Rock Pool Temperature Model

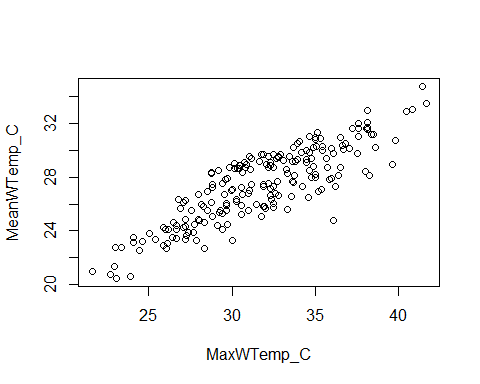
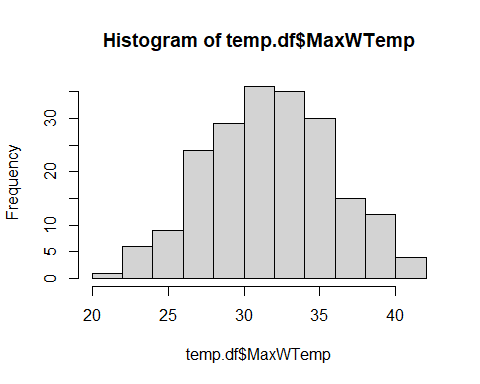
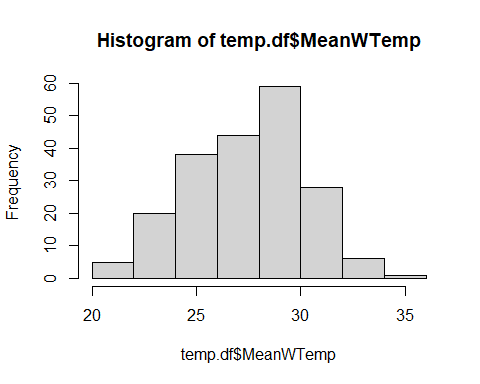
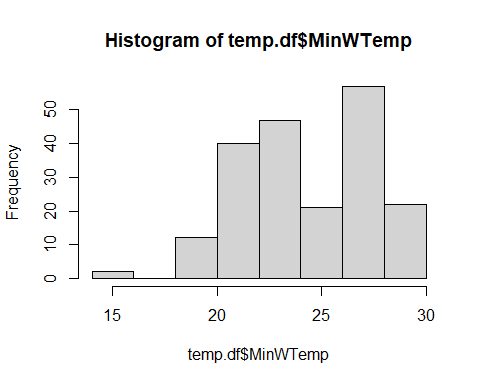
Andy

12/7/2021

# Histograms

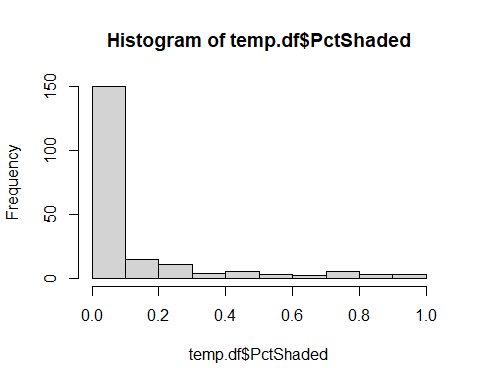
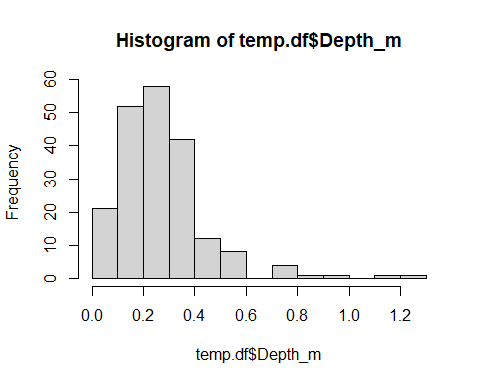
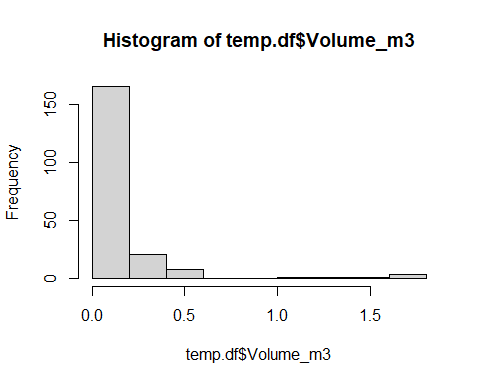
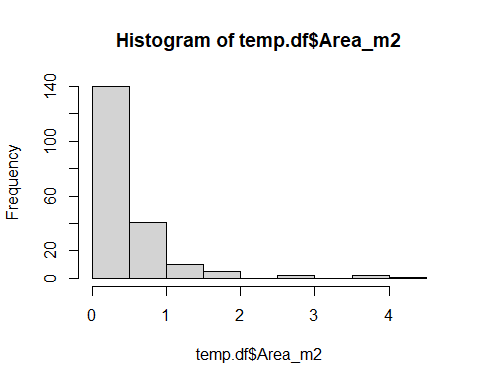
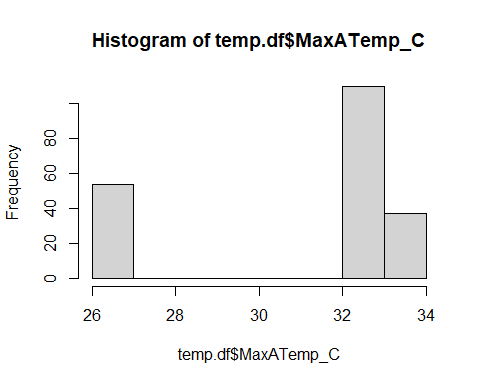
First, let’s take a look at some of the data and explore it for correlations.

## Temperature Metrics



For this analysis, I’m choosing to focus in on maximum water temperature as our dependent variable, because it has a nice and pretty distribution and correlates well with mean water temperature (which is sort of what we’re more interested in here). We could easily use mean temperature as well, but I am hesitant to use it because it does not represent a true daily “mean”, because our data were collected over 8 hour periods and not a full 24 hours.

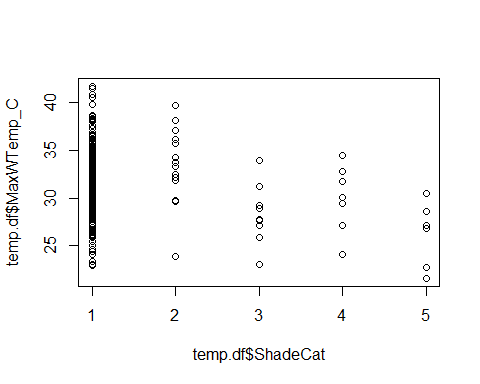
## Predictor Variables



## Applying Transformations

Lots of right-skewing going on, so we’ll do a few transformations. log for Area and Depth, and arcsine for PctShaded since it’s a proportion.

temp.df$Area\_cm2 <- temp.df$Area\_m2 \* 10000  
temp.df$logArea <- log(temp.df$Area\_cm2)  
temp.df$logDepth <- log(temp.df$Depth\_cm)  
temp.df$arcShade <- asin(sqrt(temp.df$PctShaded))  
  
temp.df$Date <- as.factor(temp.df$Date)  
  
plot(x = temp.df$ShadeCat, y = temp.df$MaxWTemp\_C)

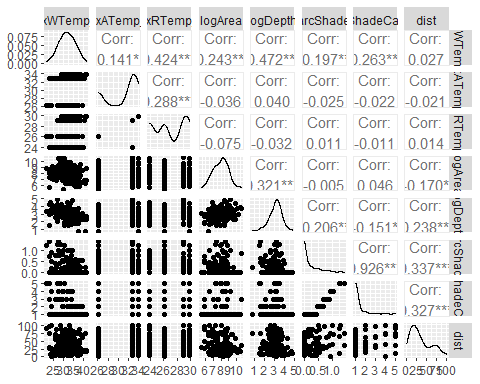


# Correlation Plots

## Loading required package: ggplot2

## Warning in register(): Can't find generic `scale\_type` in package ggplot2 to  
## register S3 method.

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2



Area and depth look like good predictors, as does maximum air temperature and shade. Oddly, there is also some colinearity going on with some of our predictor variables, too (e.g., area x depth, shade x depth).

# Preliminary Modeling

Let’s try making a basic GLMM using these variables, with area and depth as fixed effects and maximum air temperature as a random effect. We’ll take a stepwise approach to model selection.

##   
## 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

## `nlist()` is deprecated, use `tibble::lst()` instead.

## Modnames model\_vars K AICc  
## 1 mod2 MaxWTemp\_C + logDepth + arcShade + MaxATemp\_C 6 1037.879  
## 2 mod1 MaxWTemp\_C + logDepth + logArea + arcShade + MaxATemp\_C 10 1043.903  
## 3 mod6 MaxWTemp\_C + logDepth + arcShade 5 1061.329  
## 4 mod4 MaxWTemp\_C + logDepth + logArea + arcShade 9 1068.010  
## 5 mod7 MaxWTemp\_C + logDepth + MaxATemp\_C 4 1078.863  
## 6 mod5 MaxWTemp\_C + logDepth + logArea + MaxATemp\_C 6 1080.158  
## 7 mod14 MaxWTemp\_C + logDepth 3 1096.047  
## 8 mod8 MaxWTemp\_C + logDepth + logArea 5 1097.777  
## 9 mod3 MaxWTemp\_C + logArea + arcShade + MaxATemp\_C 6 1108.525  
## 10 mod10 MaxWTemp\_C + logArea + MaxATemp\_C 4 1115.886  
## 11 mod11 MaxWTemp\_C + arcShade + MaxATemp\_C 4 1119.596  
## 12 mod15 MaxWTemp\_C + MaxATemp\_C 3 1126.657  
## 13 mod9 MaxWTemp\_C + logArea + arcShade 5 1129.123  
## 14 mod12 MaxWTemp\_C + logArea 3 1134.460  
## 15 mod13 MaxWTemp\_C + arcShade 3 1138.785  
## Delta\_AICc ModelLik AICcWt LL Cum.Wt  
## 1 0.000000 1.000000e+00 9.531036e-01 -512.7232 0.9531036  
## 2 6.023907 4.919549e-02 4.688840e-02 -511.3727 0.9999920  
## 3 23.449341 8.091709e-06 7.712237e-06 -525.5105 0.9999997  
## 4 30.130251 2.866151e-07 2.731739e-07 -524.5336 1.0000000  
## 5 40.984048 1.260164e-09 1.201067e-09 -535.3297 1.0000000  
## 6 42.278889 6.595623e-10 6.286313e-10 -533.8626 1.0000000  
## 7 58.168018 2.338704e-13 2.229028e-13 -544.9628 1.0000000  
## 8 59.897378 9.850303e-14 9.388359e-14 -543.7345 1.0000000  
## 9 70.645960 4.564830e-16 4.350756e-16 -548.0462 1.0000000  
## 10 78.006638 1.150996e-17 1.097019e-17 -553.8410 1.0000000  
## 11 81.716547 1.800849e-18 1.716396e-18 -555.6959 1.0000000  
## 12 88.777262 5.275487e-20 5.028086e-20 -560.2674 1.0000000  
## 13 91.243967 1.536824e-20 1.464753e-20 -559.4078 1.0000000  
## 14 96.580968 1.065882e-21 1.015896e-21 -564.1692 1.0000000  
## 15 100.905421 1.226497e-22 1.168978e-22 -566.3315 1.0000000

Model 2, which excludes area but includes depth, has the lowest AIC. It indicates that depth and shade both negatively impact maximum water temperatures, and there is an interaction - deep, shady pools are probably especially cool. Additionally, maximum air temperature does pull out as a significant random effect.

Curiously, the max air temperature for two of the dates is the same, so I tried it with date as a random effect instead of air temperature, which returned a lower AIC value than model 2. It’s likely that date accounts for more variation in other atmospheric conditions (i.e., temperature variation and mean) than maximum air temperature alone does.

Water temperature data for the river is unavailable through the usual sources at the Westham Gage. Cartersville is available, and using max river temps from there in place of date returns similar AIC values. This might be a better predictive covariate than date and may help capture some of longer term variation in atmospheric conditions.

# River Temp as a random effect instead of date/air temp  
mod16 <- glmmTMB(MaxWTemp\_C ~ logDepth \* PctShaded + ( 1 | MaxRTemp\_C ), data = temp.df )  
summary(mod16)

## Family: gaussian ( identity )  
## Formula: MaxWTemp\_C ~ logDepth \* PctShaded + (1 | MaxRTemp\_C)  
## Data: temp.df  
##   
## AIC BIC logLik deviance df.resid   
## 1014.6 1034.4 -501.3 1002.6 195   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## MaxRTemp\_C (Intercept) 3.073 1.753   
## Residual 8.089 2.844   
## Number of obs: 201, groups: MaxRTemp\_C, 4  
##   
## Dispersion estimate for gaussian family (sigma^2): 8.09   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 44.5424 1.4400 30.931 < 2e-16 \*\*\*  
## logDepth -3.8063 0.3535 -10.766 < 2e-16 \*\*\*  
## PctShaded -20.5380 4.0077 -5.125 2.98e-07 \*\*\*  
## logDepth:PctShaded 4.9875 1.3459 3.706 0.000211 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

temp.df$residual <- residuals(mod16)

Continuous, untransformed shade actually provides a lower AIC value than binning it (0-20%, 20-40%, 40-60%, 60-80%, 80-100%) or arcsine transforming it.

mod17 <- glmmTMB(MaxWTemp\_C ~ logDepth \* ShadeCat + ( 1 | MaxRTemp\_C ), data = temp.df )  
summary(mod17)

## Family: gaussian ( identity )  
## Formula: MaxWTemp\_C ~ logDepth \* ShadeCat + (1 | MaxRTemp\_C)  
## Data: temp.df  
##   
## AIC BIC logLik deviance df.resid   
## 1014.6 1034.4 -501.3 1002.6 195   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## MaxRTemp\_C (Intercept) 3.112 1.764   
## Residual 8.088 2.844   
## Number of obs: 201, groups: MaxRTemp\_C, 4  
##   
## Dispersion estimate for gaussian family (sigma^2): 8.09   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 48.0784 1.8969 25.346 < 2e-16 \*\*\*  
## logDepth -4.5854 0.5374 -8.533 < 2e-16 \*\*\*  
## ShadeCat -4.3988 0.9347 -4.706 2.53e-06 \*\*\*  
## logDepth:ShadeCat 1.0219 0.3170 3.224 0.00127 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Including distance from river channel improves things too, but only marginally, and oddly, the parameter estimates for it aren’t significant.

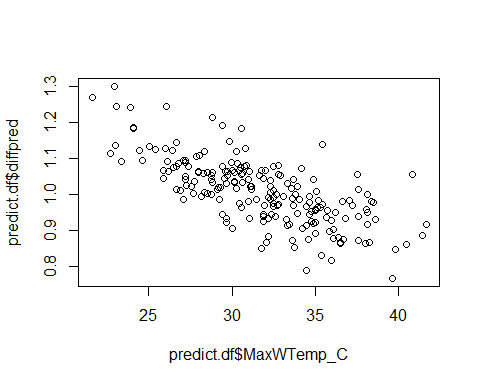
mod18 <- glmmTMB(MaxWTemp\_C ~ logDepth \* PctShaded \* dist + ( 1 | MaxRTemp\_C ), data = temp.df )  
summary(mod18)

## Family: gaussian ( identity )  
## Formula: MaxWTemp\_C ~ logDepth \* PctShaded \* dist + (1 | MaxRTemp\_C)  
## Data: temp.df  
##   
## AIC BIC logLik deviance df.resid   
## 1019.7 1052.8 -499.9 999.7 191   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## MaxRTemp\_C (Intercept) 3.079 1.755   
## Residual 7.973 2.824   
## Number of obs: 201, groups: MaxRTemp\_C, 4  
##   
## Dispersion estimate for gaussian family (sigma^2): 7.97   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 46.02870 2.18762 21.041 < 2e-16 \*\*\*  
## logDepth -4.30447 0.60317 -7.136 9.58e-13 \*\*\*  
## PctShaded -23.79674 7.29429 -3.262 0.00110 \*\*   
## dist -0.04296 0.04505 -0.954 0.34030   
## logDepth:PctShaded 6.89121 2.54880 2.704 0.00686 \*\*   
## logDepth:dist 0.01465 0.01425 1.028 0.30393   
## PctShaded:dist 0.07914 0.13870 0.571 0.56827   
## logDepth:PctShaded:dist -0.04058 0.04773 -0.850 0.39525   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

temp.df$residual <- residuals(mod18)

## Predicting from the Model

library(prediction)  
predict.df <- prediction(mod16, data = temp.df, calculate\_se = TRUE)  
predict.df$diffpred <- predict.df$fitted/predict.df$MaxWTemp\_C  
  
plot(x = predict.df$MaxWTemp\_C, y = predict.df$diffpred)



mean(predict.df$diffpred)

## [1] 1.008486

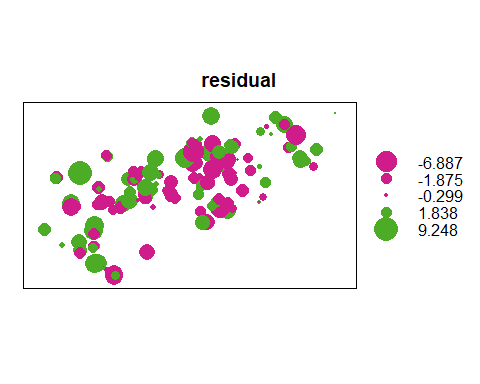
Yep, so there’s some bias here. It overestimates cool pools (by as much as 30%) and underestimates warm pools (by as much as 20%). Could be better. It looks like on average it overestimates things slightly, but not badly. This might not be a concern if there wasn’t such a clear, linear trend between bias and temperature.

This isn’t necessarily the end of the world, because we can be careful about how we propagate error. The predictions include std. errors, which could be incorporated into any future models stemming from this one.

# Spatially Explicit Modeling

In an attempt to solve our issues with overdispersion and biased estimates, let’s try accounting for spatial autocorrelation. First, let’s check for spatial autocorrelation in the residuals of the model. Similar residuals in neighboring pools may suggest autocorrelation.

library(sp)  
coordinates(temp.df) <- ~long + lat  
bubble(temp.df, "residual")



I’m not sure how to incorporate this into a GLMM, but this can maybe be done in glmmtmb. Below is a stab at this, using the lat/long coordinates as a covariate structure.

## #refugeeswelcome

MaxWTemp\_C

Predictors

Estimates

CI

p

(Intercept)

43.98

40.94 – 47.03

<0.001

logDepth

-3.66

-4.45 – -2.87

<0.001

PctShaded

-18.67

-26.98 – -10.36

<0.001

logDepth \* PctShaded

4.35

1.53 – 7.17

0.002

N MaxRTemp\_C

4

N group

1

Observations

201

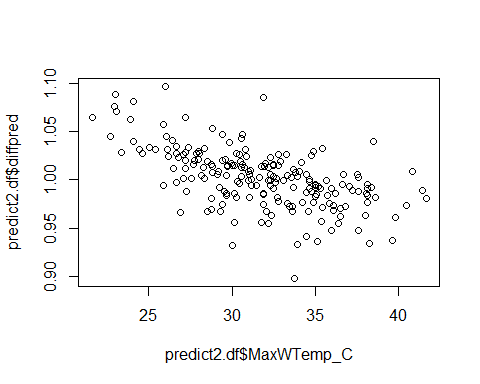
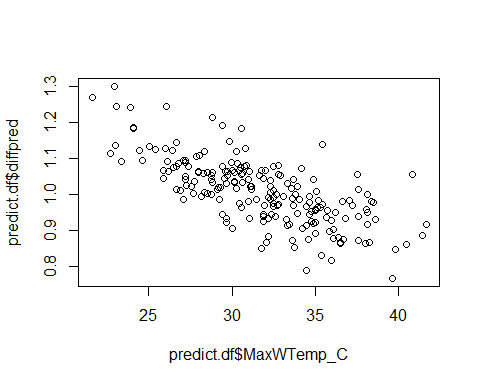
AIC

983.684

log-Likelihood

-483.842

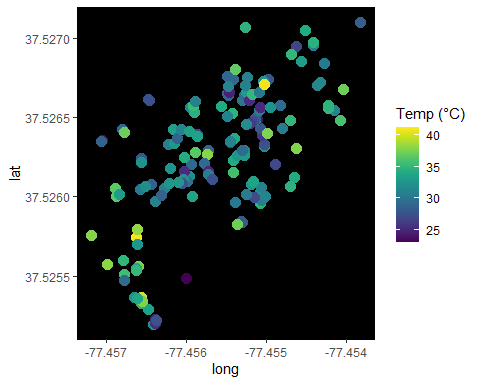
Better AIC and a much smaller dispersion estimate, at around 2.17. Let’s see how it handles predictions.



Interesting. It does seem to have tamped down on the over/underestimating some. It is still present, but most pools are within +/- 5%.

Million dollar question: can we plot it?

## Loading required package: viridisLite



The next step will be to challenge the model (and ourselves) a bit. Can we simulate a day on which the river temperature is 25°C, as an example, and predict temperatures for the whole system? I will need a data set containing pool dimensions and the densiometer data for the full system.