Black/Asian carp model selection

Eddie Wu

2023-08-29

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 subsample.

For each species, we have four different models:

1. Simple linear model (same slope, same intercept)
2. Linear additive model (same slope, different intercept)
3. Interaction model (different slope, same intercept)
4. Group-specific model (different slope, different intercept)

And we consider two temperature metrics:

1. Annual temperature
2. Winter temperature (temperature from the coldest quarter)

```r
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
```

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.

Using annual temperature - no subsample

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

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

# 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)


## Compare the AICs
AIC(black.simple, black.linear, black.int, black.group)
```

```
##              df        AIC
## black.simple  3 -10.243418
## black.linear  4  -8.744915
## black.int     4  -9.987204
## black.group   5  -8.736108
```

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

Model	R2
Simple linear	0.3408870
Linear additive	0.3557418
Interaction	0.3911135
Grouped	0.4114918

```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
```

```
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
```

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

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

Using cold temperature - no subsample

```
# 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)
```

```r
# Compare the AICs
AIC(black.simple, black.linear, black.int, black.group)
```

```
##              df       AIC
## black.simple  3 -12.56341
## black.linear  4 -11.02060
## black.int     4 -12.29867
## black.group   5 -10.97660
```

```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)$r.squared, summary(black.linear)$r.squared,
         summary(black.int)$r.squared, summary(black.group)$r.squared)
)
kable(r_2)
```

Model	R2
Simple linear	0.4068538
Linear additive	0.4190531
Interaction	0.4518409
Grouped	0.4684751

```r
## 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
```

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

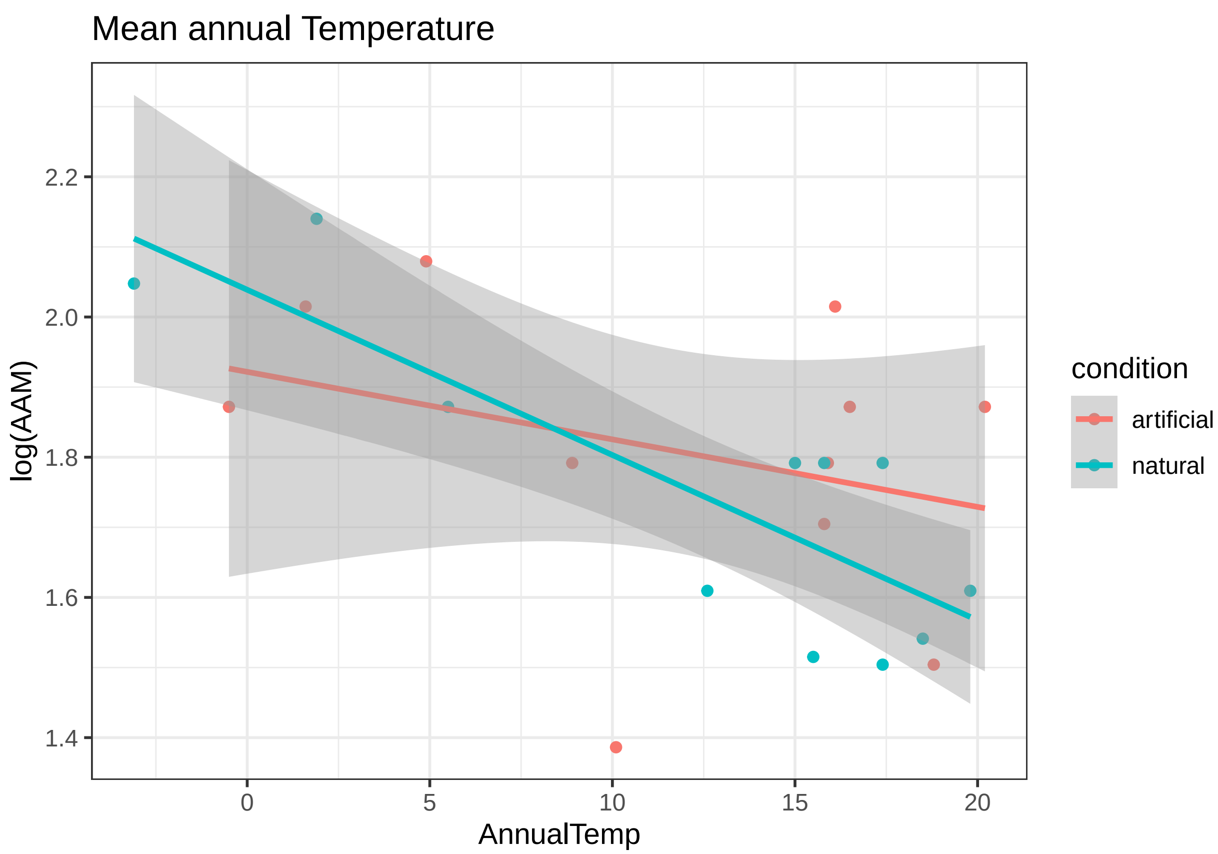
```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 20
##              f2 = 0.6056428
##       sig.level = 0.05
##           power = 0.9344838
```

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.

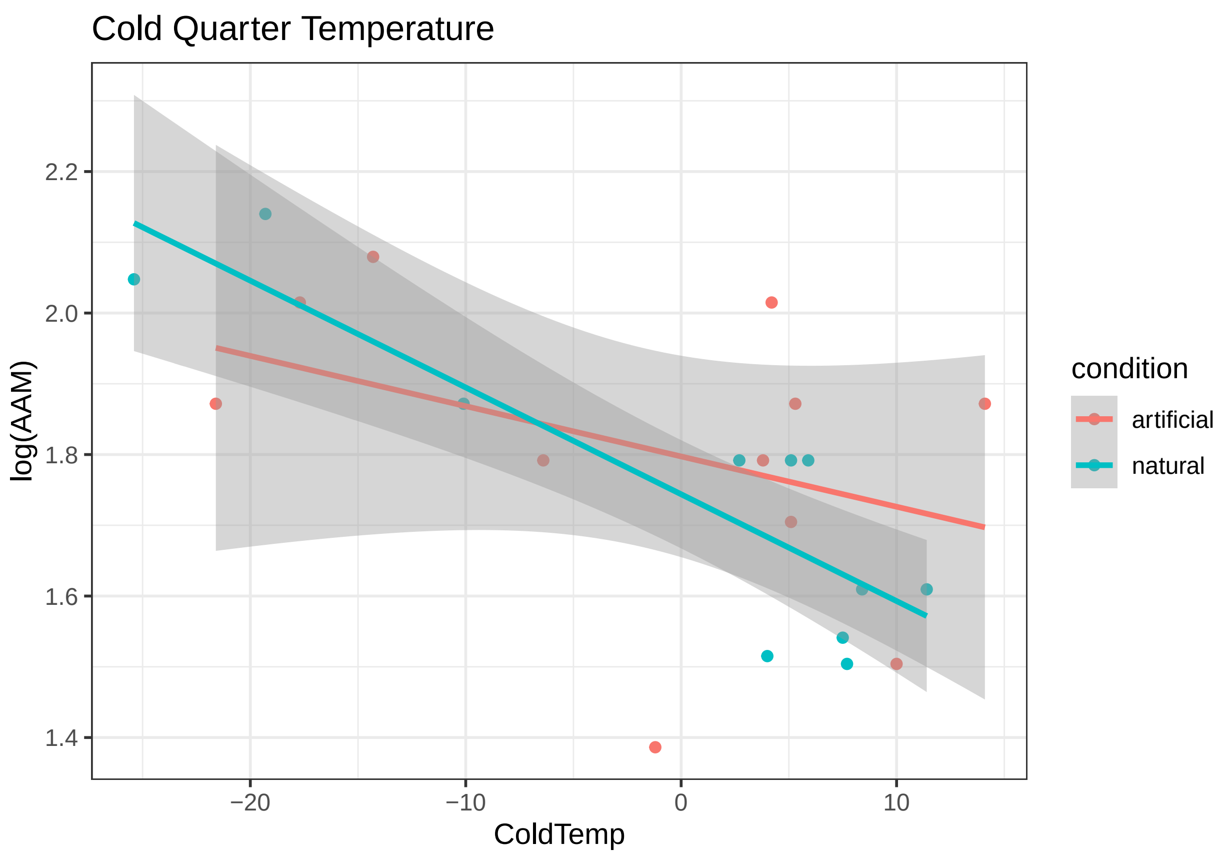
```
## Annual temperature
ggplot(black.clean, aes(x = AnnualTemp, y = log(AAM), color = condition))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_bw()+
  labs(title = "Mean annual Temperature")
```

```
## `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'
```



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.

Asian carp

Define the models and check the R2

```
## 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)
```

```
##
##  A AA AB AC AD AE AF AG AH AI AL AM AN AO AP  B  C  D  E  F  G  H  J  K  L  M
##  7  1  1  1  1  1  3  3  1  1  2  1  1  1  2  6  3  4  3  1  2  1  2  1  1  3
##  N  O  Q  S  X  Y  Z
##  1  3  1  3  1  1  3
```

```
## Subsampling for 1000 times (define all four models in one iteration!)
# Create the matrices to store the results
asian.linear.results <- matrix(NA,1000,4)
asian.add.results <- matrix(NA,1000,4)
asian.int.results <- matrix(NA,1000,4)
asian.group.results <- matrix(NA,1000,4)
```

```r
# Slopes for all models


# Iteration (put both annual and cold inside one iteration)
for(i in 1:1000){
  sub <- asian.carp.clean %>% group_by(Code) %>% 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)
  reg.group.annual <- lm(log(AAM)~AnnualTemp*Condition, data = sub)

  # AICs for annual
  asian.linear.results[i,1]<-as.numeric(AIC(reg.linear.annual))
  asian.add.results[i,1]<-as.numeric(AIC(reg.add.annual))
  asian.int.results[i,1]<-as.numeric(AIC(reg.int.annual))
  asian.group.results[i,1]<-as.numeric(AIC(reg.group.annual))

  # R2 for annual
  asian.linear.results[i,2]<-summary(reg.linear.annual)$adj.r.squared
  asian.add.results[i,2]<-summary(reg.add.annual)$adj.r.squared
  asian.int.results[i,2]<-summary(reg.int.annual)$adj.r.squared
  asian.group.results[i,2]<-summary(reg.group.annual)$adj.r.squared

  # 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)
  reg.group.cold <- lm(log(AAM)~ColdTemp*Condition, data = sub)

  # AICs for cold
  asian.linear.results[i,3]<-as.numeric(AIC(reg.linear.cold))
  asian.add.results[i,3]<-as.numeric(AIC(reg.add.cold))
  asian.int.results[i,3]<-as.numeric(AIC(reg.int.cold))
  asian.group.results[i,3]<-as.numeric(AIC(reg.group.cold))

  # R2 for cold
  asian.linear.results[i,4]<-summary(reg.linear.cold)$adj.r.squared
  asian.add.results[i,4]<-summary(reg.add.cold)$adj.r.squared
  asian.int.results[i,4]<-summary(reg.int.cold)$adj.r.squared
  asian.group.results[i,4]<-summary(reg.group.cold)$adj.r.squared
}


## R^2 values for the four models
# annual
r2annual <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction", "Grouped"),
  R2 = c(mean(unique(asian.linear.results[,2])),
         mean(unique(asian.add.results[,2])),
         mean(unique(asian.int.results[,2])),
         mean(unique(asian.group.results[,2])))
```

```
)
kable(r2annual)
```

Model	R2
Simple linear	0.5473078
Linear additive	0.5382194
Interaction	0.5467822
Grouped	0.5351531

```
# cold
r2cold <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction", "Grouped"),
  R2 = c(mean(unique(asian.linear.results[,4])),
         mean(unique(asian.add.results[,4])),
         mean(unique(asian.int.results[,4])),
         mean(unique(asian.group.results[,4])))
)
kable(r2cold)
```

Model	R2
Simple linear	0.5290966
Linear additive	0.5307475
Interaction	0.5164729
Grouped	0.5231827

Check the slopes for all Asian carp models

```
summary(reg.linear.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp, data = sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7570 -0.2010  0.1068  0.1957  0.4964
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.970269   0.087892  22.417  < 2e-16 ***
## AnnualTemp  -0.040316   0.006397  -6.302 5.21e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2872 on 31 degrees of freedom
## Multiple R-squared:  0.5617, Adjusted R-squared:  0.5475
## F-statistic: 39.72 on 1 and 31 DF,  p-value: 5.206e-07
```

```
summary(reg.add.annual)
```

```
##
## Call:
```

```
## lm(formula = log(AAM) ~ AnnualTemp + Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65988 -0.17276  0.05559  0.23052  0.43280
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.062319   0.112757  18.290  < 2e-16 ***
## AnnualTemp       -0.043834   0.006899  -6.353 5.22e-07 ***
## Conditionnatural -0.143829   0.112096  -1.283    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2843 on 30 degrees of freedom
## Multiple R-squared:  0.5845, Adjusted R-squared:  0.5568
## F-statistic:  21.1 on 2 and 30 DF,  p-value: 1.902e-06
```

```r
summary(reg.int.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp:Condition, data = sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5896 -0.1632  0.0636  0.1827  0.4564
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.002505   0.087215  22.961  < 2e-16 ***
## AnnualTemp:Conditionartificial -0.039368   0.006225  -6.324 5.66e-07 ***
## AnnualTemp:Conditionnatural    -0.055793   0.010888  -5.124 1.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2784 on 30 degrees of freedom
## Multiple R-squared:  0.6014, Adjusted R-squared:  0.5748
## F-statistic: 22.63 on 2 and 30 DF,  p-value: 1.019e-06
```

```r
summary(reg.group.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp * Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58954 -0.16235  0.06566  0.18010  0.45905
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.997744   0.126465  15.797 8.77e-16 ***
## AnnualTemp                  -0.039102   0.008085  -4.836 4.00e-05 ***
## Conditionnatural             0.009371   0.177426   0.053    0.958
```

```
## AnnualTemp:Conditionnatural -0.017055   0.015350  -1.111    0.276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2832 on 29 degrees of freedom
## Multiple R-squared:  0.6014, Adjusted R-squared:  0.5602
## F-statistic: 14.59 on 3 and 29 DF,  p-value: 5.626e-06
```

```r
summary(reg.linear.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77746 -0.21637  0.09048  0.18367  0.56114
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.485375   0.050847  29.213  < 2e-16 ***
## ColdTemp    -0.024618   0.003995  -6.163 7.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2908 on 31 degrees of freedom
## Multiple R-squared:  0.5506, Adjusted R-squared:  0.5361
## F-statistic: 37.98 on 1 and 31 DF,  p-value: 7.731e-07
```

```r
summary(reg.add.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp + Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63845 -0.21495  0.08095  0.23403  0.47587
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.556662   0.062878  24.757  < 2e-16 ***
## ColdTemp         -0.028298   0.004357  -6.495 3.52e-07 ***
## Conditionnatural -0.208090   0.114769  -1.813   0.0798 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2806 on 30 degrees of freedom
## Multiple R-squared:  0.595,  Adjusted R-squared:  0.568
## F-statistic: 22.03 on 2 and 30 DF,  p-value: 1.295e-06
```

```r
summary(reg.int.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp:Condition, data = sub)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77481 -0.20823  0.09672  0.18034  0.56473
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.483036   0.057725  25.692  < 2e-16 ***
## ColdTemp:Conditionartificial -0.024290   0.005428  -4.475 0.000102 ***
## ColdTemp:Conditionnatural    -0.025124   0.006889  -3.647 0.000997 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2956 on 30 degrees of freedom
## Multiple R-squared:  0.5507, Adjusted R-squared:  0.5208
## F-statistic: 18.39 on 2 and 30 DF,  p-value: 6.137e-06
```

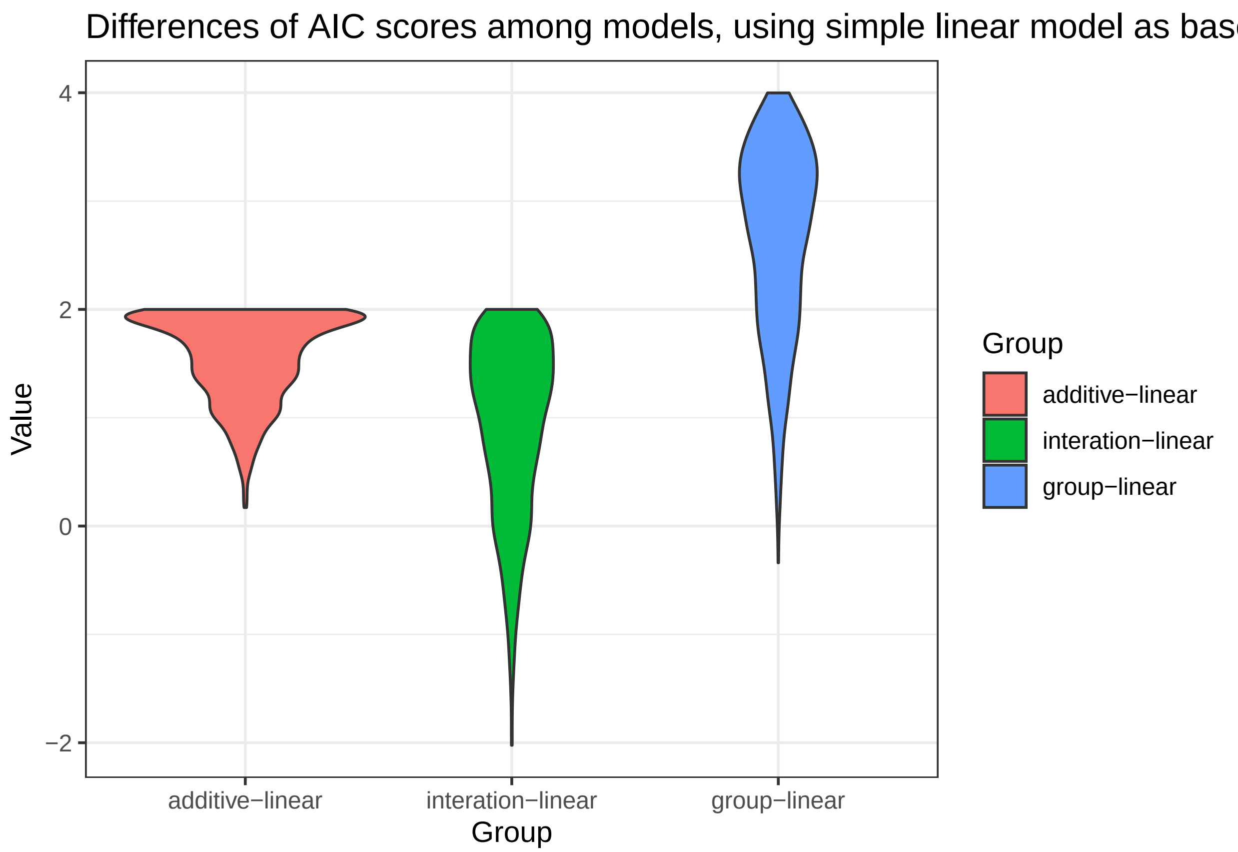
```r
summary(reg.group.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp * Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58030 -0.21211  0.05431  0.16923  0.49411
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.548226   0.063756  24.284  < 2e-16 ***
## ColdTemp                  -0.025719   0.005218  -4.929 3.09e-05 ***
## Conditionnatural          -0.254163   0.125888  -2.019   0.0528 .
## ColdTemp:Conditionnatural -0.008635   0.009548  -0.904   0.3732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2815 on 29 degrees of freedom
## Multiple R-squared:  0.6061, Adjusted R-squared:  0.5653
## F-statistic: 14.87 on 3 and 29 DF,  p-value: 4.761e-06
```

Compare AICs for annual

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

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




```
## Checking AIC values in each iteration for AnnualTemp
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(asian.linear.results[i,1], asian.add.results[i,1],
                 asian.int.results[i,1], asian.group.results[i,1])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])
```

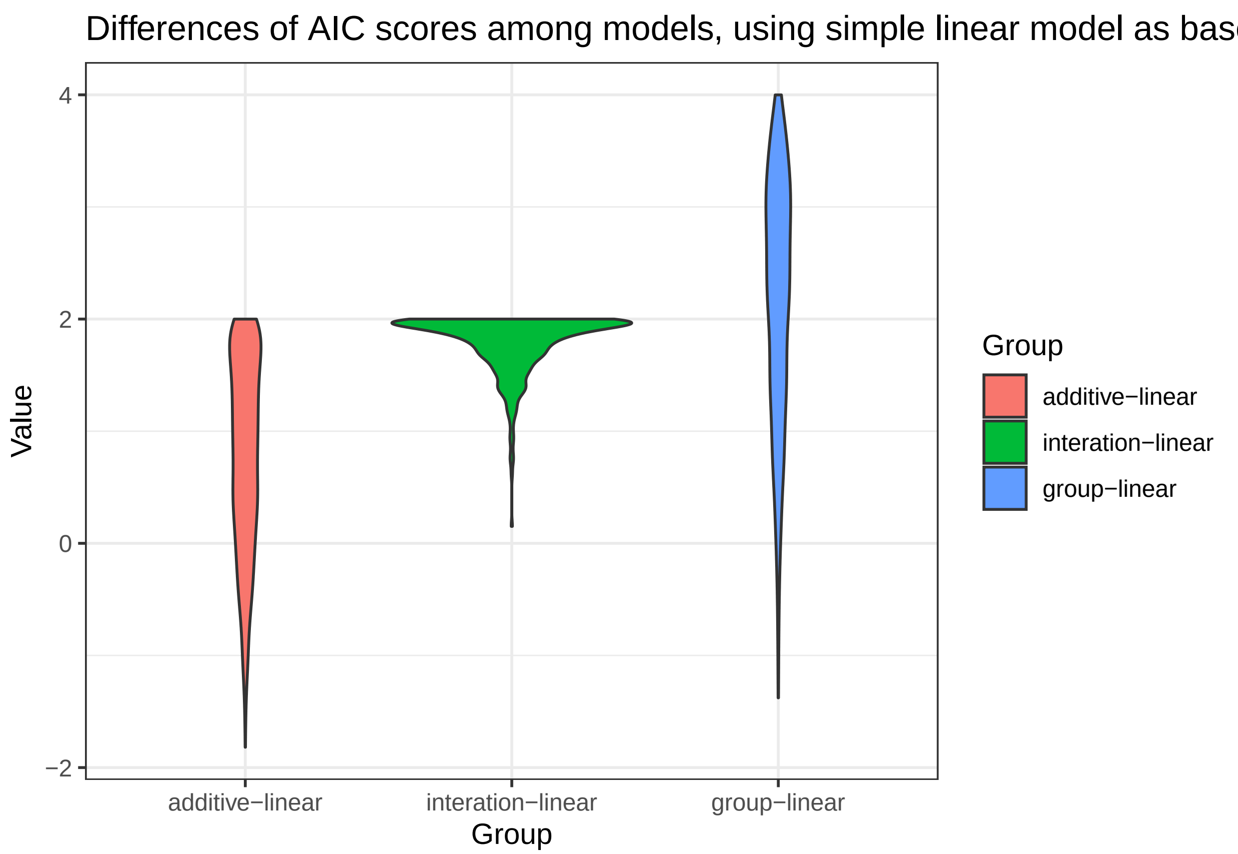
```
  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

```
## [1] NA
```

- When looking at each literation, we saw there were zero times where one AIC value was smaller than all other AIC values by two units.
- In general, the linear model had the smallest AIC values.

Compare AICs for the cold

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

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



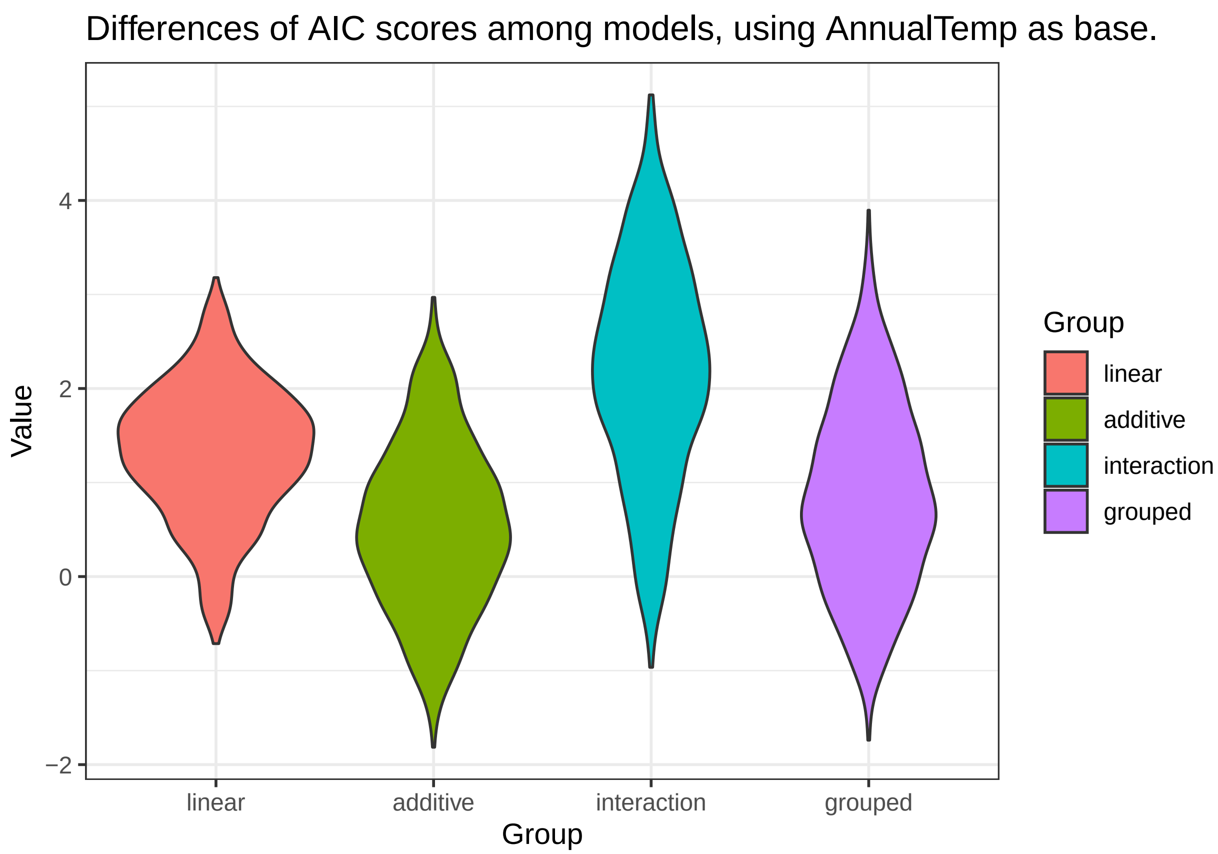

```r
## Checking AIC values in each iteration for ColdTemp
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(asian.linear.results[i,3], asian.add.results[i,3],
                 asian.int.results[i,3], asian.group.results[i,3])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])

  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

```
## [1] NA
```

- When looking at each literation, we saw there were zero times where one AIC value was smaller than all other AIC values by two units.
- In general, the linear model had the smallest AIC values.

Compare between annual and cold

```
# Calculate the differences of AIC values
aic.asian <- matrix(NA,1000,4) # store the differences in AIC values
aic.asian[,1] <- asian.linear.results[,3] - asian.linear.results[,1]
aic.asian[,2] <- asian.add.results[,3] - asian.add.results[,1]
aic.asian[,3] <- asian.int.results[,3] - asian.int.results[,1]
aic.asian[,4] <- asian.group.results[,3] - asian.group.results[,1]

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

# 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","grouped")

# 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
## Checking AIC values in each iteration
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(asian.linear.results[i,1], asian.linear.results[i,3])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])

  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}

count
```

```
##   [1] NA  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [26]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [51]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [76]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
## [101]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
## [126]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
## [151]  1  1  1  1  1  1  1  1  1  1  1  1
```

- With the simple linear model, around 15% of the time when using Coldtemp is preferred over using

AnnualTemp.

Grass carp

Define and models and check the R2

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

table(Grass.clean$Code)
```

```
##
##  A AA AB AE AF AG AH AL AM AN AO AP  B  C  D  E  F  G  J  K  L  M  O  Q  S  X
##  3  1  1  1  1  1  1  1  1  1  1  1  3  2  2  1  1  2  1  1  1  1  1  1  1  1
##  Y  Z
##  1  1
```

```
## Subsampling for 1000 times (define all four models in one iteration!)
# Create the matrices to store the results
grass.linear.results <- matrix(NA,1000,4)
grass.add.results <- matrix(NA,1000,4)
grass.int.results <- matrix(NA,1000,4)
grass.group.results <- matrix(NA,1000,4)

# Iteration (put both annual and cold inside one iteration)
for(i in 1:1000){
  sub <- Grass.clean %>% group_by(Code) %>% 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)
  reg.group.annual <- lm(log(AAM)~AnnualTemp*Condition, data = sub)

  # AICs for annual
  grass.linear.results[i,1]<-as.numeric(AIC(reg.linear.annual))
  grass.add.results[i,1]<-as.numeric(AIC(reg.add.annual))
  grass.int.results[i,1]<-as.numeric(AIC(reg.int.annual))
  grass.group.results[i,1]<-as.numeric(AIC(reg.group.annual))

  # R2 for annual
  grass.linear.results[i,2]<-summary(reg.linear.annual)$adj.r.squared
  grass.add.results[i,2]<-summary(reg.add.annual)$adj.r.squared
  grass.int.results[i,2]<-summary(reg.int.annual)$adj.r.squared
  grass.group.results[i,2]<-summary(reg.group.annual)$adj.r.squared

  # 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)
  reg.group.cold <- lm(log(AAM)~ColdTemp*Condition, data = sub)

  # AICs for cold
  grass.linear.results[i,3]<-as.numeric(AIC(reg.linear.cold))
  grass.add.results[i,3]<-as.numeric(AIC(reg.add.cold))
```

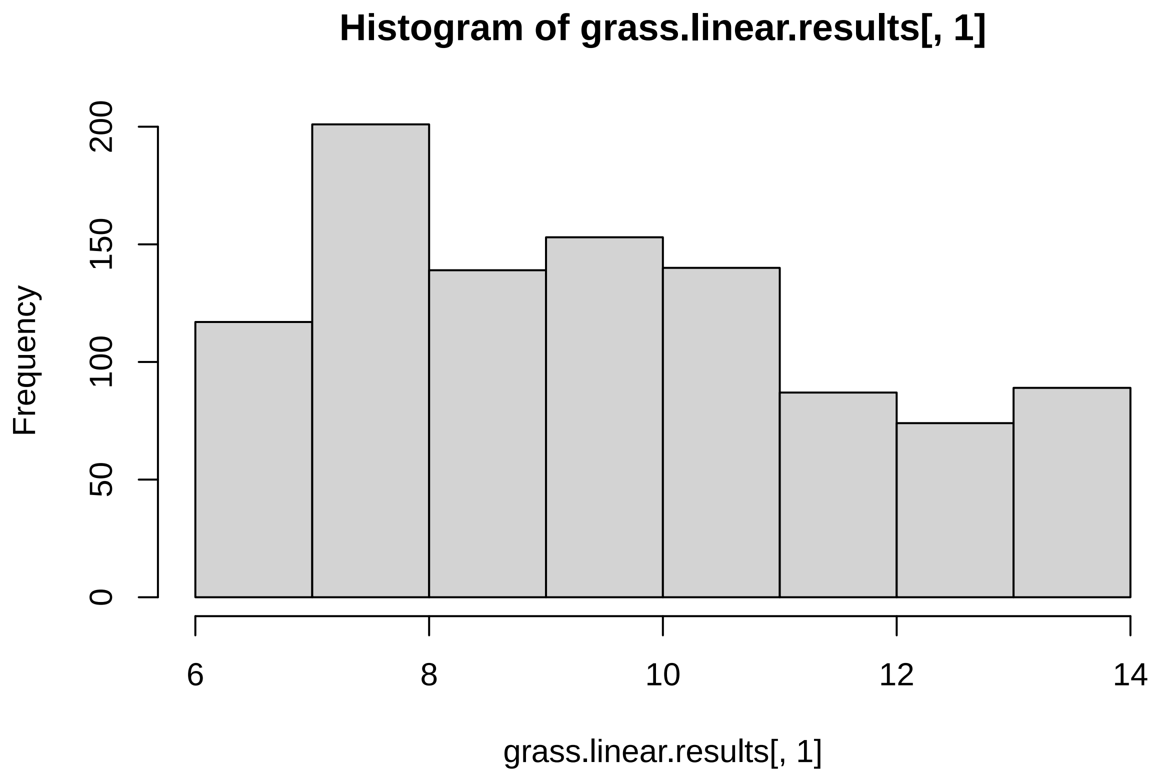
```r
  grass.int.results[i,3]<-as.numeric(AIC(reg.int.cold))
  grass.group.results[i,3]<-as.numeric(AIC(reg.group.cold))

  # R2 for cold
  grass.linear.results[i,4]<-summary(reg.linear.cold)$adj.r.squared
  grass.add.results[i,4]<-summary(reg.add.cold)$adj.r.squared
  grass.int.results[i,4]<-summary(reg.int.cold)$adj.r.squared
  grass.group.results[i,4]<-summary(reg.group.cold)$adj.r.squared
}


## R^2 values for the four models
# annual
r2annual <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction", "Grouped"),
  R2 = c(mean(unique(grass.linear.results[,2])),
         mean(unique(grass.add.results[,2])),
         mean(unique(grass.int.results[,2])),
         mean(unique(grass.group.results[,2])))
)
kable(r2annual)
```

Model	R2
Simple linear	0.6346557
Linear additive	0.6299637
Interaction	0.6274822
Grouped	0.6156102

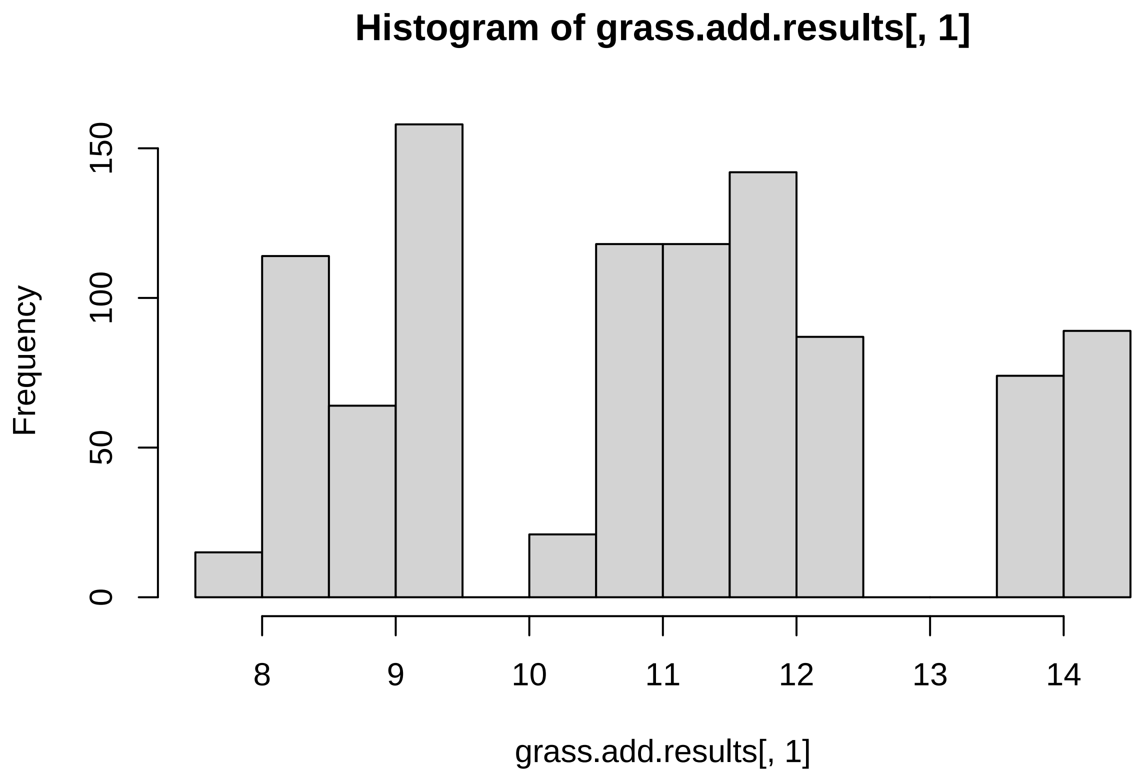
```r
# cold
r2cold <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction", "Grouped"),
  R2 = c(mean(unique(grass.linear.results[,4])),
         mean(unique(grass.add.results[,4])),
         mean(unique(grass.int.results[,4])),
         mean(unique(grass.group.results[,4])))
)
kable(r2cold)
```

Model	R2
Simple linear	0.6399651
Linear additive	0.6283568
Interaction	0.6277322
Grouped	0.6137069

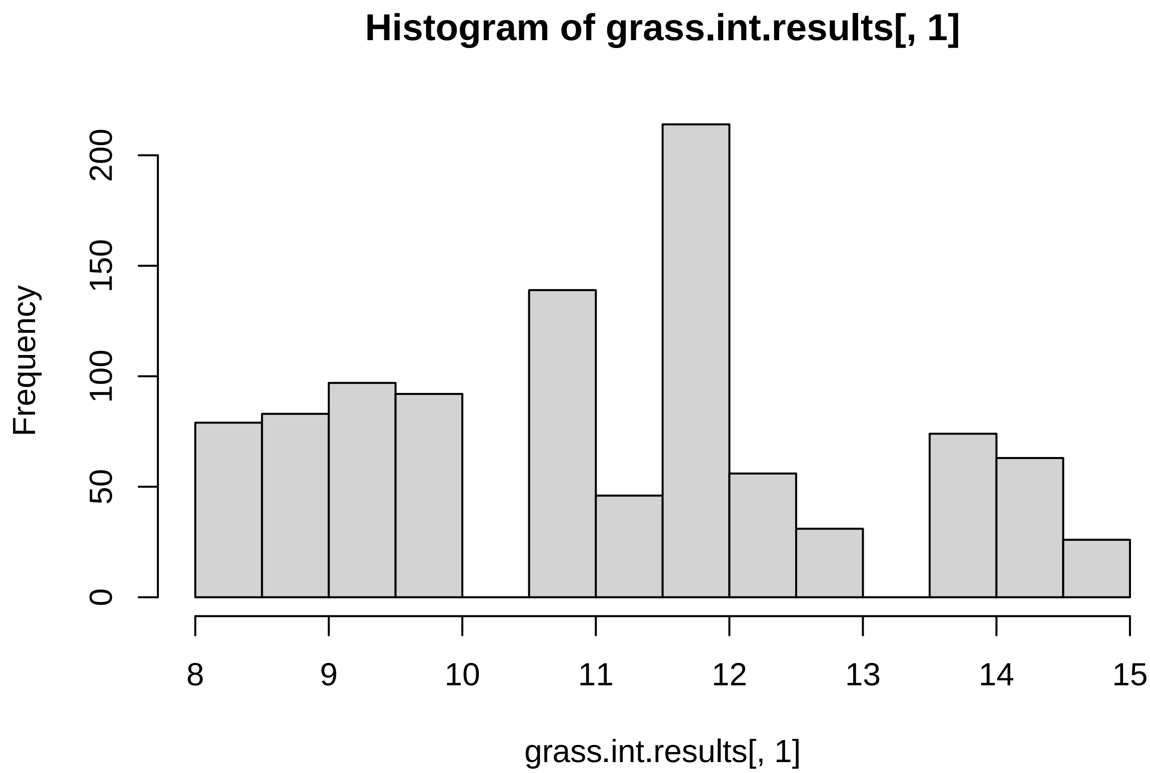
```r
hist(grass.linear.results[,1], breaks = 10)
```



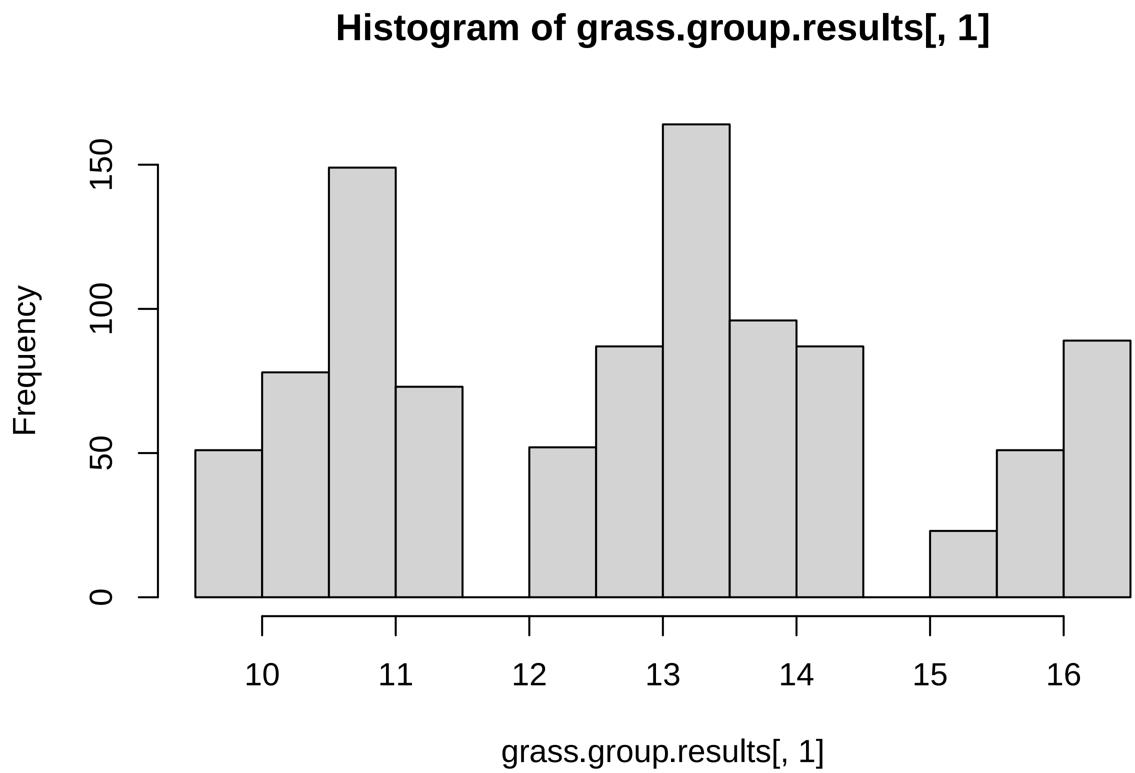

```r
hist(grass.add.results[,1], breaks = 10)
```




```r
hist(grass.int.results[,1], breaks = 10)
```




```
hist(grass.group.results[,1], breaks = 10)
```



Check the slopes for all Grass carp models

```r
summary(reg.linear.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp, data = sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5990 -0.1753  0.0403  0.1453  0.4938
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.04946    0.08994  22.787  < 2e-16 ***
## AnnualTemp  -0.04206    0.00645  -6.522  6.5e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2851 on 26 degrees of freedom
## Multiple R-squared:  0.6206, Adjusted R-squared:  0.606
## F-statistic: 42.53 on 1 and 26 DF,  p-value: 6.5e-07
```

```r
summary(reg.add.annual)
```

```
##
```

```
## Call:
## lm(formula = log(AAM) ~ AnnualTemp + Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54118 -0.16426  0.03606  0.13139  0.51666
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.98240    0.11631  17.044 2.83e-15 ***
## AnnualTemp        -0.04010    0.00682  -5.879 3.91e-06 ***
## Conditionnatural   0.10519    0.11513   0.914     0.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.286 on 25 degrees of freedom
## Multiple R-squared:  0.6329, Adjusted R-squared:  0.6035
## F-statistic: 21.55 on 2 and 25 DF,  p-value: 3.633e-06
```

```r
summary(reg.int.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp:Condition, data = sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5820 -0.1488  0.0177  0.1496  0.4765
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.038943   0.091343  22.322  < 2e-16 ***
## AnnualTemp:Conditionartificial -0.043433   0.006692  -6.490 8.49e-07 ***
## AnnualTemp:Conditionnatural    -0.036021   0.009724  -3.704  0.00105 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2868 on 25 degrees of freedom
## Multiple R-squared:  0.6309, Adjusted R-squared:  0.6014
## F-statistic: 21.37 on 2 and 25 DF,  p-value: 3.886e-06
```

```r
summary(reg.group.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp * Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55042 -0.15247  0.03245  0.12756  0.50763
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.996596   0.138893  14.375 2.73e-13 ***
## AnnualTemp                 -0.041152   0.008789  -4.682 9.31e-05 ***
```

```
## Conditionnatural             0.076630   0.186841   0.410    0.685
## AnnualTemp:Conditionnatural  0.002825   0.014376   0.196    0.846
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2917 on 24 degrees of freedom
## Multiple R-squared:  0.6335, Adjusted R-squared:  0.5876
## F-statistic: 13.83 on 3 and 24 DF,  p-value: 1.927e-05
```

```
summary(reg.linear.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54692 -0.20497  0.05263  0.15517  0.55238
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.544491   0.053207  29.028  < 2e-16 ***
## ColdTemp    -0.026040   0.003882  -6.707 4.08e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2802 on 26 degrees of freedom
## Multiple R-squared:  0.6337, Adjusted R-squared:  0.6196
## F-statistic: 44.99 on 1 and 26 DF,  p-value: 4.075e-07
```

```
summary(reg.add.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp + Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52856 -0.20092  0.04573  0.15489  0.56980
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.530931   0.072851  21.015  < 2e-16 ***
## ColdTemp         -0.025551   0.004326  -5.907 3.65e-06 ***
## Conditionnatural  0.033189   0.119203   0.278    0.783
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2853 on 25 degrees of freedom
## Multiple R-squared:  0.6349, Adjusted R-squared:  0.6057
## F-statistic: 21.73 on 2 and 25 DF,  p-value: 3.394e-06
```

```
summary(reg.int.cold)
```

```
##
## Call:
```

```
## lm(formula = log(AAM) ~ ColdTemp:Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55531 -0.20272  0.04686  0.15275  0.54497
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.547784   0.059149  26.168  < 2e-16 ***
## ColdTemp:Conditionartificial -0.026560   0.005440  -4.882 5.06e-05 ***
## ColdTemp:Conditionnatural    -0.025356   0.006296  -4.028 0.000461 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2856 on 25 degrees of freedom
## Multiple R-squared:  0.634,  Adjusted R-squared:  0.6047
## F-statistic: 21.65 on 2 and 25 DF,  p-value: 3.494e-06
```

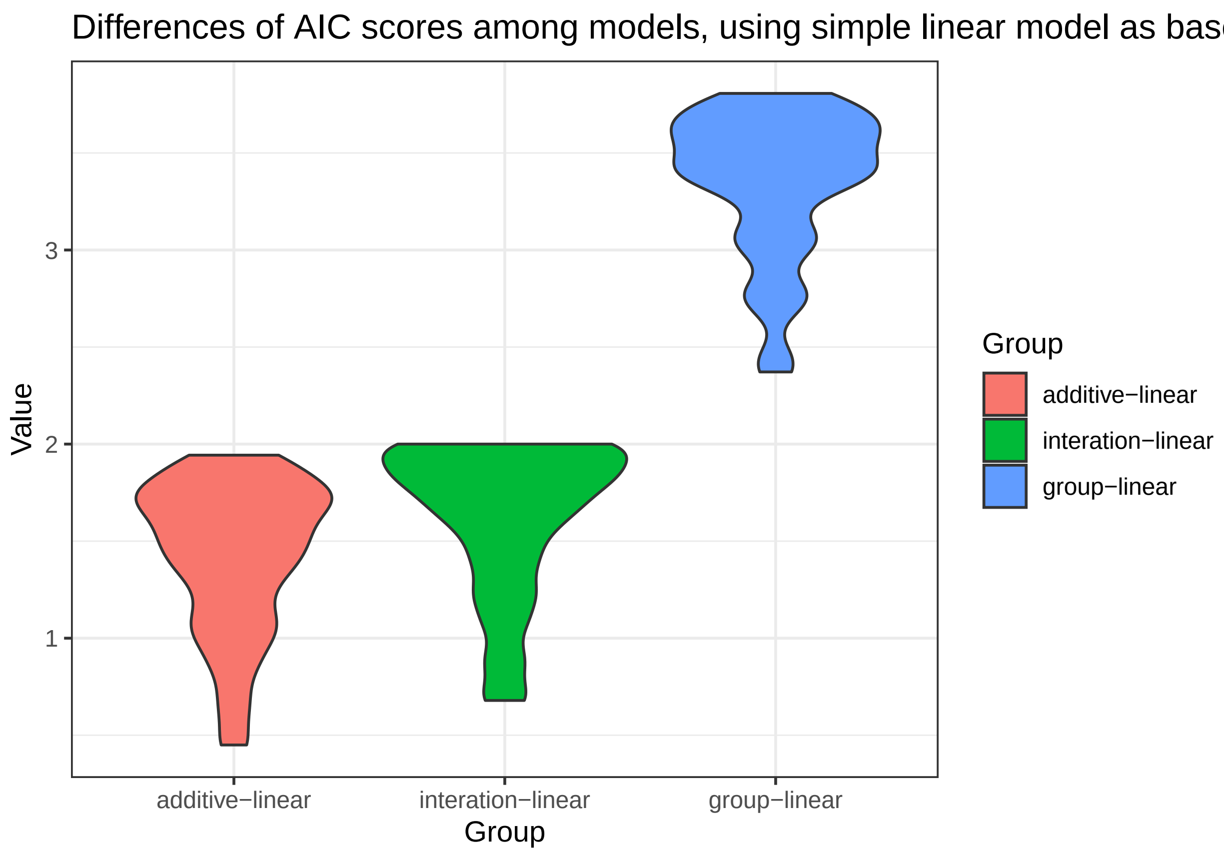
```r
summary(reg.group.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp * Condition, data = sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5383 -0.1893  0.0363  0.1639  0.5615
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.533462   0.075230  20.384  < 2e-16 ***
## ColdTemp                 -0.026287   0.005607  -4.688 9.18e-05 ***
## Conditionnatural          0.039918   0.125599   0.318    0.753
## ColdTemp:Conditionnatural  0.001931   0.009081   0.213    0.833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2909 on 24 degrees of freedom
## Multiple R-squared:  0.6355, Adjusted R-squared:   0.59
## F-statistic: 13.95 on 3 and 24 DF,  p-value: 1.802e-05
```

Compare AICs for annual

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

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

```r
# 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","group-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
## Checking AIC values in each iteration for ColdTemp
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(grass.linear.results[i,1], grass.add.results[i,1],
                 grass.int.results[i,1], grass.group.results[i,1])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])
```

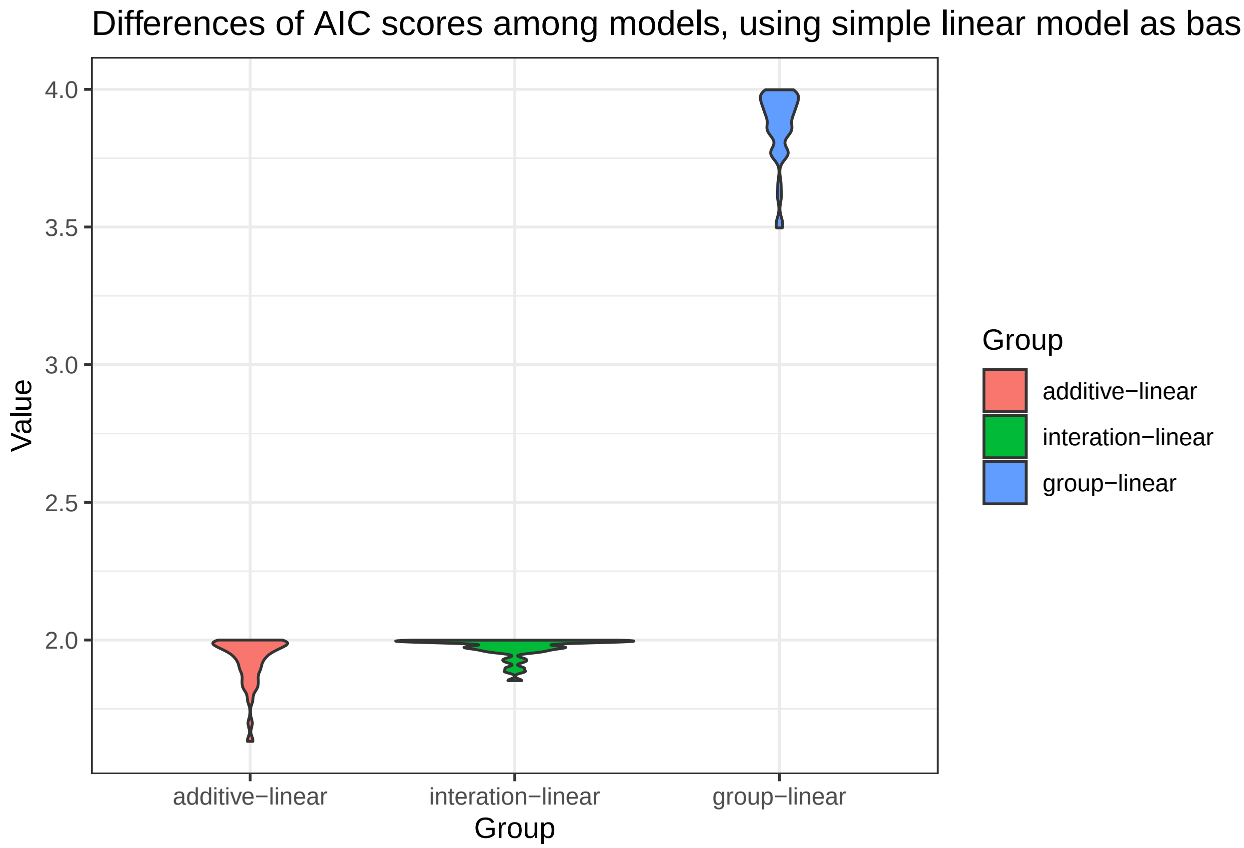
```r
  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

```
## [1] NA
```

- 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).
- In each iteration, there was never once that one model was effectively preferred (AIC smaller than that of all the other models).

Compare AICs for the cold

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

# Look at the distribution of differences
# Create a data frame
data <- as.data.frame(aic.grass)
colnames(data) <- c("additive-linear","interation-linear","group-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","group-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
## Checking AIC values in each iteration for ColdTemp
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(grass.linear.results[i,3], grass.add.results[i,3],
                 grass.int.results[i,3], grass.group.results[i,3])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])

  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

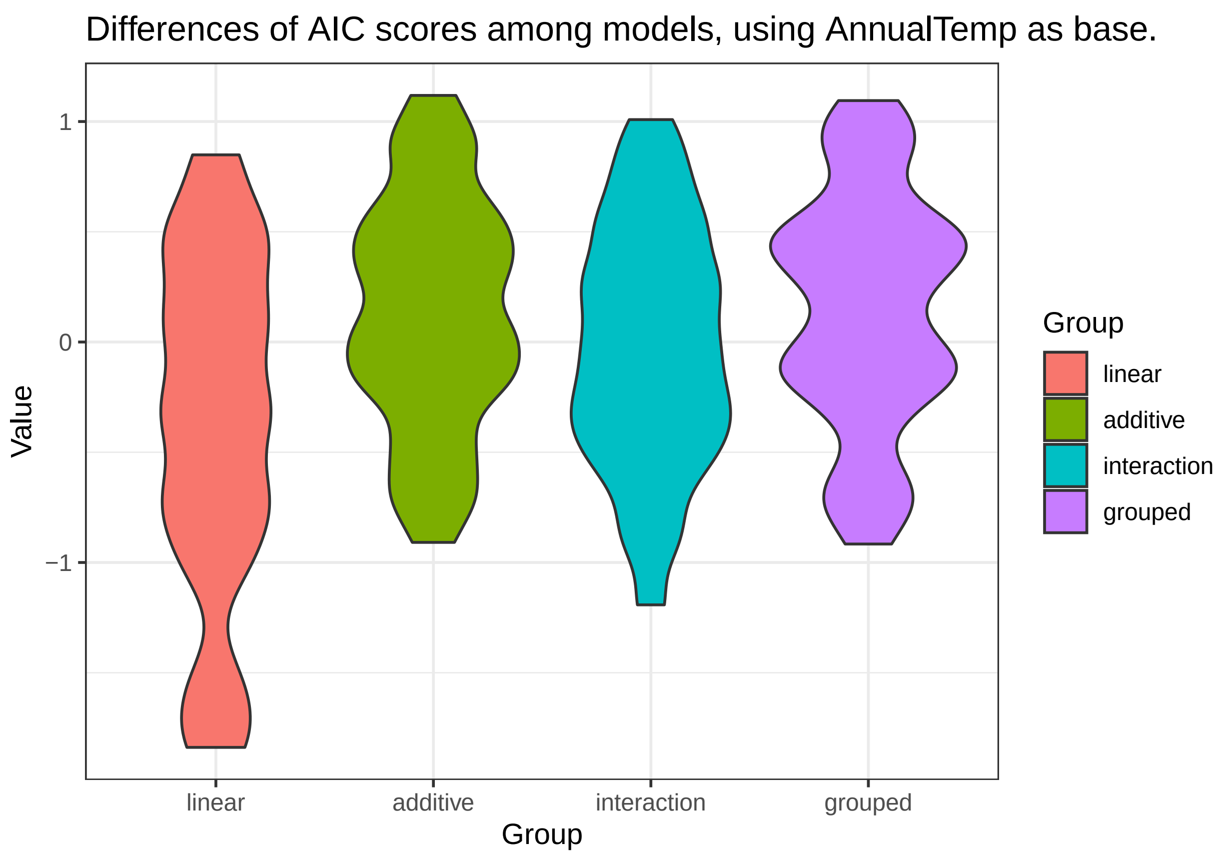
```
## [1] NA
```

- Same conclusion as AnnualTemp.

Compare between annual and cold

```r
# Calculate the differences of AIC values
aic.grass <- matrix(NA,1000,4) # store the differences in AIC values
aic.grass[,1] <- grass.linear.results[,3] - grass.linear.results[,1]
```

```r
aic.grass[,2] <- grass.add.results[,3] - grass.add.results[,1]
aic.grass[,3] <- grass.int.results[,3] - grass.int.results[,1]
aic.grass[,4] <- grass.group.results[,3] - grass.group.results[,1]

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

# 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","grouped")

# 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
## Checking AIC values in each iteration
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(grass.linear.results[i,1], grass.linear.results[i,3])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])

  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

```
## [1] NA
```

```r
summary(count)
```

```
##    Mode    NA's
## logical       1
```

- Cold temperature did not show any preference over Annual temperature.

Bighead and silver carp

Define the models and compare the R2

```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
## Subsampling for 1000 times (define all four models in one iteration!)
# Create the matrices to store the results
bs.linear.results <- matrix(NA,1000,4)
bs.add.results <- matrix(NA,1000,4)
bs.int.results <- matrix(NA,1000,4)
bs.group.results <- matrix(NA,1000,4)

# Iteration (put both annual and cold inside one iteration)
for(i in 1:1000){
  sub <- Big.sil.clean %>% group_by(Code) %>% 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)
  reg.group.annual <- lm(log(AAM)~AnnualTemp*Condition, data = sub)
```

```
  # AICs for annual
  bs.linear.results[i,1]<-as.numeric(AIC(reg.linear.annual))
  bs.add.results[i,1]<-as.numeric(AIC(reg.add.annual))
  bs.int.results[i,1]<-as.numeric(AIC(reg.int.annual))
  bs.group.results[i,1]<-as.numeric(AIC(reg.group.annual))

  # R2 for annual
  bs.linear.results[i,2]<-summary(reg.linear.annual)$adj.r.squared
  bs.add.results[i,2]<-summary(reg.add.annual)$adj.r.squared
  bs.int.results[i,2]<-summary(reg.int.annual)$adj.r.squared
  bs.group.results[i,2]<-summary(reg.group.annual)$adj.r.squared

  # 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)
  reg.group.cold <- lm(log(AAM)~ColdTemp*Condition, data = sub)

  # AICs for cold
  bs.linear.results[i,3]<-as.numeric(AIC(reg.linear.cold))
  bs.add.results[i,3]<-as.numeric(AIC(reg.add.cold))
  bs.int.results[i,3]<-as.numeric(AIC(reg.int.cold))
  bs.group.results[i,3]<-as.numeric(AIC(reg.group.cold))

  # R2 for cold
  bs.linear.results[i,4]<-summary(reg.linear.cold)$adj.r.squared
  bs.add.results[i,4]<-summary(reg.add.cold)$adj.r.squared
  bs.int.results[i,4]<-summary(reg.int.cold)$adj.r.squared
  bs.group.results[i,4]<-summary(reg.group.cold)$adj.r.squared
}


## R^2 values for the four models
# annual
r2annual <- data.frame(
  Model = c("Simple linear", "Linear additive", "Interaction", "Grouped"),
  R2 = c(mean(unique(bs.linear.results[,2])),
         mean(unique(bs.add.results[,2])),
         mean(unique(bs.int.results[,2])),
         mean(unique(bs.group.results[,2])))
)
kable(r2annual)
```

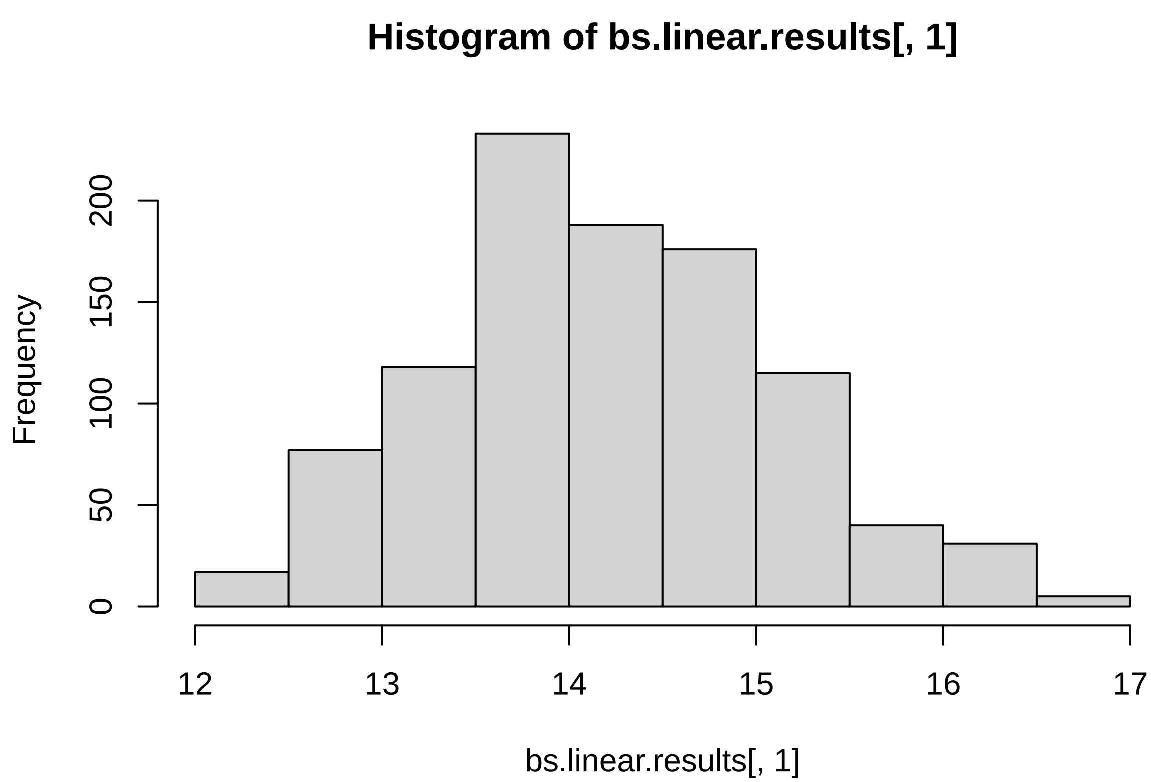
Model	R2
Simple linear	0.3385743
Linear additive	0.4228221
Interaction	0.3875880
Grouped	0.3872831

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

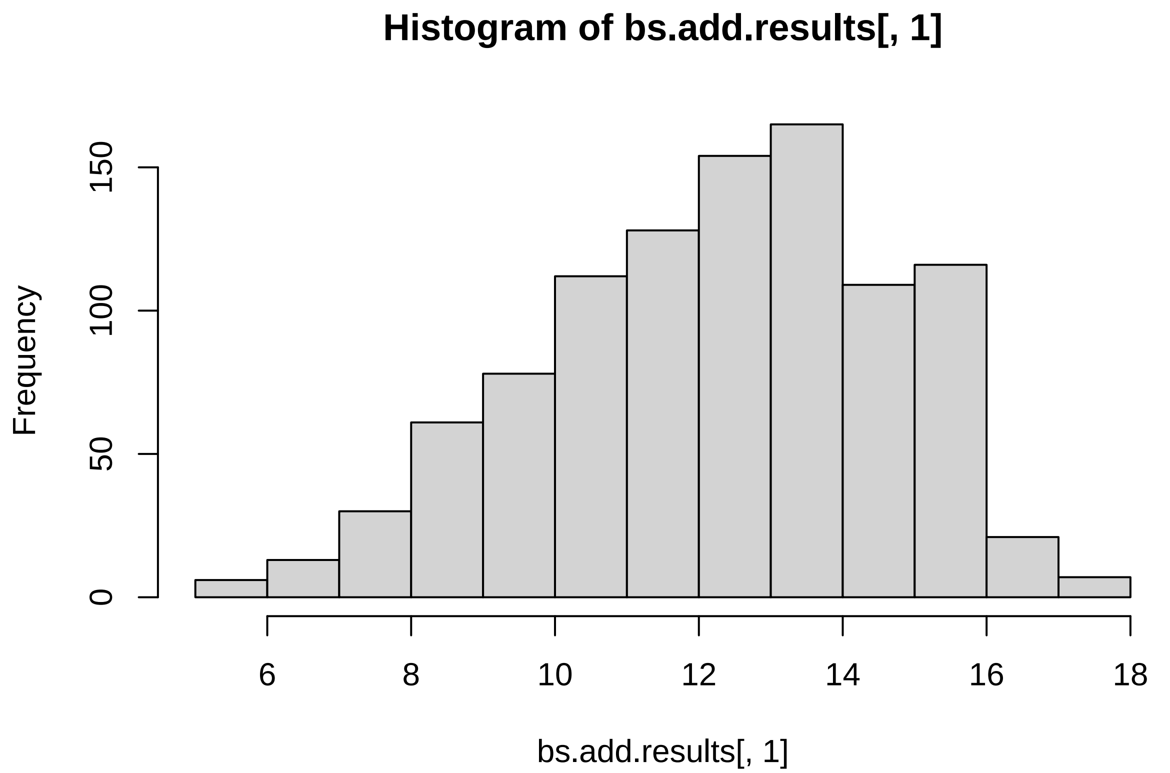
```r
  R2 = c(mean(unique(bs.linear.results[,4])),
         mean(unique(bs.add.results[,4])),
         mean(unique(bs.int.results[,4])),
         mean(unique(bs.group.results[,4])))
)
kable(r2cold)
```

Model	R2
Simple linear	0.2314162
Linear additive	0.3158189
Interaction	0.1929733
Grouped	0.2719971

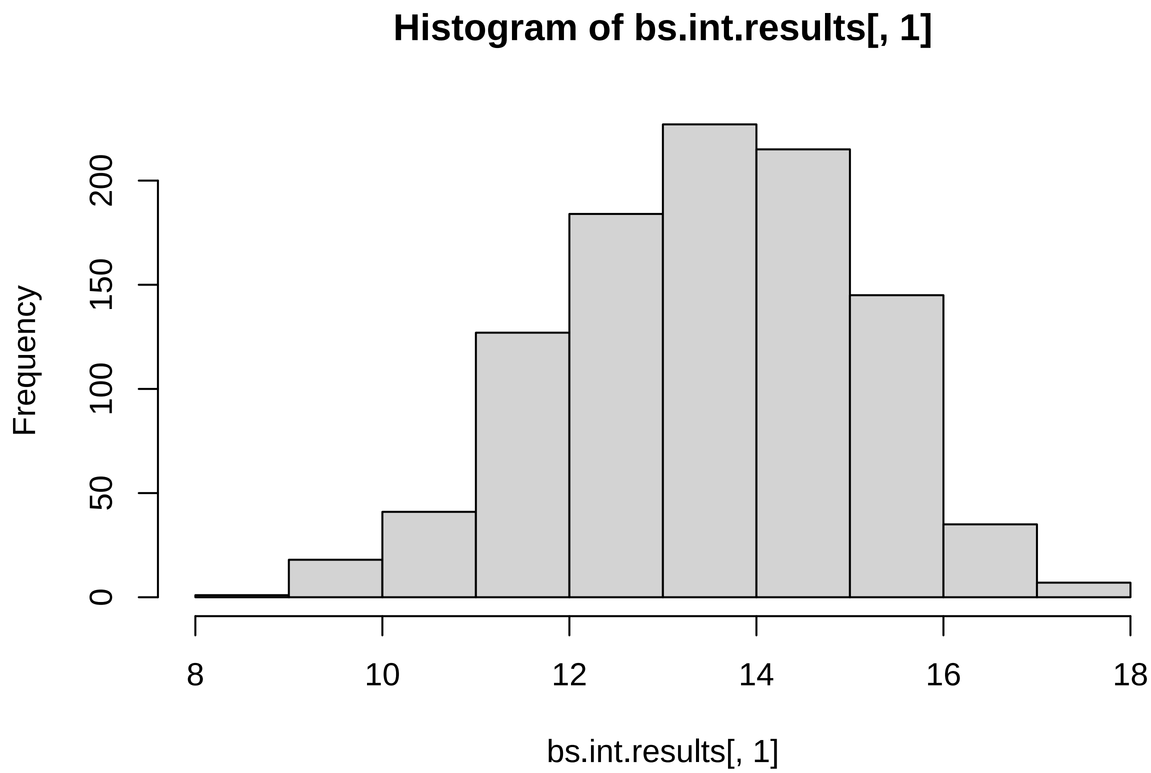
```r
hist(bs.linear.results[,1], breaks = 10)
```



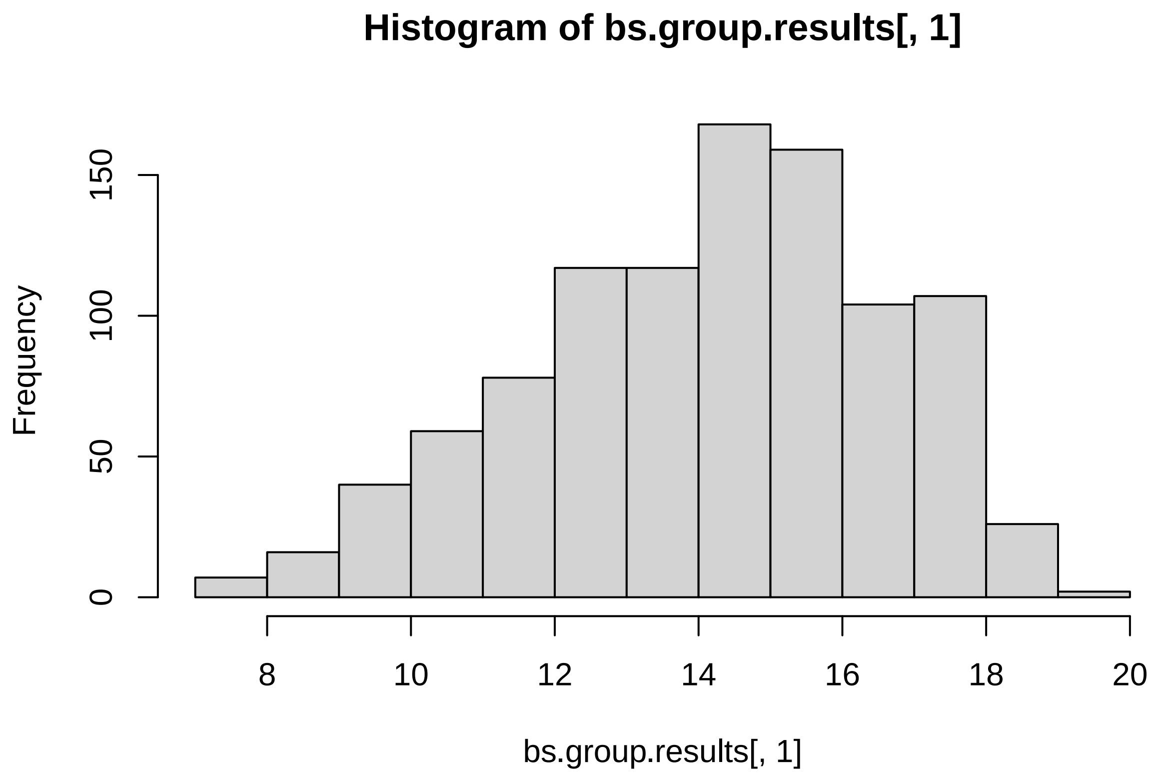

```r
hist(bs.add.results[,1], breaks = 10)
```




```r
hist(bs.int.results[,1], breaks = 10)
```




```r
hist(bs.group.results[,1], breaks = 10)
```



Check the slopes for all bighead and silver carp models

```r
summary(reg.linear.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79589 -0.17160  0.05009  0.13851  0.54185
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.92715    0.14289  13.487 1.65e-10 ***
## AnnualTemp  -0.03396    0.01058  -3.209  0.00515 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3154 on 17 degrees of freedom
## Multiple R-squared:  0.3772, Adjusted R-squared:  0.3406
## F-statistic:  10.3 on 1 and 17 DF,  p-value: 0.005147
```

```r
summary(reg.add.annual)
```

```
##
```

```
## Call:
## lm(formula = log(AAM) ~ AnnualTemp + Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67966 -0.14275  0.02962  0.10705  0.57158
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.05566    0.16129  12.745 8.56e-10 ***
## AnnualTemp       -0.03624    0.01030  -3.518  0.00285 **
## Conditionnatural -0.21539    0.14104  -1.527  0.14624
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3037 on 16 degrees of freedom
## Multiple R-squared:  0.4565, Adjusted R-squared:  0.3885
## F-statistic: 6.718 on 2 and 16 DF,  p-value: 0.007618
```

```r
summary(reg.int.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp:Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67847 -0.20931  0.04612  0.10394  0.51769
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.97064    0.14268  13.812 2.62e-10 ***
## AnnualTemp:Conditionartificial -0.03107    0.01052  -2.954  0.00933 **
## AnnualTemp:Conditionnatural    -0.04644    0.01369  -3.393  0.00372 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3072 on 16 degrees of freedom
## Multiple R-squared:  0.4439, Adjusted R-squared:  0.3744
## F-statistic: 6.387 on 2 and 16 DF,  p-value: 0.009143
```

```r
summary(reg.group.annual)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp * Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67454 -0.15745  0.02831  0.10653  0.56467
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.043274   0.188724  10.827 1.74e-08 ***
## AnnualTemp                 -0.035252   0.012765  -2.762   0.0145 *
```

```
## Conditionnatural            -0.179393   0.296602  -0.605   0.5543
## AnnualTemp:Conditionnatural -0.003212   0.023059  -0.139   0.8911
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3135 on 15 degrees of freedom
## Multiple R-squared:  0.4572, Adjusted R-squared:  0.3486
## F-statistic: 4.211 on 3 and 15 DF,  p-value: 0.02391
```

```
summary(reg.linear.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80209 -0.14713  0.02903  0.13536  0.64216
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.506902   0.077973   19.33 5.24e-13 ***
## ColdTemp    -0.019434   0.007418   -2.62   0.0179 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3374 on 17 degrees of freedom
## Multiple R-squared:  0.2876, Adjusted R-squared:  0.2457
## F-statistic: 6.863 on 1 and 17 DF,  p-value: 0.01794
```

```
summary(reg.add.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp + Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68817 -0.15713 -0.02781  0.13176  0.65174
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.604835   0.103966  15.436 4.97e-11 ***
## ColdTemp         -0.020983   0.007315  -2.868   0.0111 *
## Conditionnatural -0.210929   0.152848  -1.380   0.1866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3287 on 16 degrees of freedom
## Multiple R-squared:  0.3634, Adjusted R-squared:  0.2838
## F-statistic: 4.566 on 2 and 16 DF,  p-value: 0.02699
```

```
summary(reg.int.cold)
```

```
##
## Call:
```

```
## lm(formula = log(AAM) ~ ColdTemp:Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81077 -0.15611  0.03142  0.13896  0.61847
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.513064   0.081454  18.576 2.98e-12 ***
## ColdTemp:Conditionartificial -0.021652   0.009424  -2.298   0.0354 *
## ColdTemp:Conditionnatural    -0.015249   0.012959  -1.177   0.2565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.346 on 16 degrees of freedom
## Multiple R-squared:  0.2946, Adjusted R-squared:  0.2064
## F-statistic: 3.341 on 2 and 16 DF,  p-value: 0.0613
```

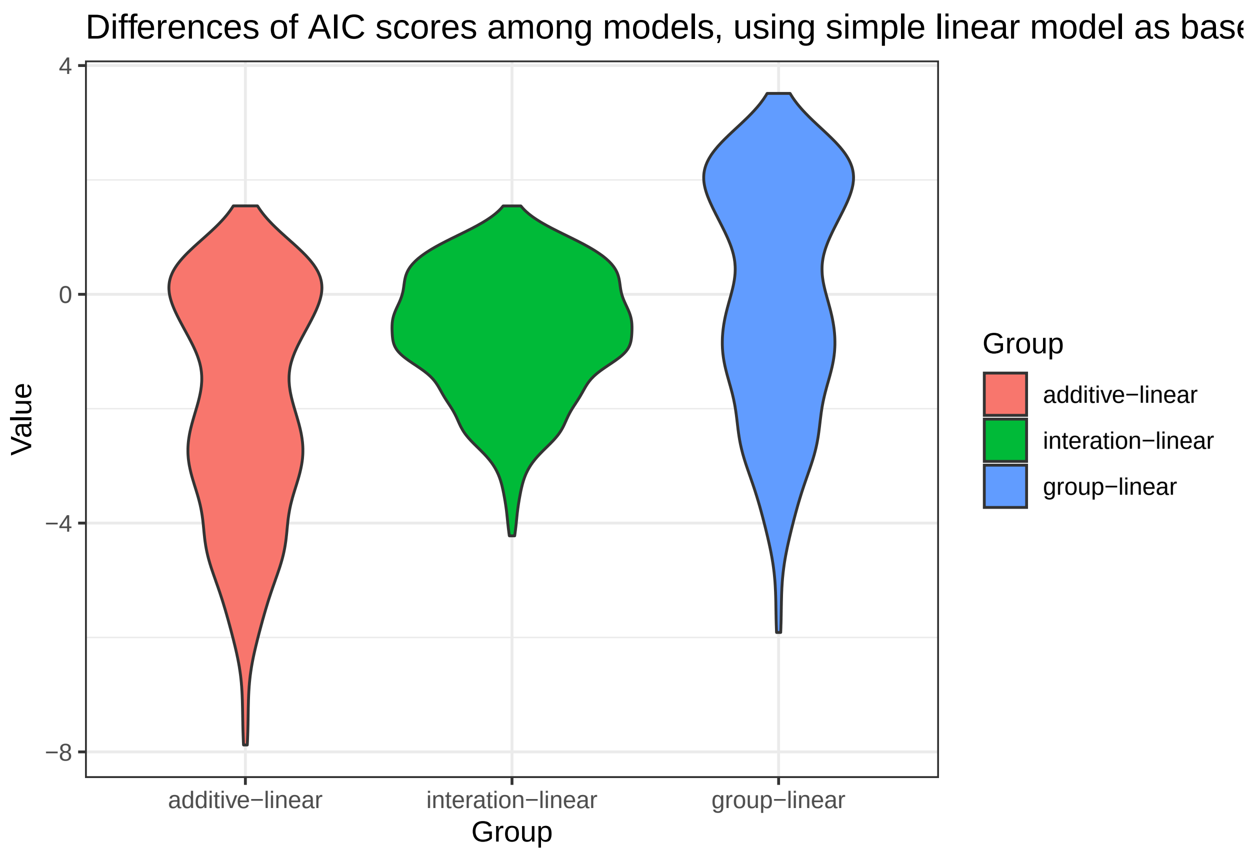
```
summary(reg.group.cold)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp * Condition, data = sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69413 -0.14895 -0.00903  0.12714  0.65430
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.605034   0.107296  14.959 2.02e-10 ***
## ColdTemp                 -0.021815   0.009239  -2.361   0.0322 *
## Conditionnatural         -0.206170   0.160646  -1.283   0.2188
## ColdTemp:Conditionnatural  0.002504   0.016025   0.156   0.8779
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3392 on 15 degrees of freedom
## Multiple R-squared:  0.3644, Adjusted R-squared:  0.2373
## F-statistic: 2.867 on 3 and 15 DF,  p-value: 0.07159
```

Compare AICs for annual

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

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

```r
# 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","group-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
## Checking AIC values in each iteration for AnnualTemp
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(bs.linear.results[i,1], bs.add.results[i,1],
                 bs.int.results[i,1], bs.group.results[i,1])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])
```

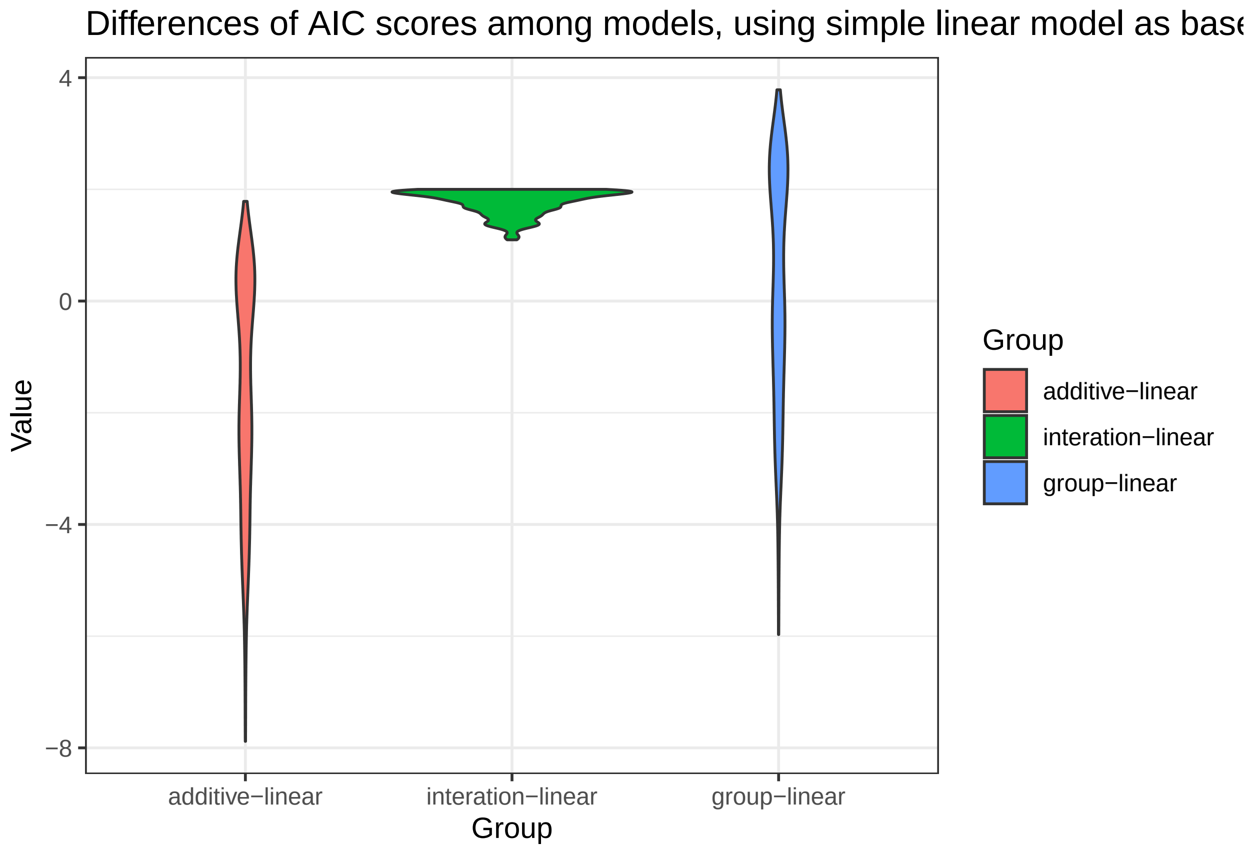
```
  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

```
## [1] NA
```

- We saw a large range in the difference of AIC values due to a larger number of combinations for subsampling sets. However, in each iteration, there was not once when the AIC of one model is more than 2 units smaller than that of the others.

Compare AICs for the cold

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

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



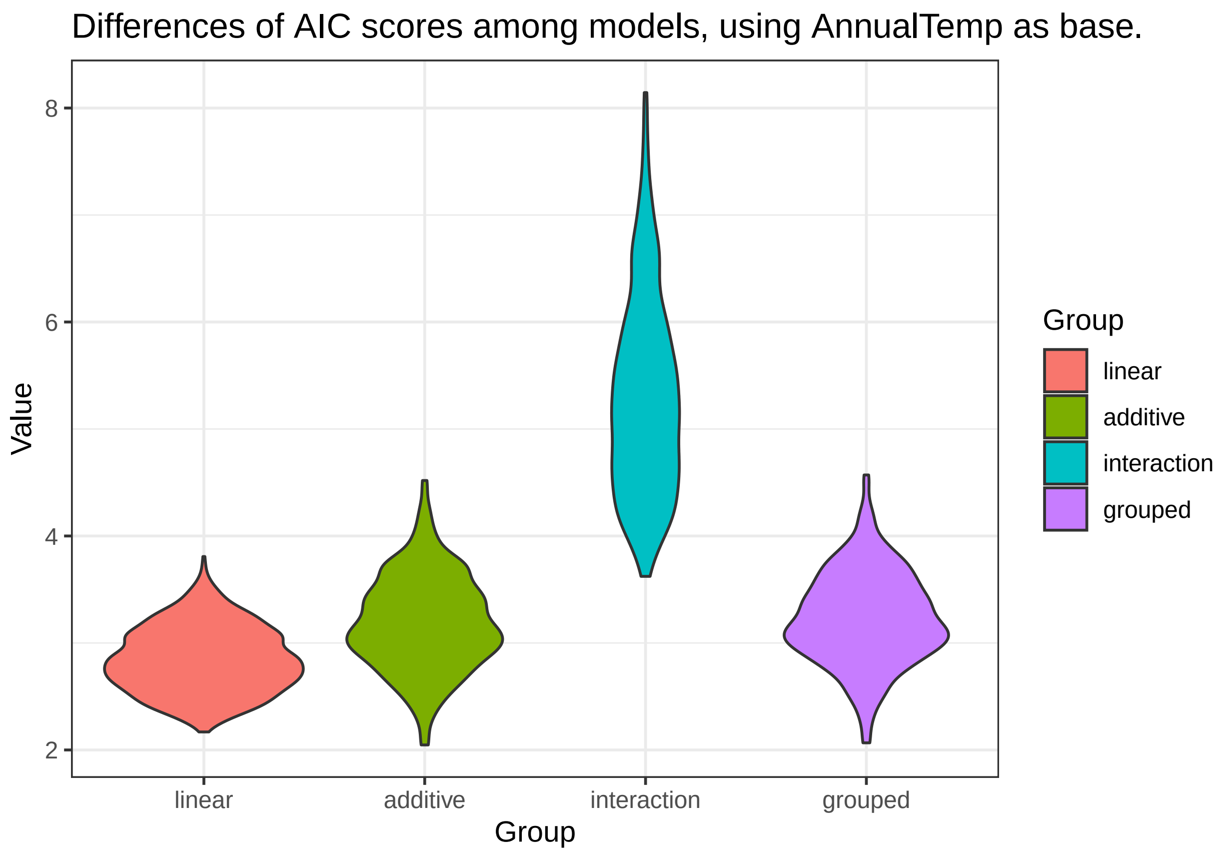

```r
## Checking AIC values in each iteration for ColdTemp
count <- NA

for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(bs.linear.results[i,3], bs.add.results[i,3],
                 bs.int.results[i,3], bs.group.results[i,3])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])

  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

```
## [1] NA
```

Compare between annual and cold

```r
# Calculate the differences of AIC values
aic.bs <- matrix(NA,1000,4) # store the differences in AIC values
aic.bs[,1] <- bs.linear.results[,3] - bs.linear.results[,1]
aic.bs[,2] <- bs.add.results[,3] - bs.add.results[,1]
```

```r
aic.bs[,3] <- bs.int.results[,3] - bs.int.results[,1]
aic.bs[,4] <- bs.group.results[,3] - bs.group.results[,1]

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

# 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","grouped")

# 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
## Checking AIC values in each iteration
count <- NA
```

```r
for (i in 1:1000) {
  # Create a list of the aic values of the current iteration
  aic_value <- c(bs.linear.results[i,1], bs.linear.results[i,3])
  smallest_aic <- min(aic_value)
  # Determining if the smallest value is 2 units smaller than the others
  is_smaller_by_two <- all(smallest_aic + 2 <= aic_value[aic_value != smallest_aic])

  # Append the index of the current list if smaller than 2 units
  if (is_smaller_by_two) {
    count <- c(count, which(aic_value == smallest_aic))
  }
}
count
```

```
##    [1] NA  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   [25]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   [49]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   [73]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   [97]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [121]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [145]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [169]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [193]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [217]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [241]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [265]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [289]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [313]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [337]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [361]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [385]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [409]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [433]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [457]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [481]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [505]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [529]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [553]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [577]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [601]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [625]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [649]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [673]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [697]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [721]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [745]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [769]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [793]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [817]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [841]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [865]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [889]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [913]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
## [937] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [961] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [985] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

- 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).

Concluding points

1. Black carp: using cold temperature have a better fit (higher R2). No preference over the four types of models.
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. Aisan carp: No preference over the four models.