Black/Asian carp model selection

Eddie Wu

2023-10-11

Introduction

This .Rmd file is to show the progress on black carp and asian carp temperature and condition analyses. Since sub-sampling from spatial autocorrelation does not give significantly different results from normal analysis for black carp, we present the results without sub-sampling here. For other Asian carp species, we still sub-sample at a distance of 250 km.

SECTION 1 - Black Carp:

1. We first check how condition affects the relationship between black carp age at maturity and temperature by looking at three candidate models, for all six temperatures (annual, cold, warm, gdd0, water annual, water cold):

- Simple linear model (same slope, same intercept)
- Linear additive model (same slope, different intercept)
- Interaction model (different slope, different intercept)

2. Since we found that none of the interaction terms is significant, we will use only the simple linear model for prediction.

3. Since there is no significant relationship between warm temperature and age at maturity, we remove in from the following analyses.

SECTION 2 - Asian carp:

1. We conducted the same analyses on other asian carp species.

```
library(ggplot2)
library(ggfortify)
library(dplyr)
library(knitr)
library(tidyverse)
library(AICcmodavg) # for AICc and akaike weights
library(pwr)

## Import data
asian.carp <- read.csv("asian carp final.csv")
asian.carp$Condition <- as.factor(asian.carp$Condition)

Black <- read.csv("eddie_carp_new.csv")
Black$condition <- as.factor(Black$condition)

## Separate by species
Grass <- asian.carp[asian.carp$Species=="Grass",]
Bighead <- asian.carp[asian.carp$Species=="Bighead",]
Silver <- asian.carp[asian.carp$Species=="Silver",]
```

```r
Big.sil <- rbind(Bighead, Silver) # combine the two groups


## Define two functions for AICs
compute_akaike_weights <- function(aic_scores) {
  # Find the AIC of the best model
  aic_min <- min(aic_scores)

  # Calculate delta AIC values
  d_aic <- aic_scores - aic_min

  # Compute Akaike weights
  akaike_weights <- exp(-0.5 * d_aic) / sum(exp(-0.5 * d_aic))

  return(akaike_weights)
}
compare_aic_scores <- function(aic_scores) {
  # Find the AIC of the best model
  aic_min <- min(aic_scores)

  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(aic_min + 2 <= aic_scores[aic_scores != aic_min])

  # Return the index if
  if (is_smaller_by_two) {
    min_index <- which(aic_scores == aic_min)
    return(min_index)
  } else {
    return(-999)
  }
}
```

SECTION 1: Black carp

For black carp data, we do not subsample at any distances. But we removed the South Ukraine data point for all the following analyses.

Data cleaning

```r
# Clean data
Black <- Black %>% filter(!row_number() == 5) %>% filter(sex != "male")

# Remove the South Ukraine data point
black.clean <- Black %>% filter(!row_number() == 20)
```

Check slopes for three models (all six temperatures)

```r
# Build the models
black.simple <- lm(log(AAM)~AnnualTemp, data = black.clean)
black.linear <- lm(log(AAM)~AnnualTemp+condition, data = black.clean)
black.int <- lm(log(AAM)~AnnualTemp*condition, data = black.clean)
```

```r
## Look at the summary (especially the slope for each model)
summary(black.simple)
```

Annual Temperature

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42489 -0.12464  0.00059  0.09959  0.30683
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.984762   0.074361  26.691  < 2e-16 ***
## AnnualTemp  -0.017186   0.005344  -3.216  0.00433 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1754 on 20 degrees of freedom
## Multiple R-squared:  0.3409, Adjusted R-squared:  0.3079
## F-statistic: 10.34 on 1 and 20 DF,  p-value: 0.004333
```

```r
summary(black.linear)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp + condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44968 -0.12574  0.02118  0.12338  0.28093
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.007664   0.082986  24.193 9.76e-16 ***
## AnnualTemp       -0.016999   0.005428  -3.132  0.00549 **
## conditionnatural -0.050293   0.075985  -0.662  0.51600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.178 on 19 degrees of freedom
## Multiple R-squared:  0.3557, Adjusted R-squared:  0.2879
## F-statistic: 5.246 on 2 and 19 DF,  p-value: 0.01535
```

```r
summary(black.int)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp * condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43816 -0.06466 -0.00710  0.12129  0.24825
```

```
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.921742   0.104737  18.348 4.24e-13 ***
## AnnualTemp                  -0.009633   0.007761  -1.241    0.230
## conditionnatural             0.117098   0.148321   0.789    0.440
## AnnualTemp:conditionnatural -0.013941   0.010676  -1.306    0.208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1747 on 18 degrees of freedom
## Multiple R-squared:  0.4115, Adjusted R-squared:  0.3134
## F-statistic: 4.195 on 3 and 18 DF,  p-value: 0.02043
```

- There is no significant interaction term or additive term in black carp using **annual temperature**, thus the simple regression model is the best.

```
# Build the models
black.simple <- lm(log(AAM)~ColdTemp, data = black.clean)
black.linear <- lm(log(AAM)~ColdTemp+condition, data = black.clean)
black.int <- lm(log(AAM)~ColdTemp*condition, data = black.clean)


## Look at the summary (especially the slope for each model)
summary(black.simple)
```

Cold temperature

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39468 -0.12079 -0.00699  0.08961  0.29562
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.767262   0.035603  49.638   <2e-16 ***
## ColdTemp    -0.011423   0.003084  -3.704   0.0014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1664 on 20 degrees of freedom
## Multiple R-squared:  0.4069, Adjusted R-squared:  0.3772
## F-statistic: 13.72 on 1 and 20 DF,  p-value: 0.001405
```

```
summary(black.linear)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp + condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.41745 -0.10672  0.01471  0.11155  0.27214
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.790191   0.051231  34.944  < 2e-16 ***
## ColdTemp         -0.011293   0.003138  -3.598  0.00192 **
## conditionnatural -0.045613   0.072213  -0.632  0.53514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.169 on 19 degrees of freedom
## Multiple R-squared:  0.4191, Adjusted R-squared:  0.3579
## F-statistic: 6.853 on 2 and 19 DF,  p-value: 0.005745
```

```r
summary(black.int)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp * condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41954 -0.07945  0.00692  0.11033  0.24744
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.797309   0.050646  35.488   <2e-16 ***
## ColdTemp                  -0.007107   0.004471  -1.590    0.129
## conditionnatural          -0.053456   0.071224  -0.751    0.463
## ColdTemp:conditionnatural -0.007989   0.006176  -1.294    0.212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1661 on 18 degrees of freedom
## Multiple R-squared:  0.4685, Adjusted R-squared:  0.3799
## F-statistic: 5.288 on 3 and 18 DF,  p-value: 0.00861
```

- There is no significant interaction term or additive term in black carp using cold temperature, thus the simple regression model is the best.

```r
# Build the models
black.simple <- lm(log(AAM)~WarmTemp, data = black.clean)
black.linear <- lm(log(AAM)~WarmTemp+condition, data = black.clean)
black.int <- lm(log(AAM)~WarmTemp*condition, data = black.clean)


## Look at the summary (especially the slope for each model)
summary(black.simple)
```

Warm temperature

```
##
## Call:
## lm(formula = log(AAM) ~ WarmTemp, data = black.clean)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44602 -0.15359  0.03875  0.13845  0.30624
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.13752    0.26482   8.072 1.02e-07 ***
## WarmTemp    -0.01511    0.01098  -1.377    0.184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2065 on 20 degrees of freedom
## Multiple R-squared:  0.08655,    Adjusted R-squared:  0.04087
## F-statistic: 1.895 on 1 and 20 DF,  p-value: 0.1839
```

```r
summary(black.linear)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WarmTemp + condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47796 -0.12162  0.04703  0.11302  0.33778
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.17054    0.27213   7.976 1.75e-07 ***
## WarmTemp         -0.01516    0.01111  -1.364    0.188
## conditionnatural -0.06348    0.08918  -0.712    0.485
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2091 on 19 degrees of freedom
## Multiple R-squared:  0.1103, Adjusted R-squared:  0.01662
## F-statistic: 1.177 on 2 and 19 DF,  p-value: 0.3296
```

```r
summary(black.int)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WarmTemp * condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42989 -0.09849  0.03779  0.13637  0.30886
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.854098   0.440016   4.214 0.000522 ***
## WarmTemp                  -0.001877   0.018282  -0.103 0.919358
## conditionnatural           0.440537   0.556557   0.792 0.438936
## WarmTemp:conditionnatural -0.021179   0.023082  -0.918 0.370982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.21 on 18 degrees of freedom
## Multiple R-squared:   0.15,  Adjusted R-squared:  0.008371
## F-statistic: 1.059 on 3 and 18 DF,  p-value: 0.391
```

- Warm temperature is not a significant predictor of black carp age at maturity (same conclusion as Madison found on other Asian carps), thus we remove it in the following analyses.

```r
# Build the models
black.simple <- lm(log(AAM)~average_gdd_0, data = black.clean)
black.linear <- lm(log(AAM)~average_gdd_0+condition, data = black.clean)
black.int <- lm(log(AAM)~average_gdd_0*condition, data = black.clean)


## Look at the summary (especially the slope for each model)
summary(black.simple)
```

Base 0 annual growing degree day

```
##
## Call:
## lm(formula = log(AAM) ~ average_gdd_0, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46712 -0.09978  0.00423  0.08019  0.32039
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.142e+00  1.276e-01  16.787 2.97e-13 ***
## average_gdd_0 -6.895e-05  2.307e-05  -2.988  0.00727 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1797 on 20 degrees of freedom
## Multiple R-squared:  0.3086, Adjusted R-squared:  0.2741
## F-statistic: 8.929 on 1 and 20 DF,  p-value: 0.007268
```

```r
summary(black.linear)
```

```
##
## Call:
## lm(formula = log(AAM) ~ average_gdd_0 + condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49300 -0.09642  0.02122  0.10002  0.29298
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.165e+00  1.336e-01  16.200 1.41e-12 ***
## average_gdd_0    -6.828e-05  2.341e-05  -2.917  0.00884 **
## conditionnatural -5.321e-02  7.772e-02  -0.685  0.50183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1821 on 19 degrees of freedom
## Multiple R-squared:  0.3253, Adjusted R-squared:  0.2543
## F-statistic:  4.58 on 2 and 19 DF,  p-value: 0.0238
```

```r
summary(black.int)
```

```
##
## Call:
## lm(formula = log(AAM) ~ average_gdd_0 * condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45437 -0.07235 -0.00356  0.12141  0.24464
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.968e+00  1.731e-01  11.374 1.19e-09 ***
## average_gdd_0                 -3.055e-05  3.169e-05  -0.964    0.348
## conditionnatural               3.431e-01  2.472e-01   1.388    0.182
## average_gdd_0:conditionnatural -7.515e-05  4.472e-05  -1.681    0.110
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.174 on 18 degrees of freedom
## Multiple R-squared:  0.4168, Adjusted R-squared:  0.3196
## F-statistic: 4.288 on 3 and 18 DF,  p-value: 0.01893
```

- There is no significant interaction term or additive term in black carp ggd0, thus the simple regression model is the best.

```r
# Build the models
black.simple <- lm(log(AAM)~WaterTemp, data = black.clean)
black.linear <- lm(log(AAM)~WaterTemp+condition, data = black.clean)
black.int <- lm(log(AAM)~WaterTemp*condition, data = black.clean)


## Look at the summary (especially the slope for each model)
summary(black.simple)
```

Annual water temperature

```
##
## Call:
## lm(formula = log(AAM) ~ WaterTemp, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43891 -0.14624  0.01017  0.12283  0.35815
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.077127   0.137615  15.094 2.15e-12 ***
## WaterTemp   -0.020957   0.009204  -2.277   0.0339 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1926 on 20 degrees of freedom
## Multiple R-squared:  0.2059, Adjusted R-squared:  0.1661
## F-statistic: 5.184 on 1 and 20 DF,  p-value: 0.03393
```

```r
summary(black.linear)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterTemp + condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47350 -0.12440  0.00971  0.11910  0.32535
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.114401   0.145940  14.488 1.01e-11 ***
## WaterTemp        -0.021181   0.009284  -2.282   0.0342 *
## conditionnatural -0.068176   0.082820  -0.823   0.4206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1941 on 19 degrees of freedom
## Multiple R-squared:  0.2332, Adjusted R-squared:  0.1525
## F-statistic: 2.889 on 2 and 19 DF,  p-value: 0.08026
```

```r
summary(black.int)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterTemp * condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44069 -0.09479  0.00059  0.13315  0.24735
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.915864   0.178963  10.705  3.1e-09 ***
## WaterTemp                  -0.007394   0.011813  -0.626   0.5392
## conditionnatural            0.375292   0.264899   1.417   0.1736
## WaterTemp:conditionnatural -0.031109   0.017745  -1.753   0.0966 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1843 on 18 degrees of freedom
## Multiple R-squared:  0.345,  Adjusted R-squared:  0.2359
## F-statistic: 3.161 on 3 and 18 DF,  p-value: 0.04996
```

- There is no significant interaction term or additive term in black carp annual water temperature, thus the simple regression model is the best.

```r
# Build the models
```

```r
black.simple <- lm(log(AAM)~WaterCold, data = black.clean)
black.linear <- lm(log(AAM)~WaterCold+condition, data = black.clean)
black.int <- lm(log(AAM)~WaterCold*condition, data = black.clean)


## Look at the summary (especially the slope for each model)
summary(black.simple)
```

Cold water temperature

```
##
## Call:
## lm(formula = log(AAM) ~ WaterCold, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49139 -0.09544 -0.02325  0.11838  0.29075
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.923428   0.059900  32.111  < 2e-16 ***
## WaterCold   -0.023142   0.007403  -3.126  0.00532 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1771 on 20 degrees of freedom
## Multiple R-squared:  0.3283, Adjusted R-squared:  0.2947
## F-statistic: 9.773 on 1 and 20 DF,  p-value: 0.005318
```

```r
summary(black.linear)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterCold + condition, data = black.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50901 -0.10625 -0.01576  0.12532  0.26811
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.940266   0.069769  27.810  < 2e-16 ***
## WaterCold        -0.022750   0.007586  -2.999  0.00738 **
## conditionnatural -0.038602   0.077398  -0.499  0.62369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1805 on 19 degrees of freedom
## Multiple R-squared:  0.3369, Adjusted R-squared:  0.2671
## F-statistic: 4.828 on 2 and 19 DF,  p-value: 0.02017
```

```r
summary(black.int)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterCold * condition, data = black.clean)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45824 -0.04944 -0.00493  0.10864  0.21744
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.862927   0.073076  25.493 1.41e-15 ***
## WaterCold                 -0.009306   0.009299  -1.001   0.3302
## conditionnatural           0.153727   0.113381   1.356   0.1919
## WaterCold:conditionnatural -0.030329   0.013967  -2.171   0.0435 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1651 on 18 degrees of freedom
## Multiple R-squared:  0.4746, Adjusted R-squared:  0.387
## F-statistic: 5.419 on 3 and 18 DF,  p-value: 0.0078
```

- There is a significant interaction term in black carp using cold quarter water temperature, thus we need to further decide based on R2 and AICc values.

```
## Get a table of corrected AICs and their Akaike weights
models <- list(black.simple, black.linear, black.int)
mod.names <- c('simple linear', 'linear additive', 'interaction')
aictab(cand.set = models, modnames = mod.names, sort = FALSE)
```

```
##
## Model selection based on AICc:
##
##                 K  AICc Delta_AICc AICcWt    LL
## simple linear   3 -8.49       0.00   0.54  7.91
## linear additive 4 -5.76       2.73   0.14  8.06
## interaction     5 -7.48       1.01   0.32 10.62
```

```
## R^2 value for three models
r_2 <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(summary(black.simple)$adj.r.squared,
         summary(black.linear)$adj.r.squared,
         summary(black.int)$adj.r.squared)
)
kable(r_2)
```

Model	R2
Simple linear	0.2946727
Linear additive	0.2671446
Interaction	0.3870111

- There was no preference between the two models, as the AICc values are within two units of each other.

Temperature predictions and power analyses

```
## Build the models with three temperature metrics (simple model here)
black.annual <- lm(log(AAM)~AnnualTemp, data = black.clean)
black.cold <- lm(log(AAM)~ColdTemp, data = black.clean)
```

```r
black.gdd <- lm(log(AAM)~average_gdd_0, data = black.clean)
black.water <- lm(log(AAM)~WaterTemp, data = black.clean)
black.waterC <- lm(log(AAM)~WaterCold, data = black.clean)


## Power analyses - annual
# calculate the coefficient of determination
coe.annual <- summary(black.annual)$adj.r.squared
pwr.f2.test(u = 1, v = 22 - 1 - 1, f2 = coe.annual/(1 -coe.annual),
            sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 20
##              f2 = 0.4449434
##       sig.level = 0.05
##           power = 0.8450604
```

```r
## Power analyses - cold
# calculate the coefficient of determination
coe.cold <- summary(black.cold)$adj.r.squared
pwr.f2.test(u = 1, v = 22 - 1 - 1, f2 = coe.cold/(1 -coe.cold),
            sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 20
##              f2 = 0.6056428
##       sig.level = 0.05
##           power = 0.9344838
```

```r
## Power analyses - gdd0
# calculate the coefficient of determination
coe.gdd <- summary(black.gdd)$adj.r.squared
pwr.f2.test(u = 1, v = 22 - 1 - 1, f2 = coe.gdd/(1 -coe.gdd),
            sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 20
##              f2 = 0.37756
##       sig.level = 0.05
##           power = 0.7827331
```

```r
pwr.f2.test(u = 1, f2 = coe.gdd/(1 -coe.gdd),
            sig.level = 0.05, power = 0.8)
```

```
##
##      Multiple regression power calculation
##
```

```
##               u = 1
##               v = 20.87807
##              f2 = 0.37756
##       sig.level = 0.05
##           power = 0.8
```

```r
## Power analyses - water annual
# calculate the coefficient of determination
coe.water <- summary(black.water)$adj.r.squared
pwr.f2.test(u = 1, v = 22 - 1 - 1, f2 = coe.water/(1 -coe.water),
            sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 20
##              f2 = 0.1992484
##       sig.level = 0.05
##           power = 0.5131047
```

```r
pwr.f2.test(u = 1, f2 = coe.water/(1 -coe.water),
            sig.level = 0.05, power = 0.8)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 39.40371
##              f2 = 0.1992484
##       sig.level = 0.05
##           power = 0.8
```

```r
## Power analyses - water cold
# calculate the coefficient of determination
coe.waterC <- summary(black.waterC)$adj.r.squared
pwr.f2.test(u = 1, v = 22 - 1 - 1, f2 = coe.waterC/(1 -coe.waterC),
            sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 20
##              f2 = 0.4177814
##       sig.level = 0.05
##           power = 0.8221123
```

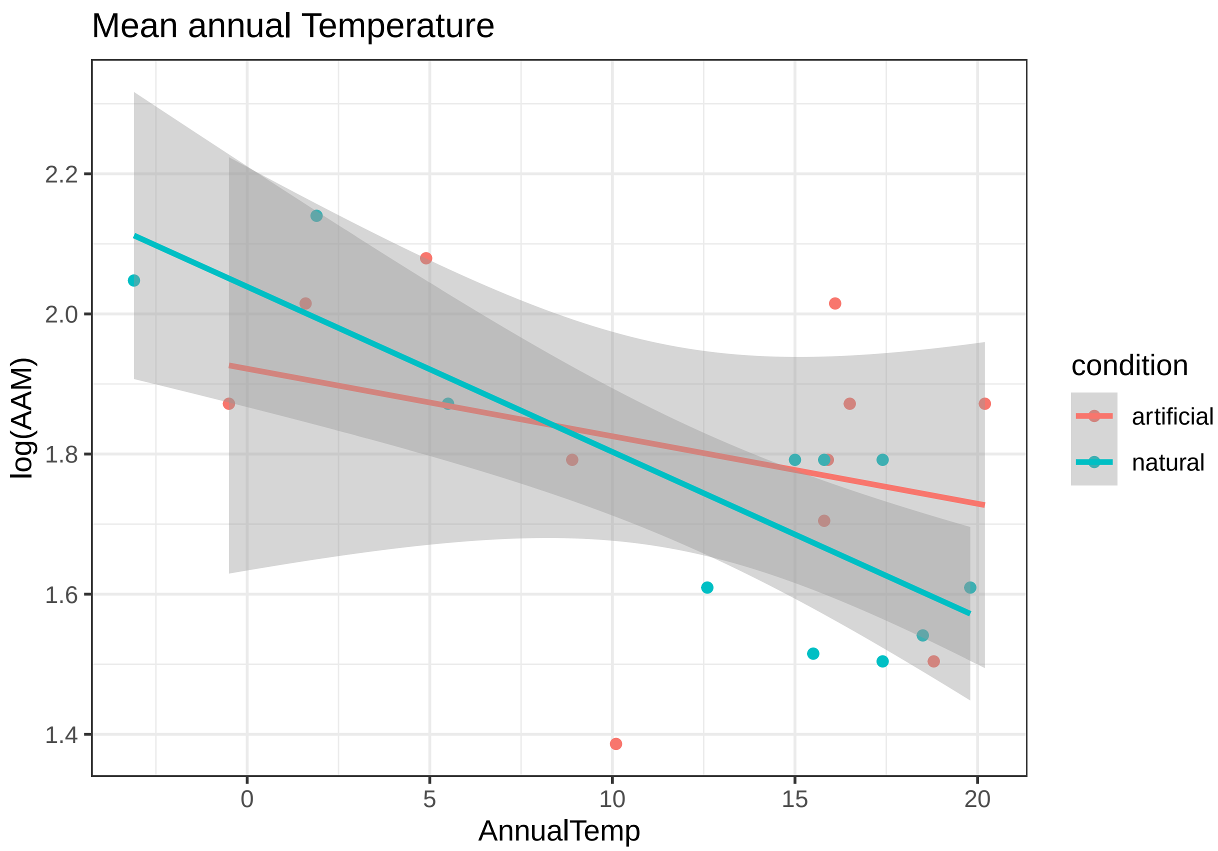
- Power analyses suggested that our current sample size is sufficient enough to produce a strong statistical power for annual temperauture, cold temperature, gdd0 (?), and cold water temperature.
- Annual water?

Black carp graphs with two conditions separated

We separated the black carp dataset into two based on conditions. Since there was no preference over the four models, we used the simple linear model on each set of the data.

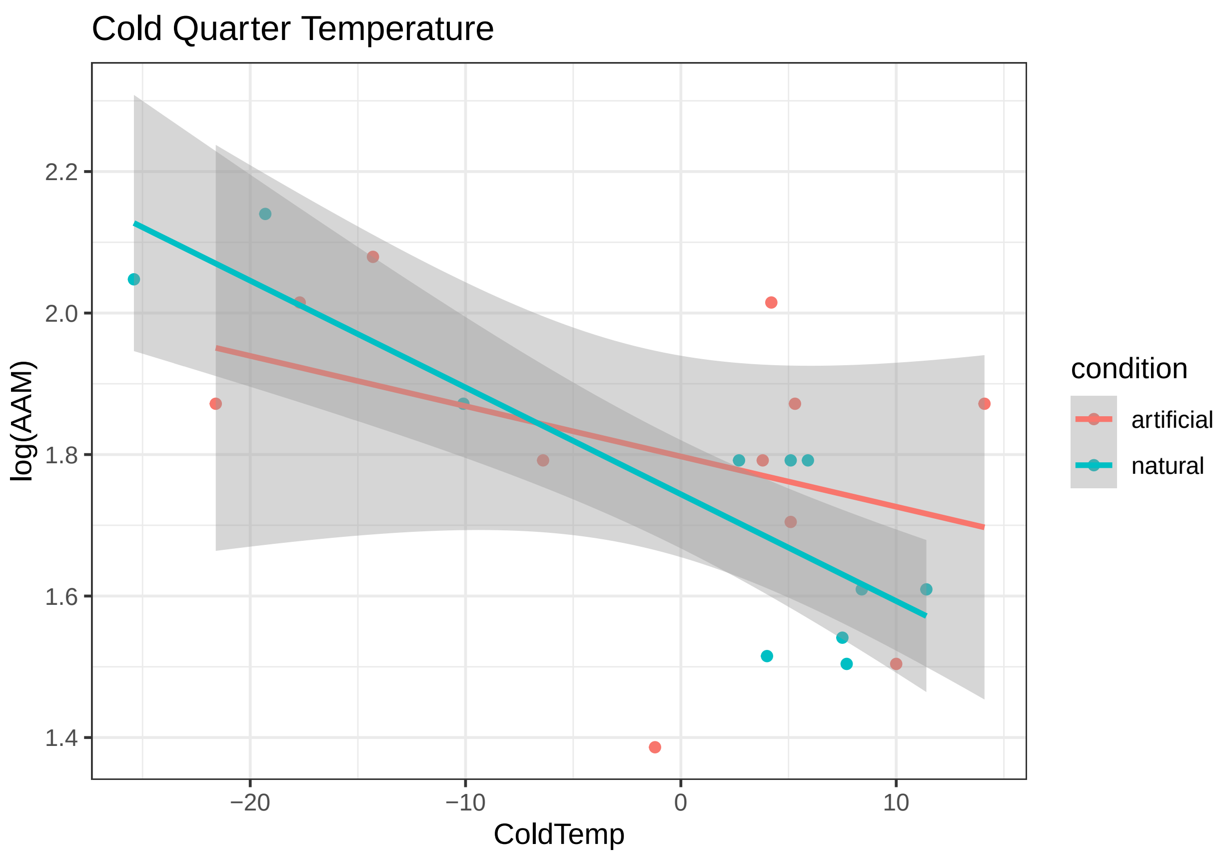
```r
## Annual temperature
ggplot(black.clean, aes(x = AnnualTemp, y = log(AAM), color = condition))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()+
  labs(title = "Mean annual Temperature")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



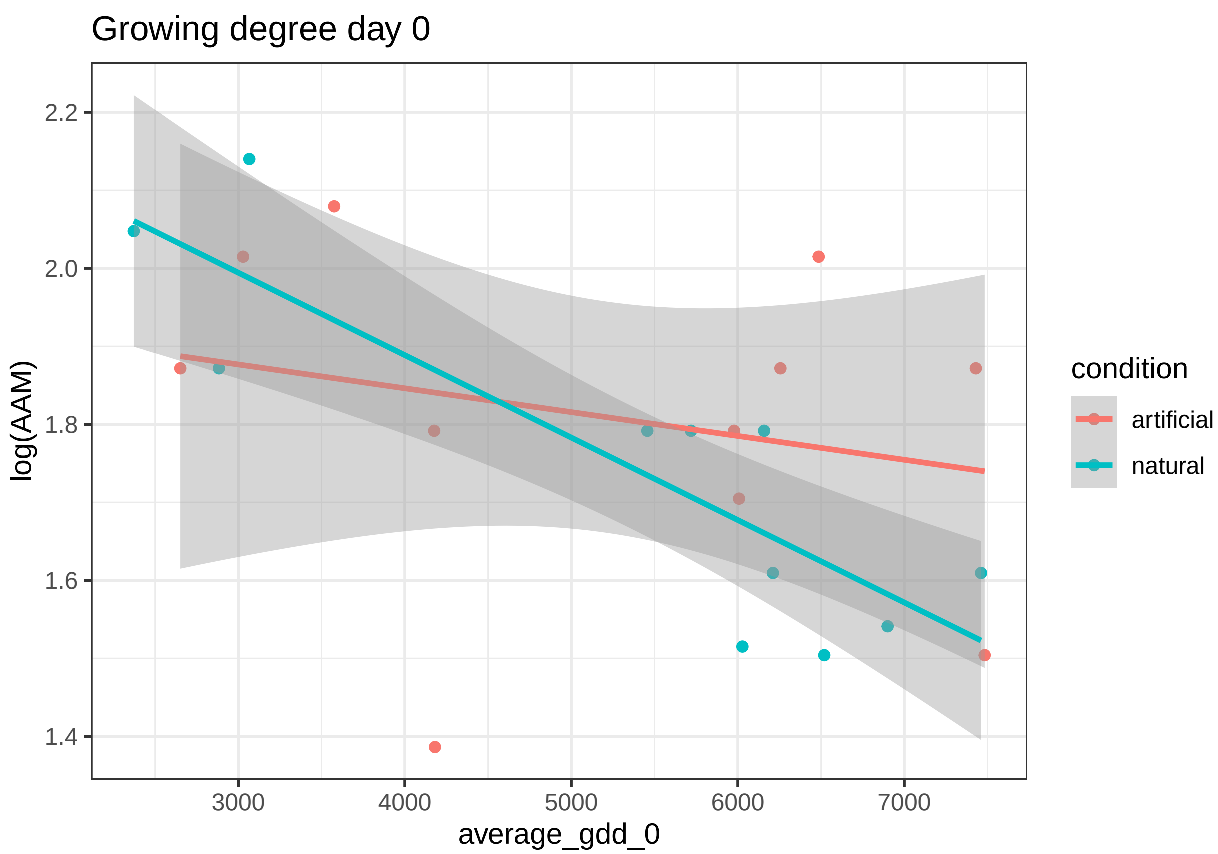

```r
## Cold temperature
ggplot(black.clean, aes(x = ColdTemp, y = log(AAM), color = condition))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()+
  labs(title = "Cold Quarter Temperature")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



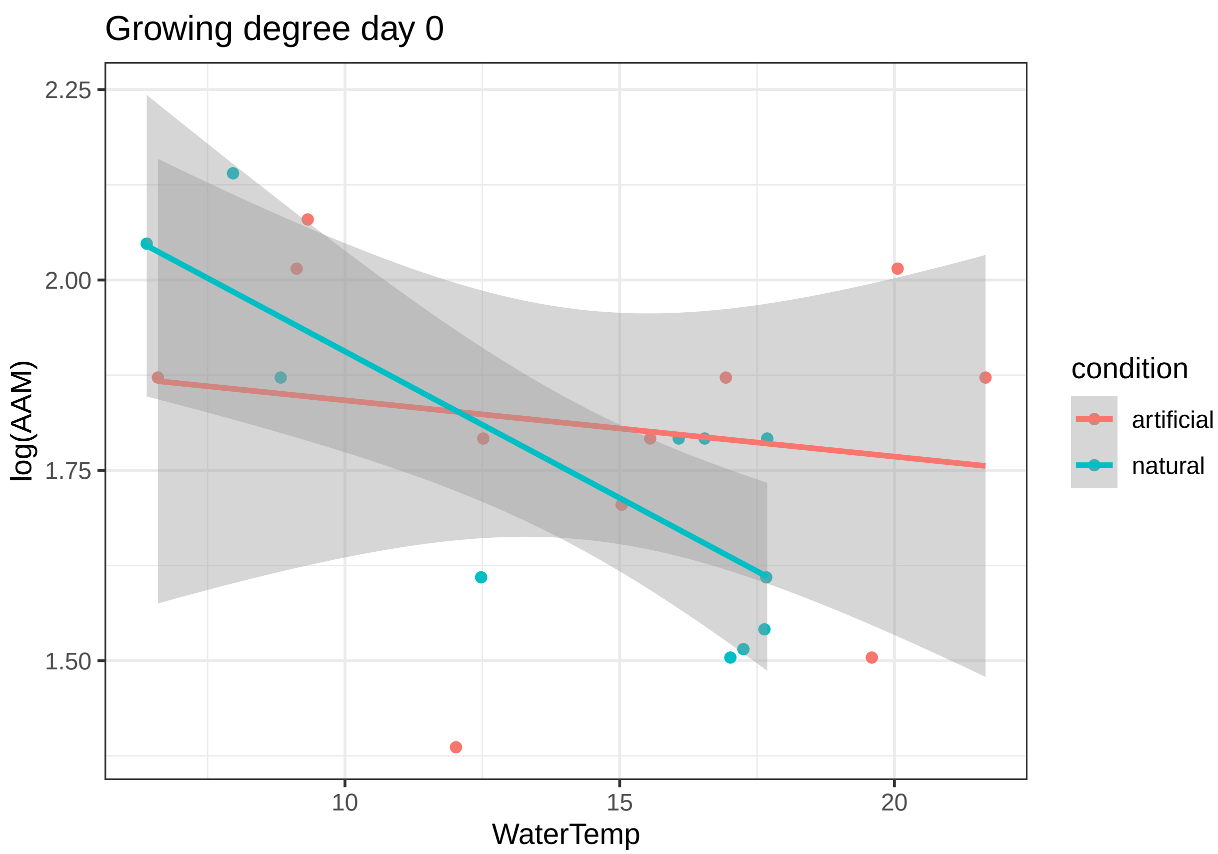

```r
## GDD0
ggplot(black.clean, aes(x = average_gdd_0, y = log(AAM), color = condition))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()+
  labs(title = "Growing degree day 0")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



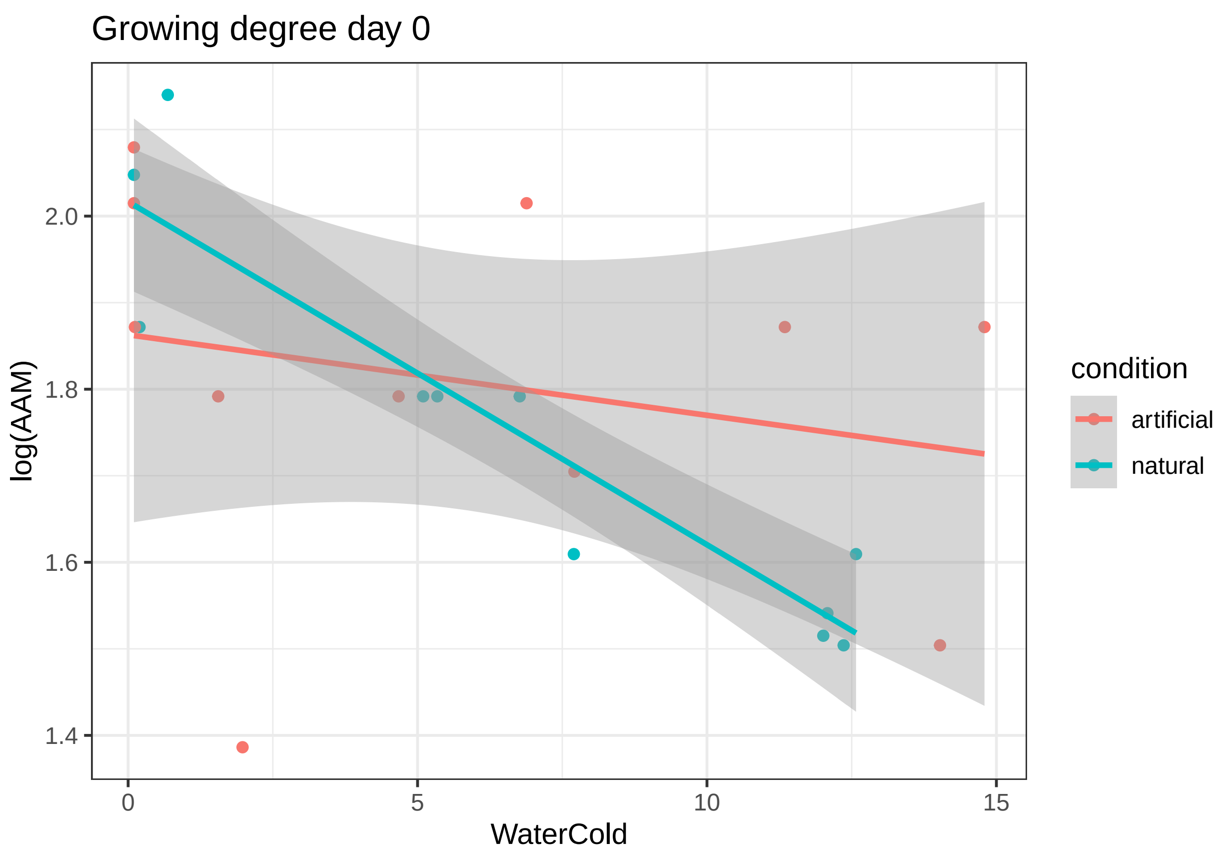

```r
## Annual water temperature
ggplot(black.clean, aes(x = WaterTemp, y = log(AAM), color = condition))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()+
  labs(title = "Growing degree day 0")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```r
## Cold water temperature
ggplot(black.clean, aes(x = WaterCold, y = log(AAM), color = condition))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()+
  labs(title = "Growing degree day 0")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Now that we have seen that the artificial condition data seems to have a larger spread, we would like to run a regression with only the natural conditions to see how much the R2 value can improve? Would it be similar to other Asian carp species (around 0.6)?

```r
## Separate into two data sets
black.natural <- black.clean[black.clean$condition == "natural",]
black.artificial <- black.clean[black.clean$condition == "artificial",]


## Run the models
black.annual.n <- lm(log(AAM)~AnnualTemp, data = black.natural)
black.cold.n <- lm(log(AAM)~ColdTemp, data = black.natural)
black.gdd.n <- lm(log(AAM)~average_gdd_0, data = black.natural)
black.water.n <- lm(log(AAM)~WaterTemp, data = black.natural)
black.waterC.n <- lm(log(AAM)~WaterCold, data = black.natural)


black.annual.a <- lm(log(AAM)~AnnualTemp, data = black.artificial)
black.cold.a <- lm(log(AAM)~ColdTemp, data = black.artificial)
black.gdd.a <- lm(log(AAM)~average_gdd_0, data = black.artificial)
black.water.a <- lm(log(AAM)~WaterTemp, data = black.artificial)
black.waterC.a <- lm(log(AAM)~WaterCold, data = black.artificial)


## Compare the model parameters
summary(black.annual.n)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp, data = black.natural)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15831 -0.09440 -0.03738  0.11596  0.16311
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.038839   0.075970  26.838  6.7e-10 ***
## AnnualTemp  -0.023574   0.005304  -4.445  0.00161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1264 on 9 degrees of freedom
## Multiple R-squared:  0.687,  Adjusted R-squared:  0.6523
## F-statistic: 19.76 on 1 and 9 DF,  p-value: 0.001612
```

```r
summary(black.annual.a)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp, data = black.artificial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43816 -0.05978  0.02318  0.12682  0.24825
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.921742   0.127277  15.099 1.07e-07 ***
## AnnualTemp  -0.009633   0.009431  -1.021    0.334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2124 on 9 degrees of freedom
## Multiple R-squared:  0.1039, Adjusted R-squared:  0.004303
## F-statistic: 1.043 on 1 and 9 DF,  p-value: 0.3337
```

```r
summary(black.cold.n)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp, data = black.natural)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.168342 -0.084535 -0.007609  0.096764  0.136973
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.743853   0.033825  51.555 1.95e-12 ***
## ColdTemp    -0.015096   0.002878  -5.246 0.000531 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1122 on 9 degrees of freedom
## Multiple R-squared:  0.7536, Adjusted R-squared:  0.7262
## F-statistic: 27.52 on 1 and 9 DF,  p-value: 0.0005305
```

```r
summary(black.cold.a)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp, data = black.artificial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41954 -0.06766  0.02146  0.14343  0.24744
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.797309   0.062927  28.562 3.85e-10 ***
## ColdTemp    -0.007107   0.005555  -1.279    0.233
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2063 on 9 degrees of freedom
## Multiple R-squared:  0.1539, Adjusted R-squared:  0.05987
## F-statistic: 1.637 on 1 and 9 DF,  p-value: 0.2328
```

```r
summary(black.gdd.n)
```

```
##
## Call:
## lm(formula = log(AAM) ~ average_gdd_0, data = black.natural)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15923 -0.08195 -0.01309  0.08566  0.15266
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.311e+00  1.172e-01  19.729 1.02e-08 ***
## average_gdd_0 -1.057e-04  2.094e-05  -5.048 0.000692 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1154 on 9 degrees of freedom
## Multiple R-squared:  0.739,  Adjusted R-squared:   0.71
## F-statistic: 25.48 on 1 and 9 DF,  p-value: 0.000692
```

```r
summary(black.gdd.a)
```

```
##
## Call:
## lm(formula = log(AAM) ~ average_gdd_0, data = black.artificial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.45437 -0.06458  0.00598  0.13471  0.24464
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.968e+00  2.161e-01   9.108 7.75e-06 ***
## average_gdd_0 -3.055e-05  3.957e-05  -0.772     0.46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2172 on 9 degrees of freedom
## Multiple R-squared:  0.0621, Adjusted R-squared:  -0.04212
## F-statistic: 0.5959 on 1 and 9 DF,  p-value: 0.4599
```

```r
summary(black.water.n)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterTemp, data = black.natural)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20126 -0.09558 -0.00148  0.12860  0.18153
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.291156   0.146522  15.637 7.86e-08 ***
## WaterTemp   -0.038503   0.009934  -3.876  0.00375 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1383 on 9 degrees of freedom
## Multiple R-squared:  0.6254, Adjusted R-squared:  0.5837
## F-statistic: 15.02 on 1 and 9 DF,  p-value: 0.003755
```

```r
summary(black.water.a)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterTemp, data = black.artificial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44069 -0.06575  0.00470  0.14127  0.24735
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.915864   0.214544   8.930  9.1e-06 ***
## WaterTemp   -0.007394   0.014162  -0.522    0.614
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.221 on 9 degrees of freedom
## Multiple R-squared:  0.0294, Adjusted R-squared:  -0.07845
## F-statistic: 0.2726 on 1 and 9 DF,  p-value: 0.6142
```

```r
summary(black.waterC.n)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterCold, data = black.natural)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.13705 -0.02420 -0.01319  0.03911  0.15057
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.01665    0.04467  45.145 6.42e-12 ***
## WaterCold   -0.03963    0.00537  -7.381 4.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08508 on 9 degrees of freedom
## Multiple R-squared:  0.8582, Adjusted R-squared:  0.8425
## F-statistic: 54.48 on 1 and 9 DF,  p-value: 4.188e-05
```

```r
summary(black.waterC.a)
```

```
##
## Call:
## lm(formula = log(AAM) ~ WaterCold, data = black.artificial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45824 -0.07156  0.00996  0.14973  0.21744
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.862927   0.096240   19.36 1.21e-08 ***
## WaterCold   -0.009306   0.012247   -0.76    0.467
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2175 on 9 degrees of freedom
## Multiple R-squared:  0.06029,   Adjusted R-squared:  -0.04412
## F-statistic: 0.5775 on 1 and 9 DF,  p-value: 0.4667
```

```r
## Comparing R^2
r_2 <- data.frame(
  Temperature = c("Annual", "Cold", "GGDO", "Water", "Water cold"),
  Black.combined = c(0.31, 0.38, 0.27, 0.17, 0.29),
  Black.natural = c(summary(black.annual.n)$adj.r.squared,
                    summary(black.cold.n)$adj.r.squared,
                    summary(black.gdd.n)$adj.r.squared,
                    summary(black.water.n)$adj.r.squared,
                    summary(black.waterC.n)$adj.r.squared),
  Black.artificial = c(summary(black.annual.a)$adj.r.squared,
                       summary(black.cold.a)$adj.r.squared,
                       summary(black.gdd.a)$adj.r.squared,
                       summary(black.water.a)$adj.r.squared,
```

```r
                        summary(black.waterC.a)$adj.r.squared),
  Asian.total = c(0.6, 0.57, 0.62, NA, NA)
)
kable(r_2)
```

Temperature	Black.combined	Black.natural	Black.artificial	Asian.total
Annual	0.31	0.6522554	0.0043026	0.60
Cold	0.38	0.7261770	0.0598742	0.57
GGD0	0.27	0.7100129	-0.0421154	0.62
Water	0.17	0.5837250	-0.0784478	NA
Water cold	0.29	0.8424617	-0.0441176	NA

- It turned out that removing the artificial conditions greatly increase the predictive power of the model for all five temperatures. While the artificial model alone did not even have a significant relationship.
- The resulting R2 (for the natural environments) is similar to that of the Asian carp.

SECTION 2: Asian carp

1. We followed the same work flow for other Asian carp species. However, for Asian carp, we subsample at a distance of 250 km to avoid spatial autocorrelation.

Data cleaning and matrices for results

```r
## Look at the spatial codes for the current asian carp data
asian.carp.clean <- asian.carp %>%
  filter(Condition %in% c("natural", "artificial"))

table(asian.carp.clean$Code_Str)
```

```
##
##  A AA AB AC AD AE AF AG AH AI AJ AK AL  B  C  D  E  F  G  H  I  J  K  L  M  N
##  4  3  3  2  1  1  1  3  1  3  1  3  2  1  1  1  1  2  1  1  3  4  1  2  2  1
##  O  P  Q  R  S  T  U  V  W  X  Y  Z
##  1  1  1  3  1  1  3  3  1  1  1  1
```

```r
# Stratified sub-sampling gives 21 artificial and 17 natural conditions.

# Simple linear model - slope, intercept, p, blank, r2, AICc
linear.results <- matrix(NA,1000,18)
colnames(linear.results) <- c("slope.a",
                              "intercept.a",
                              "p for slope.a",
                              "blank.a",
                              "r2.a",
                              "AICc.a",
                              "slope.c",
                              "intercept.c",
                              "p for slope.c",
                              "blank.c",
                              "r2.c",
                              "AICc.c",
                              "slope.g",
                              "intercept.g",
```

```
                                  "p for slope.g",
                                  "blank.g",
                                  "r2.g",
                                  "AICc.g")
# Linear additive model - slope, intercept, p(slope), p(additive), r2, AICc
add.results <- matrix(NA,1000,18)
colnames(add.results) <- c("slope.a",
                                  "intercept.a",
                                  "p for slope.a",
                                  "additive term.a",
                                  "r2.a",
                                  "AICc.a",
                                  "slope.c",
                                  "intercept.c",
                                  "p for slope.c",
                                  "additive term.c",
                                  "r2.c",
                                  "AICc.c",
                                  "slope.g",
                                  "intercept.g",
                                  "p for slope.g",
                                  "additive term.g",
                                  "r2.g",
                                  "AICc.g")
# Interaction model - slope, intercept, p(slope), p(interaction), r2, AICc
int.results <- matrix(NA,1000,18)
colnames(int.results) <- c("slope.a",
                                  "intercept.a",
                                  "p for slope.a",
                                  "interaction term.a",
                                  "r2.a",
                                  "AICc.a",
                                  "slope.c",
                                  "intercept.c",
                                  "p for slope.c",
                                  "interaction term.c",
                                  "r2.c",
                                  "AICc.c",
                                  "slope.g",
                                  "intercept.g",
                                  "p for slope.g",
                                  "interaction term.g",
                                  "r2.g",
                                  "AICc.g")
```

- Stratified sub-sampling gives 21 artificial and 17 natural conditions.

Define the models

```
# For 1000 iterations
for(i in 1:1000){
  sub <- asian.carp.clean %>% group_by(Code_Str) %>% sample_n(size=1)

  ## annual
```

```r
reg.linear.annual <- lm(log(AAM)~AnnualTemp, data = sub)
reg.add.annual <- lm(log(AAM)~AnnualTemp+Condition, data = sub)
reg.int.annual <- lm(log(AAM)~AnnualTemp*Condition, data = sub)

# simple linear model
linear.results[i,1]<-summary(reg.linear.annual)$coef[2,1] #slope
linear.results[i,2]<-summary(reg.linear.annual)$coef[1,1] #intercept
linear.results[i,3]<-summary(reg.linear.annual)$coef[2,4] #p-value
linear.results[i,4]<-0 #blank
linear.results[i,5]<-summary(reg.linear.annual)$adj.r.squared #r2
linear.results[i,6]<-as.numeric(AICc(reg.linear.annual)) #AICc

# linear additive model
add.results[i,1]<-summary(reg.add.annual)$coef[2,1] #slope
add.results[i,2]<-summary(reg.add.annual)$coef[1,1] #intercept
add.results[i,3]<-summary(reg.add.annual)$coef[2,4] #p(slope)
add.results[i,4]<-summary(reg.add.annual)$coef[3,4] #p(additive term)
add.results[i,5]<-summary(reg.add.annual)$adj.r.squared #r2
add.results[i,6]<-as.numeric(AICc(reg.add.annual)) #AICc

# interaction model
int.results[i,1]<-summary(reg.int.annual)$coef[2,1] #slope
int.results[i,2]<-summary(reg.int.annual)$coef[1,1] #intercept
int.results[i,3]<-summary(reg.int.annual)$coef[2,4] #p(slope)
int.results[i,4]<-summary(reg.int.annual)$coef[4,4] #p(interact term)
int.results[i,5]<-summary(reg.int.annual)$adj.r.squared #r2
int.results[i,6]<-as.numeric(AICc(reg.int.annual)) #AICc


## cold
reg.linear.cold <- lm(log(AAM)~ColdTemp, data = sub)
reg.add.cold <- lm(log(AAM)~ColdTemp+Condition, data = sub)
reg.int.cold <- lm(log(AAM)~ColdTemp*Condition, data = sub)

# simple linear model
linear.results[i,7]<-summary(reg.linear.cold)$coef[2,1] #slope
linear.results[i,8]<-summary(reg.linear.cold)$coef[1,1] #intercept
linear.results[i,9]<-summary(reg.linear.cold)$coef[2,4] #p-value
linear.results[i,10]<-0 #blank
linear.results[i,11]<-summary(reg.linear.cold)$adj.r.squared #r2
linear.results[i,12]<-as.numeric(AICc(reg.linear.cold)) #AICc

# linear additive model
add.results[i,7]<-summary(reg.add.cold)$coef[2,1] #slope
add.results[i,8]<-summary(reg.add.cold)$coef[1,1] #intercept
add.results[i,9]<-summary(reg.add.cold)$coef[2,4] #p(slope)
add.results[i,10]<-summary(reg.add.cold)$coef[3,4] #p(additive term)
add.results[i,11]<-summary(reg.add.cold)$adj.r.squared #r2
add.results[i,12]<-as.numeric(AICc(reg.add.cold)) #AICc

# interaction model
int.results[i,7]<-summary(reg.int.cold)$coef[2,1] #slope
int.results[i,8]<-summary(reg.int.cold)$coef[1,1] #intercept
```

```
  int.results[i,9]<-summary(reg.int.cold)$coef[2,4] #p(slope)
  int.results[i,10]<-summary(reg.int.cold)$coef[4,4] #p(interact term)
  int.results[i,11]<-summary(reg.int.cold)$adj.r.squared #r2
  int.results[i,12]<-as.numeric(AICc(reg.int.cold)) #AICc


  ## gdd0
  reg.linear.gdd <- lm(log(AAM)~AnnualDD, data = sub)
  reg.add.gdd <- lm(log(AAM)~AnnualDD+Condition, data = sub)
  reg.int.gdd <- lm(log(AAM)~AnnualDD*Condition, data = sub)

  # simple linear model
  linear.results[i,13]<-summary(reg.linear.gdd)$coef[2,1] #slope
  linear.results[i,14]<-summary(reg.linear.gdd)$coef[1,1] #intercept
  linear.results[i,15]<-summary(reg.linear.gdd)$coef[2,4] #p-value
  linear.results[i,16]<-0 #blank
  linear.results[i,17]<-summary(reg.linear.gdd)$adj.r.squared #r2
  linear.results[i,18]<-as.numeric(AICc(reg.linear.gdd)) #AICc

  # linear additive model
  add.results[i,13]<-summary(reg.add.gdd)$coef[2,1] #slope
  add.results[i,14]<-summary(reg.add.gdd)$coef[1,1] #intercept
  add.results[i,15]<-summary(reg.add.gdd)$coef[2,4] #p(slope)
  add.results[i,16]<-summary(reg.add.gdd)$coef[3,4] #p(additive term)
  add.results[i,17]<-summary(reg.add.gdd)$adj.r.squared #r2
  add.results[i,18]<-as.numeric(AICc(reg.add.gdd)) #AICc

  # interaction model
  int.results[i,13]<-summary(reg.int.gdd)$coef[2,1] #slope
  int.results[i,14]<-summary(reg.int.gdd)$coef[1,1] #intercept
  int.results[i,15]<-summary(reg.int.gdd)$coef[2,4] #p(slope)
  int.results[i,16]<-summary(reg.int.gdd)$coef[4,4] #p(interact term)
  int.results[i,17]<-summary(reg.int.gdd)$adj.r.squared #r2
  int.results[i,18]<-as.numeric(AICc(reg.int.gdd)) #AICc
}
```

Check the slopes and additive/interaction terms

Now we need to:

1. Check if the additive or interaction term is significant.
2. Check if the relationship is significant between age at maturity and temperature (significant slope).

```
## Additive/interaction term
# annual
mean(unique(add.results[,"additive term.a"]))
```

```
## [1] 0.8165643
```

```
table(add.results[,"additive term.a"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
mean(unique(int.results[,"interaction term.a"]))
```

```
## [1] 0.6655419
```

```r
table(int.results[,"interaction term.a"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
# cold
mean(unique(add.results[,"additive term.c"]))
```

```
## [1] 0.6098918
```

```r
table(add.results[,"additive term.c"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
mean(unique(int.results[,"interaction term.c"]))
```

```
## [1] 0.7956028
```

```r
table(int.results[,"interaction term.c"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
# gdd
mean(unique(add.results[,"additive term.g"]))
```

```
## [1] 0.8313756
```

```r
table(add.results[,"additive term.g"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
mean(unique(int.results[,"interaction term.g"]))
```

```
## [1] 0.3371505
```

```r
table(int.results[,"interaction term.g"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
## Slope
# annual
mean(unique(linear.results[,"p for slope.a"]))
```

```
## [1] 1.49252e-07
```

```r
mean(unique(add.results[,"p for slope.a"]))
```

```
## [1] 4.819865e-07
```

```r
mean(unique(int.results[,"p for slope.a"]))
```

```
## [1] 6.052778e-05
```

```r
# cold
mean(unique(linear.results[,"p for slope.c"]))
```

```
## [1] 3.959425e-07
```

```r
mean(unique(add.results[,"p for slope.c"]))
```

```
## [1] 1.142224e-06
```

```r
mean(unique(int.results[,"p for slope.c"]))
```

```
## [1] 0.0001098977
```

```r
# gdd
mean(unique(linear.results[,"p for slope.g"]))
```

```
## [1] 6.246775e-08
```

```r
mean(unique(add.results[,"p for slope.g"]))
```

```
## [1] 2.062617e-07
```

```r
mean(unique(int.results[,"p for slope.g"]))
```

```
## [1] 1.281651e-05
```

- No significant additive or interaction term.
- Significant slope.

Compare the R2

```r
# annual
r2annual <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,5])),
         mean(unique(add.results[,5])),
         mean(unique(int.results[,5])))
)
kable(r2annual)
```

Model	R2
Simple linear	0.5399911
Linear additive	0.5279294
Interaction	0.5176888

```r
# cold
r2cold <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,11])),
         mean(unique(add.results[,11])),
         mean(unique(int.results[,11])))
)
kable(r2cold)
```

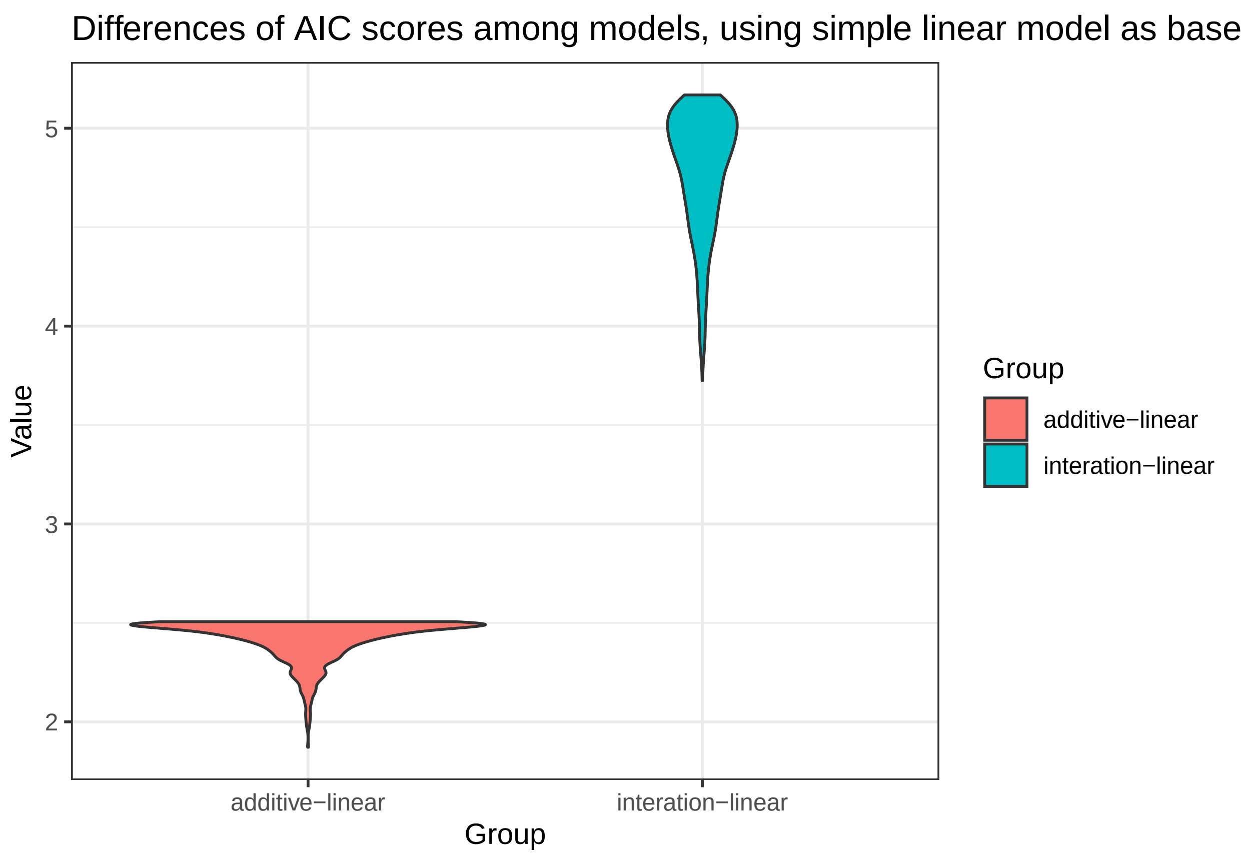
Model	R2
Simple linear	0.5123108
Linear additive	0.5032018
Interaction	0.4901441

```r
# gdd
r2gdd <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,17])),
         mean(unique(add.results[,17])),
         mean(unique(int.results[,17])))
)
kable(r2gdd)
```

Model	R2
Simple linear	0.5574711
Linear additive	0.5457403
Interaction	0.5471320

Compare AICs for annual

```r
## Look at the distribution of the differences between AIC scores
# Calculate the differences of AIC values
aic.asian <- matrix(NA,1000,2) # store the differences in AIC values
aic.asian[,1] <- add.results[,6] - linear.results[,6]
aic.asian[,2] <- int.results[,6] - linear.results[,6]

# Create a data frame
data <- as.data.frame(aic.asian)
colnames(data) <- c("additive-linear","interation-linear")

# Convert to long data format
data_long <- data %>%
  pivot_longer(cols = everything(), names_to = "Group", values_to = "Value")

# Define the desired order of groups
desired_order <- c("additive-linear","interation-linear")

# Convert "Group" to a factor with desired order
data_long$Group <- factor(data_long$Group, levels = desired_order)

# Violin plot
ggplot(data_long, aes(x = Group, y = Value, fill = Group))+
  geom_violin()+
  labs(title = "Differences of AIC scores among models, using simple linear model as base. Annual Temp")
  theme_bw()
```




```r
## Check the AICc scores and akaike weights in 1000 iterations
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,6], add.results[i,6], int.results[i,6])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##        V1               V2                V3
##  Min.   :0.663   Min.   :0.2000   Min.   :0.05500
##  1st Qu.:0.712   1st Qu.:0.2090   1st Qu.:0.05900
##  Median :0.722   Median :0.2120   Median :0.06400
##  Mean   :0.719   Mean   :0.2147   Mean   :0.06634
```

```
##   3rd Qu.:0.729   3rd Qu.:0.2180   3rd Qu.:0.07100
##   Max.   :0.735   Max.   :0.2600   Max.   :0.10600
```

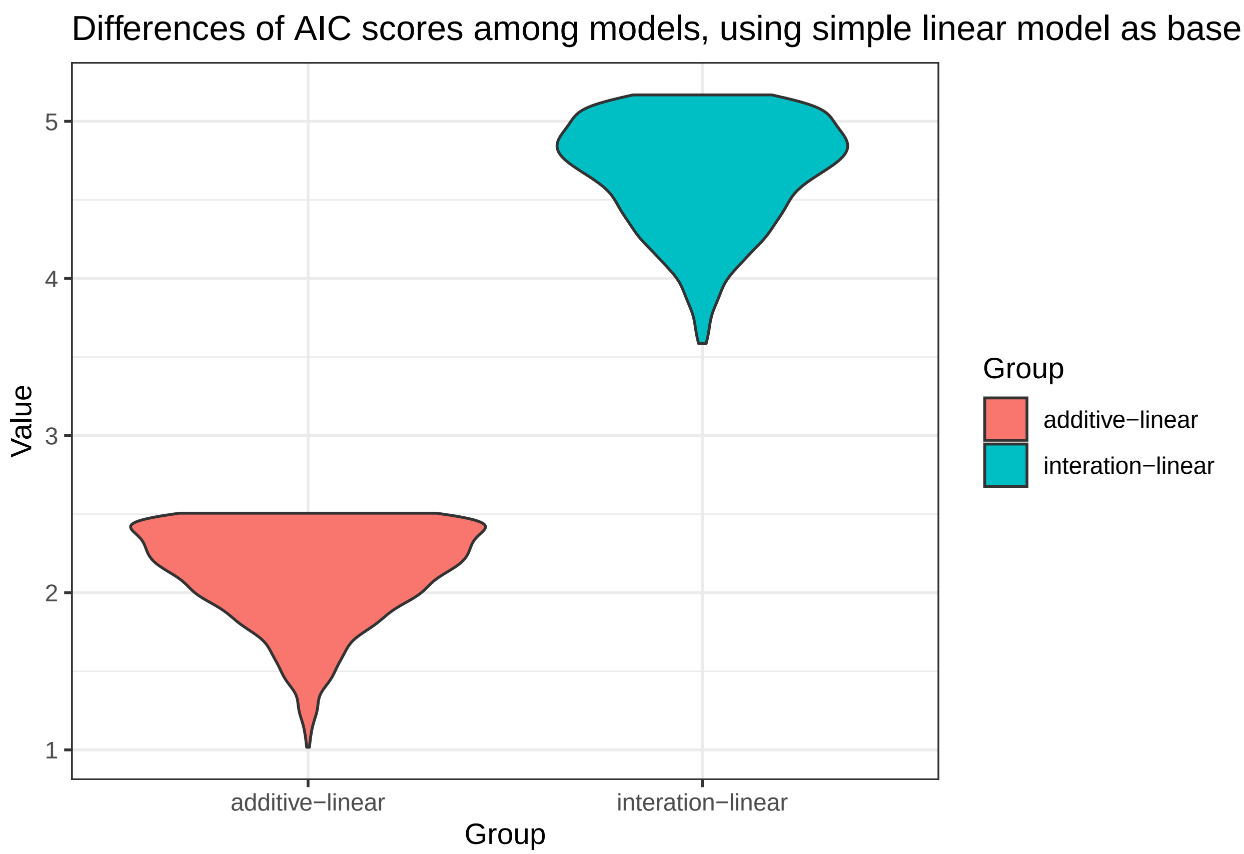
```r
table(count)
```

```
## count
##   1
## 995
```

- When looking at each iteration, we saw that around 85% of the times the simple linear model is the best.
- Akaike weight is about 60% for simple linear model.

Compare AICs for the cold

```r
# Calculate the differences of AIC values
aic.asian <- matrix(NA,1000,2) # store the differences in AIC values
aic.asian[,1] <- add.results[,12] - linear.results[,12]
aic.asian[,2] <- int.results[,12] - linear.results[,12]

# Look at the distribution of differences
# Create a data frame
data <- as.data.frame(aic.asian)
colnames(data) <- c("additive-linear","interation-linear")

# Convert to long data format
data_long <- data %>%
  pivot_longer(cols = everything(), names_to = "Group", values_to = "Value")

# Define the desired order of groups
desired_order <- c("additive-linear","interation-linear")

# Convert "Group" to a factor with desired order
data_long$Group <- factor(data_long$Group, levels = desired_order)

# Violin plot
ggplot(data_long, aes(x = Group, y = Value, fill = Group))+
  geom_violin()+
  labs(title = "Differences of AIC scores among models, using simple linear model as base. Cold Temp")+
  theme_bw()
```




```r
## Check the AICc scores and akaike weights in 1000 iterations
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,12], add.results[i,12], int.results[i,12])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##        V1               V2               V3
##  Min.   :0.5680   Min.   :0.2070   Min.   :0.0550
##  1st Qu.:0.6740   1st Qu.:0.2190   1st Qu.:0.0610
##  Median :0.7000   Median :0.2340   Median :0.0650
##  Mean   :0.6929   Mean   :0.2397   Mean   :0.0674
```

```
##   3rd Qu.:0.7200   3rd Qu.:0.2540   3rd Qu.:0.0730
##   Max.   :0.7350   Max.   :0.3410   Max.   :0.0980
```

```r
table(count)
```

```
## count
##   1
## 707
```

- When looking at each iteration, we saw that around 70% of the times the simple linear model is the best.
- Akaike weight is around 57% for simple linear models.

Compare among the temperatures (SIMPLE MODEL ONLY)

```r
## Compare the R2 for three temperatures
r2 <- data.frame(
  Temperature = c("Annual", "Cold", "GDD0"),
  R2 = c(mean(unique(linear.results[,"r2.a"])),
         mean(unique(linear.results[,"r2.c"])),
         mean(unique(linear.results[,"r2.g"])))
)
kable(r2)
```

Temperature	R2
Annual	0.5399911
Cold	0.5123108
GDD0	0.5574711

```r
## Check the AICc scores and akaike weights in ONLY LINEAR MODEL
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,6], linear.results[i,12],
                 linear.results[i,18])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##         V1                V2                 V3
##   Min.   :0.1160   Min.   :0.03200   Min.   :0.3160
##   1st Qu.:0.2360   1st Qu.:0.07175   1st Qu.:0.5220
```

```
##  Median :0.2900   Median :0.09400   Median :0.6090
##  Mean   :0.2989   Mean   :0.10147   Mean   :0.5996
##  3rd Qu.:0.3570   3rd Qu.:0.12500   3rd Qu.:0.6820
##  Max.   :0.5280   Max.   :0.28700   Max.   :0.8370
```

```
table(count)
```

```
## count
##   3
## 302
```

- With the simple linear model, around 30% of the time when using GDD is preferred over using ColdTemp.

Two conditions separated

```
## Separate the two conditions
asian.natural <- asian.carp.clean[asian.carp.clean$Condition == "natural",]
asian.artificial <- asian.carp.clean[asian.carp.clean$Condition == "artificial",]

# Check sp
table(asian.natural$Code_Str)
```

```
##
## A B C D E F G H I J K L M N O P Q
## 4 1 1 1 1 2 1 1 3 4 1 2 2 1 1 1 1
```

```
table(asian.artificial$Code_Str)
```

```
##
## AA AB AC AD AE AF AG AH AI AJ AK AL  R  S  T  U  V  W  X  Y  Z
##  3  3  2  1  1  1  3  1  3  1  3  2  3  1  1  3  3  1  1  1  1
```

```
# store the results
temp.natural <- matrix(NA, 1000, 3)
temp.artificial <- matrix(NA, 1000, 3)


## Removing artificial for 1000 iterations
for(i in 1:1000){

  ## natural
  sub.n <- asian.natural %>% group_by(Code_Str) %>% sample_n(size=1)

  # models
  reg.annual <- lm(log(AAM)~AnnualTemp, data = sub)
  reg.cold <- lm(log(AAM)~ColdTemp, data = sub)
  reg.gdd <- lm(log(AAM)~AnnualDD, data = sub)

  # r2
  temp.natural[i,1]<-summary(reg.annual)$adj.r.squared
  temp.natural[i,2]<-summary(reg.cold)$adj.r.squared
  temp.natural[i,3]<-summary(reg.gdd)$adj.r.squared


  ## artificial
  sub.a <- asian.artificial %>% group_by(Code_Str) %>% sample_n(size=1)
```

```
  # models
  reg.annual <- lm(log(AAM)~AnnualTemp, data = sub.a)
  reg.cold <- lm(log(AAM)~ColdTemp, data = sub.a)
  reg.gdd <- lm(log(AAM)~AnnualDD, data = sub.a)

  # r2
  temp.artificial[i,1]<-summary(reg.annual)$adj.r.squared
  temp.artificial[i,2]<-summary(reg.cold)$adj.r.squared
  temp.artificial[i,3]<-summary(reg.gdd)$adj.r.squared
}


## Compare the R2 for three temperatures
r2 <- data.frame(
  Temperature = c("Annual", "Cold", "GDD0"),
  Natural.R2 = c(mean(unique(temp.natural[,1])),
               mean(unique(temp.natural[,2])),
               mean(unique(temp.natural[,3]))),
  Artificial.R2 = c(mean(unique(temp.artificial[,1])),
                  mean(unique(temp.artificial[,2])),
                  mean(unique(temp.artificial[,3])))
)
kable(r2)
```

Temperature	Natural.R2	Artificial.R2
Annual	0.5163437	0.5261544
Cold	0.4908534	0.5090521
GDD0	0.5417452	0.6222562

- Natural and Artificial have similar R2 values when separated.

SECTION 3: Grass carp

Data cleaning and matrices for results

```
Grass.clean <- Grass %>%
  filter(Condition %in% c("natural", "artificial"))

table(Grass.clean$Code_Str)
```

```
##
##  A AA AB AC AD AF AG AI AJ AK AL  B  E  F  G  I  J  L  M  N  O  P  R  S  T  U
##  2  1  2  1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  1  1  1  1  1  1  1  1
##  V  W  Y  Z
##  1  1  1  1
```

```
# Simple linear model - slope, intercept, p, blank, r2, AICc
linear.results <- matrix(NA,1000,18)
colnames(linear.results) <- c("slope.a",
                              "intercept.a",
                              "p for slope.a",
                              "blank.a",
                              "r2.a",
```

```r
                                  "AICc.a",
                                  "slope.c",
                                  "intercept.c",
                                  "p for slope.c",
                                  "blank.c",
                                  "r2.c",
                                  "AICc.c",
                                  "slope.g",
                                  "intercept.g",
                                  "p for slope.g",
                                  "blank.g",
                                  "r2.g",
                                  "AICc.g")
# Linear additive model - slope, intercept, p(slope), p(additive), r2, AICc
add.results <- matrix(NA,1000,18)
colnames(add.results) <- c("slope.a",
                                  "intercept.a",
                                  "p for slope.a",
                                  "additive term.a",
                                  "r2.a",
                                  "AICc.a",
                                  "slope.c",
                                  "intercept.c",
                                  "p for slope.c",
                                  "additive term.c",
                                  "r2.c",
                                  "AICc.c",
                                  "slope.g",
                                  "intercept.g",
                                  "p for slope.g",
                                  "additive term.g",
                                  "r2.g",
                                  "AICc.g")
# Interaction model - slope, intercept, p(slope), p(interaction), r2, AICc
int.results <- matrix(NA,1000,18)
colnames(int.results) <- c("slope.a",
                                  "intercept.a",
                                  "p for slope.a",
                                  "interaction term.a",
                                  "r2.a",
                                  "AICc.a",
                                  "slope.c",
                                  "intercept.c",
                                  "p for slope.c",
                                  "interaction term.c",
                                  "r2.c",
                                  "AICc.c",
                                  "slope.g",
                                  "intercept.g",
                                  "p for slope.g",
                                  "interaction term.g",
                                  "r2.g",
                                  "AICc.g")
```

Define the models

```
# For 1000 iterations
for(i in 1:1000){
  sub <- Grass.clean %>% group_by(Code_Str) %>% sample_n(size=1)

  ## annual
  reg.linear.annual <- lm(log(AAM)~AnnualTemp, data = sub)
  reg.add.annual <- lm(log(AAM)~AnnualTemp+Condition, data = sub)
  reg.int.annual <- lm(log(AAM)~AnnualTemp*Condition, data = sub)

  # simple linear model
  linear.results[i,1]<-summary(reg.linear.annual)$coef[2,1] #slope
  linear.results[i,2]<-summary(reg.linear.annual)$coef[1,1] #intercept
  linear.results[i,3]<-summary(reg.linear.annual)$coef[2,4] #p-value
  linear.results[i,4]<-0 #blank
  linear.results[i,5]<-summary(reg.linear.annual)$adj.r.squared #r2
  linear.results[i,6]<-as.numeric(AICc(reg.linear.annual)) #AICc

  # linear additive model
  add.results[i,1]<-summary(reg.add.annual)$coef[2,1] #slope
  add.results[i,2]<-summary(reg.add.annual)$coef[1,1] #intercept
  add.results[i,3]<-summary(reg.add.annual)$coef[2,4] #p(slope)
  add.results[i,4]<-summary(reg.add.annual)$coef[3,4] #p(additive term)
  add.results[i,5]<-summary(reg.add.annual)$adj.r.squared #r2
  add.results[i,6]<-as.numeric(AICc(reg.add.annual)) #AICc

  # interaction model
  int.results[i,1]<-summary(reg.int.annual)$coef[2,1] #slope
  int.results[i,2]<-summary(reg.int.annual)$coef[1,1] #intercept
  int.results[i,3]<-summary(reg.int.annual)$coef[2,4] #p(slope)
  int.results[i,4]<-summary(reg.int.annual)$coef[4,4] #p(interact term)
  int.results[i,5]<-summary(reg.int.annual)$adj.r.squared #r2
  int.results[i,6]<-as.numeric(AICc(reg.int.annual)) #AICc


  ## cold
  reg.linear.cold <- lm(log(AAM)~ColdTemp, data = sub)
  reg.add.cold <- lm(log(AAM)~ColdTemp+Condition, data = sub)
  reg.int.cold <- lm(log(AAM)~ColdTemp*Condition, data = sub)

  # simple linear model
  linear.results[i,7]<-summary(reg.linear.cold)$coef[2,1] #slope
  linear.results[i,8]<-summary(reg.linear.cold)$coef[1,1] #intercept
  linear.results[i,9]<-summary(reg.linear.cold)$coef[2,4] #p-value
  linear.results[i,10]<-0 #blank
  linear.results[i,11]<-summary(reg.linear.cold)$adj.r.squared #r2
  linear.results[i,12]<-as.numeric(AICc(reg.linear.cold)) #AICc

  # linear additive model
  add.results[i,7]<-summary(reg.add.cold)$coef[2,1] #slope
  add.results[i,8]<-summary(reg.add.cold)$coef[1,1] #intercept
  add.results[i,9]<-summary(reg.add.cold)$coef[2,4] #p(slope)
  add.results[i,10]<-summary(reg.add.cold)$coef[3,4] #p(additive term)
```

```r
  add.results[i,11]<-summary(reg.add.cold)$adj.r.squared #r2
  add.results[i,12]<-as.numeric(AICc(reg.add.cold)) #AICc

  # interaction model
  int.results[i,7]<-summary(reg.int.cold)$coef[2,1] #slope
  int.results[i,8]<-summary(reg.int.cold)$coef[1,1] #intercept
  int.results[i,9]<-summary(reg.int.cold)$coef[2,4] #p(slope)
  int.results[i,10]<-summary(reg.int.cold)$coef[4,4] #p(interact term)
  int.results[i,11]<-summary(reg.int.cold)$adj.r.squared #r2
  int.results[i,12]<-as.numeric(AICc(reg.int.cold)) #AICc


  ## gdd0
  reg.linear.gdd <- lm(log(AAM)~AnnualDD, data = sub)
  reg.add.gdd <- lm(log(AAM)~AnnualDD+Condition, data = sub)
  reg.int.gdd <- lm(log(AAM)~AnnualDD*Condition, data = sub)

  # simple linear model
  linear.results[i,13]<-summary(reg.linear.gdd)$coef[2,1] #slope
  linear.results[i,14]<-summary(reg.linear.gdd)$coef[1,1] #intercept
  linear.results[i,15]<-summary(reg.linear.gdd)$coef[2,4] #p-value
  linear.results[i,16]<-0 #blank
  linear.results[i,17]<-summary(reg.linear.gdd)$adj.r.squared #r2
  linear.results[i,18]<-as.numeric(AICc(reg.linear.gdd)) #AICc

  # linear additive model
  add.results[i,13]<-summary(reg.add.gdd)$coef[2,1] #slope
  add.results[i,14]<-summary(reg.add.gdd)$coef[1,1] #intercept
  add.results[i,15]<-summary(reg.add.gdd)$coef[2,4] #p(slope)
  add.results[i,16]<-summary(reg.add.gdd)$coef[3,4] #p(additive term)
  add.results[i,17]<-summary(reg.add.gdd)$adj.r.squared #r2
  add.results[i,18]<-as.numeric(AICc(reg.add.gdd)) #AICc

  # interaction model
  int.results[i,13]<-summary(reg.int.gdd)$coef[2,1] #slope
  int.results[i,14]<-summary(reg.int.gdd)$coef[1,1] #intercept
  int.results[i,15]<-summary(reg.int.gdd)$coef[2,4] #p(slope)
  int.results[i,16]<-summary(reg.int.gdd)$coef[4,4] #p(interact term)
  int.results[i,17]<-summary(reg.int.gdd)$adj.r.squared #r2
  int.results[i,18]<-as.numeric(AICc(reg.int.gdd)) #AICc
}
```

Check the slopes and additive/interaction terms

Now we need to:

1. Check if the additive or interaction term is significant.
2. Check if the relationship is significant between age at maturity and temperature (significant slope).

```r
## Additive/interaction term
# annual
mean(unique(add.results[,"additive term.a"]))
```

```
## [1] 0.4175639
```

```r
table(add.results[,"additive term.a"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
mean(unique(int.results[,"interaction term.a"]))
```

```
## [1] 0.8550577
```

```r
table(int.results[,"interaction term.a"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
# cold
mean(unique(add.results[,"additive term.c"]))
```

```
## [1] 0.7727948
```

```r
table(add.results[,"additive term.c"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
mean(unique(int.results[,"interaction term.c"]))
```

```
## [1] 0.8573869
```

```r
table(int.results[,"interaction term.c"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
# gdd
mean(unique(add.results[,"additive term.g"]))
```

```
## [1] 0.3210204
```

```r
table(add.results[,"additive term.g"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
mean(unique(int.results[,"interaction term.g"]))
```

```
## [1] 0.6886118
```

```r
table(int.results[,"interaction term.g"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
## Slope
# annual
mean(unique(linear.results[,"p for slope.a"]))
```

```
## [1] 1.123214e-07
```

```r
mean(unique(add.results[,"p for slope.a"]))
```

```
## [1] 1.039898e-06
```

```r
mean(unique(int.results[,"p for slope.a"]))
```

```
## [1] 3.097377e-05
```

```r
# cold
mean(unique(linear.results[,"p for slope.c"]))
```

```
## [1] 8.397821e-08
```

```r
mean(unique(add.results[,"p for slope.c"]))
```

```
## [1] 1.12254e-06
```

```r
mean(unique(int.results[,"p for slope.c"]))
```

```
## [1] 3.38067e-05
```

```r
# gdd
mean(unique(linear.results[,"p for slope.g"]))
```

```
## [1] 3.814636e-08
```

```r
mean(unique(add.results[,"p for slope.g"]))
```

```
## [1] 3.021915e-07
```

```r
mean(unique(int.results[,"p for slope.g"]))
```

```
## [1] 6.901261e-06
```

- No significant additive or interaction term.
- Significant slope.

Compare the R2

```r
# annual
r2annual <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,5])),
         mean(unique(add.results[,5])),
         mean(unique(int.results[,5])))
)
kable(r2annual)
```

Model	R2
Simple linear	0.6345688
Linear additive	0.6319235
Interaction	0.6184244

```r
# cold
r2cold <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
```

```r
  R2 = c(mean(unique(linear.results[,11])),
         mean(unique(add.results[,11])),
         mean(unique(int.results[,11])))
)
kable(r2cold)
```

Model	R2
Simple linear	0.6397366
Linear additive	0.6284060
Interaction	0.6148537

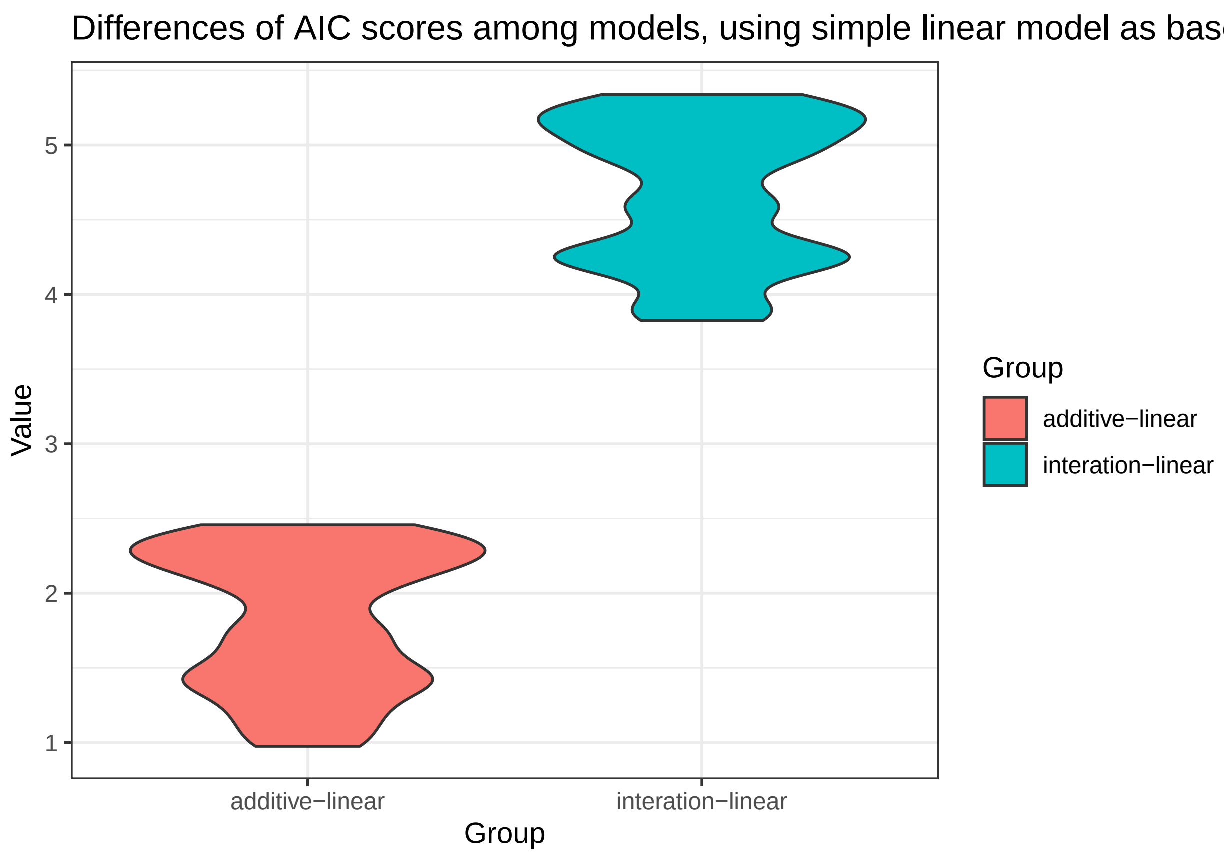
```r
# gdd
r2gdd <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,17])),
         mean(unique(add.results[,17])),
         mean(unique(int.results[,17])))
)
kable(r2gdd)
```

Model	R2
Simple linear	0.6589762
Linear additive	0.6607169
Interaction	0.6506497

Compare AICs for annual

```r
## Look at the distribution of the differences between AIC scores
# Calculate the differences of AIC values
aic.grass <- matrix(NA,1000,2) # store the differences in AIC values
aic.grass[,1] <- add.results[,6] - linear.results[,6]
aic.grass[,2] <- int.results[,6] - linear.results[,6]

# Create a data frame
data <- as.data.frame(aic.grass)
colnames(data) <- c("additive-linear","interation-linear")

# Convert to long data format
data_long <- data %>%
  pivot_longer(cols = everything(), names_to = "Group", values_to = "Value")

# Define the desired order of groups
desired_order <- c("additive-linear","interation-linear")

# Convert "Group" to a factor with desired order
data_long$Group <- factor(data_long$Group, levels = desired_order)

# Violin plot
ggplot(data_long, aes(x = Group, y = Value, fill = Group))+
  geom_violin()+
```

```r
  labs(title = "Differences of AIC scores among models, using simple linear model as base. Annual Temp")
  theme_bw()
```




```r
## Check the AICc scores and akaike weights in 1000 iterations
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,6], add.results[i,6], int.results[i,6])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##         V1               V2               V3
##  Min.   :0.5680   Min.   :0.2150   Min.   :0.05100
```

```
##  1st Qu.:0.6210   1st Qu.:0.2310   1st Qu.:0.05500
##  Median :0.6630   Median :0.2710   Median :0.06600
##  Mean   :0.6645   Mean   :0.2705   Mean   :0.06521
##  3rd Qu.:0.7140   3rd Qu.:0.3060   3rd Qu.:0.07500
##  Max.   :0.7340   Max.   :0.3490   Max.   :0.08400
```

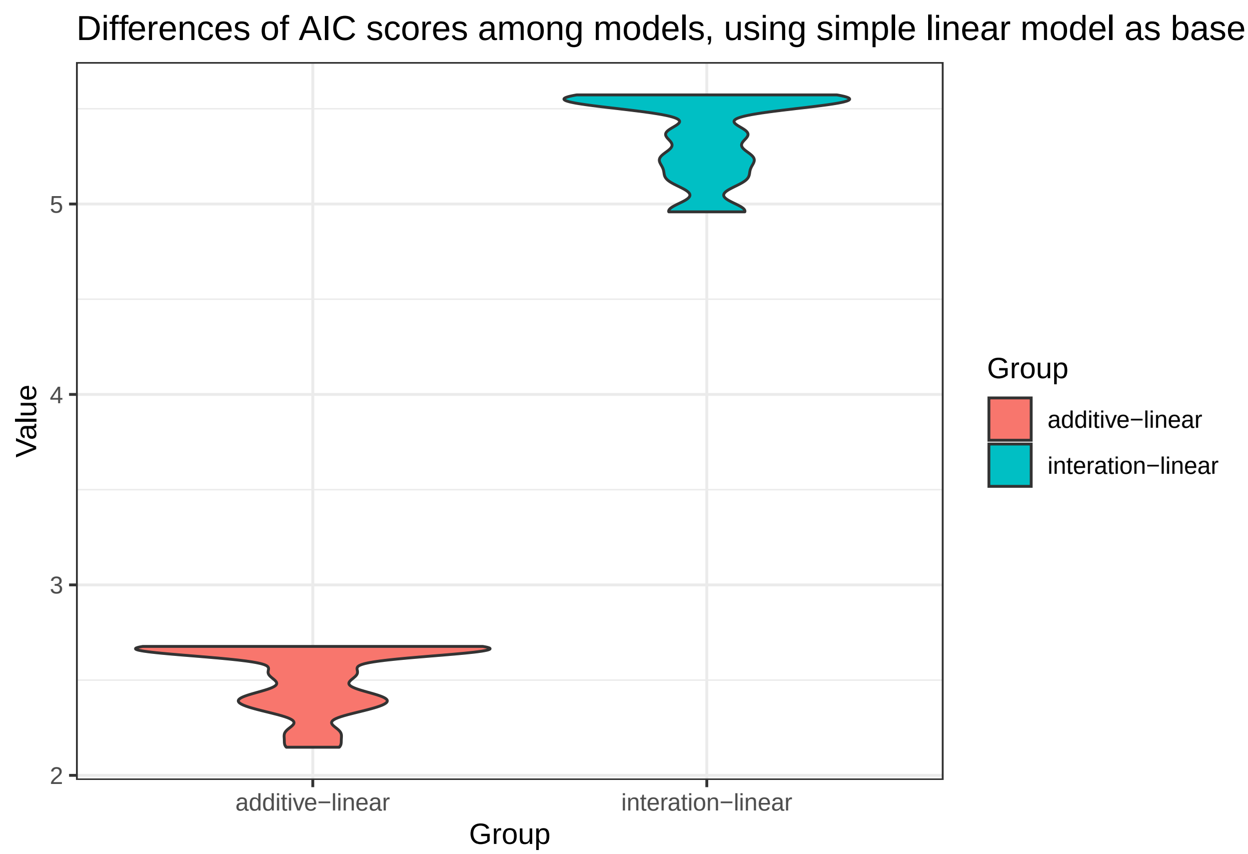
```r
table(count)
```

```
## count
##   1
## 437
```

- For grass carp, AIC for simple linear model was always smaller than the additive and interaction model, but within two units, and significantly smaller than the grouped-specific model (greater than 2 units).
- 46% of the times when the simple linear model perfroms better.

Compare AICs for the cold

```r
# Calculate the differences of AIC values
aic.grass <- matrix(NA,1000,2) # store the differences in AIC values
aic.grass[,1] <- add.results[,12] - linear.results[,12]
aic.grass[,2] <- int.results[,12] - linear.results[,12]

# Look at the distribution of differences
# Create a data frame
data <- as.data.frame(aic.grass)
colnames(data) <- c("additive-linear","interation-linear")

# Convert to long data format
data_long <- data %>%
  pivot_longer(cols = everything(), names_to = "Group", values_to = "Value")

# Define the desired order of groups
desired_order <- c("additive-linear","interation-linear")

# Convert "Group" to a factor with desired order
data_long$Group <- factor(data_long$Group, levels = desired_order)

# Violin plot
ggplot(data_long, aes(x = Group, y = Value, fill = Group))+
  geom_violin()+
  labs(title = "Differences of AIC scores among models, using simple linear model as base. Cold Temp")+
  theme_bw()
```




```
## Check the AICc scores and akaike weights in 1000 iterations
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,12], add.results[i,12], int.results[i,12])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##         V1                V2                V3
##  Min.   :0.7020   Min.   :0.1980   Min.   :0.04700
##  1st Qu.:0.7240   1st Qu.:0.1990   1st Qu.:0.04700
##  Median :0.7420   Median :0.2070   Median :0.05000
##  Mean   :0.7387   Mean   :0.2104   Mean   :0.05088
```

```
##  3rd Qu.:0.7540   3rd Qu.:0.2210   3rd Qu.:0.05300
##  Max.   :0.7550   Max.   :0.2400   Max.   :0.05900
```

```
table(count)
```

```
## count
##    1
## 1000
```

- Same conclusion as AnnualTemp. 100% of the times when the simple model performs better.

Compare among three temperatures

```
## Compare the R2 for three temperatures
r2 <- data.frame(
  Temperature = c("Annual", "Cold", "GDD0"),
  R2 = c(mean(unique(linear.results[,"r2.a"])),
         mean(unique(linear.results[,"r2.c"])),
         mean(unique(linear.results[,"r2.g"])))
)
kable(r2)
```

Temperature	R2
Annual	0.6345688
Cold	0.6397366
GDD0	0.6589762

```
## Check the AICc scores and akaike weights in ONLY LINEAR MODEL
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,6], linear.results[i,12],
                 linear.results[i,18])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##        V1               V2               V3
##  Min.   :0.1430   Min.   :0.1410   Min.   :0.4120
##  1st Qu.:0.1570   1st Qu.:0.1820   1st Qu.:0.5150
##  Median :0.1810   Median :0.2300   Median :0.5570
##  Mean   :0.1982   Mean   :0.2463   Mean   :0.5553
```

```
##  3rd Qu.:0.2330   3rd Qu.:0.2830   3rd Qu.:0.5950
##  Max.   :0.2660   Max.   :0.3750   Max.   :0.6810
```

```
table(count)
```

```
## count
##   3
## 249
```

- Cold temperature did not show any preference over Annual temperature. No significant differences.

Two conditions separated

```
## Separate the two conditions
grass.natural <- Grass.clean[Grass.clean$Condition == "natural",]
grass.artificial <- Grass.clean[Grass.clean$Condition == "artificial",]

# Check sp
table(grass.natural$Code_Str)
```

```
##
## A B E F G I J L M N O P
## 2 1 1 1 1 2 2 2 1 1 1 1
```

```
table(grass.artificial$Code_Str)
```

```
##
## AA AB AC AD AF AG AI AJ AK AL  R  S  T  U  V  W  Y  Z
##  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
# store the results
temp.natural <- matrix(NA, 1000, 3)
temp.artificial <- matrix(NA, 1000, 3)


## Removing artificial for 1000 iterations
for(i in 1:1000){

  ## natural
  sub.n <- grass.natural %>% group_by(Code_Str) %>% sample_n(size=1)

  # models
  reg.annual <- lm(log(AAM)~AnnualTemp, data = sub)
  reg.cold <- lm(log(AAM)~ColdTemp, data = sub)
  reg.gdd <- lm(log(AAM)~AnnualDD, data = sub)

  # r2
  temp.natural[i,1]<-summary(reg.annual)$adj.r.squared
  temp.natural[i,2]<-summary(reg.cold)$adj.r.squared
  temp.natural[i,3]<-summary(reg.gdd)$adj.r.squared


  ## artificial
  sub.a <- grass.artificial %>% group_by(Code_Str) %>% sample_n(size=1)

  # models
  reg.annual <- lm(log(AAM)~AnnualTemp, data = sub.a)
```

```
  reg.cold <- lm(log(AAM)~ColdTemp, data = sub.a)
  reg.gdd <- lm(log(AAM)~AnnualDD, data = sub.a)

  # r2
  temp.artificial[i,1]<-summary(reg.annual)$adj.r.squared
  temp.artificial[i,2]<-summary(reg.cold)$adj.r.squared
  temp.artificial[i,3]<-summary(reg.gdd)$adj.r.squared
}


## Compare the R2 for three temperatures
r2 <- data.frame(
  Temperature = c("Annual", "Cold", "GDD0"),
  Natural.R2 = c(mean(unique(temp.natural[,1])),
                 mean(unique(temp.natural[,2])),
                 mean(unique(temp.natural[,3]))),
  Artificial.R2 = c(mean(unique(temp.artificial[,1])),
                    mean(unique(temp.artificial[,2])),
                    mean(unique(temp.artificial[,3])))
)
kable(r2)
```

Temperature	Natural.R2	Artificial.R2
Annual	0.6047104	0.5415039
Cold	0.6141793	0.5389077
GDD0	0.6340967	0.6225748

SECTION 4: Bighead and silver carp

Data cleaning and matrices for results

```
Big.sil.clean <- Big.sil %>%
  filter(Condition %in% c("natural", "artificial"))

table(Big.sil.clean$Code)
```

```
##
##  A AC AD AF AG AI AL AP  B  C  D  E  H  J  M  N  O  S  Z
##  4  1  1  2  2  1  1  1  3  1  2  2  1  1  2  1  2  2  2
```

```
# Simple linear model - slope, intercept, p, blank, r2, AICc
linear.results <- matrix(NA,1000,18)
colnames(linear.results) <- c("slope.a",
                                   "intercept.a",
                                   "p for slope.a",
                                   "blank.a",
                                   "r2.a",
                                   "AICc.a",
                                   "slope.c",
                                   "intercept.c",
                                   "p for slope.c",
                                   "blank.c",
                                   "r2.c",
```

```r
                                    "AICc.c",
                                    "slope.g",
                                    "intercept.g",
                                    "p for slope.g",
                                    "blank.g",
                                    "r2.g",
                                    "AICc.g")
# Linear additive model - slope, intercept, p(slope), p(additive), r2, AICc
add.results <- matrix(NA,1000,18)
colnames(add.results) <- c("slope.a",
                                    "intercept.a",
                                    "p for slope.a",
                                    "additive term.a",
                                    "r2.a",
                                    "AICc.a",
                                    "slope.c",
                                    "intercept.c",
                                    "p for slope.c",
                                    "additive term.c",
                                    "r2.c",
                                    "AICc.c",
                                    "slope.g",
                                    "intercept.g",
                                    "p for slope.g",
                                    "additive term.g",
                                    "r2.g",
                                    "AICc.g")
# Interaction model - slope, intercept, p(slope), p(interaction), r2, AICc
int.results <- matrix(NA,1000,18)
colnames(int.results) <- c("slope.a",
                                    "intercept.a",
                                    "p for slope.a",
                                    "interaction term.a",
                                    "r2.a",
                                    "AICc.a",
                                    "slope.c",
                                    "intercept.c",
                                    "p for slope.c",
                                    "interaction term.c",
                                    "r2.c",
                                    "AICc.c",
                                    "slope.g",
                                    "intercept.g",
                                    "p for slope.g",
                                    "interaction term.g",
                                    "r2.g",
                                    "AICc.g")
```

- Stratified sub-sampling gives 13 artificial points and 10 natural points.

Define the models

```r
# For 1000 iterations
for(i in 1:1000){
```

```
sub <- Big.sil.clean %>% group_by(Code_Str) %>% sample_n(size=1)

## annual
reg.linear.annual <- lm(log(AAM)~AnnualTemp, data = sub)
reg.add.annual <- lm(log(AAM)~AnnualTemp+Condition, data = sub)
reg.int.annual <- lm(log(AAM)~AnnualTemp*Condition, data = sub)

# simple linear model
linear.results[i,1]<-summary(reg.linear.annual)$coef[2,1] #slope
linear.results[i,2]<-summary(reg.linear.annual)$coef[1,1] #intercept
linear.results[i,3]<-summary(reg.linear.annual)$coef[2,4] #p-value
linear.results[i,4]<-0 #blank
linear.results[i,5]<-summary(reg.linear.annual)$adj.r.squared #r2
linear.results[i,6]<-as.numeric(AICc(reg.linear.annual)) #AICc

# linear additive model
add.results[i,1]<-summary(reg.add.annual)$coef[2,1] #slope
add.results[i,2]<-summary(reg.add.annual)$coef[1,1] #intercept
add.results[i,3]<-summary(reg.add.annual)$coef[2,4] #p(slope)
add.results[i,4]<-summary(reg.add.annual)$coef[3,4] #p(additive term)
add.results[i,5]<-summary(reg.add.annual)$adj.r.squared #r2
add.results[i,6]<-as.numeric(AICc(reg.add.annual)) #AICc

# interaction model
int.results[i,1]<-summary(reg.int.annual)$coef[2,1] #slope
int.results[i,2]<-summary(reg.int.annual)$coef[1,1] #intercept
int.results[i,3]<-summary(reg.int.annual)$coef[2,4] #p(slope)
int.results[i,4]<-summary(reg.int.annual)$coef[4,4] #p(interact term)
int.results[i,5]<-summary(reg.int.annual)$adj.r.squared #r2
int.results[i,6]<-as.numeric(AICc(reg.int.annual)) #AICc


## cold
reg.linear.cold <- lm(log(AAM)~ColdTemp, data = sub)
reg.add.cold <- lm(log(AAM)~ColdTemp+Condition, data = sub)
reg.int.cold <- lm(log(AAM)~ColdTemp*Condition, data = sub)

# simple linear model
linear.results[i,7]<-summary(reg.linear.cold)$coef[2,1] #slope
linear.results[i,8]<-summary(reg.linear.cold)$coef[1,1] #intercept
linear.results[i,9]<-summary(reg.linear.cold)$coef[2,4] #p-value
linear.results[i,10]<-0 #blank
linear.results[i,11]<-summary(reg.linear.cold)$adj.r.squared #r2
linear.results[i,12]<-as.numeric(AICc(reg.linear.cold)) #AICc

# linear additive model
add.results[i,7]<-summary(reg.add.cold)$coef[2,1] #slope
add.results[i,8]<-summary(reg.add.cold)$coef[1,1] #intercept
add.results[i,9]<-summary(reg.add.cold)$coef[2,4] #p(slope)
add.results[i,10]<-summary(reg.add.cold)$coef[3,4] #p(additive term)
add.results[i,11]<-summary(reg.add.cold)$adj.r.squared #r2
add.results[i,12]<-as.numeric(AICc(reg.add.cold)) #AICc
```

```r
  # interaction model
  int.results[i,7]<-summary(reg.int.cold)$coef[2,1] #slope
  int.results[i,8]<-summary(reg.int.cold)$coef[1,1] #intercept
  int.results[i,9]<-summary(reg.int.cold)$coef[2,4] #p(slope)
  int.results[i,10]<-summary(reg.int.cold)$coef[4,4] #p(interact term)
  int.results[i,11]<-summary(reg.int.cold)$adj.r.squared #r2
  int.results[i,12]<-as.numeric(AICc(reg.int.cold)) #AICc


  ## gdd0
  reg.linear.gdd <- lm(log(AAM)~AnnualDD, data = sub)
  reg.add.gdd <- lm(log(AAM)~AnnualDD+Condition, data = sub)
  reg.int.gdd <- lm(log(AAM)~AnnualDD*Condition, data = sub)

  # simple linear model
  linear.results[i,13]<-summary(reg.linear.gdd)$coef[2,1] #slope
  linear.results[i,14]<-summary(reg.linear.gdd)$coef[1,1] #intercept
  linear.results[i,15]<-summary(reg.linear.gdd)$coef[2,4] #p-value
  linear.results[i,16]<-0 #blank
  linear.results[i,17]<-summary(reg.linear.gdd)$adj.r.squared #r2
  linear.results[i,18]<-as.numeric(AICc(reg.linear.gdd)) #AICc

  # linear additive model
  add.results[i,13]<-summary(reg.add.gdd)$coef[2,1] #slope
  add.results[i,14]<-summary(reg.add.gdd)$coef[1,1] #intercept
  add.results[i,15]<-summary(reg.add.gdd)$coef[2,4] #p(slope)
  add.results[i,16]<-summary(reg.add.gdd)$coef[3,4] #p(additive term)
  add.results[i,17]<-summary(reg.add.gdd)$adj.r.squared #r2
  add.results[i,18]<-as.numeric(AICc(reg.add.gdd)) #AICc

  # interaction model
  int.results[i,13]<-summary(reg.int.gdd)$coef[2,1] #slope
  int.results[i,14]<-summary(reg.int.gdd)$coef[1,1] #intercept
  int.results[i,15]<-summary(reg.int.gdd)$coef[2,4] #p(slope)
  int.results[i,16]<-summary(reg.int.gdd)$coef[4,4] #p(interact term)
  int.results[i,17]<-summary(reg.int.gdd)$adj.r.squared #r2
  int.results[i,18]<-as.numeric(AICc(reg.int.gdd)) #AICc
}
```

Check the slopes and additive/interaction terms

Now we need to:

1. Check if the additive or interaction term is significant.
2. Check if the relationship is significant between age at maturity and temperature (significant slope).

```r
## Additive/interaction term
# annual
mean(unique(add.results[,"additive term.a"]))
```

```
## [1] 0.1329709
```

```r
table(add.results[,"additive term.a"] < 0.05)
```

```
##
```

```
## FALSE  TRUE
##   962    38
```

```r
mean(unique(int.results[,"interaction term.a"]))
```

```
## [1] 0.9000645
```

```r
table(int.results[,"interaction term.a"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
# cold
mean(unique(add.results[,"additive term.c"]))
```

```
## [1] 0.1610584
```

```r
table(add.results[,"additive term.c"] < 0.05)
```

```
##
## FALSE  TRUE
##   993     7
```

```r
mean(unique(int.results[,"interaction term.c"]))
```

```
## [1] 0.8675939
```

```r
table(int.results[,"interaction term.c"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
# gdd
mean(unique(add.results[,"additive term.g"]))
```

```
## [1] 0.1256272
```

```r
table(add.results[,"additive term.g"] < 0.05)
```

```
##
## FALSE  TRUE
##   956    44
```

```r
mean(unique(int.results[,"interaction term.g"]))
```

```
## [1] 0.7328728
```

```r
table(int.results[,"interaction term.g"] < 0.05)
```

```
##
## FALSE
##  1000
```

```r
## Slope
# annual
mean(unique(linear.results[,"p for slope.a"]))
```

```
## [1] 0.001188979
```

```r
mean(unique(add.results[,"p for slope.a"]))
```

```
## [1] 0.0007920511
```

```r
mean(unique(int.results[,"p for slope.a"]))
```

```
## [1] 0.006319361
```

```r
# cold
mean(unique(linear.results[,"p for slope.c"]))
```

```
## [1] 0.008824907
```

```r
mean(unique(add.results[,"p for slope.c"]))
```

```
## [1] 0.006298094
```

```r
mean(unique(int.results[,"p for slope.c"]))
```

```
## [1] 0.02807173
```

```r
# gdd
mean(unique(linear.results[,"p for slope.g"]))
```

```
## [1] 0.0004937825
```

```r
mean(unique(add.results[,"p for slope.g"]))
```

```
## [1] 0.0003220879
```

```r
mean(unique(int.results[,"p for slope.g"]))
```

```
## [1] 0.003047543
```

- No significant additive or interaction term.
- Significant slope.

Compare the R2

```r
# annual
r2annual <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,5])),
         mean(unique(add.results[,5])),
         mean(unique(int.results[,5])))
)
kable(r2annual)
```

Model	R2
Simple linear	0.3793263
Linear additive	0.4244194
Interaction	0.3948840

```r
# cold
r2cold <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,11])),
         mean(unique(add.results[,11])),
         mean(unique(int.results[,11])))
```

```
)
kable(r2cold)
```

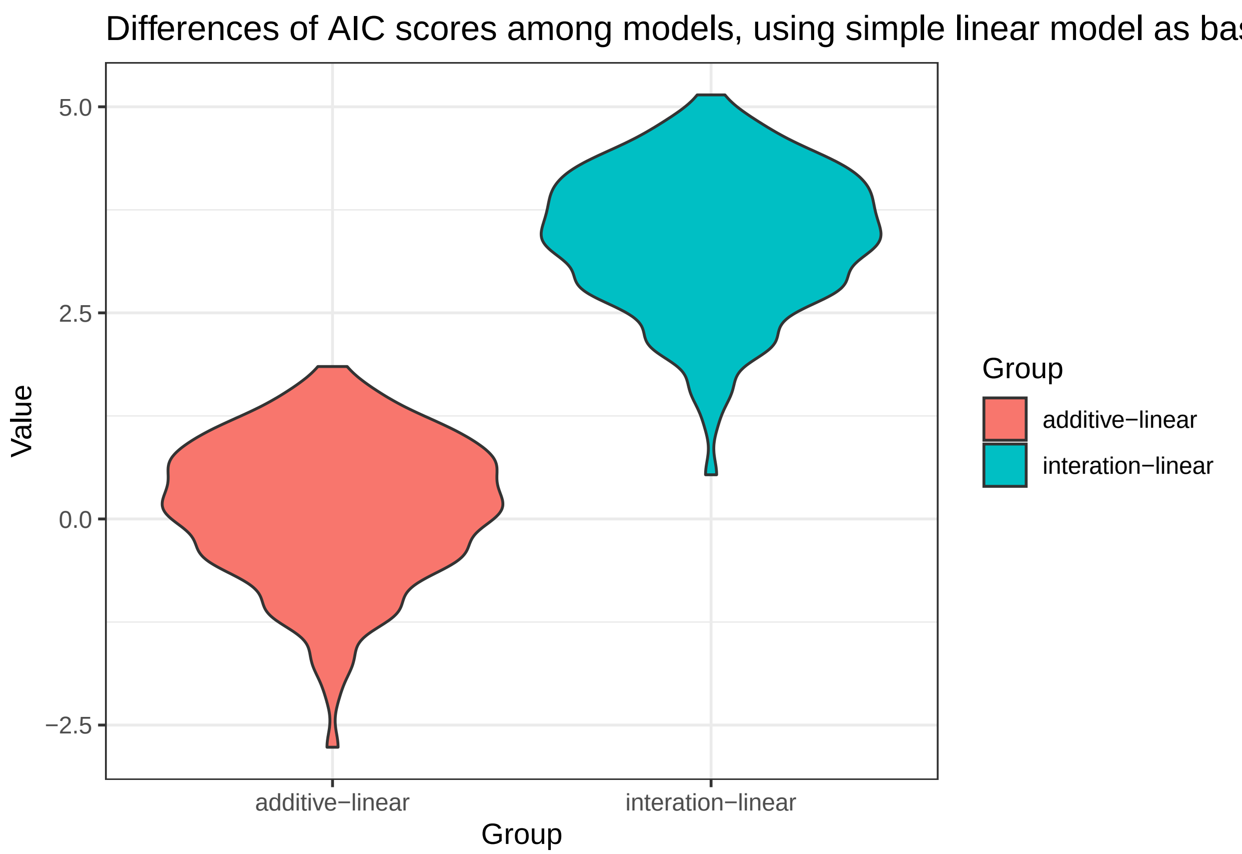
Model	R2
Simple linear	0.2559958
Linear additive	0.2983692
Interaction	0.2630150

```
# gdd
r2gdd <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction"),
  R2 = c(mean(unique(linear.results[,17])),
         mean(unique(add.results[,17])),
         mean(unique(int.results[,17])))
)
kable(r2gdd)
```

Model	R2
Simple linear	0.4257729
Linear additive	0.4705731
Interaction	0.4466748

Compare AICs for annual

```
## Look at the distribution of the differences between AIC scores
# Calculate the differences of AIC values
aic.bs <- matrix(NA,1000,2) # store the differences in AIC values
aic.bs[,1] <- add.results[,6] - linear.results[,6]
aic.bs[,2] <- int.results[,6] - linear.results[,6]

# Create a data frame
data <- as.data.frame(aic.bs)
colnames(data) <- c("additive-linear","interation-linear")

# Convert to long data format
data_long <- data %>%
  pivot_longer(cols = everything(), names_to = "Group", values_to = "Value")

# Define the desired order of groups
desired_order <- c("additive-linear","interation-linear")

# Convert "Group" to a factor with desired order
data_long$Group <- factor(data_long$Group, levels = desired_order)

# Violin plot
ggplot(data_long, aes(x = Group, y = Value, fill = Group))+
  geom_violin()+
  labs(title = "Differences of AIC scores among models, using simple linear model as base. Annual Temp")
  theme_bw()
```




```
## Check the AICc scores and akaike weights in 1000 iterations
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,6], add.results[i,6], int.results[i,6])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##        V1               V2               V3
##  Min.   :0.1740   Min.   :0.2690   Min.   :0.05200
##  1st Qu.:0.4045   1st Qu.:0.3780   1st Qu.:0.07375
##  Median :0.4780   Median :0.4370   Median :0.08500
##  Mean   :0.4725   Mean   :0.4417   Mean   :0.08577
```

```
##  3rd Qu.:0.5475   3rd Qu.:0.4980   3rd Qu.:0.09700
##  Max.   :0.6790   Max.   :0.6940   Max.   :0.13300
```

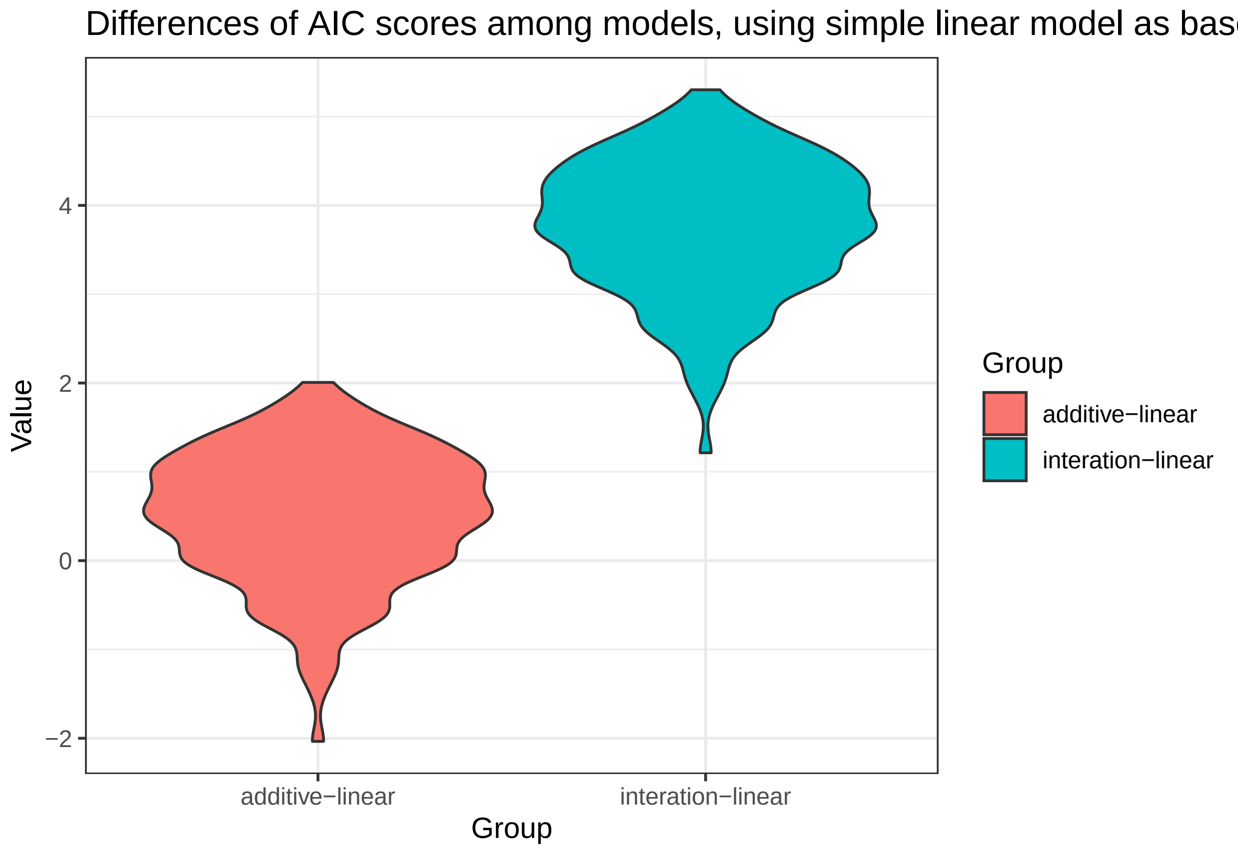
```
table(count)
```

```
## count
##  2
## 14
```

- We saw a large range in the difference of AIC values due to a larger number of combinations for subsampling sets. However, with stratified sub-sampling, we can reduce the variation for the difference in AIC scores.
- Not any preferences among the four models.

Compare AICs for the cold

```
# Calculate the differences of AIC values
aic.bs <- matrix(NA,1000,2) # store the differences in AIC values
aic.bs[,1] <- add.results[,12] - linear.results[,12]
aic.bs[,2] <- int.results[,12] - linear.results[,12]

# Look at the distribution of differences
# Create a data frame
data <- as.data.frame(aic.bs)
colnames(data) <- c("additive-linear","interation-linear")

# Convert to long data format
data_long <- data %>%
  pivot_longer(cols = everything(), names_to = "Group", values_to = "Value")

# Define the desired order of groups
desired_order <- c("additive-linear","interation-linear")

# Convert "Group" to a factor with desired order
data_long$Group <- factor(data_long$Group, levels = desired_order)

# Violin plot
ggplot(data_long, aes(x = Group, y = Value, fill = Group))+
  geom_violin()+
  labs(title = "Differences of AIC scores among models, using simple linear model as base. Cold Temp")+
  theme_bw()
```




```r
## Check the AICc scores and akaike weights in 1000 iterations
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,12], add.results[i,12], int.results[i,12])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##        V1               V2               V3
##  Min.   :0.2320   Min.   :0.2550   Min.   :0.04900
##  1st Qu.:0.4550   1st Qu.:0.3470   1st Qu.:0.06775
##  Median :0.5250   Median :0.3970   Median :0.07800
##  Mean   :0.5175   Mean   :0.4032   Mean   :0.07919
```

```
##  3rd Qu.:0.5835   3rd Qu.:0.4560   3rd Qu.:0.08900
##  Max.   :0.6960   Max.   :0.6420   Max.   :0.12600
```

```r
table(count)
```

```
## count
## 1 2
## 8 7
```

- No preference among the four models.

Compare between annual and cold

```r
## Compare the R2 for three temperatures
r2 <- data.frame(
  Temperature = c("Annual", "Cold", "GDD0"),
  R2 = c(mean(unique(linear.results[,"r2.a"])),
         mean(unique(linear.results[,"r2.c"])),
         mean(unique(linear.results[,"r2.g"])))
)
kable(r2)
```

Temperature	R2
Annual	0.3793263
Cold	0.2559958
GDD0	0.4257729

```r
## Check the AICc scores and akaike weights in ONLY LINEAR MODEL
weight.matrix <- matrix(NA, 1000, 3)
count <- numeric(0)

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(linear.results[i,6], linear.results[i,12],
                 linear.results[i,18])

  ## check the akaike weights
  weight <- compute_akaike_weights(aic_value)
  weight.matrix[i,c(1,2,3)] <- round(weight[c(1,2,3)],3)

  ## check the AICc scores
  indexing <- compare_aic_scores(aic_value)
  if (indexing != -999) {
    count <- c(count, indexing)
  }
}

summary(weight.matrix)
```

```
##        V1               V2                 V3
##  Min.   :0.2070   Min.   :0.02300   Min.   :0.5900
##  1st Qu.:0.2530   1st Qu.:0.03100   1st Qu.:0.6580
##  Median :0.2760   Median :0.03500   Median :0.6870
##  Mean   :0.2792   Mean   :0.03497   Mean   :0.6857
```

```
##   3rd Qu.:0.3020   3rd Qu.:0.04000   3rd Qu.:0.7130
##   Max.   :0.3740   Max.   :0.05000   Max.   :0.7600
```

```r
table(count)
```

```
## count
##   3
## 308
```

- For bs, using annual temperature is always better than using the cold temperature (difference in AIC > 2) for all models. This was also explained by lower R^2 values for the cold temperature models.

Two conditions separated

```r
## Separate the two conditions
bs.natural <- Big.sil.clean[Big.sil.clean$Condition == "natural",]
bs.artificial <- Big.sil.clean[Big.sil.clean$Condition == "artificial",]

# Check sp
table(bs.natural$Code_Str)
```

```
##
## A C D F H I J K M Q
## 2 1 1 1 1 1 2 1 1 1
```

```r
table(bs.artificial$Code_Str)
```

```
##
## AA AB AC AE AG AH AI AK AL  R  U  V  X
##  2  1  1  1  2  1  2  2  1  2  2  2  1
```

```r
# store the results
temp.natural <- matrix(NA, 1000, 3)
temp.artificial <- matrix(NA, 1000, 3)


## Removing artificial for 1000 iterations
for(i in 1:1000){

  ## natural
  sub.n <- bs.natural %>% group_by(Code_Str) %>% sample_n(size=1)

  # models
  reg.annual <- lm(log(AAM)~AnnualTemp, data = sub)
  reg.cold <- lm(log(AAM)~ColdTemp, data = sub)
  reg.gdd <- lm(log(AAM)~AnnualDD, data = sub)

  # r2
  temp.natural[i,1]<-summary(reg.annual)$adj.r.squared
  temp.natural[i,2]<-summary(reg.cold)$adj.r.squared
  temp.natural[i,3]<-summary(reg.gdd)$adj.r.squared


  ## artificial
  sub.a <- bs.artificial %>% group_by(Code_Str) %>% sample_n(size=1)

  # models
```

```r
  reg.annual <- lm(log(AAM)~AnnualTemp, data = sub.a)
  reg.cold <- lm(log(AAM)~ColdTemp, data = sub.a)
  reg.gdd <- lm(log(AAM)~AnnualDD, data = sub.a)

  # r2
  temp.artificial[i,1]<-summary(reg.annual)$adj.r.squared
  temp.artificial[i,2]<-summary(reg.cold)$adj.r.squared
  temp.artificial[i,3]<-summary(reg.gdd)$adj.r.squared
}


## Compare the R2 for three temperatures
r2 <- data.frame(
  Temperature = c("Annual", "Cold", "GDD0"),
  Natural.R2 = c(mean(unique(temp.natural[,1])),
                 mean(unique(temp.natural[,2])),
                 mean(unique(temp.natural[,3]))),
  Artificial.R2 = c(mean(unique(temp.artificial[,1])),
                    mean(unique(temp.artificial[,2])),
                    mean(unique(temp.artificial[,3])))
)
kable(r2)
```

Temperature	Natural.R2	Artificial.R2
Annual	0.3296859	0.5282247
Cold	0.2116542	0.3646856
GDD0	0.3823037	0.5911505

- For bighead and silver carp, there were fewer data points (32 datapoints in total), but more subsample sets (10 sets of subsamples. This gave us 19 data points after subsampling with a much larger variation (due to a larger number of combinations). At extremes, we would have 13 artificial and 6 natural (if all subseting choose artificial); or 10 natural and 9 artificial (if all subsetting choose natural).
- So we use stratified sub-sampling to reduce this effect.

Concluding points

1. Black carp: Using cold temperature have a better fit (higher R2). No preference over the four types of models. So we chose the simple linear model (Akaike weight = 51%).
2. Black carp: When separate the two conditions, we see a large increase in the R2 for the natural condition. The artificial condition alone did not have a significant relationship between log AAM and temperature.
3. Asian carp: The simple linear model is preferred (87% for annual, and 70% for cold). For grass carp, the simple linear model is preferred only for using the cold temperature (100%). For bs carp, there is no preference among the four models.
4. Asian carp: Using annual temperature is preferred (65% of times when annual temperature is the preferred). There is no preferrence for grass carp; but strong preference for bs carp using annual temperature.
5. Stratified sub-sampling reduced the large variation in AIC values in the bighead and silver carp combined dataset.