Homework 5

Noah Johnson

March 11, 2018

BYSH 4.15.1 Conceptual Exercise 5

Why is the log of means, log($\bar{Y}$), not $\bar{Y}$, plotted against X when assessing the assumptions for Poisson regression?

Because Poisson regression assumes that $log(\lambda_X)$ is linear in X. To test this assumption empirically, we estimate $log(\lambda_X)$ at each X by finding $log(\bar{Y})$. Then we plot these estimates against X looking for a linear relationship.

BYSH 4.15.1 Conceptual Exercise 8 (Fish)

A state wildlife biologist collected data from 250 park visitors as they left at the end of their stay. Each were asked to report the number of fish they caught during their one week stay. On average visitors caught 21.5 fish per week.

(a) Define the response.

We should model the response Y as a Poisson random variable, representing the number of fish a park visitor catches during one week.

(b) What are the possible values for the response?

As with all Poisson variables, the set of possible outcomes is $\{0, 1, 2, \dots\}$.

(c) What does λ represent?

$\lambda = 21.5$ represents $E[Y]$, the average number of fish a park visitor catches per week.

(d) Would a zero-inflated model be considered here? If so, what would be a "true zero?"

We should consider a zero-inflated model if during the process of exploratory data analysis it was determined that there were more zero counts of fish caught than we would expect to see if responses were coming from a Poisson distribution with $\lambda = 21.5$.

"True zeroes" would be those park visitors who will always report catching zero fish, because they never fish. Perhaps they are vegans, or have a degree in marine biology and know that fish are friends, not food. A zero-inflated Poisson model could estimate α, the proportion of "true zeroes".

BYSH 4.15.2 Guided Exercise 1 (Elephant Mating)

How does age affect male elephant mating patterns? An article by Poole (1989) investigated whether mating success in male elephants increases with age and whether there is a peak age for mating success. To address this question, the research team followed 41 elephants for one year and recorded both their ages and their number of matings. The data is found in elephant.csv, and relevant R code can be found under elephantMating.R.

The variables are:
MATINGS: the number of matings in a given year
AGE: the age of the elephant in years.

```
matings <- read.csv(paste("https://github.com/broadenyourstatisticalhorizons/bysh_book/",
    "raw/master/data/elephant.csv", sep = ""))
```

(f) Are the number of matings significantly related to age? Test with

```
matings.fit <- glm(MATINGS ~ AGE, family = poisson, data = matings)
matings.fit.sum <- summary(matings.fit)
```

i. a Wald test

```
# Grab estimate and standard error of estimate
age.slope <- matings.fit.sum$coefficients[[2, 1]]
age.se <- matings.fit.sum$coefficients[[2, 2]]

# Calculate Wald test statistic
wald.test.stat <- age.slope/age.se
print(wald.test.stat)
```

```
## [1] 4.997375
```

```
# Compare test statistic to standard normal distribution using 2-sided Z test
wald.test.p <- 2 * pnorm(abs(wald.test.stat), lower.tail = FALSE)
print(wald.test.p)
```

```
## [1] 5.81159e-07
```

With a p-value of 5.8115903×10^{-7}, the Wald test concludes that the number of matings is significantly related to age.

ii. a drop in deviance test

```
# Create a null model to compare to
null.model <- glm(MATINGS ~ 1, family = poisson, data = matings)

# Calculate the drop in deviance going from the null model to the linear model
# (lower deviance indicates a better fit).
drop.in.dev <- null.model$deviance - matings.fit$deviance
print(drop.in.dev)
```

```
## [1] 24.36011
```

```
# Calculate the change in degrees of freedom going from the null model to the
# linear model.
diff.in.df <- null.model$df.residual - matings.fit$df.residual
print(diff.in.df)
```

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

```
# Compute the p-value for this drop in deviance test by comparing the deviance
# drop to a chi-squared distribution.
p.value <- pchisq(drop.in.dev, diff.in.df, lower.tail = FALSE)
print(p.value)
```

```
## [1] 7.99062e-07
```

With a p-value of 7.9906203×10^{-7}, the drop in deviance test concludes that the linear model is significantly better than the null model, and so number of matings is significantly related to age.

(g) Add a quadratic term in AGE to determine whether there is a maximum age for the number of matings for elephants. Is a quadratic model preferred to a linear model? To investigate this

question, use

```
matings.quad.fit <- glm(MATINGS ~ AGE + I(AGE^2), family = poisson, data = matings)
matings.quad.fit.sum <- summary(matings.quad.fit)
print(matings.quad.fit$coefficients)
```

```
##  (Intercept)          AGE     I(AGE^2)
## -2.857405968  0.135954442 -0.000859507
```

```
# Coefficients of quadratic
a <- matings.quad.fit$coefficients[[3]]
b <- matings.quad.fit$coefficients[[2]]
c <- matings.quad.fit$coefficients[[1]]

# maximum of a quadratic is -b/2a
max_AGE <- -matings.quad.fit$coefficients[[2]]/(2 * matings.quad.fit$coefficients[[3]])
print(max_AGE)
```

```
## [1] 79.08861
```

```
# Find function value at maximum age
max_lambda <- predict(matings.quad.fit, data.frame(AGE = max_AGE))
print(max_lambda)
```

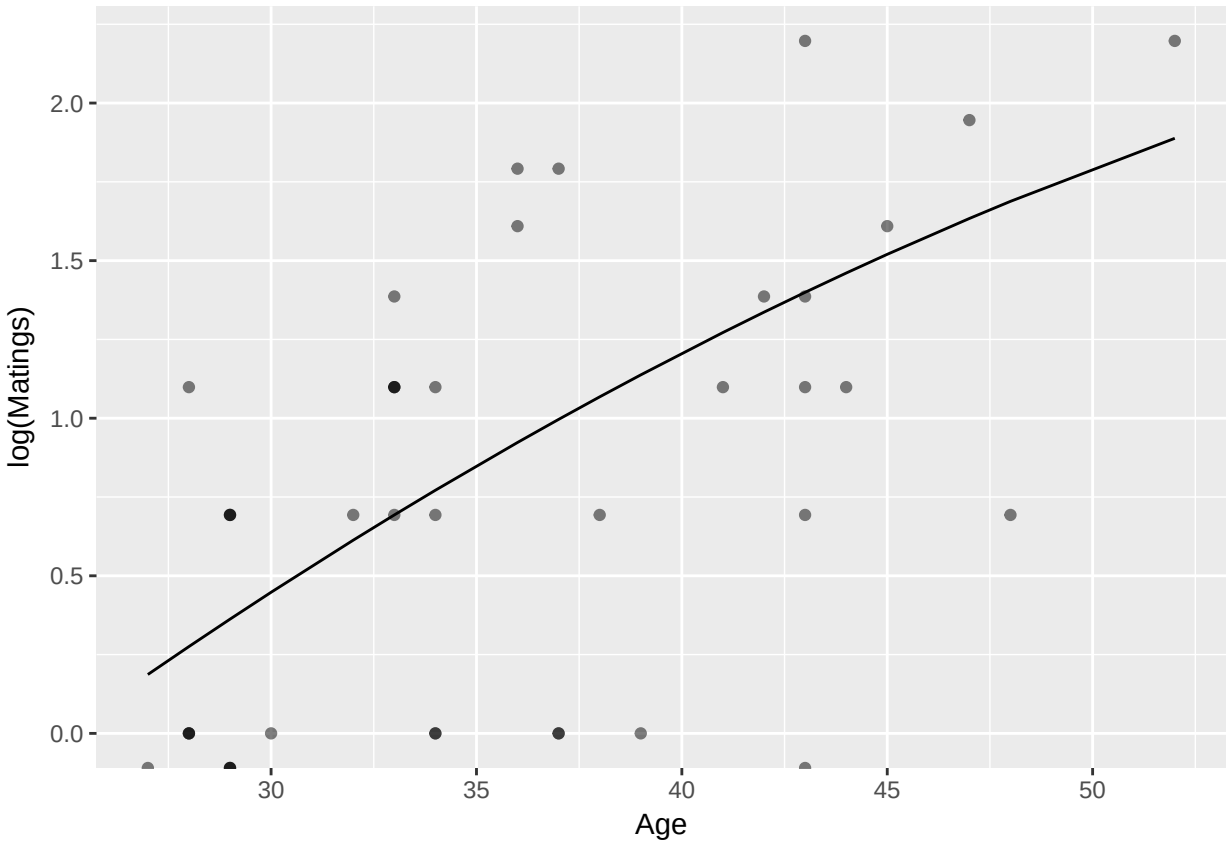
```
##        1
## 2.518818
```

The fit model is $log(meanMatings) = -8.5950705 \times 10^{-4} * Age^2 + 0.1359544$*Age + -2.857406, which has a maximum at $Age = 79.0886141$. The expected number of matings at this maximum is $e^{2.5188182} = 12.4139177$ matings per year. This is perhaps not so useful a result, since the largest observed age was 52, and the maximum observed number of matings was 9.

We can also look at this fit graphically.

```
# create new predictions dataframe with predictions for every observed age value
new_data <- data.frame(AGE = unique(matings$AGE))
new_data$log_lambda <- predict(matings.quad.fit, new_data)

# plot observations in log space, with the quadratic model's prediction curve
# overlayed on top
matings %>% ggplot(aes(x = AGE, y = log(MATINGS))) + geom_point(alpha = 0.5) + geom_line(aes(x = AGE,
    y = log_lambda), col = "black", new_data) + xlab("Age") + ylab("log(Matings)")
```



I don't see much to suggest that the quadratic fit should be used instead of a linear one. Now let's use statistical tests to look for evidence of this intuition.

i. a Wald test

```
# Grab estimate and standard error of estimate
age.sq.slope <- matings.quad.fit.sum$coefficients[[3, 1]]
age.sq.se <- matings.quad.fit.sum$coefficients[[3, 2]]

# Calculate Wald test statistic
wald.test.stat <- age.sq.slope/age.sq.se
print(wald.test.stat)
```

```
## [1] -0.4271075
```

```
# Compare test statistic to standard normal distribution using 2-sided Z test
wald.test.p <- 2 * pnorm(abs(wald.test.stat), lower.tail = FALSE)
print(wald.test.p)
```

```
## [1] 0.669301
```

With a p-value of 0.669301, the Wald test concludes that there is no evidence of a relationship between number of matings and the quadratic age variable.

ii. a drop in deviance test

```
# Calculate the drop in deviance going from the linear model to the quadratic
# model (lower deviance indicates a better fit).
drop.in.dev <- matings.fit$deviance - matings.quad.fit$deviance
print(drop.in.dev)
```

```
## [1] 0.1854446
```

```
# Calculate the change in degrees of freedom going from the ljinear model to the
# quadratic model.
diff.in.df <- matings.fit$df.residual - matings.quad.fit$df.residual
print(diff.in.df)
```

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

```
# Compute the p-value for this drop in deviance test by comparing the deviance
# drop to a chi-squared distribution.
p.value <- pchisq(drop.in.dev, diff.in.df, lower.tail = FALSE)
print(p.value)
```

```
## [1] 0.6667354
```

With a p-value of 0.6667354, the drop in deviance test concludes that there is no evidence of a relationship between number of matings and the quadratic age variable.

(h) What can we say about the goodness of fit of the model with age as the sole predictor? Compare the residual deviance for the linear model to a χ^2 distribution with the residual model degrees of freedom.

```
gof <- pchisq(matings.fit$deviance, matings.fit$df.residual, lower.tail = FALSE)
print(gof)
```

```
## [1] 0.09426231
```

We can calculate the goodness of fit by testing the null hypothesis that our observed data came from the modelled Poisson distribution. The p-value of 0.0942623 is above the standard 5% cut-off, and so we fail to reject the null hypothesis for the linear model.

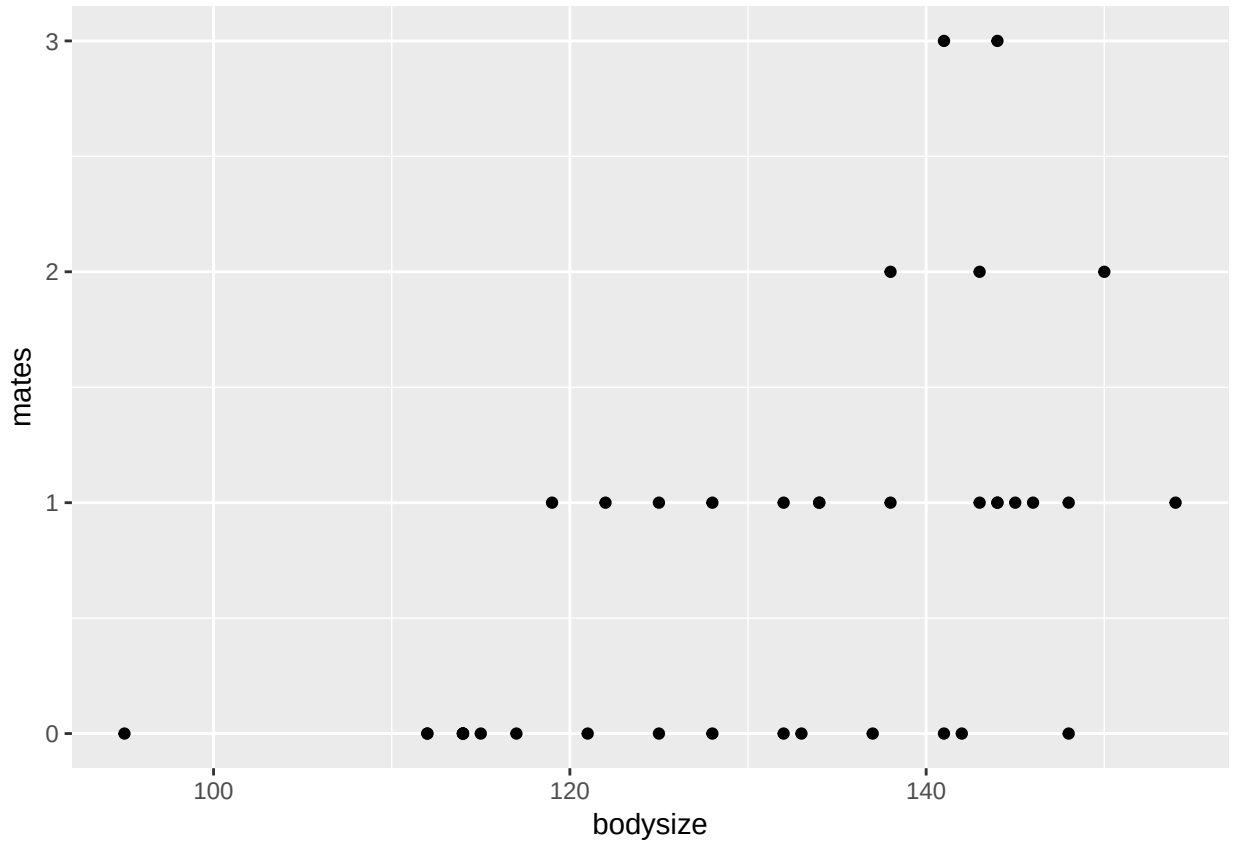
BYSH 4.15.2 Guided Exercise 3 (Bullfrogs)

***Big Bullfrog on the Pond*: Use the field observation bullfrog data bullfrogs.csv to determine whether there is convincing evidence that the number of mates is related to the size of the bullfrog.**

```
bullfrogs <- read.csv(paste("https://github.com/broadenyourstatisticalhorizons/bysh_book/",
    "raw/master/data/bullfrogs.csv", sep = ""))
```

(a) Graph number of mates by size and comment.

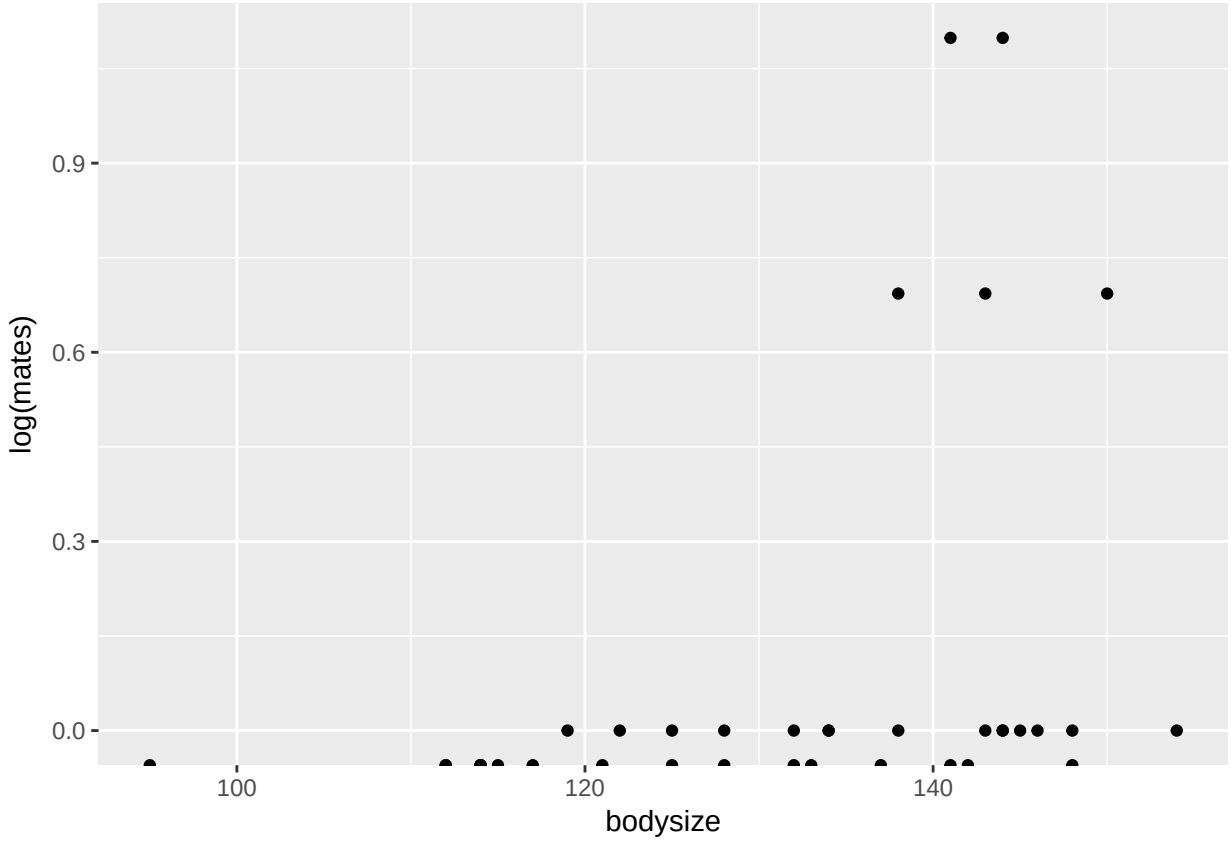
```
bullfrogs %>% ggplot(aes(x = bodysize, y = mates)) + geom_point()
```



Seems to be a skewed distribution. The mean number of mates does seem to increase as size increases. Poisson regression may be appropriate.

(b) Graph log(number of mates) by size and comment.

```
bullfrogs %>% ggplot(aes(x = bodysize, y = log(mates))) + geom_point()
```



How do we interpret the 0 counts which became negative infinity when the log transformation was applied? Anyway, this hardly looks like a linear relationship.

(c) Write out the likelihood for the Poisson regression model.

There are 18, 15, 3, and 2 observations of 0, 1, 2, and 3 matings, respectively. Therefore the likelihood is:

$$\begin{aligned} &Likelihood = \\ &P(mates = 3|size = 144) * ... * P(mates = 0|size = 112) = \\ &\prod_{i=1}^{N} \frac{e^{-e^{\beta_0+\beta_1*size_i}} * (e^{\beta_0+\beta_1*size_i})^{mates_i}}{mates_i!} \end{aligned}$$

(d) Fit the Poisson regression model and interpret the coefficient. Provide a measure of uncertainty.

```
bf.fit <- glm(mates ~ bodysize, family = poisson, data = bullfrogs)
summary(bf.fit)
```

```
##
## Call:
## glm(formula = mates ~ bodysize, family = poisson, data = bullfrogs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6866  -0.7578  -0.3722   0.4178   1.6695
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.11840    2.59380  -3.130  0.00175 **
## bodysize     0.05723    0.01851   3.092  0.00199 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 39.956  on 37  degrees of freedom
## Residual deviance: 28.003  on 36  degrees of freedom
## AIC: 75.828
##
## Number of Fisher Scoring iterations: 5
```

```
exp(bf.fit$coefficients)
```

```
##  (Intercept)     bodysize
## 0.0002980062 1.0589040321
```

```
exp(confint(bf.fit, "bodysize"))
```

```
## Waiting for profiling to be done...
```

```
##    2.5 %   97.5 %
## 1.023604 1.101159
```

(e) Conduct a goodness-of-fit test.

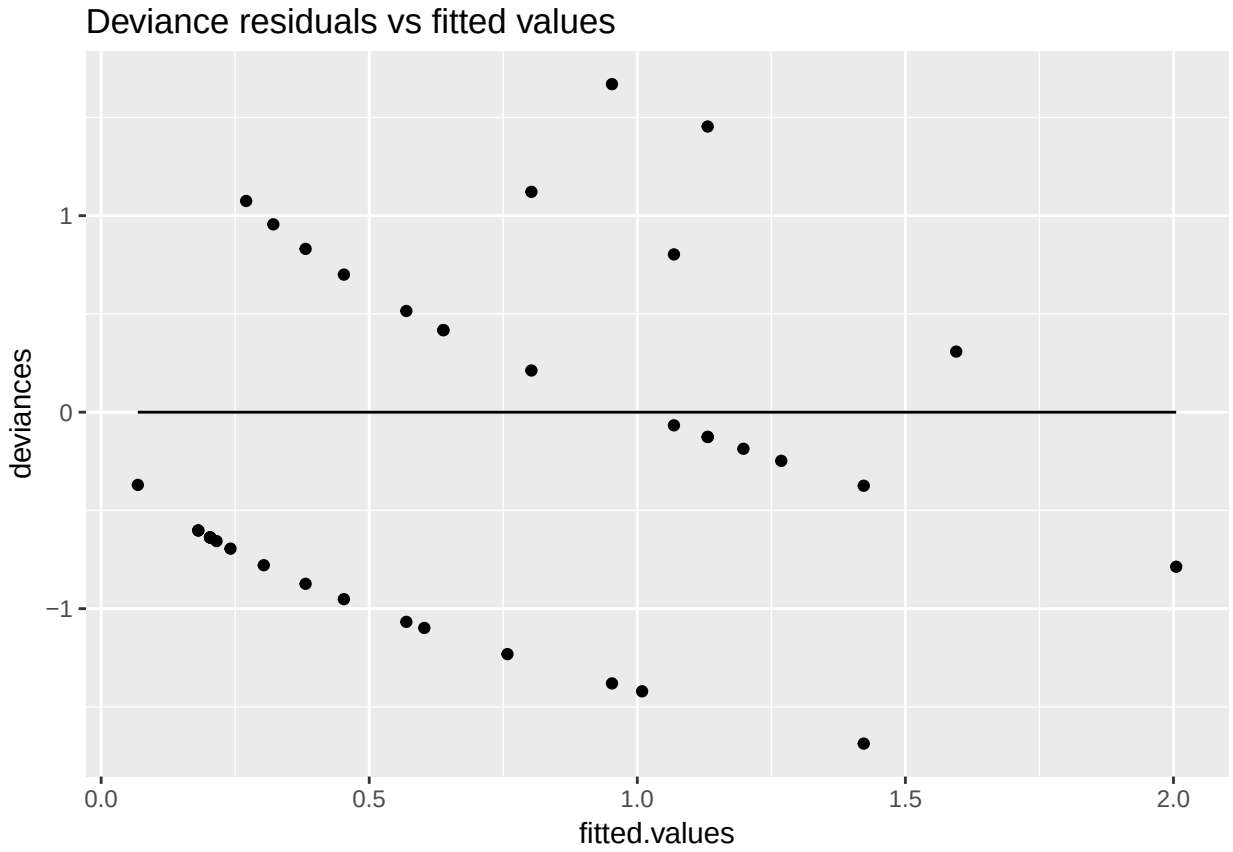
```
pchisq(bf.fit$deviance, bf.fit$df.residual, lower.tail = FALSE)
```

```
## [1] 0.827079
```

There's no evidence to reject the null hypothesis that our data is generated from this Poisson process.

(f) Look at the residuals and comment.

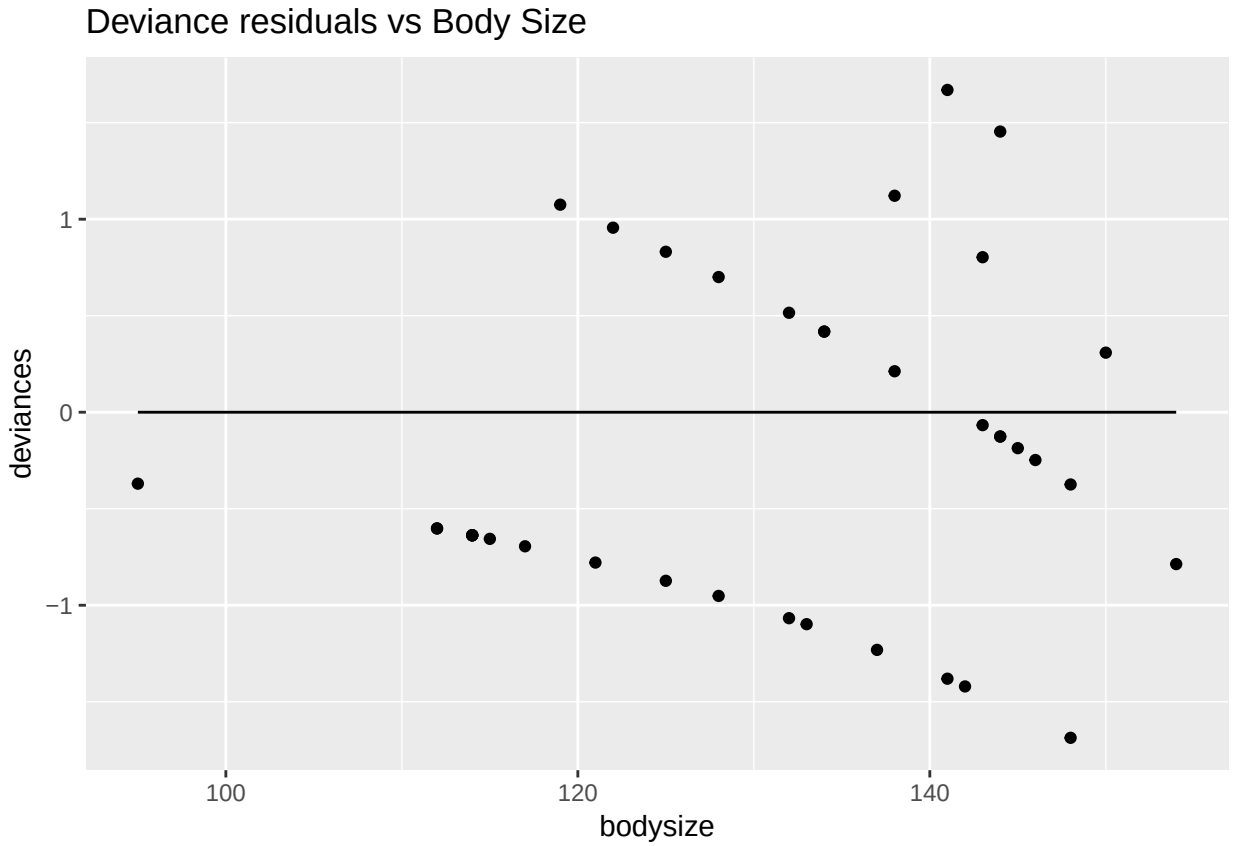
```
list(deviances = summary(bf.fit)$deviance.resid, fitted.values = bf.fit$fitted.values) %>%
    as_tibble() %>% ggplot(aes(x = fitted.values, y = deviances)) + geom_point() +
    geom_line(y = 0) + ggtitle("Deviance residuals vs fitted values")
```



Looking at the deviance residuals, we see four clear decreasing curves, which correspond with the observed counts. This is pretty cool, and occurs since the counts are distinct. As the model's predicted mean approaches the observed counts, the deviances decrease, becoming negative as the predicted mean passes over their count level. We can see that the predicted mean never reaches 2 or 3, since the top two curves never cross zero.

Since this is a model with only one predictor, we can plot the deviances vs the predictor and observe the same effect.

```
list(deviances = summary(bf.fit)$deviance.resid, bodysize = bullfrogs$bodysize) %>%
    as_tibble() %>% ggplot(aes(x = bodysize, y = deviances)) + geom_point() + geom_line(y = 0) +
    ggtitle("Deviance residuals vs Body Size")
```



It turns out this always happens in residual plots for Poisson regressions, with k curves for k different count levels. Neat stuff!