE OF THIS MYSTERIOUS T
> library(pingr)
> #Establish start and end dates
> startDate <- "2008-03-01"
> nDays <- 2200
> #Set up URL
> linkPart1 <- "http://www.wunderground.com/history/airport/KGNV/"
> linkPart3 <-  "/DailyHistory.html"
> ts <- as.data.frame(c(as.Date(startDate, format="%Y-%m-%d"),
+                       as.Date(startDate, format="%Y-%m-%d")+1:(nDays-1)))
> colnames(ts) <- "date"
> ts$dateRec <- format(ts$date, format="%Y/%m/%d")
> #RAINFALL DATA
> ts$pui <- NA
> for (i in 1:nrow(ts)){
+   linkPart2 <- ts$dateRec[i]
+   link <- paste0(linkPart1, linkPart2, linkPart3)
+   webPage <- readLines(link)
+   webPage <- webPage[grepl("  <span class=\"nobr\"><span class=\"b\">", webPage) &
+                      grepl("</span> in</span>", webPage)][1]
+   ts$pui[i] <- as.numeric(gsub(paste0("  <span class=\"nobr\"><span class=\"b\">",
+                                       "|", "</span> in</span>"),
+                                "",
+                                webPage))
+ }
> ts$rain <- ts$pui
> ts$pui <- NULL
> ts$rain[is.na(ts$rain)] <- 0
> #write.csv(ts, "C:/Users/BrewJR/Desktop/mosquito/rainFall2013/rain2008-2013.csv")
>
>
> ping(8)
> ######NOW FOR TEMPERATURE DATA
> #TEMPERATURE DATA (TAKES FOREVER)
> ts$temp <- NA
> for (i in 1:nrow(ts)){
+   linkPart2 <- ts$dateRec[i]
+   link <- paste0(linkPart1, linkPart2, linkPart3)
+   webPage <- readLines(link)
+   webPage <- webPage[grepl("  <span class=\"nobr\"><span class=\"b\">", webPage) &
+                        grepl("</span> ", webPage)][1]
+   ts$temp <- as.numeric(gsub(paste0("  <span class=\"nobr\"><span class=\"b\">",
+                                     "|", "</span> &deg;F</span>"), "", webPage))
```

```
+ }
> ts$heat[is.na(ts$heat)] <- 0
> write.csv(ts, "E:/workingdirectory/mosquito/rainAndTemp/a_rain2008-2013.csv")
> ping(8)
> ##########################################
> #BEGIN THE RAIN ANALYSIS MULTIYEAR FILE (RUNNING TOGETHER OVER WEEKEND)
> ##########################################
>
> library(pingr)
> ########################################
> #READ IN THE RAIN TIME SERIES DATA [CREATED FROM WEBSCRAPING WUNDERGROUND]
> #READ IN MOSQUITO TIME SERIES DATA [CREATED FROM 2013 MOSQ SEASON SURVEIL]
> ########################################
> tsRain <- read.csv("C:/Users/BrewJR/Desktop/mosquito/rainFall2013/rain2008-2013.csv")
> tsMosq <- read.csv("C:/Users/BrewJR/Desktop/mosquito/simple.csv")
> tsTemp <- read.csv("C:/Users/BrewJR/Desktop/mosquito/rainFall2013/rainAndTemp2008-2013.csv
> ########################################
> #CONVERT DATES INTO R DATE OBJECTS
> ########################################
> tsRain$date <- as.Date(tsRain$date, format="%Y-%m-%d")
> tsMosq$date <- as.Date(tsMosq$date, format="%m/%d/%Y")
> tsTemp$date <- as.Date(tsTemp$date, format="%m/%d/%Y")
> ########################################
> #ADD MOSQUITOES (TOTAL AND VECTOR) TO RAINFALL
> ########################################
> tsRain$total <- NA
> for (i in tsMosq$date){
+   tsRain$total[which(tsRain$date == i)] <-
+     tsMosq$total[which(tsMosq$date == i)]
+ }
> tsRain$vector <- NA
> for (i in tsMosq$date){
+   tsRain$vector[which(tsRain$date == i)] <-
+     tsMosq$vector[which(tsMosq$date == i)]
+ }
> ########################################
> #ADD DAILY MEDIAN TEMPERATURE TO tsRain
> ########################################
> tsRain$temp <- NA
> for (i in tsRain$date){
+   tsRain$temp[which(tsRain$date == i)] <-
+     tsTemp$temp[which(tsTemp$date == i)]
+ }
> ########################################
>
> ########################################
```

```
> #ADD RAINFALL RANGES
> ########################################
>
> #Make columns for a range of 5-20 days old, plus 5-20 days older than that
> for (j in 5:20){
+   for (k in 5:20){
+     tsRain[,paste0("rain", j, ".", j+k)] <- NA
+   }
+ }
> #Add rainfall for each of the columns
> for (j in colnames(tsRain)[grepl("rain", colnames(tsRain))][-1]){
+   for (i in 30:nrow(tsRain)){
+     tsRain[i,j] <-
+       sum(tsRain$rain[which(tsRain$date <=
+                               tsRain$date[i-min(as.numeric(unlist(strsplit(gsub("rain", "'
+                               tsRain$date >=
+                               tsRain$date[i-max(as.numeric(unlist(strsplit(gsub("rain", "'
+   }
+ }
> ########################################
> #ADD TEMPERATURE RANGES
> ########################################
>
> #Make columns for a range of 5-20 days old, plus 5-20 days older than that
> for (j in 5:20){
+   for (k in 5:20){
+     tsRain[,paste0("temp", j, ".", j+k)] <- NA
+   }
+ }
> #Add rainfall for each of the columns
> for (j in colnames(tsRain)[grepl("temp", colnames(tsRain))][-1]){
+   for (i in 30:nrow(tsRain)){
+     tsRain[i,j] <-
+       sum(tsRain$temp[which(tsRain$date <=
+                               tsRain$date[i-min(as.numeric(unlist(strsplit(gsub("temp", "'
+                               tsRain$date >=
+                               tsRain$date[i-max(as.numeric(unlist(strsplit(gsub("temp", "'
+   }
+ }
> ########################################
> #Test the r-squared for each column (RAIN ONLY)
> ########################################
> #Create a dataframe with the R-squared and correlation coefficient for each range
> pred <- as.data.frame(colnames(tsRain)[grepl("rain", colnames(tsRain))][-1])
> colnames(pred) <- "range"
> for (i in pred$range){
```

```
+   mylm <- summary(lm(tsRain[,"total"] ~ tsRain[,i]))
+   mycor <- cor(tsRain[,i], tsRain[,"total"], use="complete.obs")
+
+   pred$r.squared[which(pred$range == i)] <- mylm$r.squared
+   pred$cor[which(pred$range == i)] <- mycor
+
+ }
> pred <- pred[order(pred$r.squared),]
> ########################################
> #Select best predicition model
> ########################################
> best <- as.character(pred$range[which(pred$r.squared == max(pred$r.squared))])
> ########################################
> #Save it for use in Sweave (RDATA FILE)
> ########################################
> #save.image("C:/Users/BrewJR/Desktop/mosquito/rainFall2013/rainAndHeat2008-2013.RData")
> ########################################
> ping(8)
>
```

## Merging rain/precip data and calculating date range values

```
> #THE FOLLOWING TAKES ABOUT 6 HOURS TO RUN.
> #NO NEED TO RUN AGAIN... OUTPUT INTO :
> #write.csv(ts2, "E:/workingdirectory/mosquito/rainAndTemp/rainAndTemp.csv")
> #save.image("E:/workingdirectory/mosquito/rainAndTemp/rainAndTemp.RData")
>
> library(pingr)
> ########################################
> #READ IN THE RAIN TIME SERIES DATA [CREATED FROM WEBSCRAPING WUNDERGROUND]
> #READ IN MOSQUITO TIME SERIES DATA [CREATED FROM 2013 MOSQ SEASON SURVEIL]
> ########################################
> tsRain <- read.csv("E:/workingdirectory/mosquito/rainAndTemp/a_rain2008-2013.csv")
> tsMosq <- read.csv("E:/workingdirectory/mosquito/simple.csv")
> tsTemp <- read.csv("E:/workingdirectory/mosquito/rainAndTemp/a_tsTemp2008-2013.csv")
> ########################################
> #CONVERT DATES INTO R DATE OBJECTS
> ########################################
> tsRain$date <- as.Date(tsRain$date, format="%Y-%m-%d")
> tsMosq$date <- as.Date(tsMosq$date, format="%m/%d/%Y")
> tsTemp$date <- as.Date(tsTemp$date, format="%Y-%m-%d")
> ########################################
> #CREATE A MASTER TS
> ########################################
> ts <- as.data.frame(tsRain$date)
> colnames(ts) <- "date"
> ########################################
> #ADD RAIN TO TS
> ########################################
> ts$rain <- tsRain$rain
> ########################################
> #ADD MOSQUITOES (TOTAL AND VECTOR) TO TS
> # (NOTE, THESE ARE MOSQUITOES PER TRAP)
> ########################################
> ts$total <- NA
> for (i in tsMosq$date){
+   ts$total[which(ts$date == i)] <-
+     tsMosq$total[which(tsMosq$date == i)]
+ }
> ts$vector <- NA
> for (i in tsMosq$date){
+   ts$vector[which(ts$date == i)] <-
+     tsMosq$vector[which(tsMosq$date == i)]
+ }
> ########################################
> #ADD MINIMUM TEMP TO TS
```

```
> ########################################
> ts$minTemp <- NA
> for (i in ts$date){
+   ts$minTemp[which(ts$date == i)] <-
+     tsTemp$MinTemp[which(tsTemp$date == i)]
+ }
> ts$minTemp[which(ts$minTemp < -100)] <- NA
> ########################################
> #ADD MAXIMUM TEMP TO TS
> ########################################
> ts$maxTemp <- NA
> for (i in ts$date){
+   ts$maxTemp[which(ts$date == i)] <-
+     tsTemp$MaxTemp[which(tsTemp$date == i)]
+ }
> ping()
> ########################################
> #ADD COLUMNS FOR RAIN AND MINTEMP RANGES
> ########################################
>
> ts2 <- ts
> #Make columns for a range of 5-20 days old, plus 5-20 days older than that
> for (j in 5:20){
+   for (k in 5:20){
+     ts2[,paste0("rain", j, ".", j+k)] <- NA
+     ts2[,paste0("minTemp", j, ".", j+k)] <- NA
+   }
+ }
> ########################################
> #ADD CUMULATIVE RAINFALL TO CORRESPONDING COLUMNS
> ########################################
> for (j in colnames(ts2)[grepl("rain", colnames(ts2))][-1]){
+   for (i in 30:nrow(ts2)){
+     ts2[i,j] <-
+       sum(ts2$rain[which(ts2$date <=
+                            ts2$date[i-min(as.numeric(unlist(strsplit(gsub("rain", "", j),
+                            ts2$date >=
+                            ts2$date[i-max(as.numeric(unlist(strsplit(gsub("rain", "", j),
+   }
+ }
> #write.csv(ts2, "E:/workingdirectory/mosquito/rainAndTemp/rainOnly.csv")
>
>
> ########################################
> #ADD MINIMUM MINTEMP FOR EACH MINTEMP COLUMN COLUMN
> ########################################
```

```
> for (j in colnames(ts2)[grepl("minTemp", colnames(ts2))][-1]){
+   for (i in 30:nrow(ts2)){
+     ts2[i,j] <-
+       min(ts2$minTemp[which(ts2$date <=
+                               ts2$date[i-min(as.numeric(unlist(strsplit(gsub("minTemp", "'
+                             ts2$date >=
+                               ts2$date[i-max(as.numeric(unlist(strsplit(gsub("minTemp", "",
+   }
+ }
> ping(2)
> write.csv(ts2, "E:/workingdirectory/mosquito/rainAndTemp/b_rainAndTemp.csv")
> save.image("E:/workingdirectory/mosquito/rainAndTemp/b_rainAndTemp.RData")
> ping(8)
>
>
>
```

## Linear modelling

```
> library(pingr)
> ########################################
> # READ IN DATA FOR TEMP AND RAINFALL FOR LAST 6 YEARS
> # THIS DATA ALREADY HAS RANGES CALCULATED INTO IT (SEE COMBINERAINANDTEMP.R)
> ########################################
> ts <- read.csv("E:/workingdirectory/mosquito/rainAndTemp/b_rainAndTemp.csv")
> ########################################
> #### CREATE A VECTOR OF ALL THE POSSIBLE COMBINATIONS
> #OF MIN TEMP RANGES AND RAINFALL RANGES
> ########################################
>
> rainPosibs <- rep(colnames(ts)[grepl("rain", colnames(ts))][-1],
+                   length(colnames(ts)[grepl("minTemp", colnames(ts))][-1]))
> minTempPosibs <- sort(rep(colnames(ts)[grepl("minTemp", colnames(ts))][-1],
+                       length(colnames(ts)[grepl("rain", colnames(ts))][-1])))
> posibs <- paste0(rainPosibs, "AND", minTempPosibs)
> ########################################
> #CREATE A DATA FRAME FROM MY POSIBS VECTOR
> # THIS IS WHERE I'LL PUT MY MODEL QUALITY INDICATORS
> ########################################
> pred <- as.data.frame(posibs)
> ########################################
> #Test the r-squared for each column (RAIN ONLY)
> ########################################
> pred$r.squared <- NA
> for (i in 1:length(pred$posibs)){
+   mylm <- summary(lm(ts[,"total"] ~
+                        ts[,unlist(strsplit(posibs, "AND")[i])[1]] +
+                        ts[,unlist(strsplit(posibs, "AND")[i])[2]]
+                      ))
+   pred$r.squared[i] <- mylm$r.squared
+ }
> ########################################
> #Select best predicition model
> ########################################
> pred <- pred[rev(order(pred$r.squared)),]
> best <- as.character(pred$posibs[which(pred$r.squared == max(pred$r.squared))])
> myModel <- lm(ts$total ~ ts[, unlist(strsplit(best, "AND"))[1]] +
+                 ts[,unlist(strsplit(best, "AND"))[2]])
> summary(myModel)
> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
> ########################################
> # USING BEST MODEL, MAKE A PREDICTED COLUMN IN TS
> ########################################
```

```
> ts$predicted <- -234.105 + (ts$rain16.36*53.74) + (ts$minTemp16.31*3.496)
> save.image("E:/workingdirectory/mosquito/rainAndTemp/c_rainAndTempDone.Rdata")
> write.csv(pred, "E:/workingdirectory/mosquito/rainAndTemp/c_rainAndTempDone.csv")
> ping(2)
```

## Log-linear modelling

```
> library(pingr)
> ########################################
> # READ IN DATA FOR TEMP AND RAINFALL FOR LAST 6 YEARS
> ########################################
> ts <- read.csv("E:/workingdirectory/mosquito/rainAndTemp/b_rainAndTemp.csv")
> ########################################
> #LOAD THE MODEL VARIATIONS WITH THEIR R-SQUARED VALUES
> ########################################
> pred <- read.csv("E:/workingdirectory/mosquito/rainAndTemp/c_rainAndTempDone.csv")
> ########################################
> #Select best predicition model
> ########################################
> best <- as.character(pred$posibs[which(pred$r.squared == max(pred$r.squared))])
> ########################################
> #GET MODEL DETAILS
> ########################################
> bestModel <- lm(ts$total ~ ts$rain16.36 + ts$minTemp16.31)
> summary(bestModel)
> ########################################
> # USING BEST MODEL, MAKE A PREDICTED COLUMN IN TS
> ########################################
> ts$predicted <- -234.105 + (ts$rain16.36*53.74) + (ts$minTemp16.31*3.496)
> ########################################
> #PLAY AROUND WITH A FEW ALTERNATIVE MODELS
> ########################################
> #SQRT
> sqrtModel <- lm(sqrt(ts$total) ~ ts$rain16.36 + ts$minTemp16.31)
> summary(sqrtModel) #R.squared == 0.5094
> #LOG
> logModel <- lm(log(ts$total) ~ ts$rain16.36 + ts$minTemp16.31)
> summary(logModel) #R.squared == 0.5325
> #ALTMODEL
> plot(seq(6.3,6.5,0.01), seq(6.3,6.5,0.01)/6.5, type="n", ylim=c(0,1))
> for (i in seq(6.3,6.5,0.01)){
+   altModel <- summary(lm((ts$total)^(1/i) ~ ts$rain16.36 + ts$minTemp16.31))
+   #summary(altModel) #R.squared == 0.5094
+   bla <- cbind(i, altModel$r.squared)
+   points(i, altModel$r.squared, pch=".")
+   print(unlist(bla))
+ } #THIS FINDS BEST R-SQUARED USING ts$total^(1/6.4)
> altModel <- lm((ts$total)^(1/6.4) ~ ts$rain16.36 + ts$minTemp16.31)
> summary(altModel) #R.squared == 0.5408
> # Though the 6.4 model is slightly better, I'm sticking with log
> # But if taking the log of total works best,
```

```
> # maybe other predictors would have done a better job?
> # Time to re-run the 65536 simulations, this time taking the log of total
>
> ########### CREATING LOG MODEL
>
> ########################################
> #### CREATE A VECTOR OF ALL THE POSSIBLE COMBINATIONS
> #OF MIN TEMP RANGES AND RAINFALL RANGES
> ########################################
>
> rainPosibs <- rep(colnames(ts)[grepl("rain", colnames(ts))][-1],
+                   length(colnames(ts)[grepl("minTemp", colnames(ts))][-1]))
> minTempPosibs <- sort(rep(colnames(ts)[grepl("minTemp", colnames(ts))][-1],
+                           length(colnames(ts)[grepl("rain", colnames(ts))][-1])))
> posibs <- paste0(rainPosibs, "AND", minTempPosibs)
> ########################################
> #CREATE A DATA FRAME FROM MY POSIBS VECTOR
> # THIS IS WHERE I'LL PUT MY MODEL QUALITY INDICATORS
> ########################################
> predLog <- as.data.frame(posibs)
> ########################################
> #Test the r-squared for each column (RAIN ONLY)
> ########################################
> predLog$r.squared <- NA
> for (i in 1:length(predLog$posibs)){
+   mylm <- summary(lm(log(ts[,"total"]) ~
+                        ts[,unlist(strsplit(posibs, "AND")[i])[1]] +
+                        ts[,unlist(strsplit(posibs, "AND")[i])[2]]
+   ))
+   predLog$r.squared[i] <- mylm$r.squared
+ }
> ########################################
> #Select best predicition model
> ########################################
> predLog <- predLog[rev(order(predLog$r.squared)),]
> bestLog <- as.character(predLog$posibs[which(predLog$r.squared == max(predLog$r.squared))]
> bestModelLog <- lm(log(ts$total) ~ ts[, unlist(strsplit(bestLog, "AND"))[1]] +
+                 ts[,unlist(strsplit(bestLog, "AND"))[2]])
> summary(bestModelLog)
> summary(lm(log(ts$total)~
+              ts$rain17.37 +
+              ts$minTemp14.32))
> #%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
>
> ########################################
> # USING BEST LOG MODEL, MAKE A PREDICTEDLOG COLUMN IN TS
```

```
> ########################################
> ts$predictedLog <- exp(-1.121293 + (ts$rain17.37*.200055) + (ts$minTemp14.32*0.074435))
> save.image("E:/workingdirectory/mosquito/rainAndTemp/d_rainAndTempDoneLog.Rdata")
> write.csv(predLog, "E:/workingdirectory/mosquito/rainAndTemp/d_rainAndTempDoneLog.csv")
> write.csv(ts, "E:/workingdirectory/mosquito/rainAndTemp/d_tsRainTempPredPredLog.csv")
> ping(2)
>
> #The above is a simulation in which the log(total) is regressed against all the 512
> #possible combinations of minTemp and cum rain.
> #When this is done running, I'll compare it's r-squared values with those of before
> #(before being the linear, not log model)
>
>
>
```

## Visuals

```
> library(pingr)
> ########################################
> # READ IN DATA FOR TEMP AND RAINFALL FOR LAST 6 YEARS
> ########################################
> ts <- read.csv("E:/workingdirectory/mosquito/rainAndTemp/b_rainAndTemp.csv")
> ########################################
> #LOAD THE MODEL VARIATIONS WITH THEIR R-SQUARED VALUES
> ########################################
> pred <- read.csv("E:/workingdirectory/mosquito/rainAndTemp/c_rainAndTempDone.csv")
> ########################################
> # PLOT DISTRIBUTION OF ALL MODELS' R-SQUARED
> ########################################
> hist(pred$r.squared, col="darkgrey",
+      xlab="R-squared", main=NA, breaks=300, border="darkgrey")
> ########################################
> #Select best predicition model
> ########################################
> best <- as.character(pred$posibs[which(pred$r.squared == max(pred$r.squared))])
> ########################################
> #GET BEST MODEL DETAILS
> ########################################
> bestModel <- lm(ts$total ~ ts$rain16.36 + ts$minTemp16.31)
> summary(bestModel)
> ########################################
> #PLOT INDIVIDUAL PREDICTORS AGAINT OUTCOME
> ########################################
> #rain
> plot(ts$rain16.36, ts$total,
+      xlab="Cumulative rainfall 16-36 days prior",
+      ylab="Mosquitoes per trap",
+      pch=16, col=adjustcolor("darkred", alpha.f=0.4))
> abline(lm(ts$total ~ ts$rain16.36), col=adjustcolor("black", alpha.f=0.6), lwd=2)
> summary(lm(ts$total ~ ts$rain16.36)) # good = RAIN IS LINEAR
> #rain, log total
> plot(ts$rain16.36, log(ts$total))
> abline(lm(log(ts$total) ~ ts$rain16.36), col="red")
> summary(lm(log(ts$total) ~ ts$rain16.36)) # bad = RAIN IS NOT LOG
> #minTemp
> plot(ts$minTemp16.31, ts$total)
> abline(lm(ts$total ~ ts$minTemp16.31), col="red")
> summary(lm(ts$total ~ ts$minTemp16.31)) # bad = TEMP IS NOT LINEAR
> #mintemp, log total
> plot(ts$minTemp16.31, log(ts$total),
+      xlab="Minimum temperature 16-31 days prior",
```

```
+      ylab="Mosquitoes per trap", yaxt="n",
+      pch=16, col=adjustcolor("darkred", alpha.f=0.4))
> axis(side=2, at=as.numeric(quantile(log(ts$total), na.rm=TRUE)),
+      labels=as.numeric(quantile(ts$total, na.rm=TRUE)))
> abline(lm(log(ts$total) ~ ts$minTemp16.31),
+        col=adjustcolor("black", alpha.f=0.6), lwd=2)
> summary(lm(log(ts$total) ~ ts$minTemp16.31)) # good = TEMP IS LOG
> ########################################
> #SO, WHAT'S BETTER, LOG()ING THE TEMP OR LOGGING THE total?
> ########################################
> #linear
> bestModel <- lm(ts$total ~ ts$rain16.36 + ts$minTemp16.31)
> summary(bestModel) #R.squared == 0.4305
> #log
> logModel <- lm(log(ts$total) ~ ts$rain16.36 + ts$minTemp16.31)
> summary(logModel) #R.squared == 0.5325
> #logTemp
> logTempModel <- lm(ts$total ~ ts$rain16.36 + log(ts$minTemp16.31))
> summary(logTempModel) #R.squared == 0.4284
> #expTemp, no transformations for anything else
> expTempModel <- lm(ts$total ~ ts$rain16.36 + exp(ts$minTemp16.31))
> summary(logTempModel) #R.squared ==
> #ANSWER - LOGGING THE TOTAL
> ########################################
> # USING BEST MODEL, MAKE A PREDICTED COLUMN IN TS
> ########################################
> ts$predicted <- -234.105 + (ts$rain16.36*53.74) + (ts$minTemp16.31*3.496)
> ########################################
> #PLOT PREDICTED VS OBSERVED
> ########################################
> plot(ts$predicted, ts$total,
+      xlab="Predicted", ylab="Observed", pch=16,
+      col=adjustcolor("darkblue", alpha.f=0.4))
> plot(ts$predicted, ts$total^(1/2),
+      xlab="Predicted", ylab="Observed", pch=16,
+      col=adjustcolor("darkblue", alpha.f=0.4))
> #WRONG WAY - ARBITRARILY SORTS, NOT MATCHING [originally appeared in article this way]
> plot(sort(ts$predicted[which(is.na(ts$total) == FALSE)]),
+      (sort(ts$total[which(is.na(ts$total) == FALSE)])),
+      xlab="Predicted", ylab="Observed", pch=16,
+      col=adjustcolor("darkblue", alpha.f=0.4))
> ########################################
> #PLOT OBSERVED MOSQUITO NUMBERS
> ########################################
> par(mar=c(4,4,2,1))
> par(oma=c(0,0,0,0))
```

```
> par(mfrow=c(1,1))
> plot(ts$date, ts$total, type="n", xaxt="n", col="white", type="n")
> axis(side=1, at=ts$date[seq(1,2100,100)], label=ts$date[seq(1,2100,100)],
+      las=3, cex.axis=0.65, line=-0.85, tick=FALSE)
> ########################################
> #POLYGON OF OBSERVED
> ########################################
> polygon(c(ts$date[which(is.na(ts$total) == FALSE)],
+           rev(ts$date[which(is.na(ts$total) == FALSE)])),
+         c(ts$total[which(is.na(ts$total) == FALSE)],
+           rep(0, length(ts$date[which(is.na(ts$total) == FALSE)]))),
+         col= adjustcolor("darkred", alpha.f=0.6), border=NA)
> ########################################
> #LINES OF PREDICTED (ALL DATES)
> ########################################
> lines(ts$date,
+       ts$predicted,
+       col=adjustcolor("darkblue", alpha.f=0.6), lwd=2)
> ########################################
> #LINES OF OBSERVED
> ########################################
> lines(ts$date[which(is.na(ts$total) == FALSE)],
+       ts$total[which(is.na(ts$total) == FALSE)],
+       col=adjustcolor("darkred", alpha.f=0.6), lwd=2)
> ########################################
> #POLYGON OF PREDICTED
> ########################################
> polygon(c(ts$date[which(is.na(ts$total) == FALSE)],
+           rev(ts$date[which(is.na(ts$total) == FALSE)])),
+         c(ts$predicted[which(is.na(ts$total) == FALSE)],
+           rep(0, length(ts$date[which(is.na(ts$total) == FALSE)]))),
+         col= "red", border=NA)
> ########################################
> #LINES OF PREDICTED (ONLY FOR COLLECTION DATES)
> ########################################
> lines(ts$date[which(is.na(ts$total) == FALSE)],
+       ts$predicted[which(is.na(ts$total) == FALSE)],
+       col=adjustcolor("darkblue", alpha.f=0.6), lwd=2)
> #IDENTICAL, BUT USING LOG OF TOTAL
> ########################################
> #PLOT OBSERVED MOSQUITO NUMBERS
> ########################################
> par(mar=c(4,4,2,1))
> par(oma=c(0,0,0,0))
> par(mfrow=c(1,1))
> plot(ts$date, log(ts$total), type="n", xaxt="n", col="white", type="n")
```

```
> axis(side=1, at=ts$date[seq(1,2100,100)], label=ts$date[seq(1,2100,100)],
+      las=3, cex.axis=0.65, line=-0.85, tick=FALSE)
> ########################################
> #POLYGON OF OBSERVED
> ########################################
> polygon(c(ts$date[which(is.na(log(ts$total)) == FALSE)],
+           rev(ts$date[which(is.na(log(ts$total)) == FALSE)])),
+         c(log(ts$total)[which(is.na(log(ts$total)) == FALSE)],
+           rep(0, length(ts$date[which(is.na(log(ts$total)) == FALSE)]))),
+         col= adjustcolor("darkred", alpha.f=0.6), border=NA)
> ########################################
> #LINES OF PREDICTED (ALL DATES)
> ########################################
> library(splines)
> ts$predicted2 <- log(ts$predicted)
> ts$predicted2[which(is.na(ts$predicted2)==TRUE)] <- 0
> lines(ts$date,
+       ts$predicted2,
+       col=adjustcolor("darkblue", alpha.f=0.6),
+       lwd=3)
> #ONLY USING TRAP DATES
> lines(ts$date[which(is.na(ts$total) == FALSE)],
+       ts$predicted2[which(is.na(ts$total) == FALSE)],
+       col=adjustcolor("darkblue", alpha.f=0.6), lwd=2)
>
>
>
>
>
```