

Crotalus atrox Sexual Selection Analysis

This R Notebook details analyses for the Western Diamondback Rattlesnake (*Crotalus atrox*) sexual selection study (Levine et al. in prep). Raw data were originally collected, analyzed, and reported by Clark et al. (2014).

Parental table generated by Chuck Smith - file name = atrox_parental_table_draft_2.xlsx

78 males - 9 unknown and inferred from parentage analysis, 69 known

18 females - all known

.....

First, I loaded the data file

```
#Load csv file  
bateman <- read.csv("atrox_bateman_comparison_4feb20.csv")
```

Next, I subset the data to a female set, a male set that included zero mating and reproductive success, and a male set that did not include zero mating and reproductive success. The reason for analyzing non-zero male data is that I (BA)L was worried about the natural dependency of reproductive success on mating success (if mating success = 0, reproductive success has to equal 0). This dependency will be particularly important for calculations of the Bateman gradient that include number of breeding years as a covariate. If number of mates is zero, number of breeding years will also be zero, and therefore we will be basically guaranteed to have a significant interaction between mating success and number of breeding years that will make it impossible to evaluate the effect of number of mates and number of breeding years without considering the interaction between the two. This differs from Levine et al. (2015), but given the long-term nature of this data set (with individuals mating over time in multiple years) as opposed to the short-term nature of Levine et al. (2015), it's necessary.

```
####subset data  
#subset data frame and make new female data frame  
female <- subset(bateman, sex=='f')  
  
#subset data frame and make new male data frame  
male <- subset(bateman, sex=='m')  
  
#subset new male data frame and make male data frame without zero values  
male_nonzero <- subset(male, mates > 0)
```

Next, I plotted the relationships between mating success and reproductive success for the different data sets.

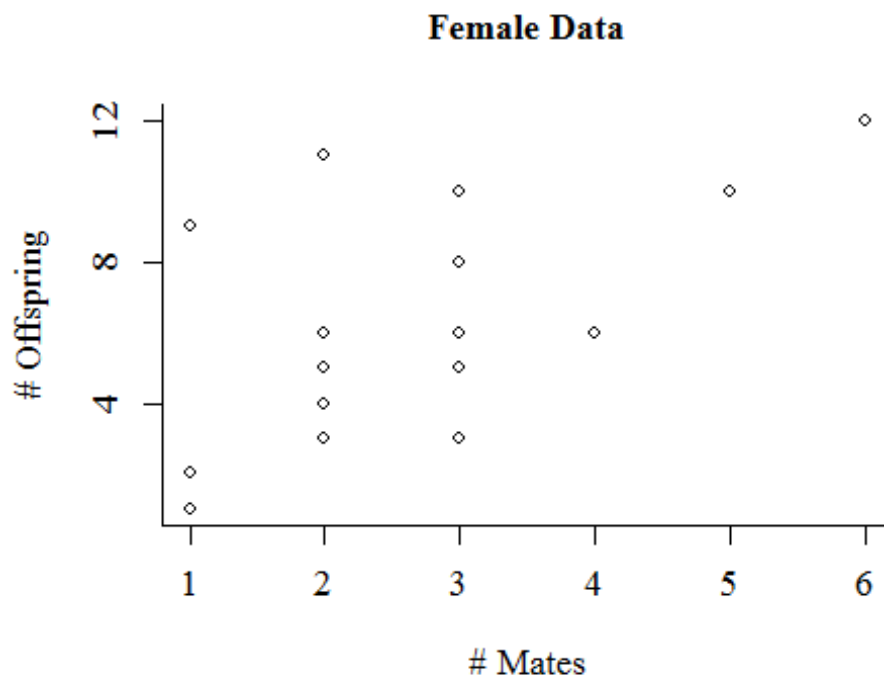
Female plot:

```
#create Times New Roman font
windowsFonts(T = "Times New Roman")

#make scatterplot of data
plot(offspring ~ mates, data = female,
      xlab = list("# Mates", cex = 1.1),
      ylab = list("# Offspring", cex = 1.1),
      main = list("Female Data", cex = 1.1), xaxt="n", yaxt="n",
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")

#Change x axis to right tic marks, font, and size
axis(1, at=0:6, family = "T", cex.axis=1.2)

#Change y axis to right tic marks, font, and size
axis(2, at=4*0:12, family = "T", cex.axis=1.2)
```

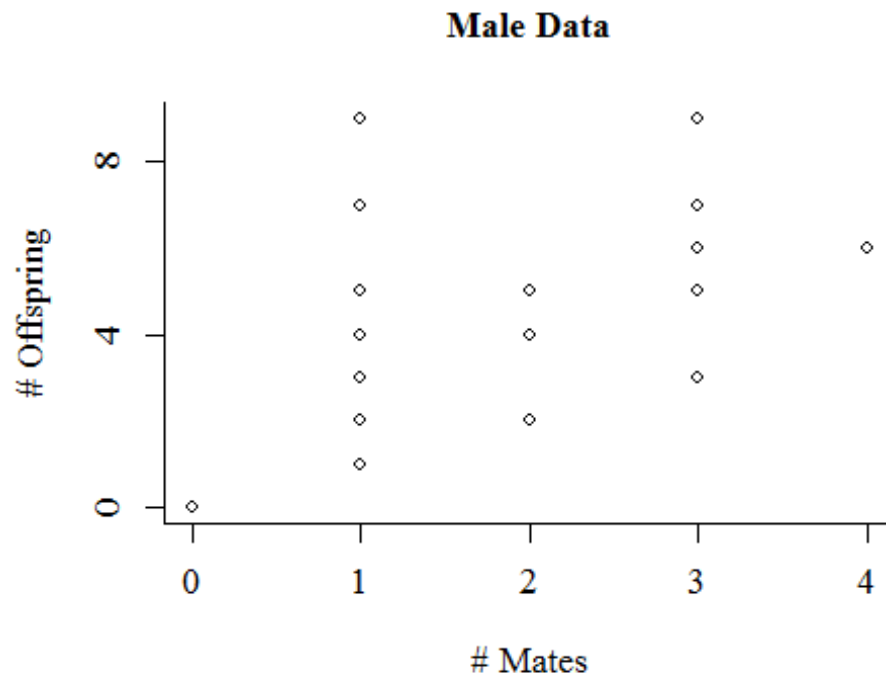


Male plot:

```
#make scatterplot of data
plot(offspring ~ mates, data = male,
      xlab = list("# Mates", cex = 1.1),
      ylab = list("# Offspring", cex = 1.1),
      main = list("Male Data", cex = 1.1), xaxt="n", yaxt="n",
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```

```
#Change x axis to right tic marks, font, and size
axis(1, at=0:6, family = "T", cex.axis=1.2)
```

```
#Change y axis to right tic marks, font, and size
axis(2, at=4*0:12, family = "T", cex.axis=1.2)
```

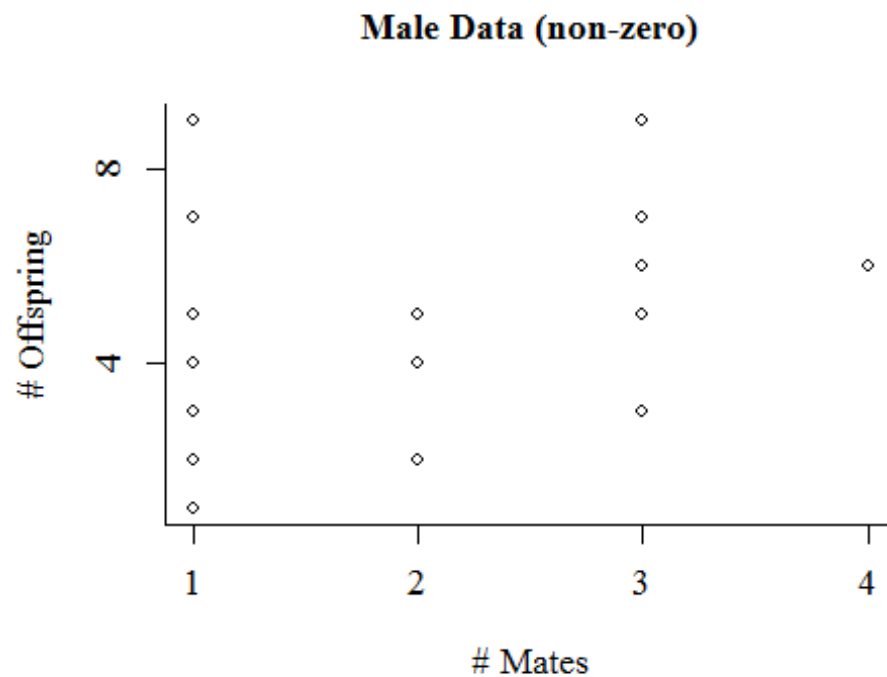


Male plot with data subset to remove zero value mating success records:

```
#make scatterplot of data
plot(offspring ~ mates, data = male_nonzero,
      xlab = list("# Mates", cex = 1.1),
      ylab = list("# Offspring", cex = 1.1),
      main = list("Male Data (non-zero)", cex = 1.1), xaxt="n", yaxt="n",
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```

```
#Change x axis to right tic marks, font, and size
axis(1, at=0:6, family = "T", cex.axis=1.2)
```

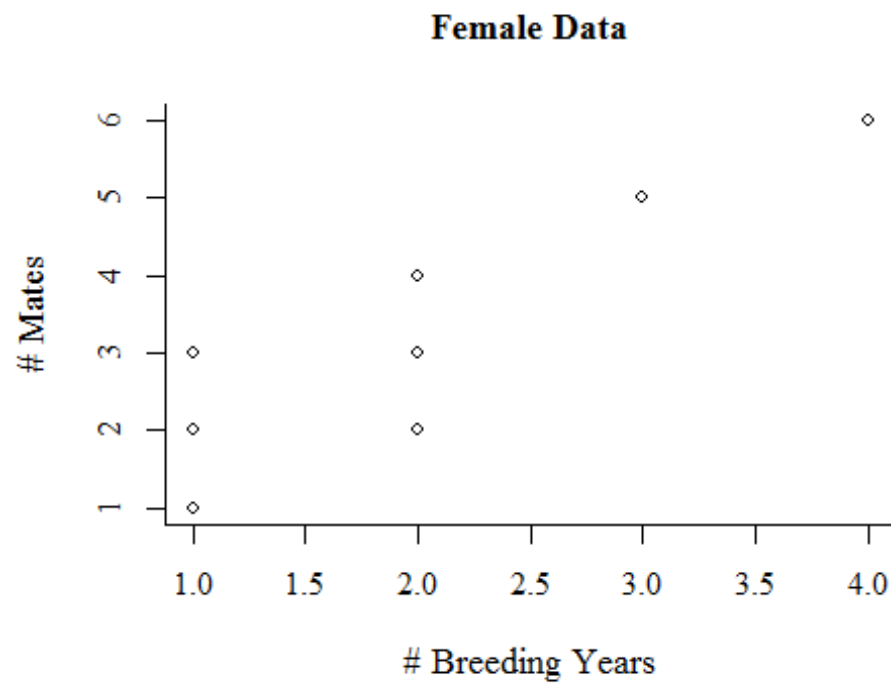
```
#Change y axis to right tic marks, font, and size
axis(2, at=4*0:12, family = "T", cex.axis=1.2)
```



I also plotted number of mates vs. number of breeding years for each data set.

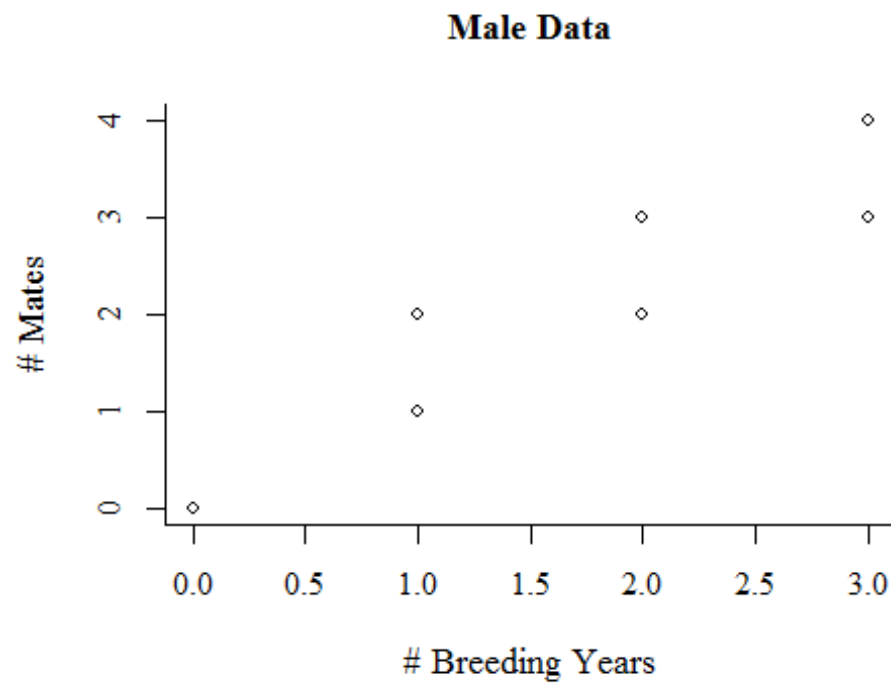
Females:

```
#make scatterplot of data
plot(mates ~ years_breeding, data = female,
      xlab = list("# Breeding Years", cex = 1.1),
      ylab = list("# Mates", cex = 1.1),
      main = list("Female Data", cex = 1.1),
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```



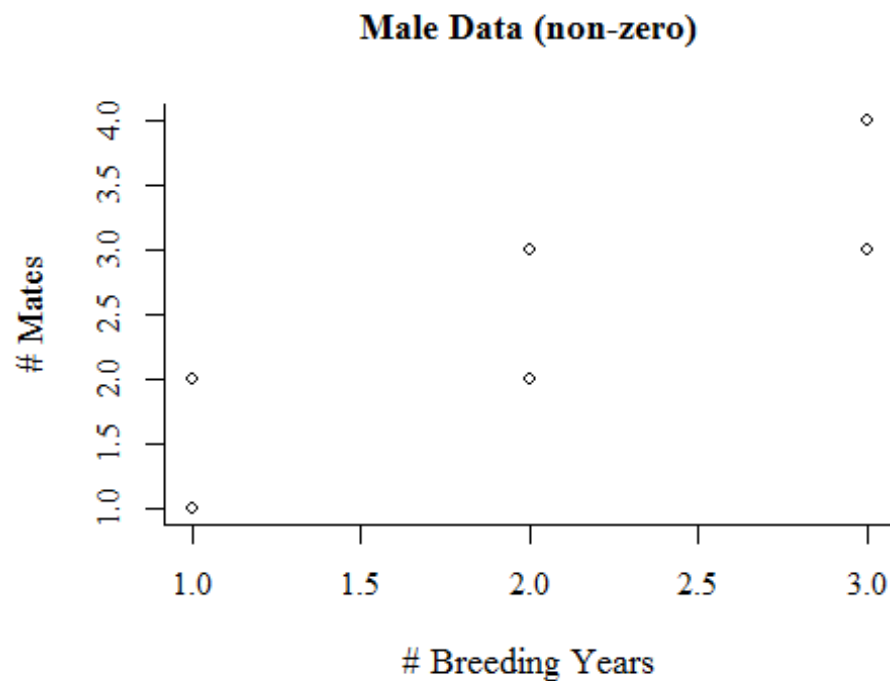
Males:

```
#make scatterplot of data  
plot(mates ~ years_breeding, data = male,  
      xlab = list("# Breeding Years", cex = 1.1),  
      ylab = list("# Mates", cex = 1.1),  
      main = list("Male Data", cex = 1.1),  
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```



Males without zero values:

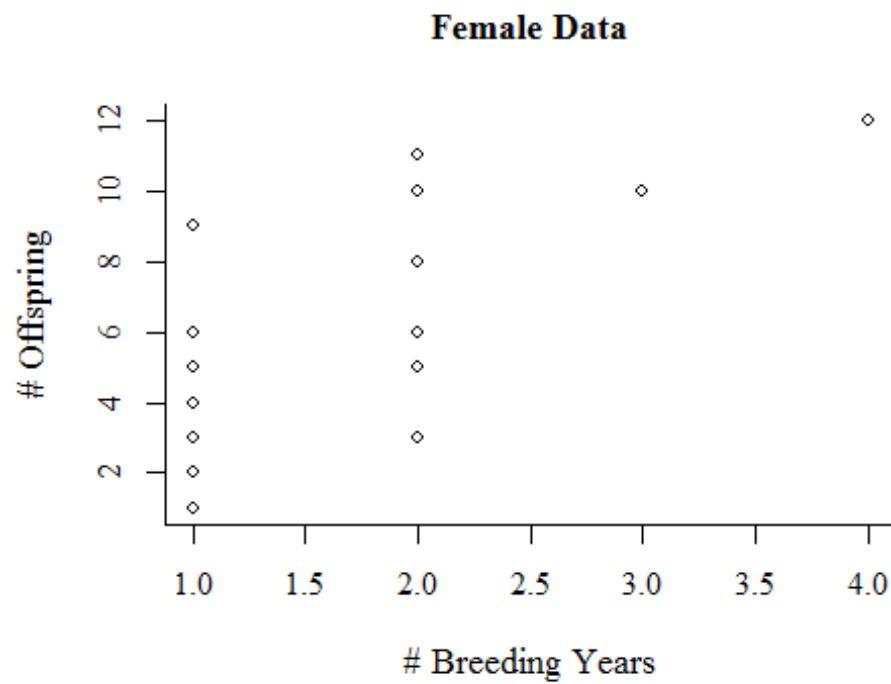
```
#make scatterplot of data  
plot(mates ~ years_breeding, data = male_nonzero,  
      xlab = list("# Breeding Years", cex = 1.1),  
      ylab = list("# Mates", cex = 1.1),  
      main = list("Male Data (non-zero)", cex = 1.1),  
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```



I also plotted number of offspring vs number of breeding years for each data set.

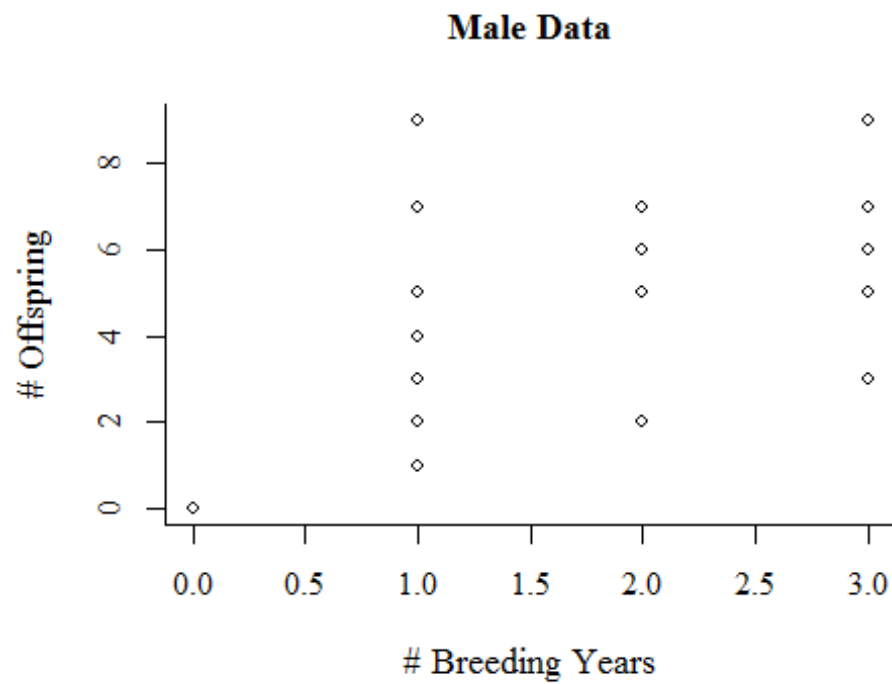
Females:

```
#make scatterplot of data  
plot(offspring ~ years_breeding, data = female,  
      xlab = list("# Breeding Years", cex = 1.1),  
      ylab = list("# Offspring", cex = 1.1),  
      main = list("Female Data", cex = 1.1),  
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```



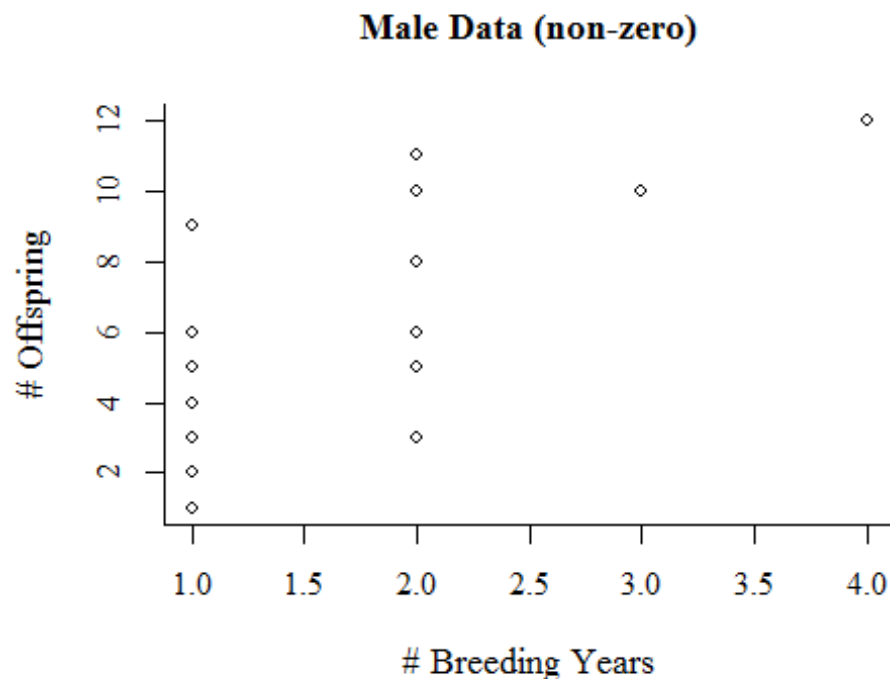
Males:

```
#make scatterplot of data  
plot(offspring ~ years_breeding, data = male,  
      xlab = list("# Breeding Years", cex = 1.1),  
      ylab = list("# Offspring", cex = 1.1),  
      main = list("Male Data", cex = 1.1),  
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```

Males without zero values:

```
#make scatterplot of data
plot(offspring ~ years_breeding, data = female,
      xlab = list("# Breeding Years", cex = 1.1),
      ylab = list("# Offspring", cex = 1.1),
      main = list("Male Data (non-zero)", cex = 1.1),
      family = "T", cex = .8, pch = 1, bty = "l", xaxs = "r")
```



.....

To test for a significant relationship between number of mates and numbers of offspring (= Bateman gradient, Bss), I ran generalized linear models with a Poisson distribution (because data are count data). This was also done by Apakupakul and Rubenstein (2015).

Starting with the female data, I first ran the GLM with an interaction between number of mates and the number of breeding years (a covariate).

```
#glm with interaction between mating success and number of years breeding
glm_female <- glm(offspring ~ mates*years_breeding, data=female,
family=poisson)
```

```
#view summary of glm
summary(glm_female)
```

```
##
## Call:
## glm(formula = offspring ~ mates * years_breeding, family = poisson,
##      data = female)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67348  -0.83275  -0.04859   0.54697   2.31655
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)          0.54763    0.55480    0.987    0.3236
## mates                0.18909    0.19502    0.970    0.3323
## years_breeding       0.65052    0.37830    1.720    0.0855 .
## mates:years_breeding -0.07573    0.06887   -1.100    0.2715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 35.535  on 17  degrees of freedom
## Residual deviance: 21.010  on 14  degrees of freedom
## AIC: 91.358
##
## Number of Fisher Scoring iterations: 5
```

There was no evidence of a significant interaction between number of mates and number of breeding years for the female data set, so this interaction was removed and the model re-run.

```
#glm without interaction of fixed effects
#Note: overwriting earlier female glm
glm_female <- glm(offspring ~ mates + years_breeding, data=female,
family=poisson)

#view summary of glm
summary(glm_female)

##
## Call:
## glm(formula = offspring ~ mates + years_breeding, family = poisson,
##      data = female)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9155  -0.7756  -0.2013   0.5948   1.9786
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.09941    0.21843   5.033 4.82e-07 ***
## mates         0.04541    0.14363   0.316   0.752
## years_breeding 0.31169    0.22774   1.369   0.171
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 35.535  on 17  degrees of freedom
## Residual deviance: 22.237  on 15  degrees of freedom
## AIC: 90.584
```

```
##  
## Number of Fisher Scoring iterations: 5
```

Then, I extracted the female Bateman gradient from the model..

```
#print the female Bateman gradient, as extracted from the model coefficients  
cat("Female Bss (absolute): ", glm_female$coefficients[2], "\n")  
## Female Bss (absolute): 0.04541403
```

Then, I tested for a significant effect of number of mates on number of offspring for females by removing mating success from the model with the drop1 function, and comparing the reduced model with a LLR test.

```
#use drop one function with chi-square statistic for LRT  
drop1(glm_female, test="Chisq")  
  
## Single term deletions  
##  
## Model:  
## offspring ~ mates + years_breeding  
##  
##          Df Deviance    AIC      LRT Pr(>Chi)  
## <none>          22.237 90.584  
## mates           1  22.337 88.684 0.09976  0.7521  
## years_breeding  1  24.163 90.511 1.92650  0.1651
```

I found that there was no significant effect of number of mates on reproductive success when adjusting for the number of years that the female bred. This is seen by the LRT which shows that the model is not significantly different if the number of mates is removed as a predictor variable. In other words, there is no significant Bateman gradient for females when including number of breeding years as a covariate. This is what we would expect based off of sexual selection theory.

I then ran a similar GLM for males, including the zero count data, and with an interaction effect between number of mates and number of breeding years.

```
#glm with interaction between fixed effects  
glm_male <- glm(offspring ~ mates*years_breeding, data=male, family=poisson)  
  
#view glm summary  
summary(glm_male)  
  
##  
## Call:  
## glm(formula = offspring ~ mates * years_breeding, family = poisson,  
##      data = male)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max  
## -2.3003  -0.4978  -0.4978  -0.4978   3.6072  
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.0883     0.3499  -5.968 2.40e-09 ***
## mates           1.5589     0.2641   5.902 3.58e-09 ***
## years_breeding  1.9808     0.3677   5.387 7.17e-08 ***
## mates:years_breeding -0.7539     0.1142  -6.601 4.08e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 267.60  on 77  degrees of freedom
## Residual deviance:  62.33  on 74  degrees of freedom
## AIC: 153.55
##
## Number of Fisher Scoring iterations: 5
```

As expected given the dependency I describe above (and the large number of zeros in this data set), there was a significant interaction between number of mates and number of breeding years.

However, this interaction disappears if I only use the non-zero male data for the GLM.

```
#glm of non-zero male data set with interaction between fixed effects
glm_male_nonzero <- glm(offspring ~ mates*years_breeding, data=male_nonzero,
family=poisson)

#view summary of glm
summary(glm_male_nonzero)

##
## Call:
## glm(formula = offspring ~ mates * years_breeding, family = poisson,
##      data = male_nonzero)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4087  -0.9095  -0.0921   0.5358   2.6932
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.92535     0.65786   1.407   0.160
## mates           0.26952     0.37709   0.715   0.475
## years_breeding -0.07217     0.52385  -0.138   0.890
## mates:years_breeding 0.01989     0.18159   0.110   0.913
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 38.450  on 26  degrees of freedom
## Residual deviance: 29.934  on 23  degrees of freedom
```

```
## AIC: 121.16
##
## Number of Fisher Scoring iterations: 5
```

Therefore, I re-ran the male GLM without the interaction term for the non-zero male data (i.e., since there was no significant interaction between number of mates and number of breeding years).

```
#glm without interaction between fixed effects
#Note: overwriting earlier male glm
glm_male_nonzero <- glm(offspring ~ mates + years_breeding,
data=male_nonzero, family=poisson)

#view summary of glm
summary(glm_male_nonzero)

##
## Call:
## glm(formula = offspring ~ mates + years_breeding, family = poisson,
##      data = male_nonzero)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -1.4010  -0.9151  -0.1086   0.5269   2.7043
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.85756    0.22284   3.848 0.000119 ***
## mates          0.30261    0.22433   1.349 0.177356
## years_breeding -0.02269    0.26527  -0.086 0.931827
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 38.450  on 26  degrees of freedom
## Residual deviance: 29.945  on 24  degrees of freedom
## AIC: 119.17
##
## Number of Fisher Scoring iterations: 5
```

Quick sidebar: I also removed the data files for males with zeros so as to not get confused later.

```
#remove unnecessary data files
rm(male)
rm(glm_male)
```

I extracted the male Bateman gradient from the model coefficients.

```
#cat the male Bateman gradient as extracted from the coefficients of the model
```

```
cat("Male Bss (absolute): ", glm_male_nonzero$coefficients[2], "\n")
```

```
## Male Bss (absolute): 0.3026129
```

I tested the significance of number of mates in the model with drop1, as I did for females.

```
#use drop1 to do LRT with chi-square
```

```
drop1(glm_male_nonzero, test="Chisq")
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## offspring ~ mates + years_breeding
```

```
##           Df Deviance    AIC      LRT Pr(>Chi)
```

```
## <none>           29.945 119.17
```

```
## mates           1   31.676 118.90 1.73035  0.1884
```

```
## years_breeding  1   29.953 117.17 0.00731  0.9319
```

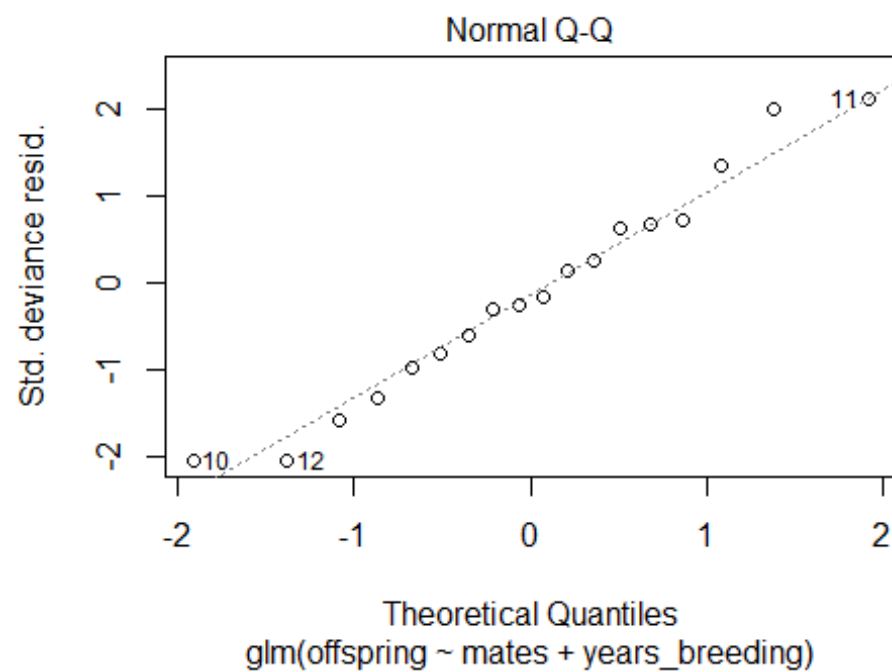
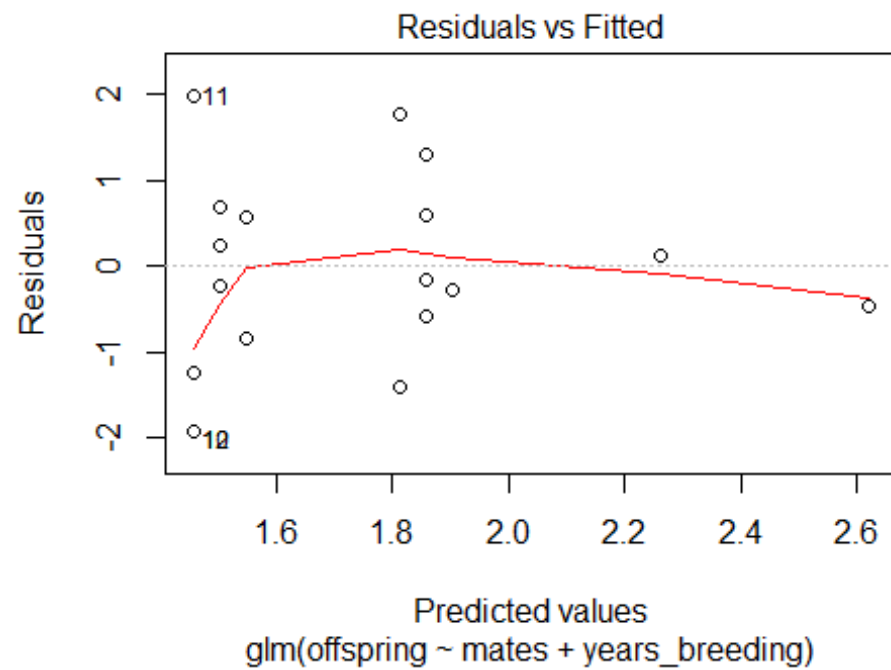
Interestingly, I found that there was no significant effect of number of mates on number of offspring for males when adjusting for the number of years that the male bred. In other words, there is no significant Bateman gradient for males when accounting for this covariate. This is a really interesting result and supports our results regarding lack of selection on male SVL (see results of SVL analyses later in the notebook). We would expect to see an effect of male SVL on number of offspring produced if males are gaining more priority access to females via male combat (i.e., acquiring more mates). However, we didn't. There is no relationship between male SVL and number of offspring produced (nor number of mates with which offspring were produced). Furthermore, there isn't dramatic sexual size dimorphism in this population, also supporting that male SVL isn't allowing males to gain more offspring via more mating opportunities. Therefore, it makes sense that we wouldn't see a relationship between number of mates (which could have been, but was not, positively related to male SVL) and number of offspring produced.

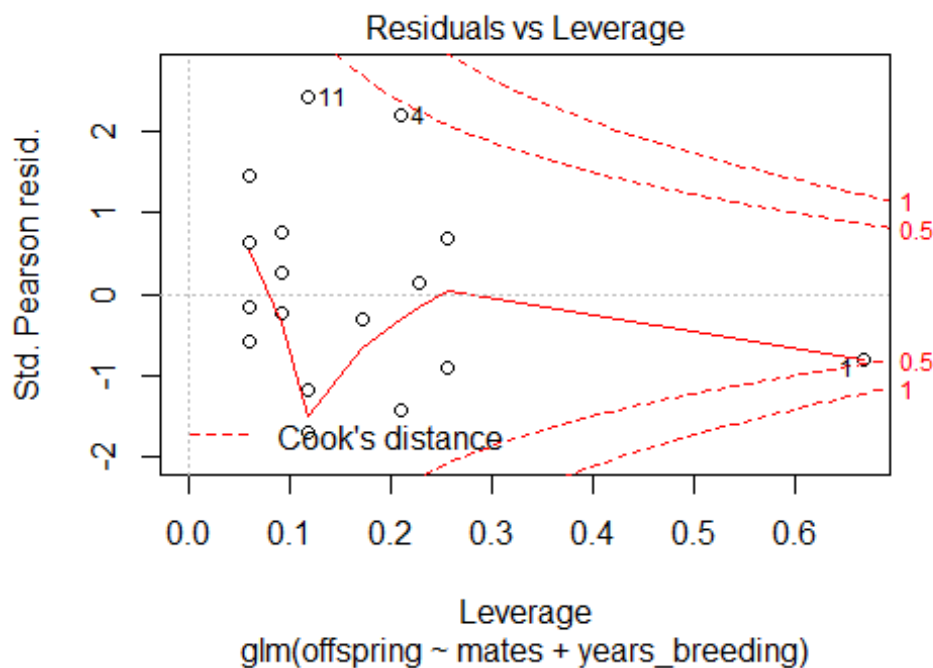
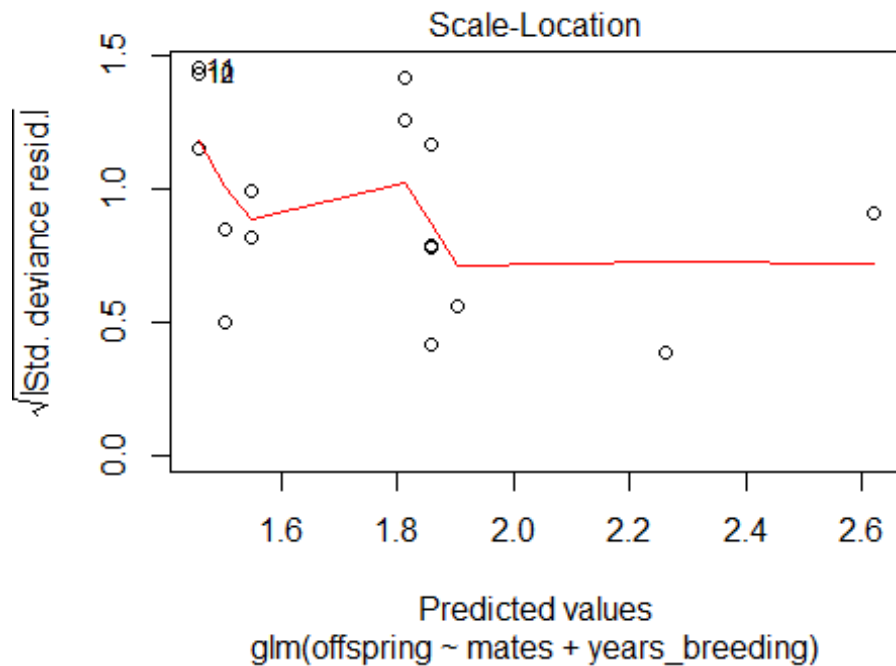
Then, I evaluated the residual diagnostics of the models.

First, for females:

```
#plot residual diagnostics for female glm
```

```
plot(glm_female)
```

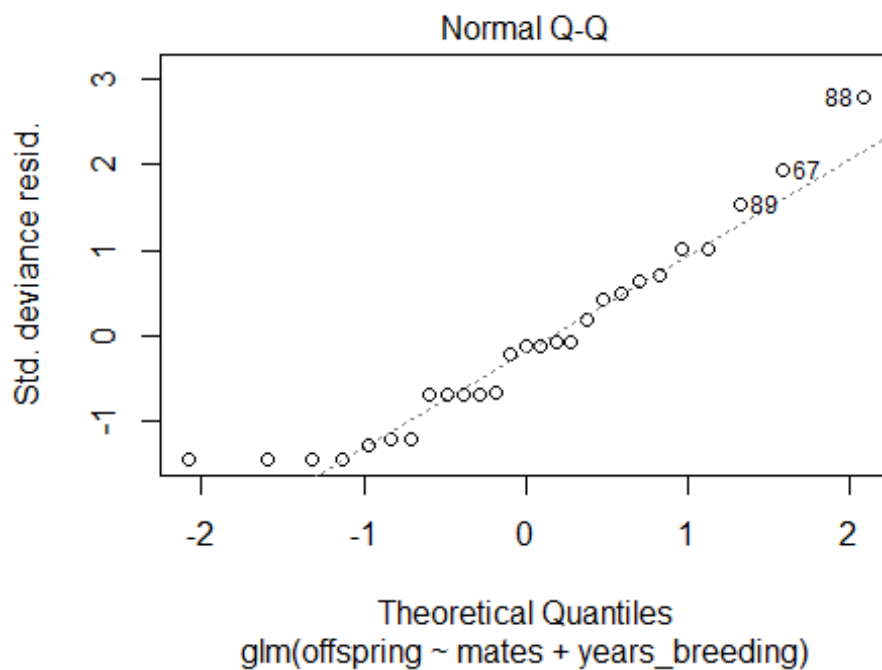
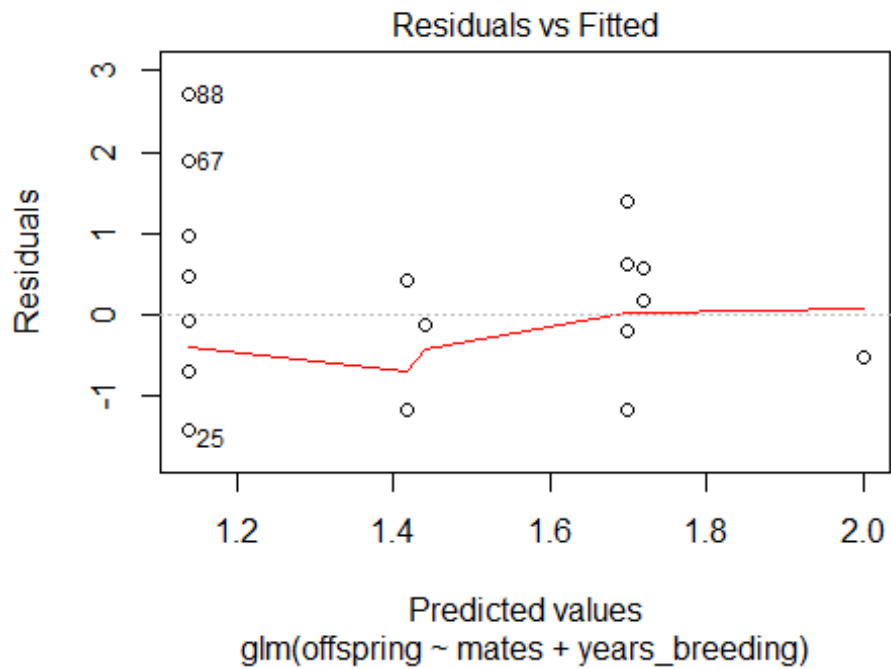


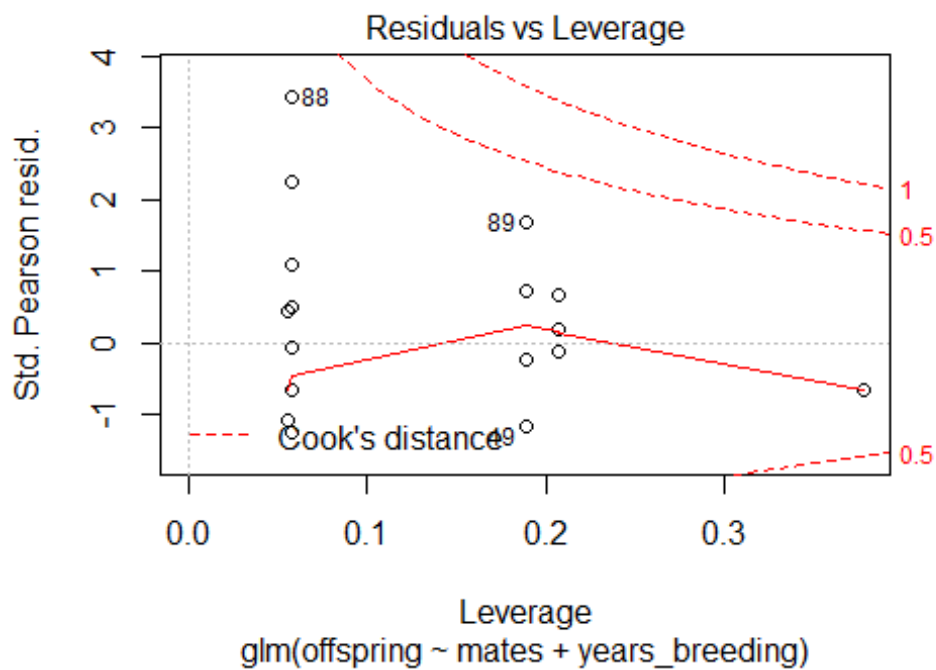
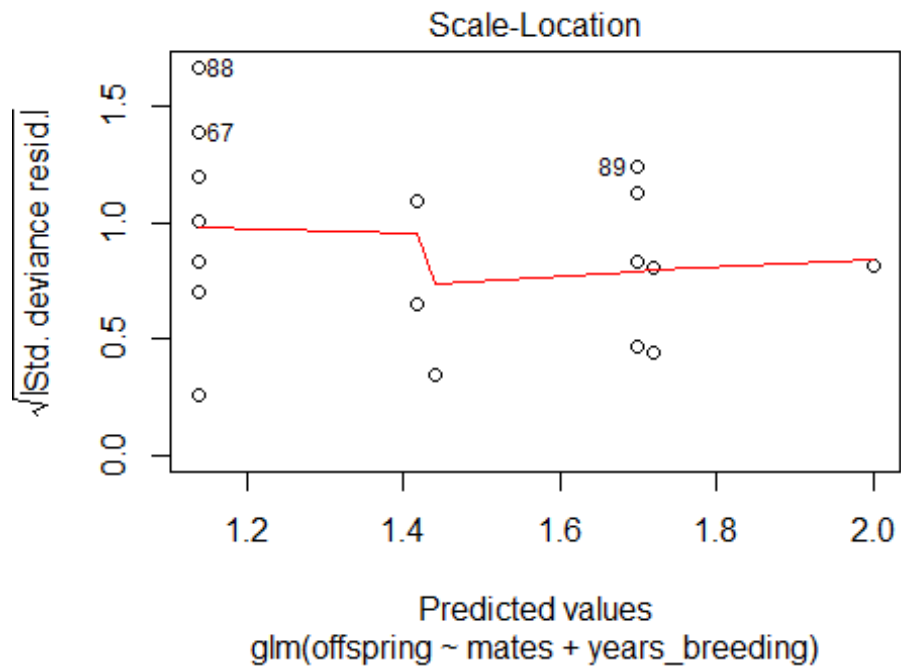


The female residuals look OK considering the sample size is so small ($N = 18$). There doesn't seem to be a clear pattern on the residuals vs. fitted curve, the qqplot pretty closely follows a straight line, the scale location line is relatively horizontal and the residuals seem randomly spread, and there aren't any points that seem to be influential to the regression results.

Then, I evaluated the residual diagnostics for the male GLM.

```
#plot residual diagnostics for male GLM  
plot(glm_male_nonzero)
```





Same interpretation of residuals as for females.

I also tested for overdispersion given the Poisson distribution.

First for females:

```

#use package AER to test for overdispersion of female glm
library(AER)

## Loading required package: car
## Loading required package: carData
## Loading required package: lmtest
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric

## Loading required package: sandwich
## Loading required package: survival

dispersiontest(glm_female)

##
## Overdispersion test
##
## data:  glm_female
## z = 0.61766, p-value = 0.2684
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
## 1.202054

```

Then for males:

```

#use package AER to test for overdispersion of male glm
dispersiontest(glm_male_nonzero)

##
## Overdispersion test
##
## data:  glm_male_nonzero
## z = 0.59554, p-value = 0.2757
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
## 1.212422

```

There is no evidence of overdispersion for either model.

.....

All analyses were repeated using relative mating success and relative reproductive success so as to quantify standardized Bateman gradients (as in Henshaw et al. 2016).

First, I calculated mean mating and reproductive success for each sex. Then, I divided each individual's mating and reproductive success by their respective means to calculate relative mating and reproductive success and added these new values in two new columns to the respective data frame for each sex.

```
##### females
#calculate mean number of mates for females
mean_mate_f <- mean(female[["mates"]])

#calculate mean number of offspring for females
mean_off_f <- mean(female[["offspring"]])

#cat mean number of mates and offspring for females
cat("Female Mean Mating Success:", mean_mate_f, "\n", "Female Mean
Reproductive Success:", mean_off_f, "\n\n")

## Female Mean Mating Success: 2.611111
## Female Mean Reproductive Success: 6

#add relative mating and reproductive success to female data frame
female$rel_mates <- female$mates/mean_mate_f #rel_mates = relative mating
success added to data frame
female$rel_off <- female$offspring/mean_off_f #rel_off = relative
reproductive success added to data frame

#view the first few rows of the new data frame to confirm new columns are
there
head(female)

##      id sex mates offspring years_breeding rel_mates  rel_off
## 1  CA-1  f     6         12             4 2.2978723 2.0000000
## 2  CA-2  f     2          5             1 0.7659574 0.8333333
## 3 CA-16  f     3          6             1 1.1489362 1.0000000
## 4 CA-30  f     2         11             2 0.7659574 1.8333333
## 5 CA-42  f     4          6             2 1.5319149 1.0000000
## 6 CA-44  f     2          3             2 0.7659574 0.5000000

cat("\n\n") #cat some new lines

##### males
#calculate mean number of mates for males
mean_mate_m <- mean(male_nonzero[["mates"]])

#calculate mean number of offspring for males
mean_off_m <- mean(male_nonzero[["offspring"]])

#cat mean number of mates and offspring for males
```

```

cat("Male Mean Mating Success:", mean_mate_m, "\n", "Male Mean Reproductive
Success:", mean_off_m, "\n\n")

## Male Mean Mating Success: 1.740741
## Male Mean Reproductive Success: 4

#add relative mating and reproductive success to male data frame
male_nonzero$rel_mates <- male_nonzero$mates/mean_mate_m #rel_mates =
relative mating success added to data frame
male_nonzero$rel_off <- male_nonzero$offspring/mean_off_m #rel_off = relative
reproductive success added to data frame

#cat the first few rows of the new data frame to confirm new columns are
there
head(male_nonzero)

##      id sex mates offspring years_breeding rel_mates rel_off
## 22  CA-5  m     3         7             2 1.7234043    1.75
## 25  CA-9  m     1         1             1 0.5744681    0.25
## 28 CA-20  m     1         2             1 0.5744681    0.50
## 30 CA-23  m     1         2             1 0.5744681    0.50
## 34 CA-27  m     2         4             1 1.1489362    1.00
## 44 CA-40  m     1         1             1 0.5744681    0.25

cat("\n\n") #cat some new lines

```

Starting with the female data, I ran the GLM with an interaction between relative number of mates and the number of breeding years. I used a Gaussian probability distribution rather than Poisson, because data are no longer count data and are continuous.

```

#glm with interaction between mating success and number of years breeding
#Note: overwriting the earlier female glm
glm_female <- glm(rel_off ~ rel_mates*years_breeding, data=female,
family=gaussian)

#view summary of glm
summary(glm_female)

##
## Call:
## glm(formula = rel_off ~ rel_mates * years_breeding, family = gaussian,
##      data = female)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64097  -0.28043  -0.00413   0.20411   0.88463
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.02760    0.55777   0.049   0.961
## rel_mates       0.29038    0.53448   0.543   0.595

```

```
## years_breeding          0.50764    0.41844    1.213    0.245
## rel_mates:years_breeding -0.08116    0.21222   -0.382    0.708
##
## (Dispersion parameter for gaussian family taken to be 0.2045135)
##
## Null deviance: 5.4444  on 17  degrees of freedom
## Residual deviance: 2.8632  on 14  degrees of freedom
## AIC: 27.99
##
## Number of Fisher Scoring iterations: 2
```

There was no evidence of a significant interaction between number of mates and number of breeding years, so this interaction was removed and the model re-run.

```
#glm without interaction of fixed effects
#Note: overwriting earlier female glm
glm_female <- glm(rel_off ~ rel_mates + years_breeding, data=female,
family=gaussian)

#view summary of glm
summary(glm_female)

##
## Call:
## glm(formula = rel_off ~ rel_mates + years_breeding, family = gaussian,
##      data = female)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59069  -0.30398  -0.06199   0.21837   0.84517
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.2190    0.2392   0.915   0.374
## rel_mates       0.1509    0.3795   0.398   0.696
## years_breeding  0.3780    0.2384   1.586   0.134
##
## (Dispersion parameter for gaussian family taken to be 0.1928734)
##
## Null deviance: 5.4444  on 17  degrees of freedom
## Residual deviance: 2.8931  on 15  degrees of freedom
## AIC: 26.177
##
## Number of Fisher Scoring iterations: 2
```

I extracted the standardized female Bss using relative values.

```
#cat the female standardized Bateman gradient as extracted from the model coefficients
cat("Standardized Female Bss (relative): ", glm_female$coefficients[2], "\n")
```

```
## Standardized Female Bss (relative): 0.1509466
```

Then, I tested for a significant effect of relative number of mates on number of offspring for females by removing relative mating success from the model with the drop1 function, and comparing the reduced model with a LLR test.

```
#use drop one function with chi-square statistic for LRT  
drop1(glm_female, test="Chisq")
```

```
## Single term deletions  
##  
## Model:  
## rel_off ~ rel_mates + years_breeding  
##           Df Deviance    AIC scaled dev. Pr(>Chi)  
## <none>           2.8931 26.177  
## rel_mates      1   2.9236 24.366    0.18883 0.66389  
## years_breeding 1   3.3782 26.967    2.79015 0.09485 .  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For male data, I ran the GLM with an interaction between relative number of mates and the number of breeding years. I used a Gaussian probability distribution rather than Poisson, because data are no longer count data and are continuous.

```
#glm of non-zero male data set with interaction between fixed effects  
#Note: overwriting earlier male glm  
glm_male_nonzero <- glm(rel_off ~ rel_mates*years_breeding,  
data=male_nonzero, family=gaussian)
```

```
#view summary of glm  
summary(glm_male_nonzero)
```

```
##  
## Call:  
## glm(formula = rel_off ~ rel_mates * years_breeding, family = gaussian,  
##      data = male_nonzero)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max   
## -0.63328 -0.42469 -0.07257  0.31348  1.46683   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)    0.7636    0.7879   0.969   0.343      
## rel_mates       0.3410    0.7861   0.434   0.668      
## years_breeding -0.2698    0.6771  -0.398   0.694      
## rel_mates:years_breeding 0.1627    0.4059   0.401   0.692      
##  
## (Dispersion parameter for gaussian family taken to be 0.3121603)  
##  
##      Null deviance: 9.5000  on 26  degrees of freedom
```



```
## Residual deviance: 7.1797 on 23 degrees of freedom
## AIC: 50.859
##
## Number of Fisher Scoring iterations: 2
```

I re-ran the male GLM without the interaction term for the non-zero male data (since there was no significant interaction between relative number of mates and number of breeding years).

```
#glm without interaction between fixed effects
#Note: overwriting earlier male glm
glm_male_nonzero <- glm(rel_off ~ rel_mates + years_breeding,
data=male_nonzero, family=gaussian)

#view summary of glm
summary(glm_male_nonzero)

##
## Call:
## glm(formula = rel_off ~ rel_mates + years_breeding, family = gaussian,
##      data = male_nonzero)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6283  -0.3928  -0.1058   0.2856   1.4822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.46263    0.23511   1.968  0.0608 .
## rel_mates       0.58834    0.47864   1.229  0.2309
## years_breeding -0.03277    0.32431  -0.101  0.9204
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3012438)
##
##      Null deviance: 9.5000 on 26 degrees of freedom
## Residual deviance: 7.2299 on 24 degrees of freedom
## AIC: 49.047
##
## Number of Fisher Scoring iterations: 2
```

I extracted the standardized Bateman gradient for males from the model coefficients.

```
#cat the male standardized Bateman gradient as extracted from the model coefficients
cat("Standardized Male Bateman Gradient (relative): ",
glm_male_nonzero$coefficients[2], "\n")

## Standardized Male Bateman Gradient (relative): 0.5883374
```

I tested the significance of relative number of mates in the model with drop1, as I did for females.

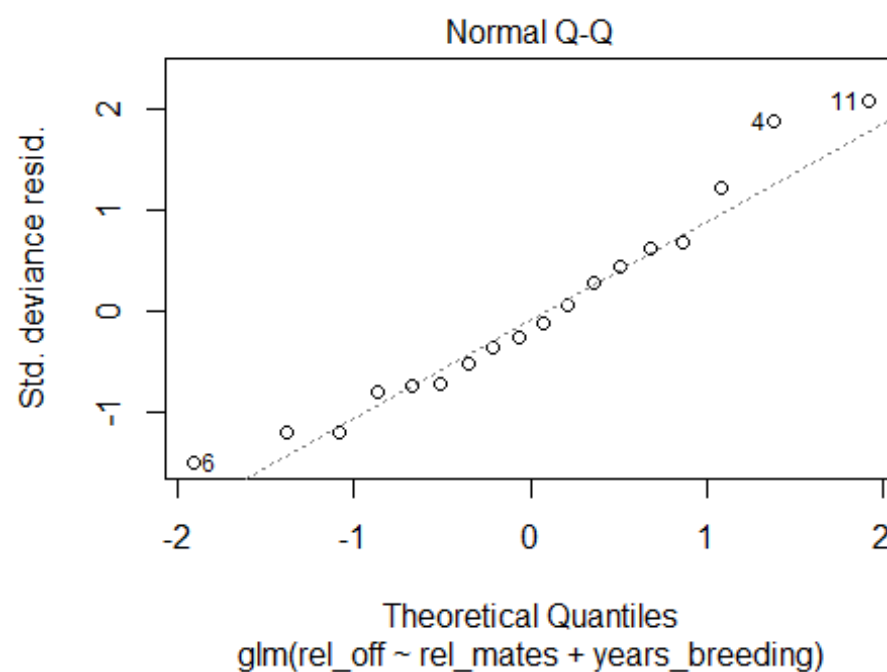
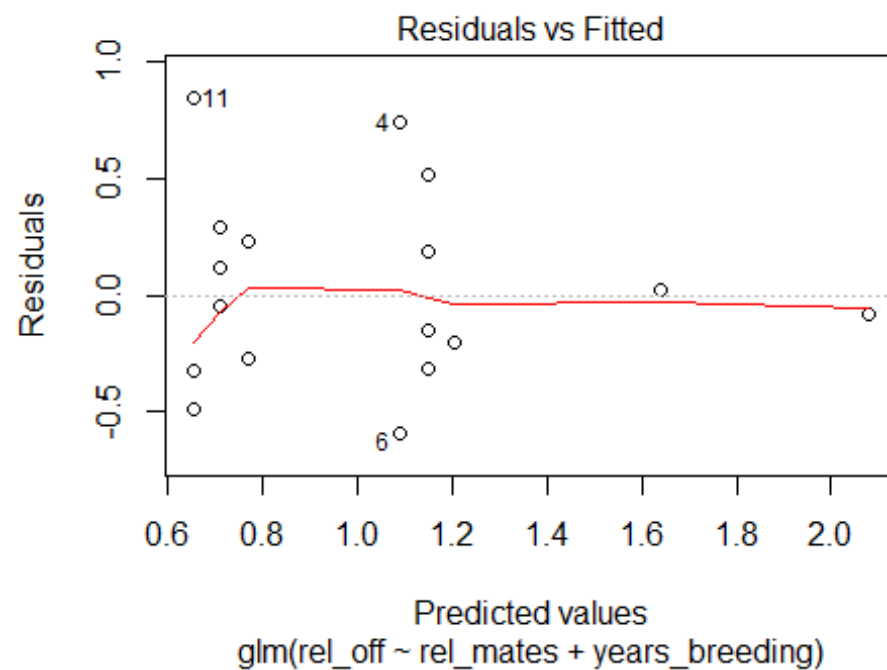
```
#use drop1 to do LRT with chi-square
drop1(glm_male_nonzero, test="Chisq")

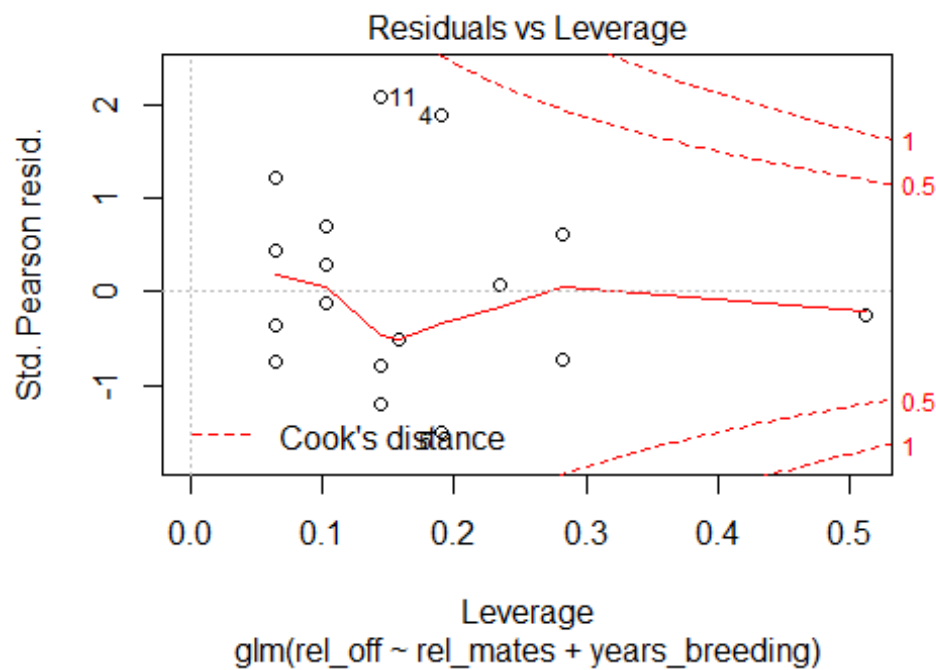
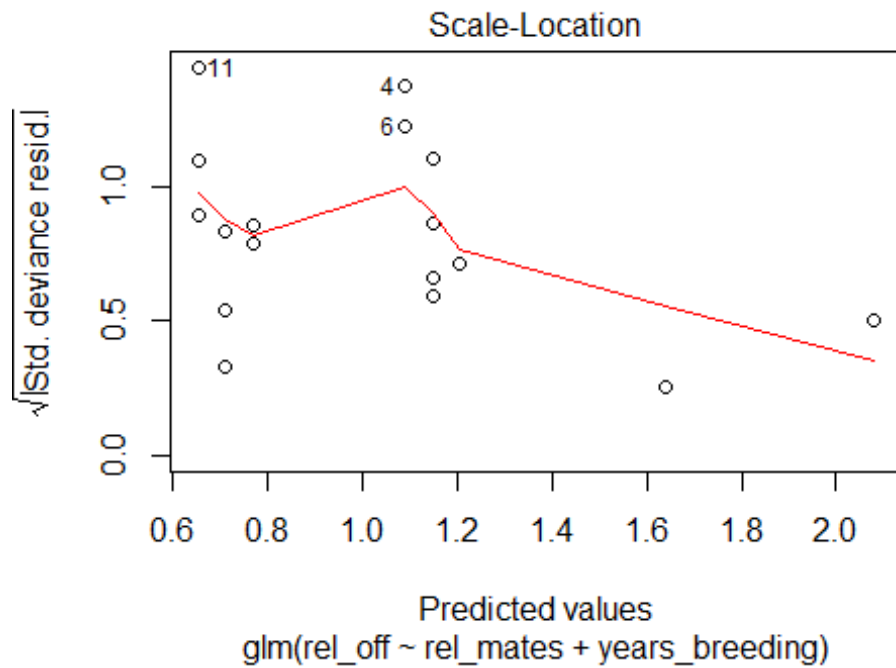
## Single term deletions
##
## Model:
## rel_off ~ rel_mates + years_breeding
##           Df Deviance    AIC scaled dev. Pr(>Chi)
## <none>           7.2299 49.047
## rel_mates      1   7.6850 48.695     1.64840    0.1992
## years_breeding 1   7.2329 47.058     0.01148    0.9147
```

Then, I evaluated the residual diagnostics of the models.

First, for females:

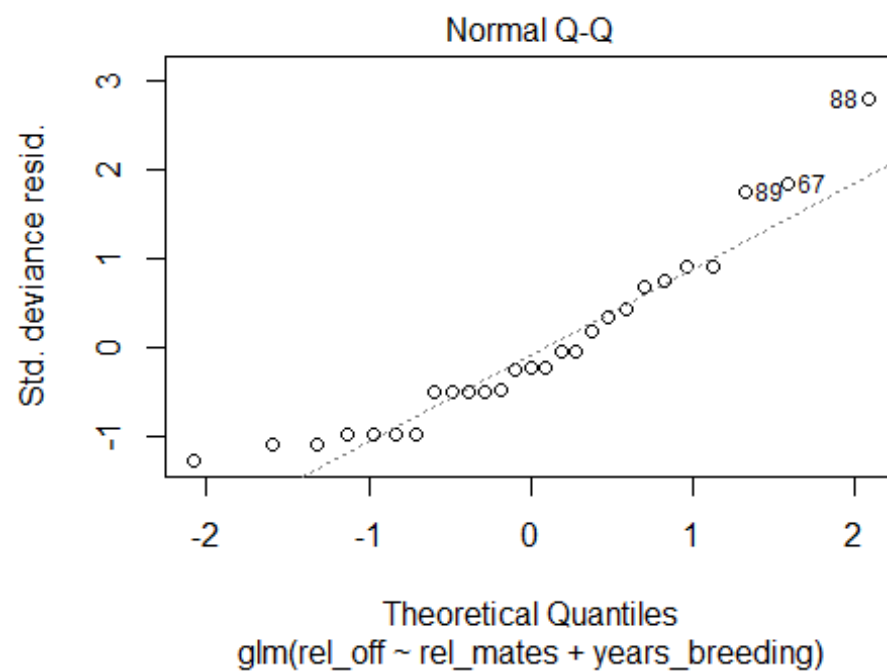
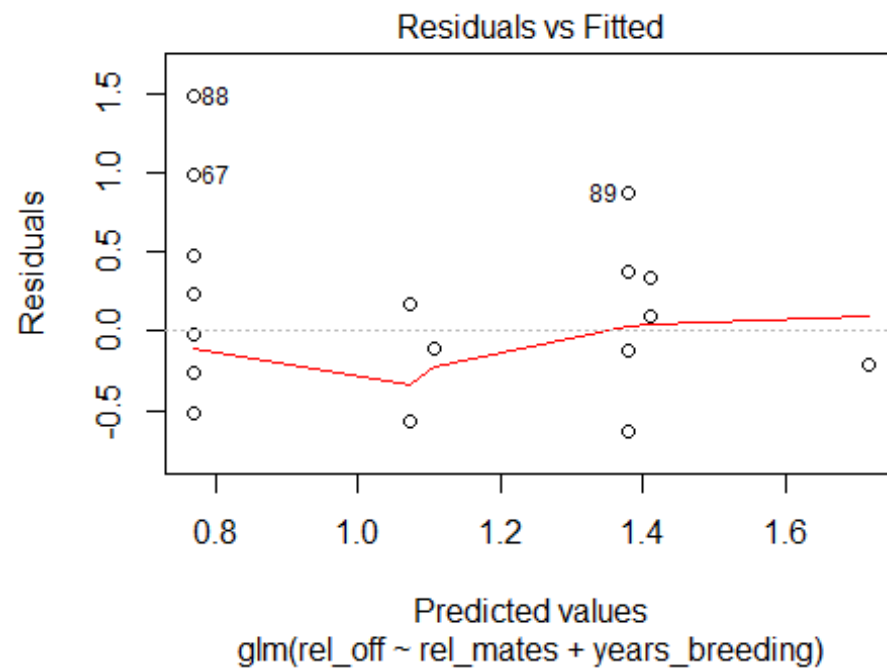
```
#plot residual diagnostics for female glm
plot(glm_female)
```

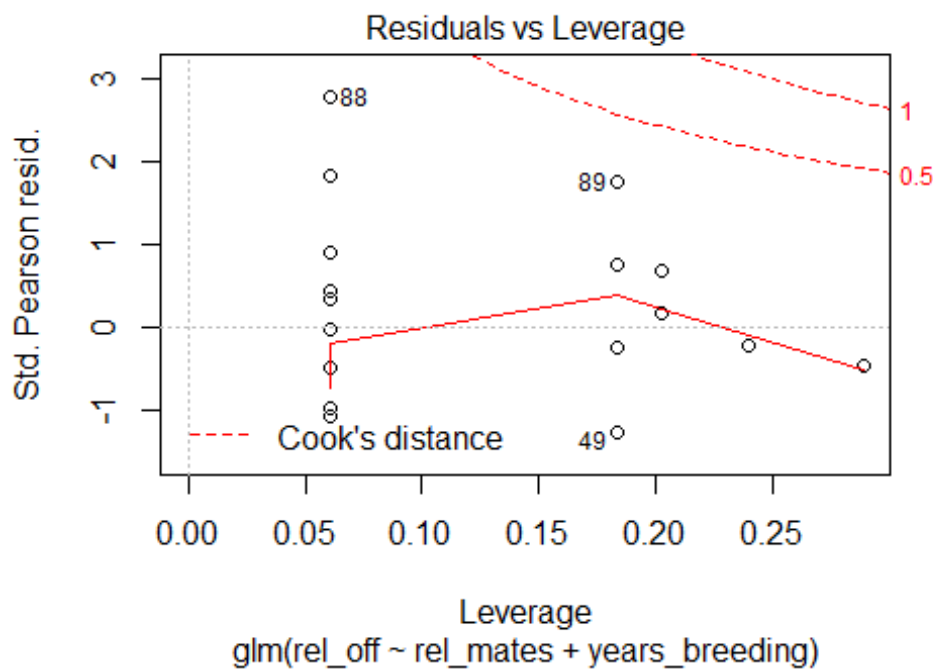
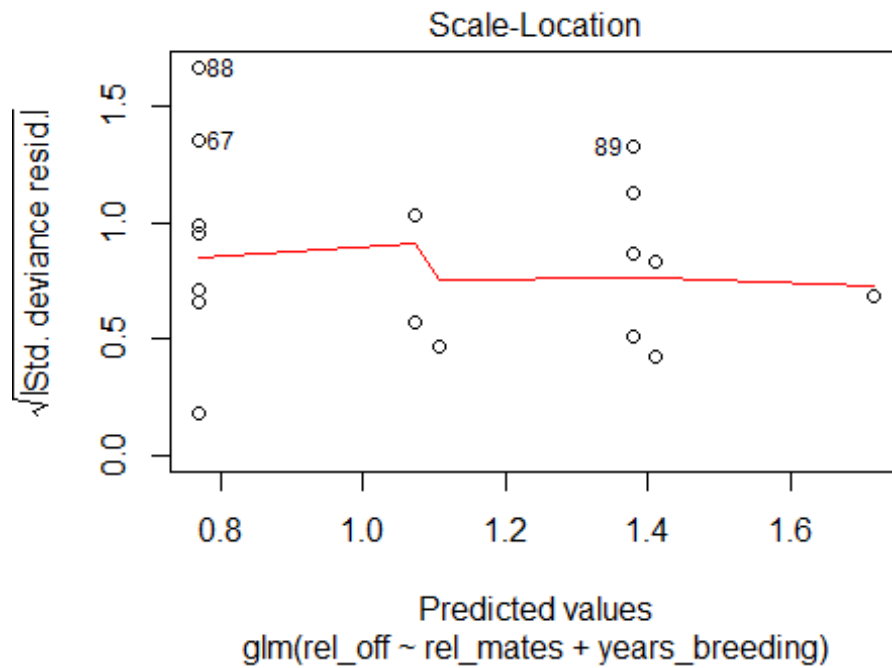




Then, for males:

```
#plot residual diagnostics for male glm
plot(glm_male_nonzero)
```





The residual diagnostics are similar for both males and females when using the relative, rather than absolute, values in the model with a Gaussian distribution.

Finally, I made some plots for the publications using ggplot2 and saves them as tiffs.

```

#Load graphics packages
library(ggplot2)
library(extrafont)

## Registering fonts with R

#save the resulting graph as a tiff
tiff(file="female_Bateman.tiff", width=6, height=4, units="in", res=100)

#plot RRS against StSVL using ggplot for females
ggplot(female, aes(x=mates, y=offspring)) + geom_point(size=2) +
theme_classic() + labs(x="Mating Sucess", y="Reproductive Success") +
theme(axis.title.x = element_text(size=14, family = "Arial"), axis.title.y =
element_text(size=14, family = "Arial"), axis.text = element_text(size=12,
family = "Arial"))

#over-ride graphics device
dev.off()

## png
## 2

#save the resulting graph as a tiff
tiff(file="male_Bateman.tiff", width=6, height=4, units="in", res=100)

#plot RRS against StSVL using ggplot for males
ggplot(male_nonzero, aes(x=mates, y=offspring)) + geom_point(size=2) +
theme_classic() + labs(x="Mating Sucess", y="Reproductive Success") +
theme(axis.title.x = element_text(size=14, family = "Arial"), axis.title.y =
element_text(size=14, family = "Arial"), axis.text = element_text(size=12,
family = "Arial"))

#over-ride graphics device
dev.off()

## png
## 2

```

.....
.....

Next, I tested for selection on male SVL.

First, I loaded the data file and set sire ID as a factor. Then, I calculated relative reproductive success, relative mating success, and mean-standardized SVL from the data.

```

#read raw data
atrox_SVL <- read.csv("atrox_RRS_StSVL.csv")

#make sire ID a factor to later fit as a random effect in the model
atrox_SVL$Sire <- as.factor(atrox_SVL$Sire)

```

```

#calculate mean-standardized SVL
mean_svl <- mean(atrox_SVL[["SVL"]]) #calculate mean SVL
stdev_svl <- sd(atrox_SVL[["SVL"]]) #calculate standard deviation SVL
atrox_SVL$StSVL <- (atrox_SVL$SVL - mean_svl)/stdev_svl #calculate mean
standardized SVL and add to data frame

#calculate relative reproductive success
mean_RS <- mean(atrox_SVL[["RS"]]) #calculate mean RS
atrox_SVL$RRS <- atrox_SVL$RS/mean_RS #calculate relative RS and add to data
frame

#calculate relative mating success
mean_MS <- mean(atrox_SVL[["MS"]]) #calculate mean MS
atrox_SVL$RMS <- atrox_SVL$MS/mean_MS #calculate relative MS and add to data
frame

#confirm all calculations added to data frame by viewing the first few rows
head(atrox_SVL)

##      Sire Year RS MS  SVL      StSVL      RRS RMS
## 1 CA-108 2010  5  1  682 -3.101429016  2.109375 0.9
## 2 CA-20  2007  2  1  850 -1.111319805  0.843750 0.9
## 3 CA-23  2003  2  1  943 -0.009652205  0.843750 0.9
## 4 CA-27  2006  4  2 1050  1.257857828  1.687500 1.8
## 5 CA-40  2006  1  1 1017  0.866943519  0.421875 0.9
## 6 CA-43  2002  3  1  935 -0.104419310  1.265625 0.9

```

Then, I ran a linear mixed effects model to test for an effect of mean-standardized annual male SVL on annual reproductive success (= linear selection differential). We have repeated measures in this data set (males siring offspring in multiple years), so I included sire ID as a random effect to account for non-independence among data points.

```

#Load package
library(lme4)

## Loading required package: Matrix

#run linear mixed effects model with sire ID set as a random effect
mod_RRS <- lmer(RRS ~ StSVL + (1|Sire), data=atrox_SVL)

#view summary of the model
summary(mod_RRS)

## Linear mixed model fit by REML ['lmerMod']
## Formula: RRS ~ StSVL + (1 | Sire)
## Data: atrox_SVL
##
## REML criterion at convergence: 52
##

```



```
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.4843 -0.2948 -0.1041  0.3766  1.5212
##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   Sire      (Intercept) 0.50744  0.7123
##   Residual                0.07272  0.2697
## Number of obs: 27, groups: Sire, 19
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  1.09463    0.17263   6.341
## StSVL        0.07024    0.13080   0.537
##
## Correlation of Fixed Effects:
##      (Intr)
## StSVL 0.035
```

Then, I tested the significance of fixed effects in the model (mean-standardized SVL) using the drop1 function and an LRT.

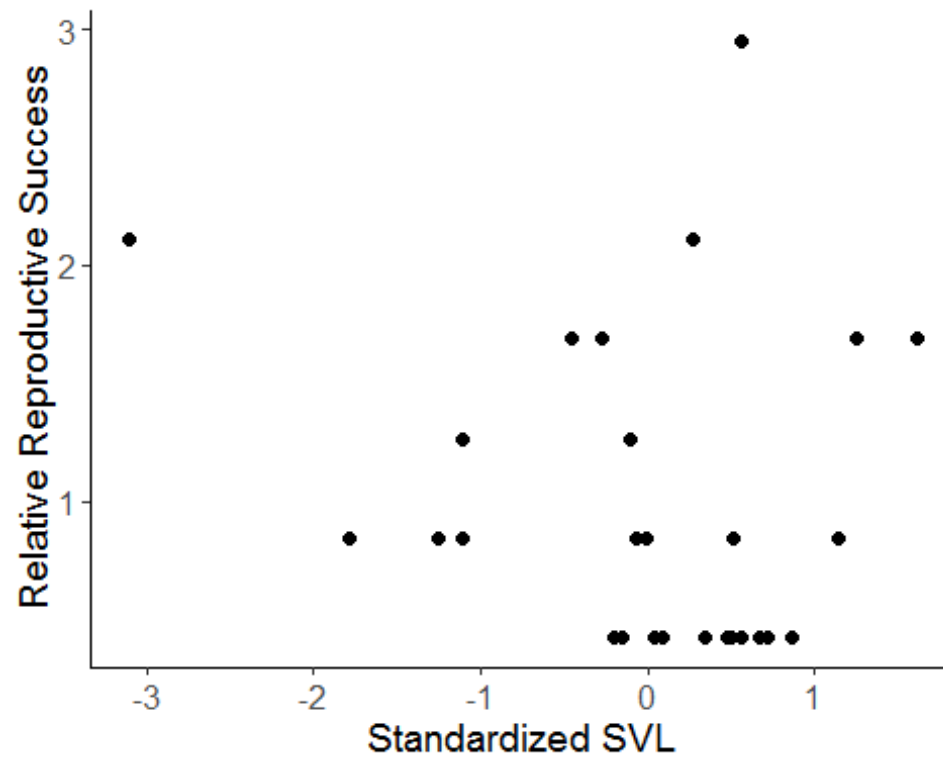
```
#drop1 to test significance of model without StSVL
drop1(mod_RRS, test="Chisq")
```

```
## Single term deletions
##
## Model:
## RRS ~ StSVL + (1 | Sire)
##      Df    AIC    LRT Pr(Chi)
## <none>  56.004
## StSVL   1 54.156 0.15206  0.6966
```

There is no significant effect of male mean-standardized SVL on male relative reproductive success.

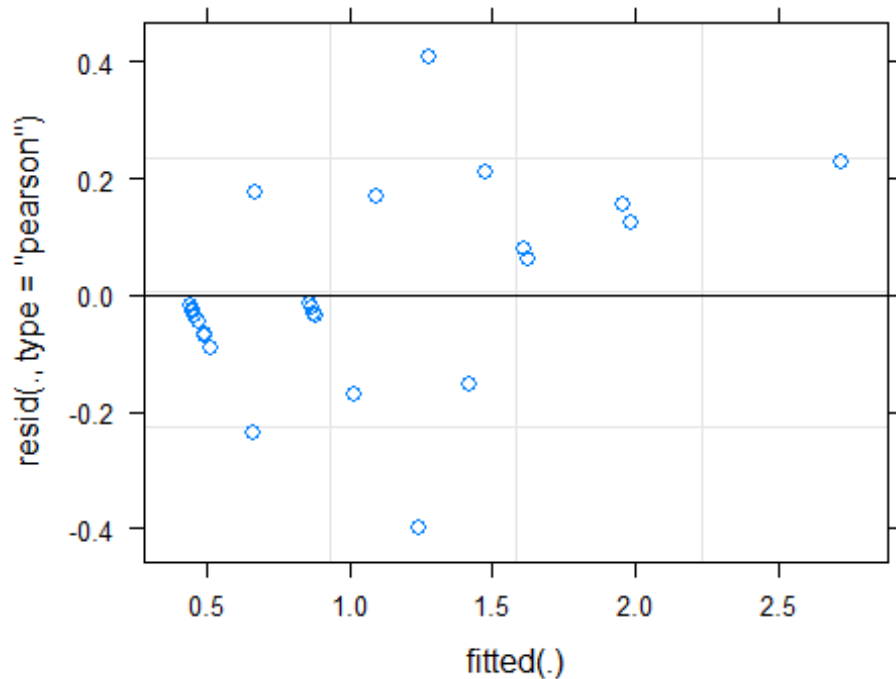
I then plotted annual relative reproductive success against mean-standardized SVL.

```
#plot RRS against StSVL using ggplot
ggplot(atrox_SVL, aes(x=StSVL, y=RRS)) + geom_point(size=2) + theme_classic()
+ labs(x="Standardized SVL", y="Relative Reproductive Success") +
theme(axis.title.x = element_text(size=14, family = "Arial"), axis.title.y =
element_text(size=14, family = "Arial"), axis.text = element_text(size=12,
family = "Arial"))
```



I also evaluated model residuals.

```
#evaluate model residuals  
plot(mod_RRS)
```



I also ran a linear mixed effects model to test for an effect of mean-standardized annual male SVL on relative mating success (= linear mating differential). We have repeated measures in this data set (males siring offspring across years), so I included sire ID as a random effect to account for non-independence among data points. I received a warning that variance of the sire random effect was very small; this was unsurprising given that almost all male records (except for 3) had the same value of RMS.

```
#use lmer (package lme4) to regress RMS on to StSVL while controlling for
repeated measures
mod_RMS <- lmer(RMS ~ StSVL + (1|Sire), data=atrox_SVL)
```

```
## singular fit
```

```
#view summary of model
summary(mod_RMS)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: RMS ~ StSVL + (1 | Sire)
## Data: atrox_SVL
##
## REML criterion at convergence: 13.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.72548 -0.52115 -0.35501  0.00018  2.95214
##
## Random effects:
```

```
## Groups      Name      Variance Std.Dev.
## Sire        (Intercept) 0.00000  0.0000
## Residual                0.07797  0.2792
## Number of obs: 27, groups: Sire, 19
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  1.00000    0.05374  18.609
## StSVL        0.09003    0.05476   1.644
##
## Correlation of Fixed Effects:
##      (Intr)
## StSVL 0.000
## convergence code: 0
## singular fit
```

Then, I tested the significance of fixed effects in the model (mean-standardized SVL) using the drop1 function and an LRT.

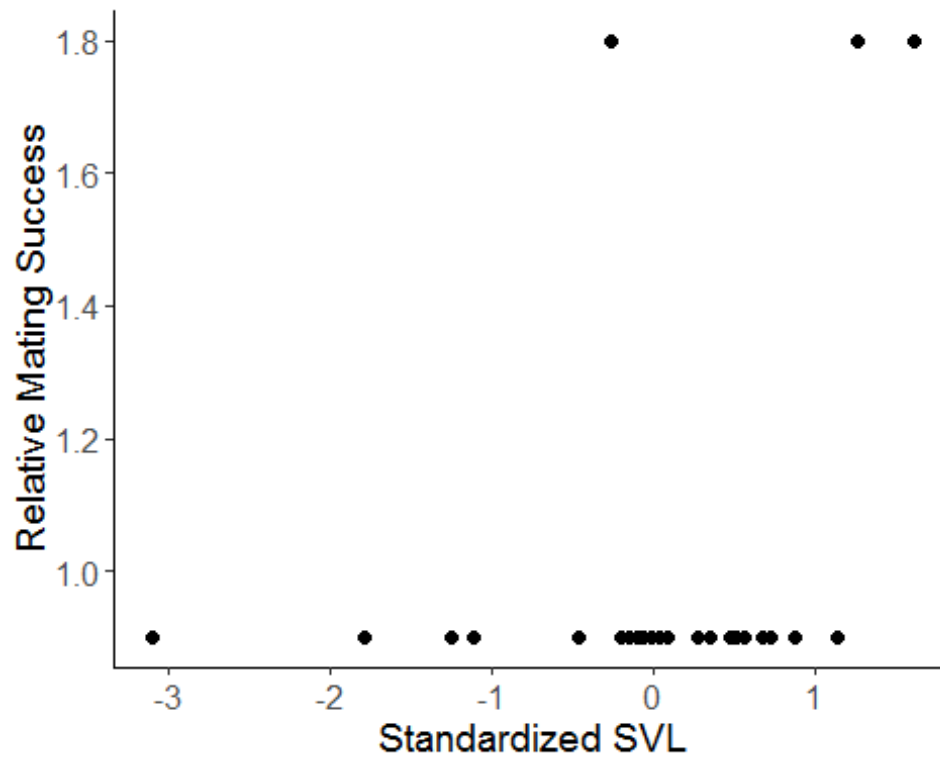
```
#drop1 to test significance of model without StSVL
drop1(mod_RMS, test="Chisq")

## singular fit

## Single term deletions
##
## Model:
## RMS ~ StSVL + (1 | Sire)
##      Df    AIC    LRT Pr(Chi)
## <none>   13.656
## StSVL   1 14.428 2.7717 0.09594 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

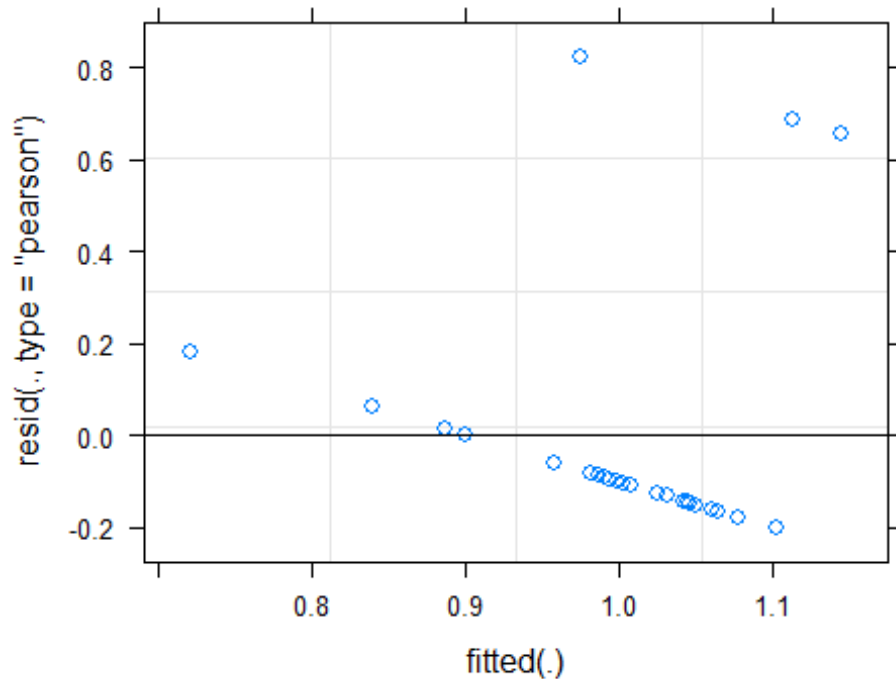
I plotted relative mating success against mean-standardized SVL.

```
#plot RMS against StSVL using ggplot
ggplot(atrox_SVL, aes(x=StSVL, y=RMS)) + geom_point(size=2) + theme_classic()
+ labs(x="Standardized SVL", y="Relative Mating Success") +
theme(axis.title.x = element_text(size=14, family = "Arial"), axis.title.y =
element_text(size=14, family = "Arial"), axis.text = element_text(size=12,
family = "Arial"))
```



I also evaluated model residuals. They don't look great, but there's not much to be done about this. This could also be because we have a small sample size and we used a mixed effects model.

```
#plot model residuals  
plot(mod_RMS)
```



Finally, I made some plots for publications and saves them as tiffs.

```
#save resulting graph as a tiff
tiff(file="mating_diff.tiff", width=6, height=4, units="in", res=100)

#plot RMS against StSVL using ggplot
ggplot(atrox_SVL, aes(x=StSVL, y=RMS)) + geom_point(size=2) + theme_classic()
+ labs(x="Mean-Standardized SVL", y="Relative Mating Success") +
theme(axis.title.x = element_text(size=14, family = "Arial"), axis.title.y =
element_text(size=14, family = "Arial"), axis.text = element_text(size=12,
family = "Arial"))

#over-ride graphics device
dev.off()

## png
## 2

#save resulting graph as a tiff
tiff(file="selection_diff.tiff", width=6, height=4, units="in", res=100)

#plot RRS against StSVL using ggplot
ggplot(atrox_SVL, aes(x=StSVL, y=RRS)) + geom_point(size=2) + theme_classic()
+ labs(x="Mean-Standardized SVL", y="Relative Reproductive Success") +
theme(axis.title.x = element_text(size=14, family = "Arial"), axis.title.y =
element_text(size=14, family = "Arial"), axis.text = element_text(size=12,
family = "Arial"))
```

```
#over-ride graphics device  
dev.off()
```

```
## png  
## 2
```

```
.....  
.....
```

Then, I moved on to quantifying opportunities for sexual selection (Is) and selection (I).

I first quantified Is and I BY YEAR for males and females.

To do so, I first read in the data file. Then, I subset the file into a male and a female file.

```
#Load csv file  
opportunities <- read.csv("opportunities_selection.csv")
```

```
##### subset data  
#subset data frame and make new female data frame  
female_opp <- subset(opportunities, Sex=='f')
```

```
#subset data frame and make new male data frame  
male_opp <- subset(opportunities, Sex=='m')
```

Starting with the female data frame, I automated a subsetting of the female file to create subsets for each year with at least 3 data points. This is because we cannot calculate variance for years in which we have 2 or fewer records.

```
#create a dataframe (c_f) of years for which there are at least 3 female data points  
a_f <- rle(sort(female_opp$Year))  
b_f <- data.frame(year=a_f$values, n=a_f$lengths)  
c_f <- subset(b_f, n>'2') #subset those years that have more than 2 records  
  
#create a subset of the original female data set for each year with greater than 2 records  
for(i in c_f$year) { #for each year in the c_f data frame  
  assign(paste("f_", i, sep = ""), subset(female_opp, Year==i)) #create a subset of the original female data set and name it with an f (for female) and the respective year  
}  
  
#remove intermediate files  
rm(a_f, b_f)
```

For each yearly female data frame, I calculated annual opportunity for sexual selection (Is).

```

#create list of year data frames (these are the years with >2 records)
df_list <- list(f_2003, f_2004, f_2005, f_2006, f_2007)

#create empty data frame called opp_sex_f in which to store the Is values
opp_sex_f = NULL

#use for loop to calculate variables for female Is for each data frame
for(i in df_list) { #for each data frame in the df_list

  variance <- var(i$MS) #calculate the variance in MS
  x <- mean(i$MS) #calculate the mean in MS
  meansq <- x^2 #calculate the squared mean of MS
  rel_var <- variance/meansq #calculate opportunity for sexual selection
  opp_sex_f = rbind(opp_sex_f, data.frame(variance, x, meansq, rel_var))
#add all variables to the new empty data frame
}

#combine the data frame containing years and number of data points for
females (c_f) with the data frame containing calculations (opp_sex_f) so that
we know what year corresponds with each value
opp_sex_f <- cbind(c_f, opp_sex_f)

#### rename columns in opp_sex_f data frame
#load necessary package to do this
library(plyr)

#rename columns of data frame
opp_sex_f <- rename(opp_sex_f, c("year"= "Year", "n"="N (f)",
"variance"="Variance MS (f)", "x"="Mean MS (f)", "meansq" = "Squared Mean MS
(f)", "rel_var" = "Is (f)"))

#view new data frame
opp_sex_f

##   Year N (f) Variance MS (f) Mean MS (f) Squared Mean MS (f)    Is (f)
## 2 2003     6      0.6666667    1.666667      2.777778 0.2400000
## 3 2004     4      1.0000000    1.500000      2.250000 0.4444444
## 4 2005     4      1.3333333    2.000000      4.000000 0.3333333
## 5 2006     8      0.5714286    1.500000      2.250000 0.2539683
## 6 2007     6      0.2666667    1.333333      1.777778 0.1500000

```

For each female year data frame, I calculated annual opportunity for selection (I).

```

#create empty data frame called opp_sel_f in which to store the I values.
opp_sel_f = NULL

for(i in df_list) { #for each data frame in the previously created df_list

  variance <- var(i$RS) #calculate the variance in RS
  x <- mean(i$RS) #calculate the mean in RS

```



```

meansq <- x^2 #calculate the squared mean of RS
rel_var <- variance/meansq #calculate opportunity for selection
opp_sel_f = rbind(opp_sel_f, data.frame(variance, x, meansq, rel_var))
#add all variables to the empty data frame
}

#combine the data frame containing years and number of data points for
females (c_f) with the data frame containing I calculations (opp_sel_f) so we
know which years correspond with which values
opp_sel_f <- cbind(c_f, opp_sel_f)

#rename columns in opp_sel_f
opp_sel_f <- rename(opp_sel_f, c("year"= "Year", "n"="N (f)",
"variance"="Variance RS (f)", "x"="Mean RS (f)", "meansq" = "Squared Mean RS
(f)", "rel_var" = "I (f)"))

#view new data frame
opp_sel_f

##   Year N (f) Variance RS (f) Mean RS (f) Squared Mean RS (f)      I (f)
## 2 2003     6      1.366667   3.166667      10.02778 0.1362881
## 3 2004     4     10.666667   5.000000      25.00000 0.4266667
## 4 2005     4      4.250000   3.250000      10.56250 0.4023669
## 5 2006     8      4.267857   3.375000      11.39062 0.3746816
## 6 2007     6      8.666667   3.333333      11.11111 0.7800000

```

I calculated MEAN opportunities for sexual selection and selection for females.

```

mean_Is_f <- mean(opp_sex_f$`Is (f)` ) #mean opportunity for sexual selection
mean_I_f <- mean(opp_sel_f$`I (f)` ) #mean opportunity for selection

#cat the means so we know what they are
cat("Female Mean Is: ", mean_Is_f, "\n", "Female Mean I: ", mean_I_f, "\n\n")

## Female Mean Is:  0.2843492
## Female Mean I:  0.4240006

```

Then, for the male data frame, I automated a subsetting of the data frame to create subsets for each year with at least 3 data points. This is because we cannot calculate variance for years with fewer than 3 records. This is the same thing I did for females.

```

#create a dataframe (c_m) of years for which there are at least 3 male data
points
a_m <- rle(sort(male_opp$Year))
b_m <- data.frame(year=a_m$values, n=a_m$lengths)
c_m <- subset(b_m, n>'2') #subset the male data frame for only years with >2
records

#for each year in the c_m data frame, create a subset of the original male

```

```

data set
for(i in c_m$year) { #for each year with more than 2 records

  assign(paste("m_", i, sep = ""), subset(male_opp, Year==i)) #create a
subset for each year for males, and name with an m and the respective year

}

#remove intermediate data frames
rm(a_m, b_m)

```

Then, for each yearly male data frame, I calculated annual opportunity for sexual selection (Is).

```

#create list of year data frames (note: overwriting the female df_list)
df_list <- list(m_2003, m_2005, m_2006, m_2007)

#create empty data frame called opp_sex_m in which to store Is values
opp_sex_m = NULL

for(i in df_list) { #for each year data frame in the df_list

  variance <- var(i$MS) #calculate the variance in MS
  x <- mean(i$MS) #calculate the mean in MS
  meansq <- x^2 #calculate the squared mean of MS
  rel_var <- variance/meansq #calculate opportunity for sexual selection
  opp_sex_m = rbind(opp_sex_m, data.frame(variance, x, meansq, rel_var))
#add all variables to the empty data frame
}

#combine the data frame containing years and number of data points (c_m) with
the data frame containing Is calculations (opp_sex_m) so we know which years
correspond with which values
opp_sex_m <- cbind(c_m, opp_sex_m)

#rename columns in opp_sex_m
opp_sex_m <- rename(opp_sex_m, c("year"= "Year", "n"="N (m)",
"variance"="Variance MS (m)", "x"="Mean MS (m)", "meansq" = "Squared Mean MS
(m)", "rel_var" = "Is (m)"))

#view new data frame
opp_sex_m

```

##	Year	N (m)	Variance MS (m)	Mean MS (m)	Squared Mean MS (m)	Is (m)
## 2	2003	6	0.0000000	1.00	1.0000	0.0000000
## 4	2005	4	0.0000000	1.00	1.0000	0.0000000
## 5	2006	8	0.2142857	1.25	1.5625	0.1371429
## 6	2007	5	0.2000000	1.20	1.4400	0.1388889

Then, for each yearly male data frame, I calculated annual opportunity for selection.

```

#create empty data frame in which to store I values
opp_sel_m = NULL

for(i in df_list) { #for each data frame in the df_list

  variance <- var(i$RS) #calculate the variance in RS
  x <- mean(i$RS) #calculate the mean in RS
  meansq <- x^2 #calculate the squared mean of RS
  rel_var <- variance/meansq #calculate opportunity for selection
  opp_sel_m = rbind(opp_sel_m, data.frame(variance, x, meansq, rel_var))
#add all variables to the empty data frame
}

#combine the data frame containing years and number of data points (c_m) with
the data frame containing I calculations (opp_sel_m) so we know which years
correspond with which values
opp_sel_m <- cbind(c_m, opp_sel_m)

#rename columns in opp_sel_m
opp_sel_m <- rename(opp_sel_m, c("year"= "Year", "n"="N (m)",
"variance"="Variance RS (m)", "x"="Mean RS (m)", "meansq" = "Squared Mean RS
(m)", "rel_var" = "I (m)"))

#view new data frame
opp_sel_m

##   Year N (m) Variance RS (m) Mean RS (m) Squared Mean RS (m)      I (m)
## 2 2003     6      1.366667    2.166667      4.694444 0.2911243
## 4 2005     4      0.250000    1.250000      1.562500 0.1600000
## 5 2006     8      4.125000    2.875000      8.265625 0.4990548
## 6 2007     5      3.300000    2.600000      6.760000 0.4881657

```

Then, I calculated MEAN opportunities for sexual selection and selection for males.

```

mean_Is_m <- mean(opp_sex_m$`Is (m)`) #mean opportunity for sexual selection
mean_I_m <- mean(opp_sel_m$I (m)`) #mean opportunity for selection

#cat the means so we know what they are
cat("Male Mean Is: ", mean_Is_m, "\n", "Male Mean I: ", mean_I_m, "\n\n")

## Male Mean Is:  0.06900794
## Male Mean I:  0.3595862

```

I statistically compared YEARLY male and female Is with an F-ratio test.

```

#create a merged data frame of yearly data that is present for BOTH males and
females
opp_sex_merged <- merge(opp_sex_f, opp_sex_m, by.x = "Year") #merge the male
and female opp_sex dataframes by year
head(opp_sex_merged) #view the merged data frame to make sure the merge
worked correctly

```

```
## Year N (f) Variance MS (f) Mean MS (f) Squared Mean MS (f) Is (f)
## 1 2003 6 0.6666667 1.666667 2.777778 0.2400000
## 2 2005 4 1.3333333 2.000000 4.000000 0.3333333
## 3 2006 8 0.5714286 1.500000 2.250000 0.2539683
## 4 2007 6 0.2666667 1.333333 1.777778 0.1500000
## N (m) Variance MS (m) Mean MS (m) Squared Mean MS (m) Is (m)
## 1 6 0.0000000 1.00 1.0000 0.0000000
## 2 4 0.0000000 1.00 1.0000 0.0000000
## 3 8 0.2142857 1.25 1.5625 0.1371429
## 4 5 0.2000000 1.20 1.4400 0.1388889
```

```
cat("\n\n") #cat some newlines
```

```
#write a function to compare male and female Is for each year with F Ratio test
```

```
func_sex_comp <- function(x, output) { #function to be applied to each row
```

```
  #calculate degrees of freedom for males and females
```

```
  df_f <- x["N (f)"] - 1 #degrees of freedom for females = #females - 1
```

```
  df_m <- x["N (m)"] - 1 #degrees of freedom for males = #males - 1
```

```
  #use if/else to determine which values to put in numerator/denominator and order for F Ratio
```

```
  if(x["Is (f)"] > x["Is (m)]) { #if female Is is greater than male Is
```

```
    crit <- qf(0.95, df1 = df_f, df2 = df_m) # use female df as df1 in calculation of critical value
```

```
    f <- x["Is (f)"]/x["Is (m)"] #use female estimate as numerator in f ratio
```

```
    cat("Note: For", x["Year"], ", female Is value is numerator.\n") #cat description
```

```
  } else { #otherwise
```

```
    crit <- qf(0.95, df1 = df_m, df2 = df_f) # use male df as df1 in calculation of critical value
```

```
    f <- x["Is (m)"]/x["Is (f)"] #use male estimate as numerator in f ratio
```

```
    cat("Note: For", x["Year"], ", male Is value is numerator.\n") #cat description
```

```
  }
```

```
#test whether male and female Is are significantly different by comparing F Ratio with critical value
```

```
if(f > crit){ #if the f ratio is larger than the critical value
```

```
  cat("For", x["Year"], ", male and female Is are different (F > critical).", "\n", "Female Is: ", round(x["Is (f)"], digits = 2), "\n", "Male Is: ", round(x["Is (m)"], digits = 2), "\n", "F: ", round(f, digits = 2),
```

```

"\n", "Critical Value: ", round(crit, digits = 2), "\n", "Female Degrees of
Freedom: ", df_f, "\n", "Male Degrees of Freedom: ", df_m, "\n\n\n") #cat
statistical metrics and parameters

} else { #otherwise

    cat("For", x["Year"], ", male and female Is are not different (F <
critical).", "\n", "Female Is: ", round(x["Is (f)"], digits = 2), "\n", "Male
Is: ", round(x["Is (m)"], digits = 2), "\n", "F: ", round(f, digits = 2),
"\n", "Critical Value: ", round(crit, digits = 2), "\n", "Female Degrees of
Freedom: ", df_f, "\n", "Male Degrees of Freedom: ", df_m, "\n\n\n") #cat
statistical metrics and parameters

}

}

#run the function on the opp_sex_merged data frame
apply(opp_sex_merged, 1, func_sex_comp)

## Note: For 2003 , female Is value is numerator.
## For 2003 , male and female Is are different (F > critical).
## Female Is: 0.24
## Male Is: 0
## F: Inf
## Critical Value: 5.05
## Female Degrees of Freedom: 5
## Male Degrees of Freedom: 5
##
##
## Note: For 2005 , female Is value is numerator.
## For 2005 , male and female Is are different (F > critical).
## Female Is: 0.33
## Male Is: 0
## F: Inf
## Critical Value: 9.28
## Female Degrees of Freedom: 3
## Male Degrees of Freedom: 3
##
##
## Note: For 2006 , female Is value is numerator.
## For 2006 , male and female Is are not different (F < critical).
## Female Is: 0.25
## Male Is: 0.14
## F: 1.85
## Critical Value: 3.79
## Female Degrees of Freedom: 7
## Male Degrees of Freedom: 7
##
##

```

```
## Note: For 2007 , female Is value is numerator.
## For 2007 , male and female Is are not different (F < critical).
## Female Is: 0.15
## Male Is: 0.14
## F: 1.08
## Critical Value: 6.26
## Female Degrees of Freedom: 5
## Male Degrees of Freedom: 4

## NULL
```

Then, I statistically compared YEARLY male and female I with an F-ratio test.

```
#create a merged data frame of yearly data that is present for BOTH males and females
```

```
opp_sel_merged <- merge(opp_sel_f, opp_sel_m, by.x = "Year") #merge the male and female opp_sel dataframes by year
```

```
head(opp_sel_merged) #view the merged data frame to make sure the merge worked correctly
```

```
##   Year N (f) Variance RS (f) Mean RS (f) Squared Mean RS (f)    I (f)
## 1 2003     6      1.366667   3.166667      10.02778 0.1362881
## 2 2005     4      4.250000   3.250000      10.56250 0.4023669
## 3 2006     8      4.267857   3.375000      11.39062 0.3746816
## 4 2007     6      8.666667   3.333333      11.11111 0.7800000
##   N (m) Variance RS (m) Mean RS (m) Squared Mean RS (m)    I (m)
## 1     6      1.366667   2.166667      4.694444 0.2911243
## 2     4      0.250000   1.250000      1.562500 0.1600000
## 3     8      4.125000   2.875000      8.265625 0.4990548
## 4     5      3.300000   2.600000      6.760000 0.4881657
```

```
cat("\n\n") #cat some new lines
```

```
#write a function to compare male and female I for each year with an F ratio test
```

```
func_sel_comp <- function(x, output) { #function to be applied to each row
```

```
  #calculate degrees of freedom for males and females
```

```
  df_f <- x["N (f)"] - 1 #degrees of freedom for females = #females - 1
```

```
  df_m <- x["N (m)"] - 1 #degrees of freedom for males = #males - 1
```

```
  #use if/else to determine which values to put in numerator/denominator and order for F Ratio
```

```
  if(x["I (f)"] > x["I (m)]) { #if female I is greater than male I
```

```
    crit <- qf(0.95, df1 = df_f, df2 = df_m) # use female df as df1 in calculation of critical value
```

```
    f <- x["I (f)"]/x["I (m)"] #use female estimate as numerator in f ratio
```

```
    cat("Note: For", x["Year"],", female I value is numerator.\n") #cat description
```

```

} else { #otherwise

    crit <- qf(0.95, df1 = df_m, df2 = df_f) # use male df as df1 in
calculation of critical value
    f <- x["I (m)"]/x["I (f)"] #use male estimate as numerator in f ratio
    cat("Note: For", x["Year"],", male I value is numerator.\n") #cat
description

}

#test whether male and female I are significantly different by comparing F
Ratio with critical value
if(f > crit){ #if the f ratio is larger than the critical value

    cat("For", x["Year"], ", male and female I are different (F >
critical).", "\n", "Female I: ", round(x["I (f)"], digits = 2), "\n", "Male
I: ", round(x["I (m)"], digits = 2), "\n", "F: ", round(f, digits = 2), "\n",
"Critical Value: ", round(crit, digits = 2), "\n", "Female Degrees of
Freedom: ", df_f, "\n", "Male Degrees of Freedom: ", df_m, "\n\n\n") #cat
statistical metrics and parameters

} else { #otherwise

    cat("For", x["Year"], ", male and female I are not different (F <
critical).", "\n", "Female I: ", round(x["I (f)"], digits = 2), "\n", "Male
I: ", round(x["I (m)"], digits = 2), "\n", "F: ", round(f, digits = 2), "\n",
"Critical Value: ", round(crit, digits = 2), "\n", "Female Degrees of
Freedom: ", df_f, "\n", "Male Degrees of Freedom: ", df_m, "\n\n\n") #cat
statistical metrics and parameters

}

}

#run the function on the opp_sel_merged data frame
apply(opp_sel_merged, 1, func_sel_comp)

## Note: For 2003 , male I value is numerator.
## For 2003 , male and female I are not different (F < critical).
## Female I: 0.14
## Male I: 0.29
## F: 2.14
## Critical Value: 5.05
## Female Degrees of Freedom: 5
## Male Degrees of Freedom: 5
##
##
## Note: For 2005 , female I value is numerator.

```

```

## For 2005 , male and female I are not different (F < critical).
## Female I: 0.4
## Male I: 0.16
## F: 2.51
## Critical Value: 9.28
## Female Degrees of Freedom: 3
## Male Degrees of Freedom: 3
##
##
## Note: For 2006 , male I value is numerator.
## For 2006 , male and female I are not different (F < critical).
## Female I: 0.37
## Male I: 0.5
## F: 1.33
## Critical Value: 3.79
## Female Degrees of Freedom: 7
## Male Degrees of Freedom: 7
##
##
## Note: For 2007 , female I value is numerator.
## For 2007 , male and female I are not different (F < critical).
## Female I: 0.78
## Male I: 0.49
## F: 1.6
## Critical Value: 6.26
## Female Degrees of Freedom: 5
## Male Degrees of Freedom: 4

## NULL

```

I also statistically compared MEAN annual male and female Is with an F-ratio test.

```

## degrees of freedom for females calculated as 1 - MEAN number of females
per year
df_f <- mean(c_f$n)-1 #number of records per year are still stored in the c_f
data frame
df_f <- round(df_f, digits = 0) #round to the nearest integer

## degrees of freedom for males calculated as 1 - MEAN number of males per
year
df_m <- mean(c_m$n)-1 #number of records per year are still stored in c_m
data frame
df_m <- round(df_m, digits = 0) #round to the nearest integer

#calculate F-ratio and critical value for Is, again paying attention to
whether males or females have the greater Is value
if(mean_Is_f > mean_Is_m){ #if mean female Is is greater than mean male Is

  crit <- qf(0.95, df1 = df_f, df2 = df_m) # use female df as df1 in
calculation of critical value

```



```

f <- mean_Is_f/mean_Is_m #use female estimate as numerator in f ratio
cat("Note: Female Is value is numerator.\n\n") #cat description

} else { #otherwise

  crit <- qf(0.95, df1 = df_m, df2 = df_f) # use male df as df1 in
calculation of critical value
  f <- mean_Is_m/mean_Is_f #use male estimate as numerator in f ratio
  cat("Note: Male Is value is numerator.\n\n") #cat description

}

## Note: Female Is value is numerator.

#test whether male and female Is are significantly different by comparing F
ratio to critical value
if(f > crit){ #if the f ratio is larger than the critical value

  cat("Mean male and female Is are different (F > critical).", "\n", "Male
Is: ", round(mean_Is_m, digits = 2), "\n", "Female Is: ", round(mean_Is_f,
digits = 2), "\n", "F-Ratio: ", round(f, digits = 2), "\n", "Critical Value:
", round(crit, digits = 2), "\n", "Male Degrees of Freedom: ", df_m, "\n",
"Female Degrees of Freedom: ", df_f, "\n\n\n") #cat statistical metrics and
parameters

} else { #otherwise

  cat("Mean male and female Is are not different (F < critical).", "\n",
"Male Is: ", round(mean_Is_m, digits = 2), "\n", "Female Is: ",
round(mean_Is_f, digits = 2), "\n", "F-Ratio: ", round(f, digits = 2), "\n",
"Critical Value: ", round(crit, digits = 2), "\n", "Male Degrees of Freedom:
", df_m, "\n", "Female Degrees of Freedom:", df_f, "\n\n\n" ) #cat
statistical metrics and parameters

}

## Mean male and female Is are not different (F < critical).
## Male Is: 0.07
## Female Is: 0.28
## F-Ratio: 4.12
## Critical Value: 5.05
## Male Degrees of Freedom: 5
## Female Degrees of Freedom: 5

```

Then, I statistically compared MEAN annual male and female I with an F-ratio test.

```

#Note: I can use the same degrees of freedom as for Is

#calculate F-ratio and critical value for I
if(mean_I_f > mean_I_m){ #if mean female I is greater than male I

```

```

    crit <- qf(0.95, df1 = df_f, df2 = df_m) # use female df as df1 in
calculation of critical value
    f <- mean_I_f/mean_I_m #use female estimate as numerator in f ratio
    cat("Note: Female I value is numerator.\n\n") #cat description

} else { #otherwise

    crit <- qf(0.95, df1 = df_m, df2 = df_f) # use male df as df1 in
calculation of critical value
    f <- mean_I_m/mean_I_f #use male estimate as numerator in f ratio
    cat("Note: Male I value is numerator.\n\n") #cat description

}

## Note: Female I value is numerator.

#test whether male and female Is are significantly different by comparing F
ratio to critical value
if(f > crit){ #if the f ratio is larger than the critical value

    cat("Mean male and female I are different (F > critical).", "\n", "Male I:
", round(mean_I_m, digits = 2), "\n", "Female I: ", round(mean_I_f, digits =
2), "\n", "F-Ratio: ", round(f, digits = 2), "\n", "Critical Value: ",
round(crit, digits = 2), "\n", "Male Degrees of Freedom: ", df_m, "\n",
"Female Degrees of Freedom: ", df_f, "\n\n\n") #cat statistical metrics and
parameters

} else { #otherwise

    cat("Mean male and female I are not different (F < critical).", "\n", "Male
I: ", round(mean_I_m, digits = 2), "\n", "Female I: ", round(mean_I_f, digits
= 2), "\n", "F-Ratio: ", round(f, digits = 2), "\n", "Critical Value: ",
round(crit, digits = 2), "\n", "Male Degrees of Freedom: ", df_m, "\n",
"Female Degrees of Freedom: ", df_f, "\n\n\n") #cat statistical metrics and
parameters

}

## Mean male and female I are not different (F < critical).
## Male I: 0.36
## Female I: 0.42
## F-Ratio: 1.18
## Critical Value: 5.05
## Male Degrees of Freedom: 5
## Female Degrees of Freedom: 5

```

```

.....
.....

```

Finally, using the means for males and females of annual opportunities for sexual selection and the previously calculated STANDARDIZED Bateman gradients, I calculated the Jones index ($s'max = Bss * \sqrt{Is}$). See Jones (2009).

I first calculated the Jones index for females.

```
#run glm with relative mating and reproductive success and extract
coefficients
coef_f <- coef(glm(rel_off ~ rel_mates + years_breeding, data=female,
family=gaussian))

#extract Bss from coefficients
Bss_f <- as.numeric(coef_f[2])

#calculate Jones index
Jones_f <- Bss_f*sqrt(mean_Is_f)

#cat the results of the analysis
cat("Female Standardized Bateman Gradient: ", Bss_f, "\n", "Female Mean Annual
Opportunity for Sexual Selection: ", mean_Is_f, "\n", "Square Root Mean Annual
Female Is: ", sqrt(mean_Is_f), "\n", "Female Jones Index: ", Jones_f)

## Female Standardized Bateman Gradient:  0.1509466
##  Female Mean Annual Opportunity for Sexual Selection:  0.2843492
##  Square Root Mean Annual Female Is:  0.533244
##  Female Jones Index:  0.08049137
```

Then, I calculated the Jones index for males.

```
#run glm with relative mating and reproductive success and extract
coefficients
coef_m <- coef(glm(rel_off ~ rel_mates + years_breeding, data=male_nonzero,
family=gaussian))

#extract Bss from coefficients
Bss_m <- as.numeric(coef_m[2])

#calculate Jones index
Jones_m <- Bss_m*sqrt(mean_Is_m)

#cat the results of the analysis
cat("Male Standardized Bateman Gradient: ", Bss_m, "\n", "Male Mean Annual
Opportunity for Sexual Selection: ", mean_Is_m, "\n", "Square Root Mean Annual
Male Is: ", sqrt(mean_Is_m), "\n", "Male Jones Index: ", Jones_m)

## Male Standardized Bateman Gradient:  0.5883374
##  Male Mean Annual Opportunity for Sexual Selection:  0.06900794
##  Square Root Mean Annual Male Is:  0.2626936
##  Male Jones Index:  0.1545525
```