# Black/Asian carp model selection

Eddie Wu

2023-09-20

## 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 three temperatures (annual, cold, warm):

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

```r
## 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 Ukarine data point for all the following analyses.

### Temperature prediction of black carp AAM

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

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

### Model selection using annual temperature - no subsample

```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)
black.group <- lm(log(AAM)~AnnualTemp*condition, data = black.clean)
```

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

```
##
## 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
## -0.45878 -0.10360  0.01486  0.12414  0.25006
##
## Coefficients:
```

```
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      1.980132   0.073422  26.969  < 2e-16 ***
## AnnualTemp:conditionartificial -0.013372   0.006087  -2.197  0.04063 *
## AnnualTemp:conditionnatural    -0.020029   0.005738  -3.491  0.00245 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.173 on 19 degrees of freedom
## Multiple R-squared:  0.3911, Adjusted R-squared:  0.327
## F-statistic: 6.102 on 2 and 19 DF,  p-value: 0.008976
```

```r
summary(black.group)
```

```
##
## 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
```

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

```
##
## Model selection based on AICc:
##
##                  K  AICc Delta_AICc AICcWt   LL
## simple linear    3 -8.91       0.00   0.51 8.12
## linear additive  4 -6.39       2.52   0.15 8.37
## interaction      4 -7.63       1.28   0.27 8.99
## grouped-specific 5 -4.99       3.92   0.07 9.37
```

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

```r
)
kable(r_2)
```

| Model | R2 |
|---|---|
| Simple linear | 0.3079314 |
| Linear additive | 0.2879251 |
| Interaction | 0.3270202 |
| Grouped | 0.3134071 |

- There is no significant interaction term or additive term, thus the simple linear model is the best.

### Model selection using cold temperature - no subsample

```r
# 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)
black.group <- lm(log(AAM)~ColdTemp*condition, data = black.clean)


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

```
##
## 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
```

```r
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
## -0.39294 -0.09375 -0.00773  0.10732  0.27597
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.770280   0.035198  50.295  < 2e-16 ***
## ColdTemp:conditionartificial -0.007465   0.004394  -1.699  0.10564
## ColdTemp:conditionnatural    -0.015059   0.004211  -3.576  0.00201 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1642 on 19 degrees of freedom
## Multiple R-squared:  0.4518, Adjusted R-squared:  0.3941
## F-statistic: 7.831 on 2 and 19 DF,  p-value: 0.003308
```

```r
summary(black.group)
```

```
##
## 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
```

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

```
##
## Model selection based on AICc:
##
##                  K   AICc Delta_AICc AICcWt    LL
## simple linear    3 -11.23       0.00   0.52  9.28
## linear additive  4  -8.67       2.56   0.14  9.51
## interaction      4  -9.95       1.28   0.27 10.15
## grouped-specific 5  -7.23       4.00   0.07 10.49
```

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

| Model | R2 |
|---|---|
| Simple linear | 0.3771965 |
| Linear additive | 0.3579008 |
| Interaction | 0.3941400 |
| Grouped | 0.3798876 |

- There is no significant interaction term or additive term, thus the simple linear model is the best.
- Cold temperature in general gives better predictions (lower AICc and higher R2).

### Model selection using warm temperature - no subsample

```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)
black.group <- lm(log(AAM)~WarmTemp*condition, data = black.clean)


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

```
##
## 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
## -0.47655 -0.11070  0.04103  0.11833  0.33915
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.12946    0.26677   7.982 1.73e-07 ***
## WarmTemp:conditionartificial -0.01320    0.01127  -1.171    0.256
## WarmTemp:conditionnatural    -0.01635    0.01114  -1.467    0.159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2079 on 19 degrees of freedom
## Multiple R-squared:  0.1204, Adjusted R-squared:  0.02786
## F-statistic: 1.301 on 2 and 19 DF,  p-value: 0.2955
```

```r
summary(black.group)
```

```
##
## 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
```

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

```
##
## Model selection based on AICc:
##
##                  K  AICc Delta_AICc AICcWt   LL
## simple linear    3 -1.73       0.00   0.58 4.53
## linear additive  4  0.71       2.44   0.17 4.82
## interaction      4  0.46       2.19   0.19 4.95
## grouped-specific 5  3.10       4.83   0.05 5.32
```

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

| Model | R2 |
|---|---|
| Simple linear | 0.0408726 |
| Linear additive | 0.0166216 |
| Interaction | 0.0278624 |
| Grouped | 0.0083708 |

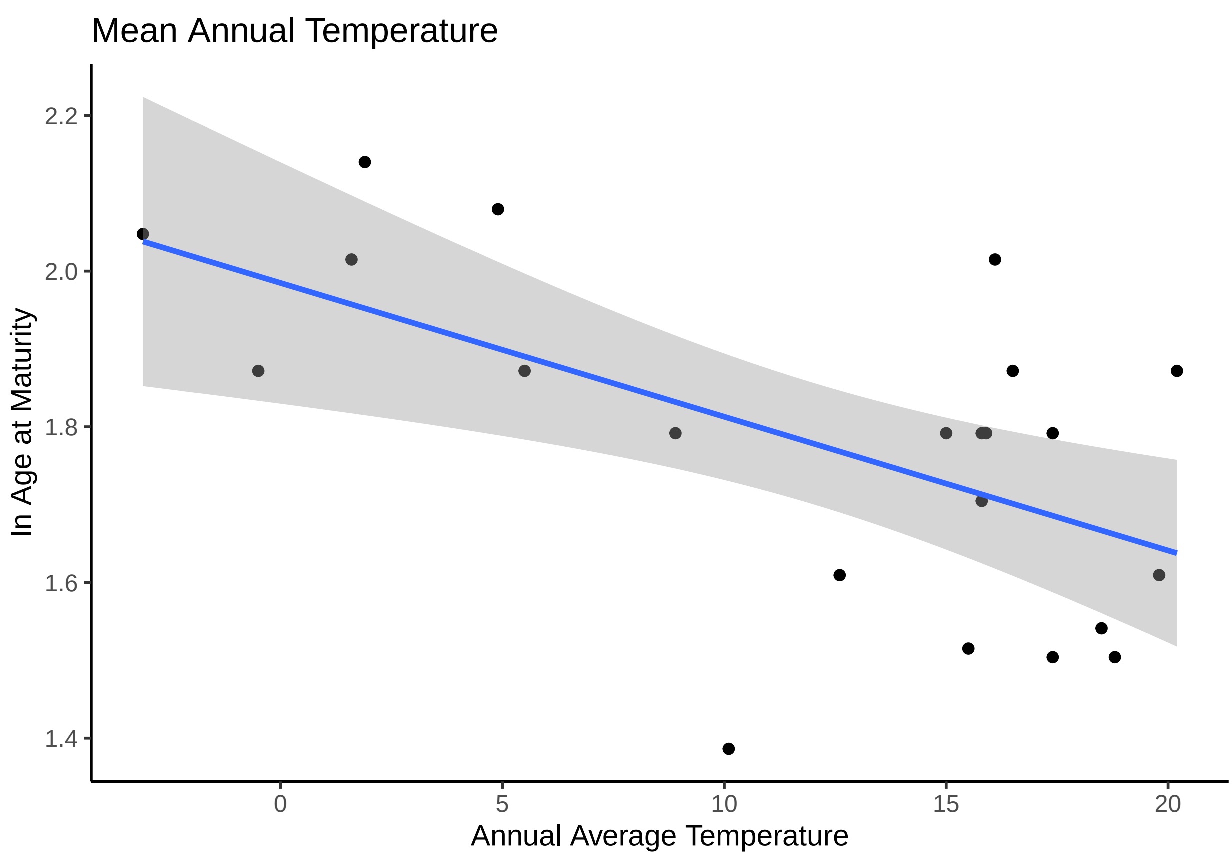
- Warm temperature is not a significant predictor of black carp age at maturity (same conclusion as

Madison found on other Asian carps), thus we remove it in the following analyses.

**Temperature predictions, plots, and power analyses**

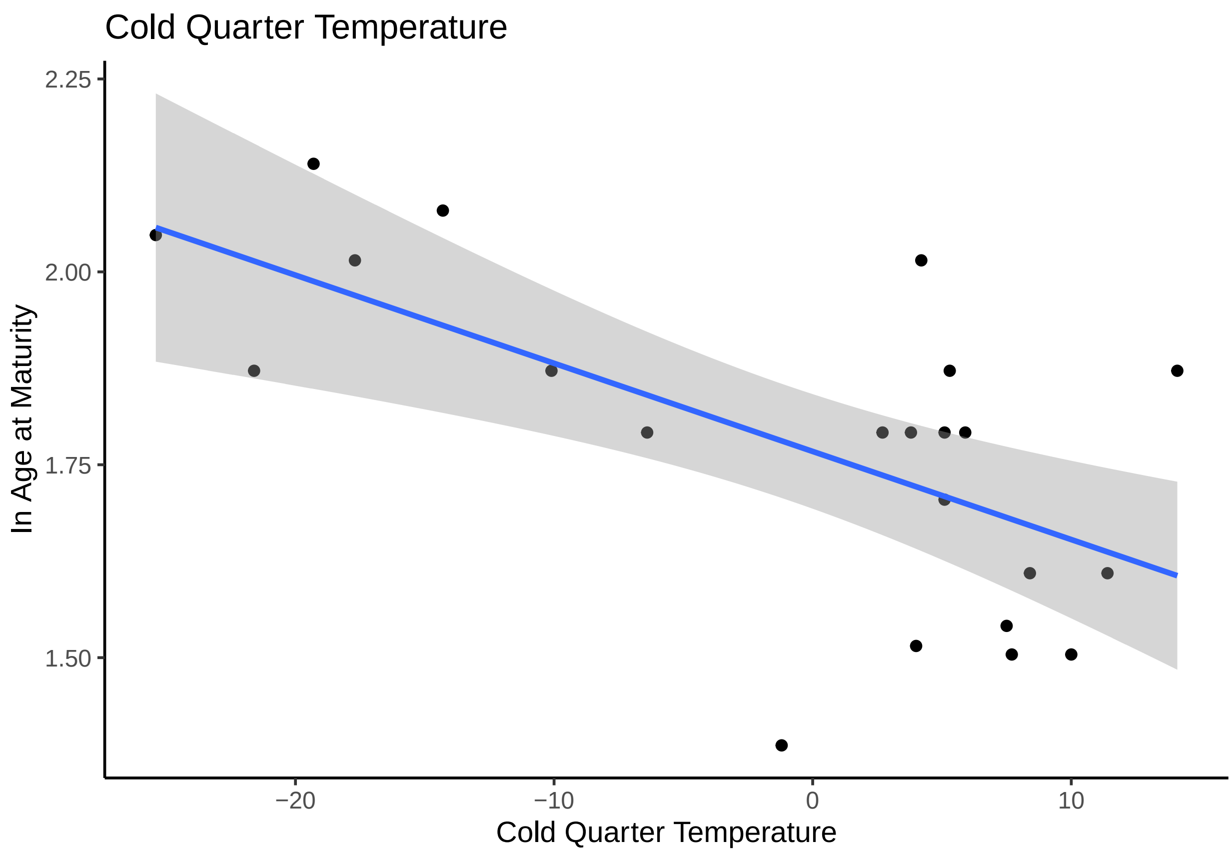
```
## Build the models with three temperature metrics (simple model here)
black.annual <- lm(log(AAM)~AnnualTemp, data = black.clean)
black.cold <- lm(log(AAM)~ColdTemp, data = black.clean)

## Temperature plots
# Annual temperature
ggplot(black.clean, aes(x = AnnualTemp, y = log(AAM)))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_classic()+
  labs(title = "Mean Annual Temperature",
       x = "Annual Average Temperature", y = "ln Age at Maturity")
```

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




```
# Cold temperature
ggplot(black.clean, aes(x = ColdTemp, y = log(AAM)))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_classic()+
  labs(title = "Cold Quarter Temperature",
       x = "Cold Quarter Temperature", y = "ln Age at Maturity")
```

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




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

```
##
##      Multiple regression power calculation
##
##                u = 1
##                v = 13.14405
##               f2 = 0.6056428
##        sig.level = 0.05
##            power = 0.8
```

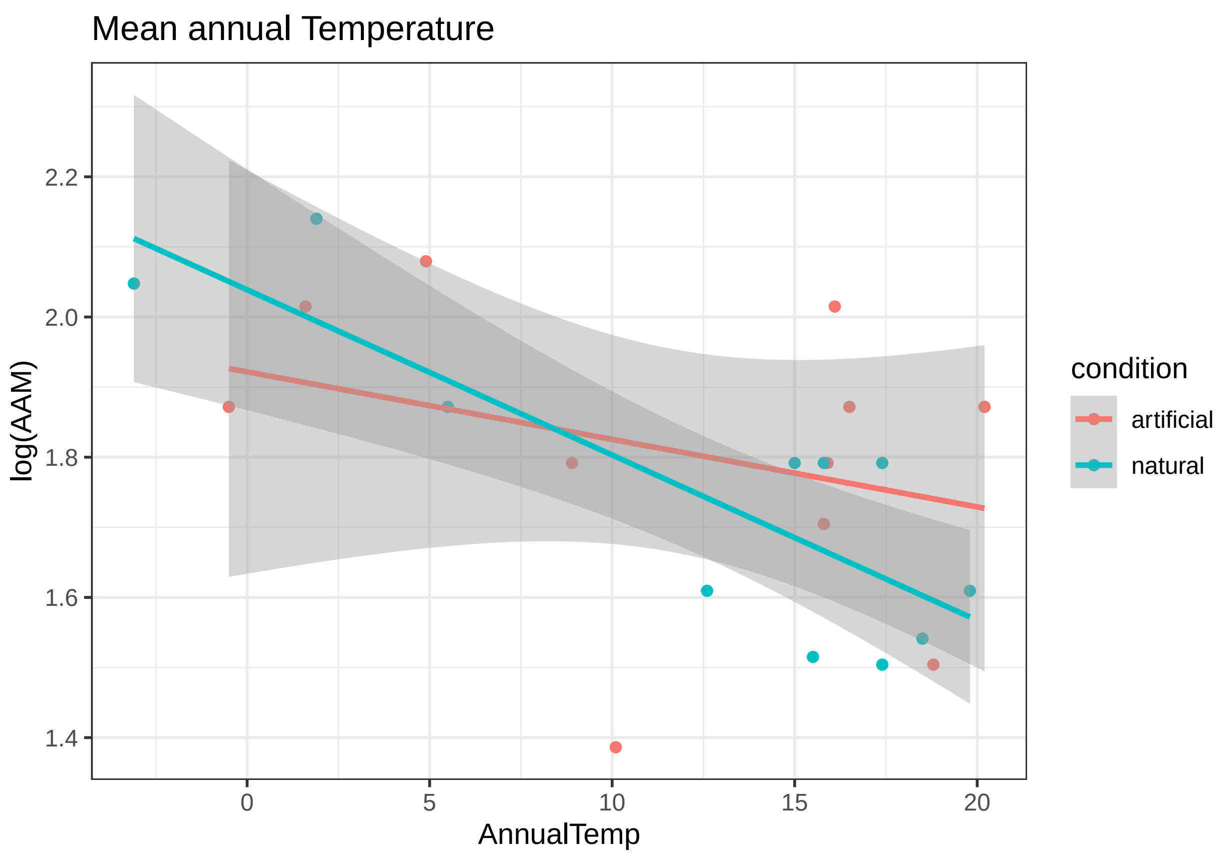
- We can see that annual temperature and cold temperature are significant predictors of black carp AAM. Warm temperature is not.
- Power analyses suggested that our current sample size is sufficient enough to produce a strong statistical power.

### 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.

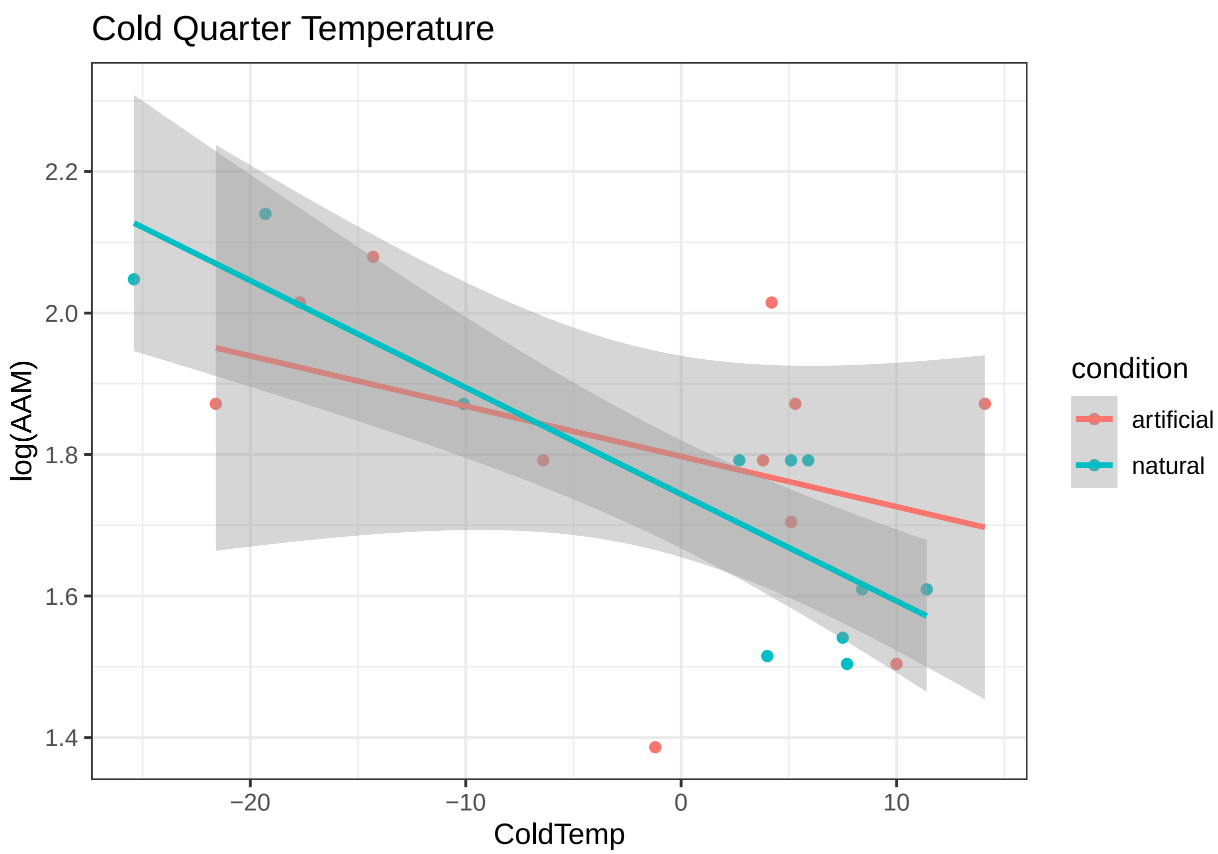
```r
## Annual temperature
ggplot(black.clean, aes(x = AnnualTemp, y = log(AAM), color = condition))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()+
  labs(title = "Mean annual Temperature")
```

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




```
## 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'
```




Now that we have seen that the artificial condition data seems to have a larger spread, we would like to run the simple linear model to take a look.

```
## 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.annual.a <- lm(log(AAM)~AnnualTemp, data = black.artificial)
black.cold.a <- lm(log(AAM)~ColdTemp, data = black.artificial)


## Compare the AIC scores
AIC(black.annual.n, black.annual.a) #for annual temperature
```

```
##                df         AIC
## black.annual.n  3 -10.4922037
## black.annual.a  3   0.9200476
```

```
AIC(black.cold.n, black.cold.a) #for cold temperature
```

```
##              df        AIC
## black.cold.n  3 -13.121053
## black.cold.a  3   0.288321
```

```r
## 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
```

```
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
```

- It turned put that after separating out the artificial condition, the model performed much better. While the artificial model alone did not even have a significant relationship.

## 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

```
## 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
```

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

# Simple linear model - slope, intercept, p, blank, r2, AICc
linear.results <- matrix(NA,1000,12)
```

```r
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")
# Linear additive model - slope, intercept, p(slope), p(additive), r2, AICc
add.results <- matrix(NA,1000,12)
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")
# Interaction model - slope, intercept, p(slope), p(interaction), r2, AICc
int.results <- matrix(NA,1000,12)
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")
```

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

#### Define the models

```r
# For 1000 iterations
for(i in 1:1000){
  sub <- asian.carp.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

# 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
```

```
}
```

### 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] 0.8264942
```

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

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

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

```
## [1] 0.6651748
```

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

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

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

```
## [1] 0.6212652
```

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

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

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

```
## [1] 0.7959113
```

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

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

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

```
## [1] 1.604829e-07
```

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

```
## [1] 5.207614e-07
```

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

```
## [1] 6.310934e-05
```

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

```
## [1] 4.19733e-07
```

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

```
## [1] 1.22763e-06
```

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

```
## [1] 0.000114186
```

- 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)
```

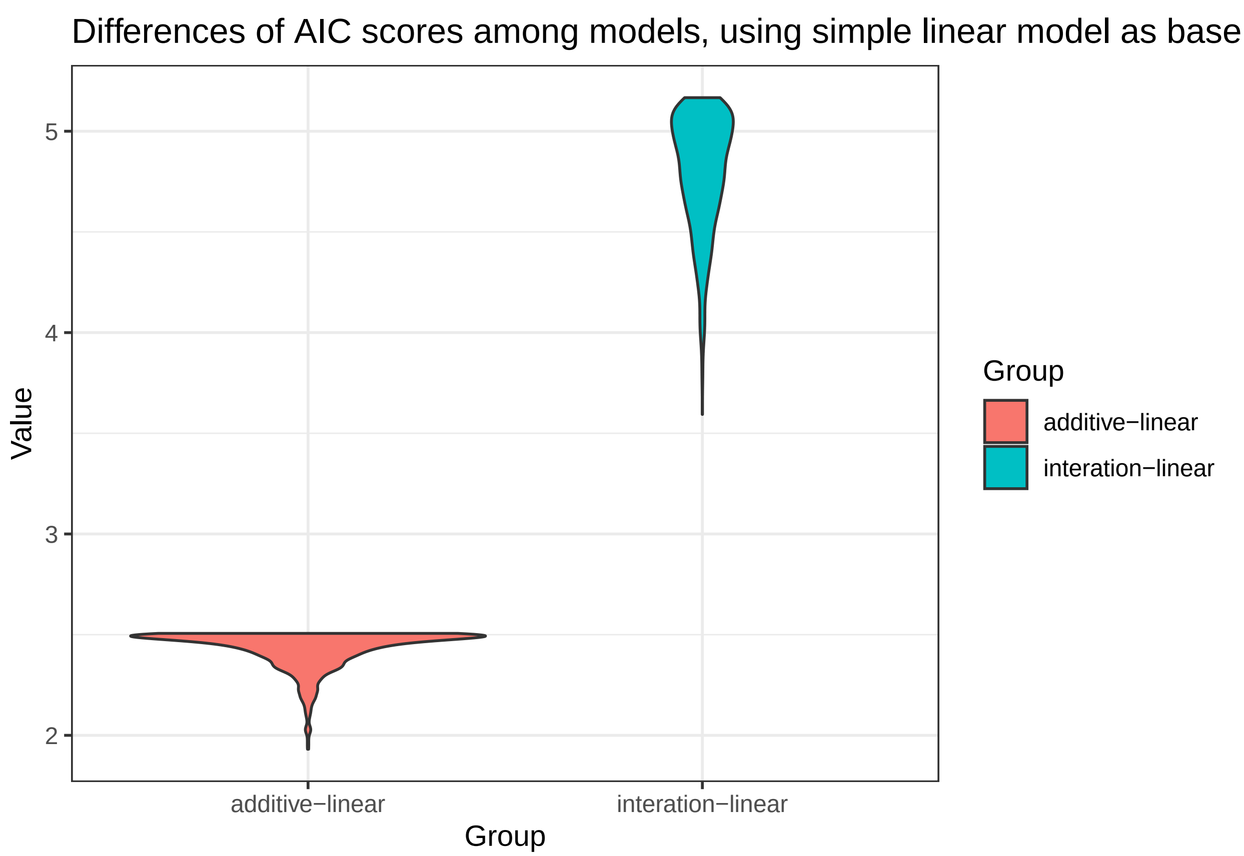
| Model | R2 |
|---|---|
| Simple linear | 0.5386165 |
| Linear additive | 0.5264130 |
| Interaction | 0.5161327 |

```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 | 0.5113136 |
| Linear additive | 0.5018924 |
| Interaction | 0.4888129 |

### 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.   :0.6600   Min.   :0.199   Min.   :0.05500
##  1st Qu.:0.7140   1st Qu.:0.209   1st Qu.:0.05900
##  Median :0.7230   Median :0.212   Median :0.06400
##  Mean   :0.7199   Mean   :0.214   Mean   :0.06605
##  3rd Qu.:0.7290   3rd Qu.:0.217   3rd Qu.:0.07100
##  Max.   :0.7350   Max.   :0.255   Max.   :0.11400
```

```r
table(count)
```

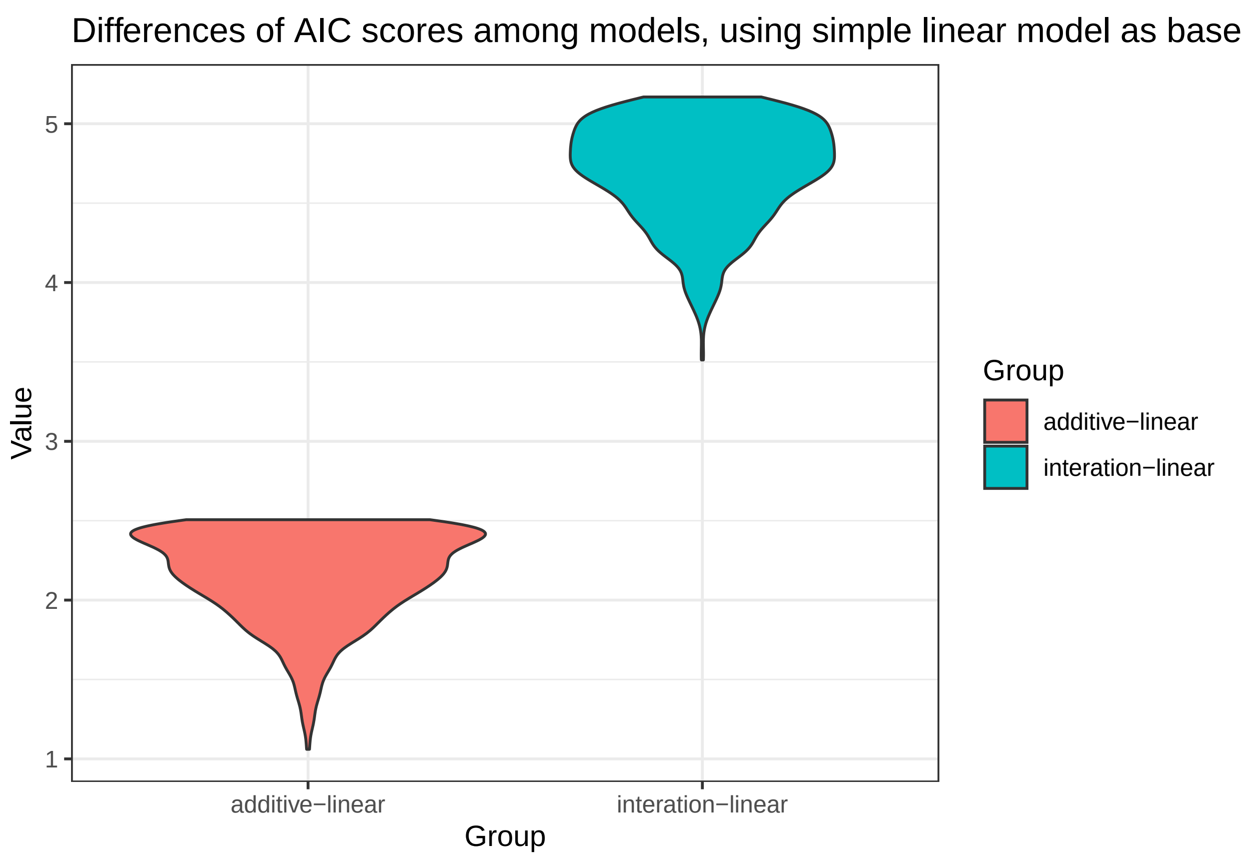
```
## count
##   1
## 997
```

- 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")
```

```r
# 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.   :0.05500
##  1st Qu.:0.6780   1st Qu.:0.2180   1st Qu.:0.06075
##  Median :0.7010   Median :0.2330   Median :0.06500
##  Mean   :0.6955   Mean   :0.2377   Mean   :0.06682
##  3rd Qu.:0.7210   3rd Qu.:0.2510   3rd Qu.:0.07200
##  Max.   :0.7350   Max.   :0.3340   Max.   :0.10000
```

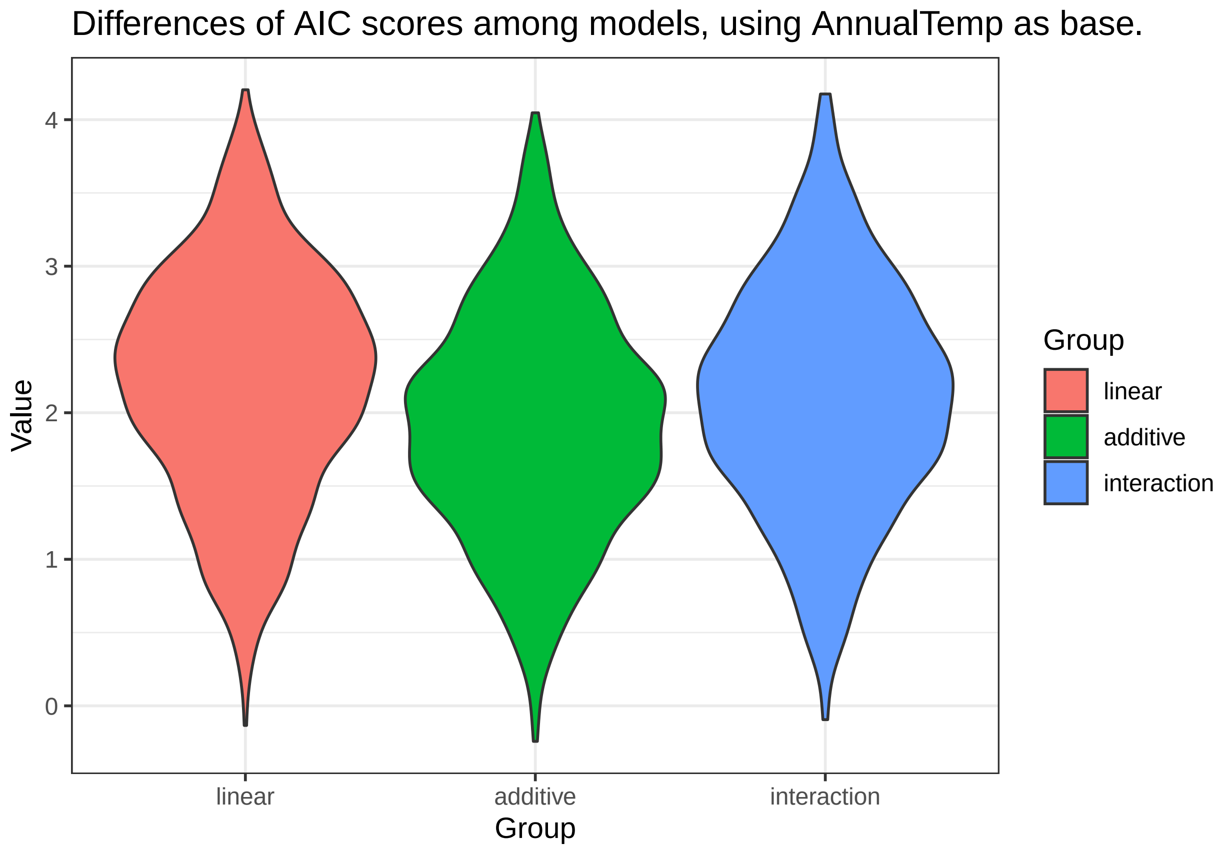
```
table(count)
```

```
## count
##   1
## 741
```

- 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 between annual and cold

```
# Calculate the differences of AIC values
aic.asian <- matrix(NA,1000,3) # store the differences in AIC values
aic.asian[,1] <- linear.results[,12] - linear.results[,6]
aic.asian[,2] <- add.results[,12] - add.results[,6]
aic.asian[,3] <- int.results[,12] - int.results[,6]

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

# 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("linear","additive","interaction")

# 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 AnnualTemp as base.")+
  theme_bw()
```




```r
## Check the AICc scores and akaike weights in ONLY LINEAR MODEL
weight.matrix <- matrix(NA, 1000, 2)
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])

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

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

summary(weight.matrix)
```

```
##        V1               V2
##  Min.   :0.4830   Min.   :0.1090
##  1st Qu.:0.7000   1st Qu.:0.2010
##  Median :0.7550   Median :0.2450
##  Mean   :0.7432   Mean   :0.2568
```

```
##   3rd Qu.:0.7990   3rd Qu.:0.3000
##   Max.   :0.8910   Max.   :0.5170
```

```
table(count)
```

```
## count
##   1
## 620
```

- With the simple linear model, around 65% of the time when using Annualtemp is preferred over using ColdTemp.

## 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,1000,12)
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")
# Linear additive model - slope, intercept, p(slope), p(additive), r2, AICc
add.results <- matrix(NA,1000,12)
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")
# Interaction model - slope, intercept, p(slope), p(interaction), r2, AICc
```

```r
int.results <- matrix(NA,1000,12)
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")
```

**Define the models**

```r
# 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)
  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
}
```

### 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] 0.4175639
```

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

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

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

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

```
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
## 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
```

- 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)
```

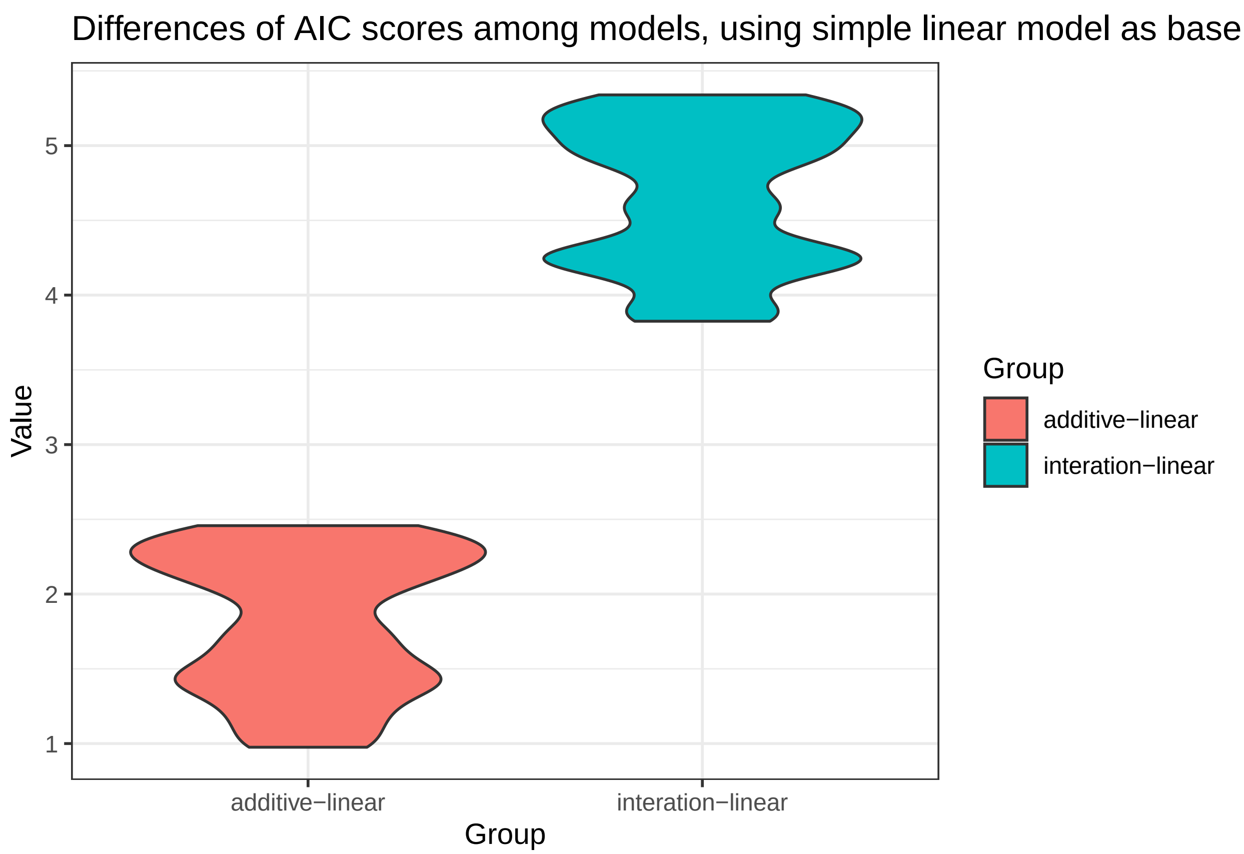
| Model | R2 |
|---|---|
| Simple linear | 0.6345688 |
| Linear additive | 0.6319235 |
| Interaction | 0.6184244 |

```
# 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 | 0.6397366 |
| Linear additive | 0.6284060 |
| Interaction | 0.6148537 |

**Compare AICs for annual**

```
## 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()+
  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.   :0.5680   Min.   :0.2150   Min.   :0.05100
##  1st Qu.:0.6210   1st Qu.:0.2320   1st Qu.:0.05600
##  Median :0.6630   Median :0.2710   Median :0.06600
##  Mean   :0.6635   Mean   :0.2712   Mean   :0.06546
```

```
##  3rd Qu.:0.7130   3rd Qu.:0.3053   3rd Qu.:0.07500
##  Max.   :0.7340   Max.   :0.3490   Max.   :0.08400
```

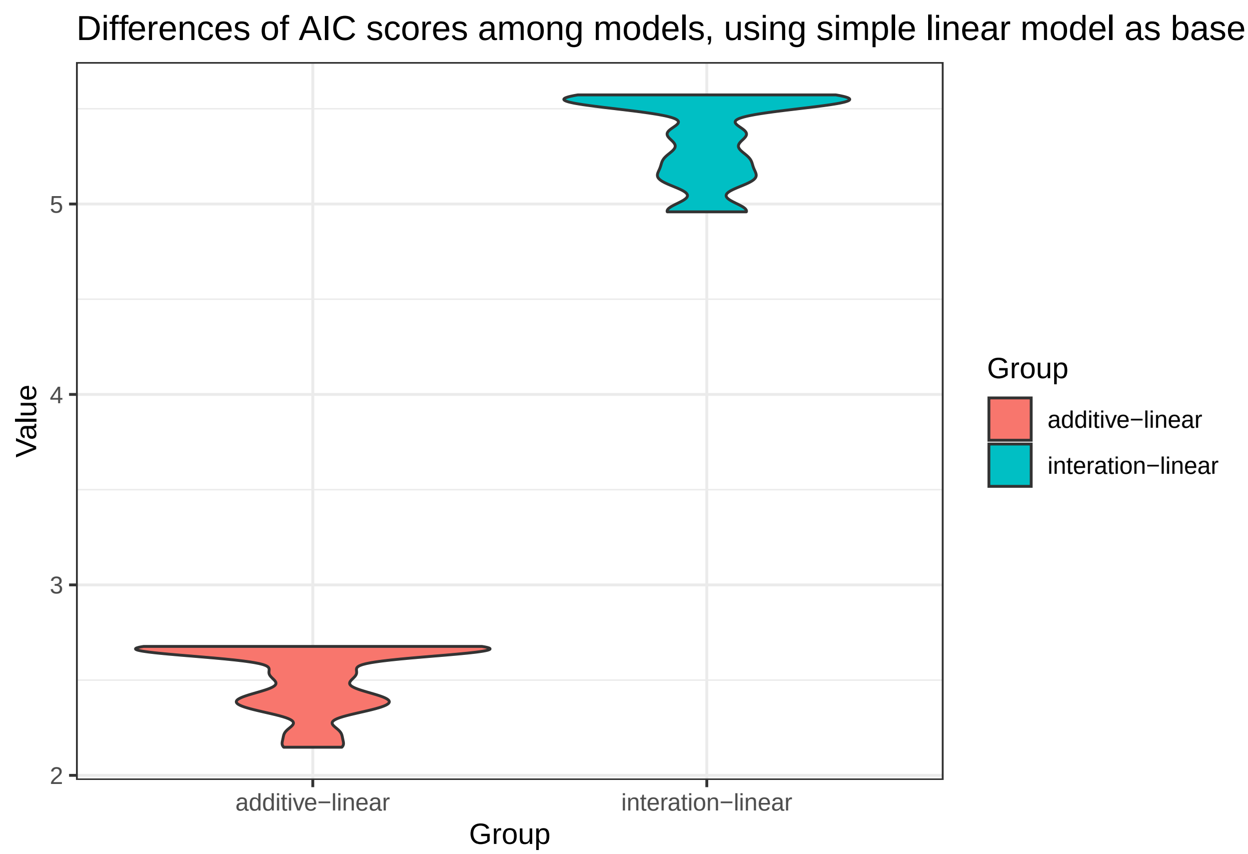
```
table(count)
```

```
## count
##   1
## 428
```

- 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

```
# 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   :0.7383   Mean   :0.2107   Mean   :0.05099
```

```
##   3rd Qu.:0.7540   3rd Qu.:0.2210   3rd Qu.:0.05500
##   Max.   :0.7550   Max.   :0.2400   Max.   :0.05900
```

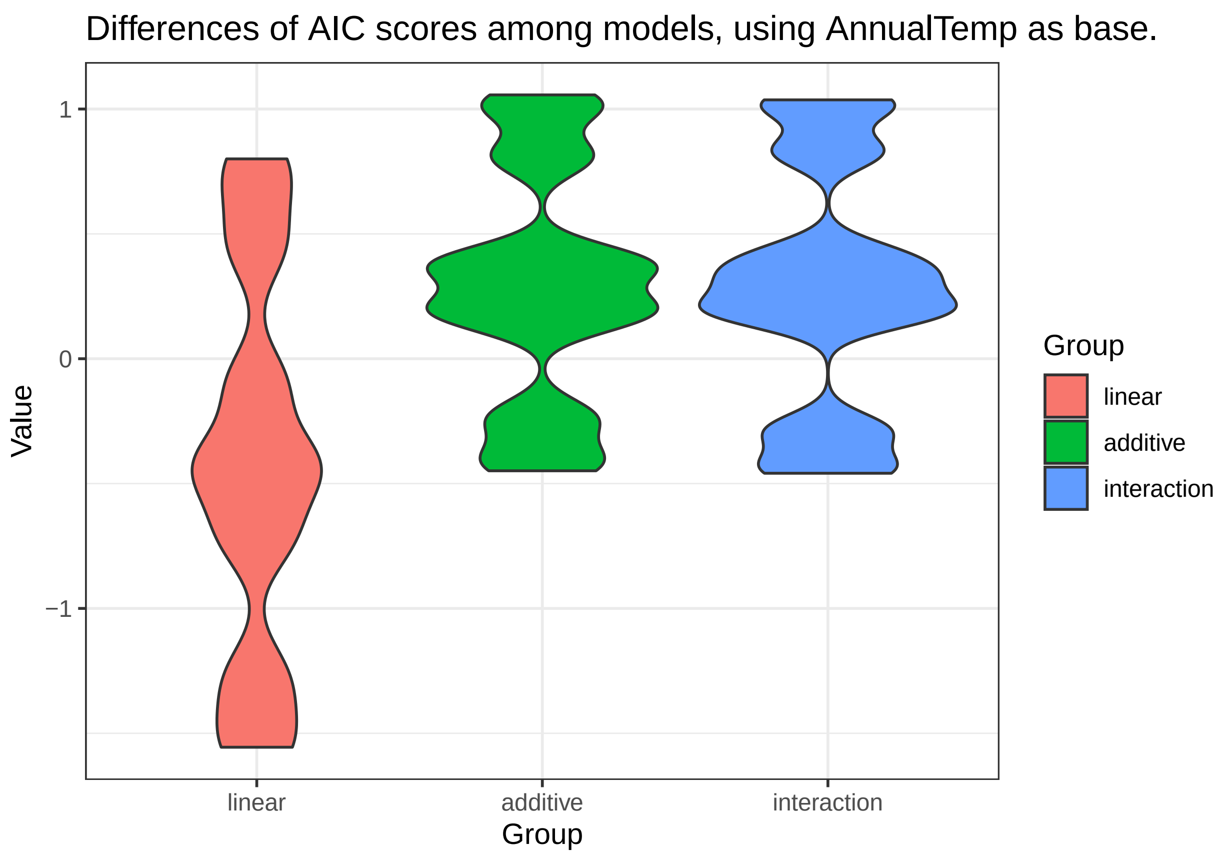
```r
table(count)
```

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

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

#### Compare between annual and cold

```r
# Calculate the differences of AIC values
aic.grass <- matrix(NA,1000,3) # store the differences in AIC values
aic.grass[,1] <- linear.results[,12] - linear.results[,6]
aic.grass[,2] <- add.results[,12] - add.results[,6]
aic.grass[,3] <- int.results[,12] - int.results[,6]

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

# 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("linear","additive","interaction")

# 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 AnnualTemp as base.")+
  theme_bw()
```




```r
## Check the AICc scores and akaike weights in ONLY LINEAR MODEL
weight.matrix <- matrix(NA, 1000, 2)
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])

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

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

summary(weight.matrix)
```

```
##        V1               V2
##  Min.   :0.3150   Min.   :0.4010
##  1st Qu.:0.3937   1st Qu.:0.5070
##  Median :0.4460   Median :0.5540
##  Mean   :0.4493   Mean   :0.5507
```

```
##   3rd Qu.:0.4930   3rd Qu.:0.6062
##   Max.   :0.5990   Max.   :0.6850
```

```r
table(count)
```

```
## < table of extent 0 >
```

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

## SECTION 4: Bighead and silver carp

### Data cleaning and matrices for results

```r
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
```

```r
## Simple linear model - slope, intercept, p, blank, r2, AICc
linear.results <- matrix(NA,1000,12)
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")
# Linear additive model - slope, intercept, p(slope), p(additive), r2, AICc
add.results <- matrix(NA,1000,12)
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")
# Interaction model - slope, intercept, p(slope), p(interaction), r2, AICc
int.results <- matrix(NA,1000,12)
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")
```

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

#### Define the models

```
# 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

  # 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
}
```

### 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] 0.1329709
```

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

```
##
## FALSE  TRUE
##   950    50
```

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

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

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

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

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

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

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

```
##
## FALSE  TRUE
##   991     9
```

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

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

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

- 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 |

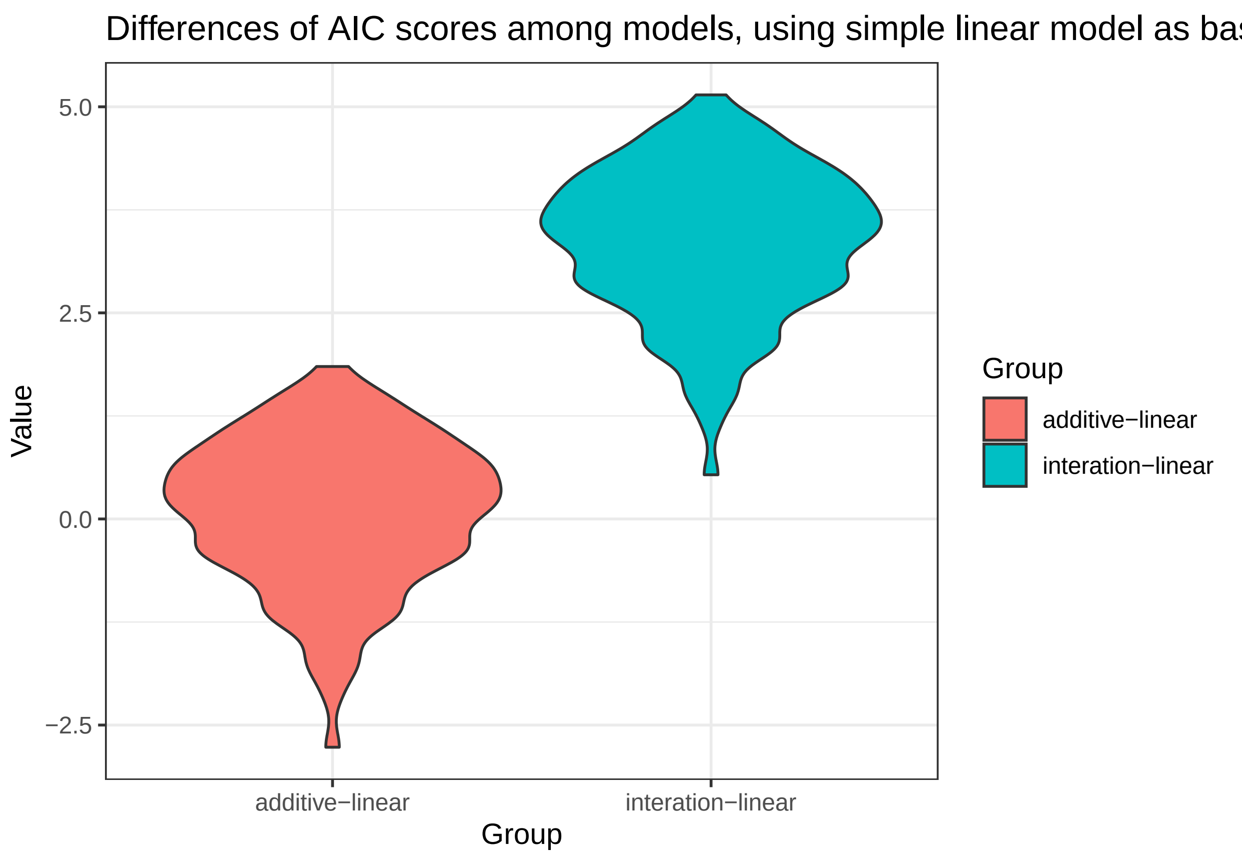
| Model | R2 |
|---|---|
| Linear additive | 0.4244194 |
| Interaction | 0.3948840 |

```
# 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 |

**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.   :0.0520
##  1st Qu.:0.4000   1st Qu.:0.3790   1st Qu.:0.0740
##  Median :0.4780   Median :0.4380   Median :0.0850
##  Mean   :0.4684   Mean   :0.4451   Mean   :0.0865
```

```
##  3rd Qu.:0.5470   3rd Qu.:0.5040   3rd Qu.:0.0980
##  Max.   :0.6790   Max.   :0.6940   Max.   :0.1330
```

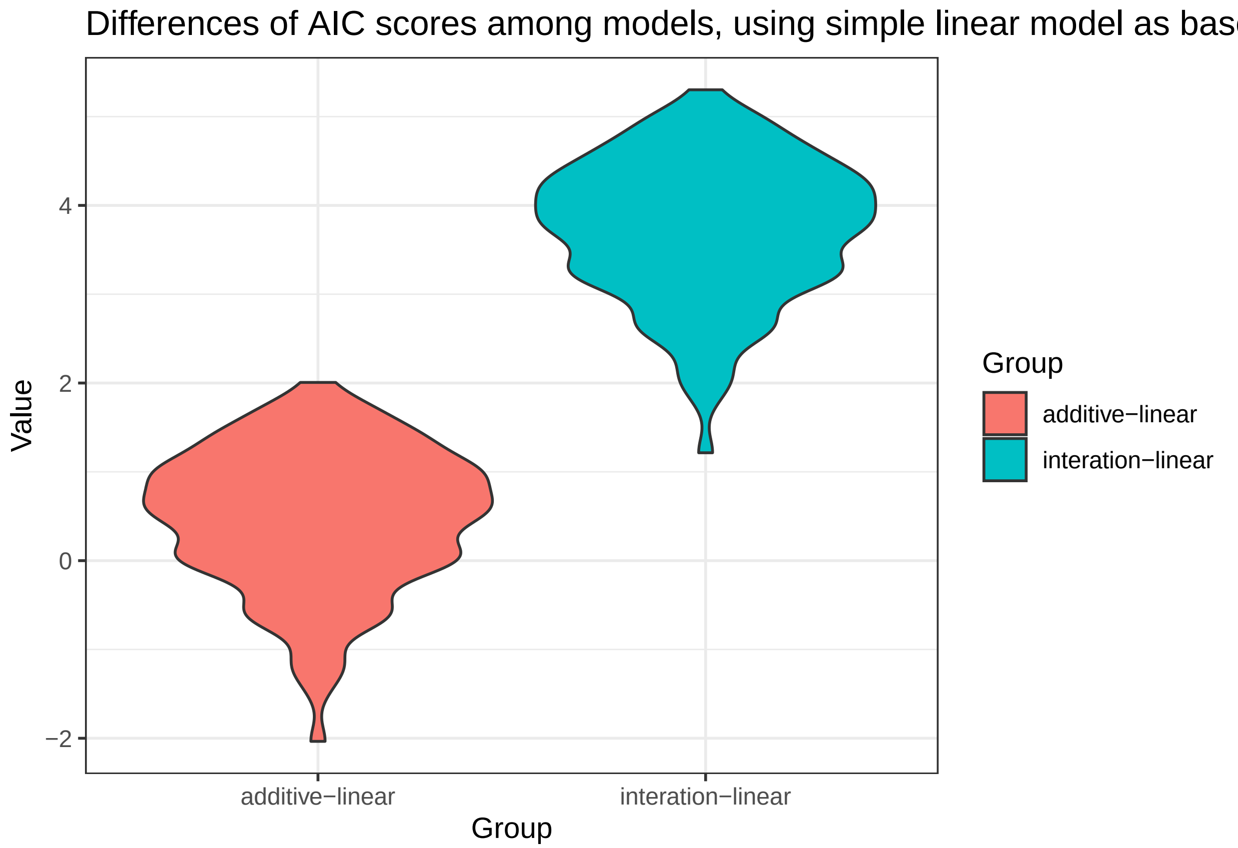
```r
table(count)
```

```
## count
##  2
## 19
```

- 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

```r
# 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()
```




```
## 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.:0.4530   1st Qu.:0.3470   1st Qu.:0.06800
##  Median :0.5250   Median :0.3970   Median :0.07800
##  Mean   :0.5139   Mean   :0.4061   Mean   :0.07988
```

```
##   3rd Qu.:0.5830   3rd Qu.:0.4570   3rd Qu.:0.09000
##   Max.   :0.6960   Max.   :0.6420   Max.   :0.12600
```

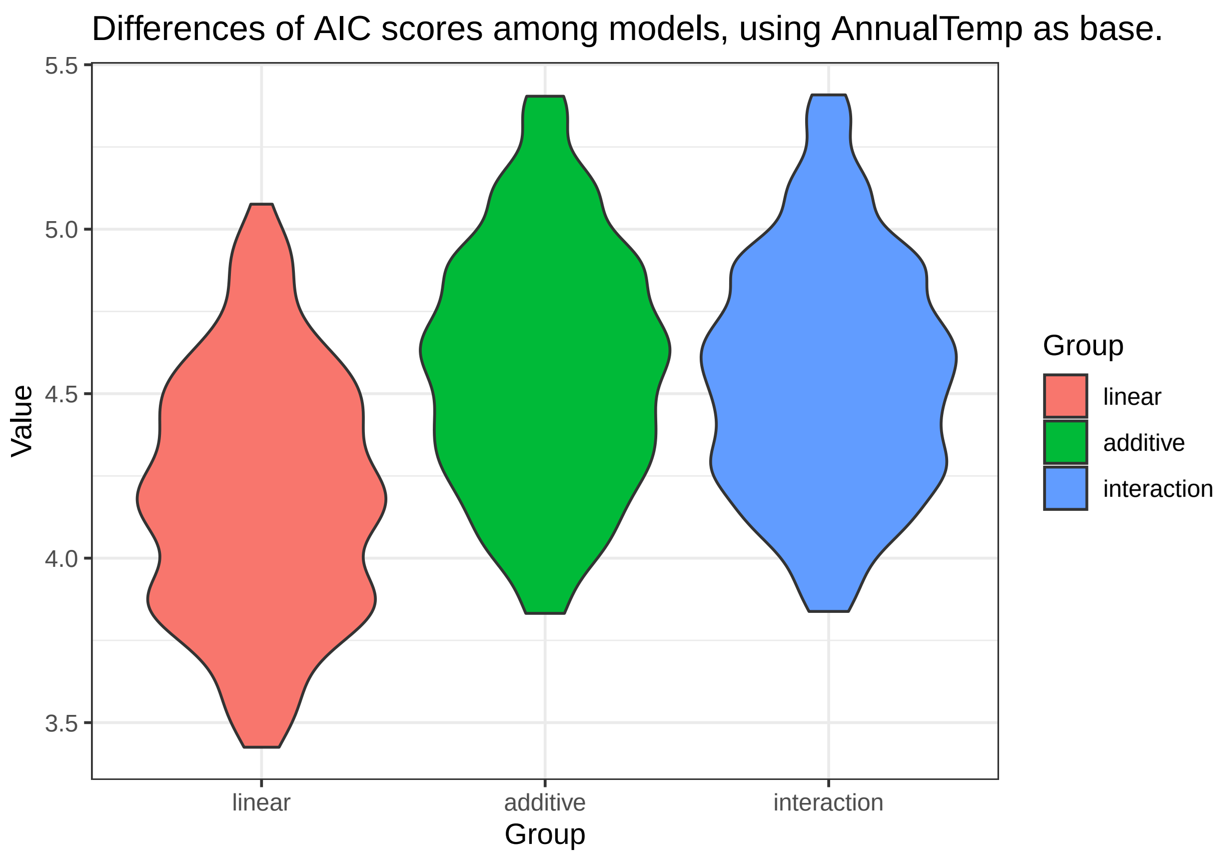
```r
table(count)
```

```
## count
## 1 2
## 9 9
```

- No preference among the four models.

#### Compare between annual and cold

```r
# Calculate the differences of AIC values
aic.bs <- matrix(NA,1000,3) # store the differences in AIC values
aic.bs[,1] <- linear.results[,12] - linear.results[,6]
aic.bs[,2] <- add.results[,12] - add.results[,6]
aic.bs[,3] <- int.results[,12] - int.results[,6]

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

# 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("linear","additive","interaction")

# 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 AnnualTemp as base.")+
  theme_bw()
```




```r
## Check the AICc scores and akaike weights in ONLY LINEAR MODEL
weight.matrix <- matrix(NA, 1000, 2)
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])

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

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

summary(weight.matrix)
```

```
##         V1               V2
##  Min.   :0.8470   Min.   :0.0730
##  1st Qu.:0.8750   1st Qu.:0.0970
##  Median :0.8900   Median :0.1100
##  Mean   :0.8891   Mean   :0.1109
```

```
##   3rd Qu.:0.9030   3rd Qu.:0.1250
##   Max.   :0.9270   Max.   :0.1530
```

```r
table(count)
```

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

- 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.
- 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.