# Modeling Assignment 9: Poisson and ZIP Regression Models

Reed Ballesteros

8/21/2022

MSDS-410-DL, Summer 2022

Dr. Mickelson

## Assignment Overview

In this assignment we will be fitting models and calculating the various summative statistics that are associated with Poisson and Zero-Inflated Poisson Regression. The data set for this assignment, STRESS, includes information from about 650 adolescents in the United States who were surveyed about the number of stressful life events they had experienced in the past year (STRESS). STRESS is also an integer variable that represents counts of stressful events. The dataset also includes school and family related variables, which are assumed to be continuously distributed. The variables in this data set are:

- COHES = measure of how well the adolescent gets along with their family (coded low to high)
- ESTEEM = measure of self-esteem (coded low to high)
- GRADES = past year's school grades (coded low to high)
- SATTACH = measure of how well the adolescent likes and is attached to their school (coded low to high)

There is no other information about this data or the variables.

```
STRESS <- read_excel("STRESS.xlsx")
mydata<-data.frame(STRESS)
```

## Assignment Tasks

### Part 1

**For the STRESS variable, make a histogram and obtain summary statistics.**

```
summary(mydata$STRESS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    1.00    1.73    3.00    9.00
```

```
describe(mydata$STRESS)
```

```
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 651 1.73 1.85      1    1.44 1.48   0   9     9 1.27     1.61 0.07
```

We will create the NUMSTRESS variable in order to get the standard deviation of non-zero STRESS values from the dataset:

```
mydata$HASSTRESS <- ifelse(mydata$STRESS>0, 1, 0)
mydata$NUMSTRESS <- ifelse(mydata$HASSTRESS==1,mydata$STRESS,NA)
```

```
summary(mydata$NUMSTRESS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   1.000   2.000   2.619   3.000   9.000     221
```

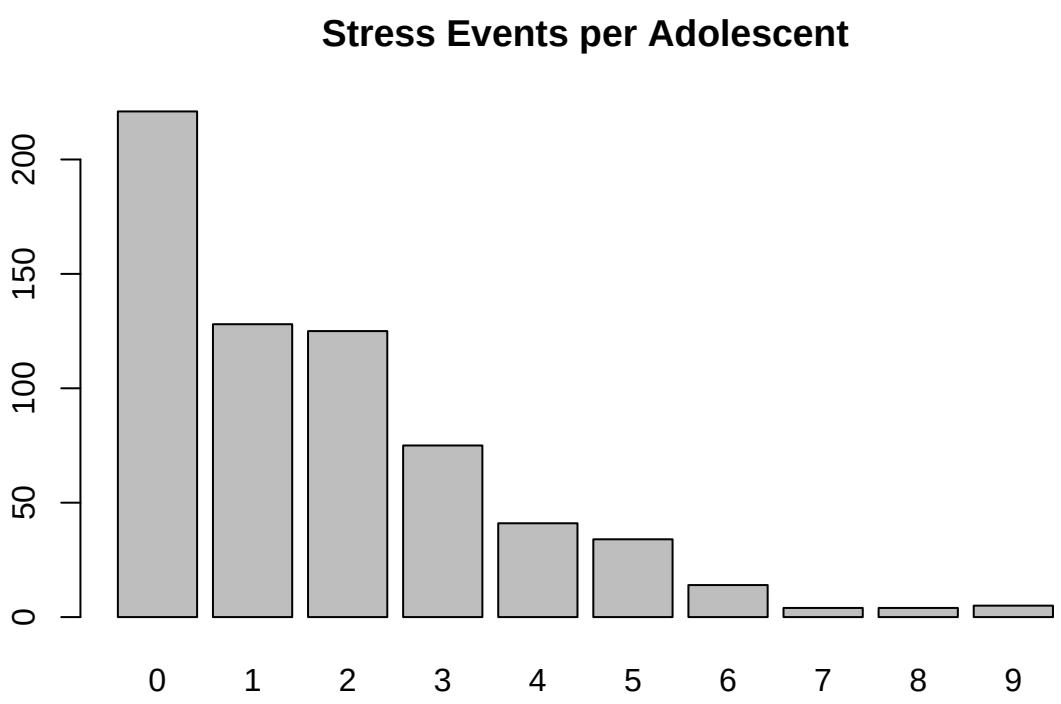
```
describe(mydata$NUMSTRESS)
```

```
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 430 2.62 1.69      2    2.37 1.48   1   9     8 1.36      1.9 0.08
```

Based on the NUMBSTRESS variable, we have a mean of 2.62 and a standard deviation of 1.69.

The histogram of STRESS is as follows:

```
# hist() doesn't work well with discrete variables (creates its own intervals),
# try bar plots instead
stressTbl <- table(mydata$STRESS)
barplot(stressTbl, main="Stress Events per Adolescent")
```



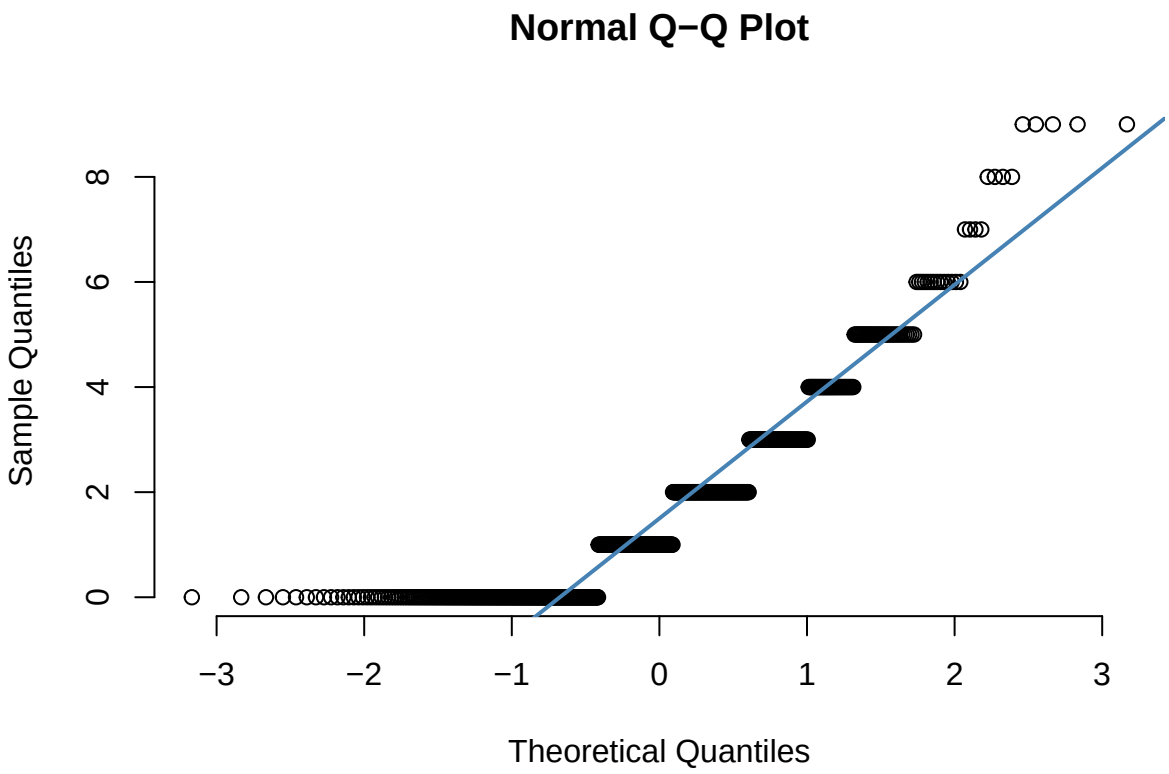

```
stressTbl
```

```
##
##   0   1   2   3   4   5   6   7   8   9
## 221 128 125  75  41  34  14   4   4   5
```

From the histogram we see a large number of 0 STRESS events per adolescent, larger than any other count.

**Obtain a normal probability (Q-Q) plot for the STRESS variable.**

```
qqnorm(mydata$STRESS, pch = 1, frame = FALSE)
qqline(mydata$STRESS, col = "steelblue", lwd = 2)
```




**Is STRESS a normally distributed variable? What do you think is its most likely probability distribution for STRESS? Give a justification for the distribution you selected.**

STRESS is not a normally distributed variable, but its right skewness with the mean towards the lower end shows signs of a Poisson Distribution.

The variance of NUMSTRESS which filters out 0-values is:

$$var(NUMSTRESS) = sd(NUMSTRESS)^2 = 1.69^2 = 2.8561$$

With the mean=2.62, the variance 2.86 is fairly close to the mean such that the dataset of STRESS can be reasonably considered to be either a Poisson Distribution, or even a Negative Binomial Distribution.

**Part 2**

**Fit an OLS regression model to predict STRESS (Y) using COHES, ESTEEM, GRADES, SATTACH as explanatory variables (X).**

```
model1 <- lm(STRESS ~ COHES + ESTEEM + GRADES + SATTACH, data=mydata)
```

**Obtain the typical diagnostic information and graphs.**

```
summary(model1)
```

```
##
```

```
## Call:
## lm(formula = STRESS ~ COHES + ESTEEM + GRADES + SATTACH, data = mydata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1447 -1.3827 -0.3819  0.9504  6.9525
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  5.71281    0.58118   9.830 < 0.0000000000000002 ***
## COHES       -0.02319    0.00703  -3.298              0.00103 **
## ESTEEM      -0.04129    0.01933  -2.136              0.03305 *
## GRADES      -0.04170    0.02352  -1.773              0.07670 .
## SATTACH     -0.03042    0.01412  -2.154              0.03160 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.776 on 646 degrees of freedom
## Multiple R-squared:  0.08319,    Adjusted R-squared:  0.07751
## F-statistic: 14.65 on 4 and 646 DF,  p-value: 0.00000000001826
```

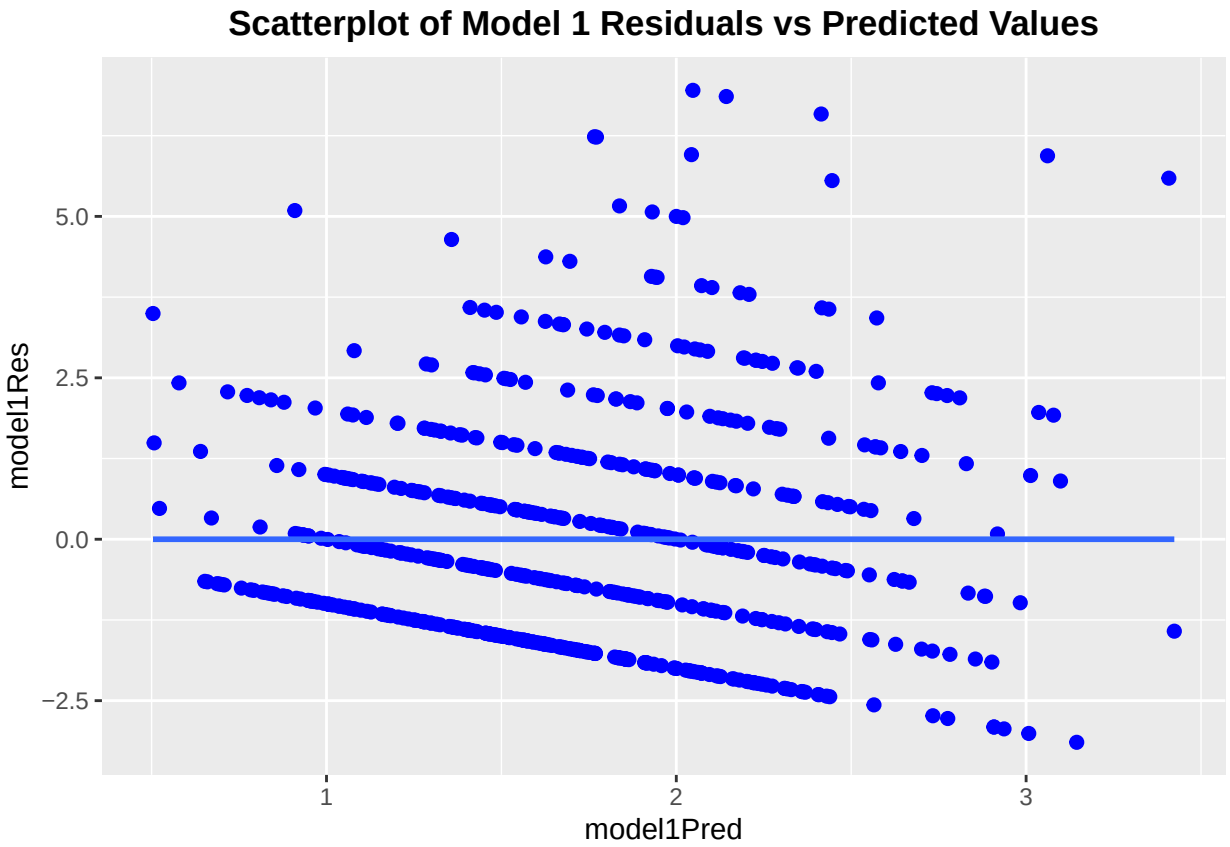
```
anova(model1)
```

```
## Analysis of Variance Table
##
## Response: STRESS
##            Df  Sum Sq Mean Sq F value           Pr(>F)
## COHES       1  122.93 122.930 38.9749 0.0000000007777 ***
## ESTEEM      1   31.26  31.264  9.9122         0.001718 **
## GRADES      1   16.05  16.052  5.0894         0.024407 *
## SATTACH     1   14.64  14.635  4.6401         0.031602 *
## Residuals 646 2037.54   3.154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As a linear regression model, its residuals vs predictor plot below doesn't show a random output, indicating a not-so-great fit of this model over the data.

```
model1Pred <- predict(model1,newdata=mydata)
mydata$model1Pred <- model1Pred
model1Res <- resid(model1)
ggplot(mydata, aes(x=model1Pred, y=model1Res)) +
  geom_point(color="blue", size=2) +
  ggtitle("Scatterplot of Model 1 Residuals vs Predicted Values") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5)) +
  geom_smooth(method=lm,se=FALSE)
```

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




**Discuss how well this model fits.**

The linear equation from the linear regression model is:

$$STRESS = 5.71281 - 0.02319 * COHES - 0.04129 * ESTEEM - 0.04170 * GRADES - 0.03042 * SATTACH$$

For every unit of COHES, STRESS goes down 0.02319 units; for every unit of ESTEEM, STRESS goes down 0.04129 units; for every unit of GRADES, STRESS goes down 0.04170 units; and for every unit of SATTACH, STRESS goes down 0.03042 units. The model shows that each of the explanatory variables can help reduce the number of STRESS events for an adolescent, a positive sign for relieving stress.

Though the intercept and each variable in the model has a fairly low p-value showing statistical significance, the R-Squared value for the model is also very low (0.08319), indicating that the model doesn't explain much of the variability around its mean.

**Obtain predicted values (Y_hat) and plot them in a histogram.**

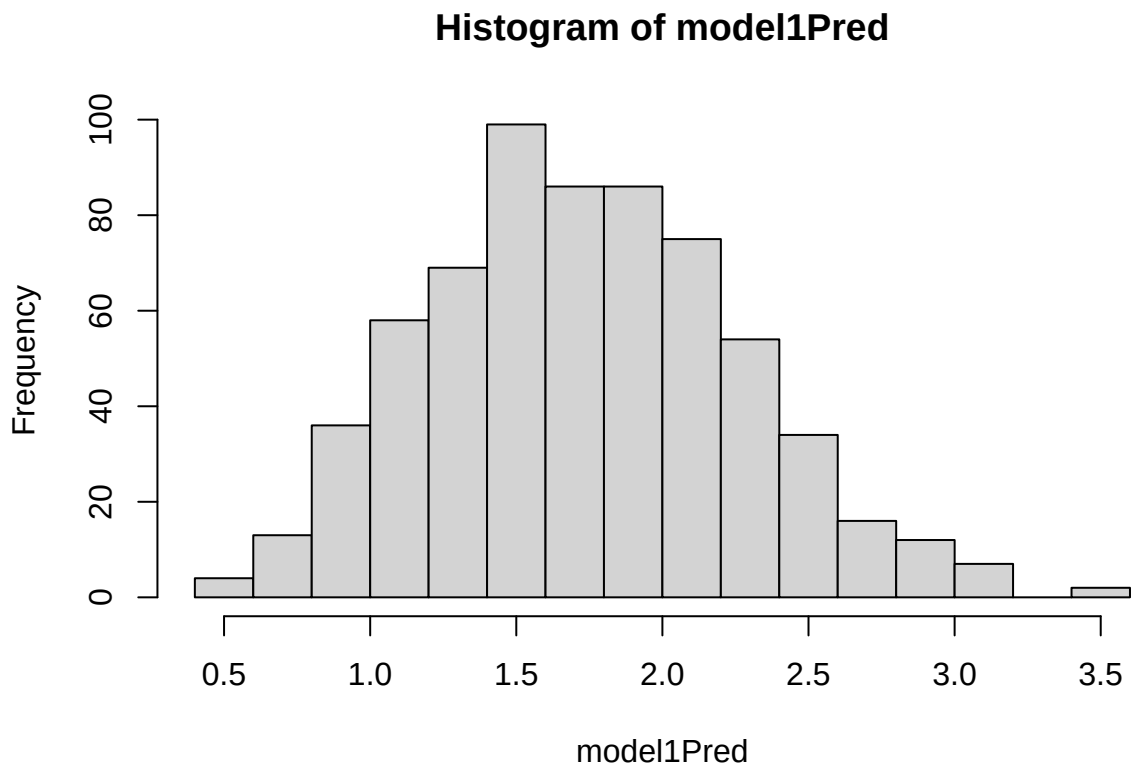
```
summary(model1Pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5041  1.3541  1.7055  1.7296  2.0971  3.4237
```

```
describe(model1Pred)
```

```
##    vars   n mean   sd median trimmed  mad min  max range skew kurtosis   se
## X1    1 651 1.73 0.53   1.71    1.71 0.56 0.5 3.42  2.92 0.26    -0.23 0.02
```

```
hist(model1Pred)
```




**What issues do you see?**

The histogram of predicted values shows more of a normal distribution compared to the histogram of STRESS, and doesn't account for the large number of 0 STRESS events per adolescent recorded in the observations. The variance of the predicted values (sd^2= 0.53^2 = 0.2809) does is not near the mean, further not indicating a Poisson Distribution.

And, while STRESS is a countable, discrete dependent variable, linear regression is used to predict a continuous dependent variable, in which STRESS is not. Because of this, fitting a linear regression model over the data with a discrete dependent variable is a not a great fit, as this exercise shows.

**Part 3**

**Create a transformed variable on Y that is LN(Y).**

```
# need to add 1 to compensate for 0 STRESS - log(0) doesn't work well
# will subrtract 1 from the predicted values after
STRESSCONST <- mydata$STRESS + 1
mydata$STRESSPLUSONELN <- log(STRESSCONST)
describe(mydata$STRESSPLUSONELN)
```

```
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 651 0.79 0.66   0.69    0.75 1.03   0 2.3   2.3 0.15    -1.17 0.03
```

**Fit an OLS regression model to predict LN(Y) using COHES, ESTEEM, GRADES, SATTACH as explanatory variables (X).**

```
model2 <- lm(STRESSPLUSONELN ~ COHES + ESTEEM + GRADES + SATTACH, data=mydata)
```

**Obtain the typical diagnostic information and graphs.**

```
summary(model2)
```

```
##
## Call:
## lm(formula = STRESSPLUSONELN ~ COHES + ESTEEM + GRADES + SATTACH,
##     data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22362 -0.63438  0.04982  0.51763  1.44040
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  2.073322   0.209097   9.916 < 0.0000000000000002 ***
## COHES       -0.007947   0.002529  -3.142              0.00175 **
## ESTEEM      -0.010915   0.006955  -1.569              0.11706
## GRADES      -0.014336   0.008462  -1.694              0.09072 .
## SATTACH     -0.011283   0.005081  -2.220              0.02674 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.639 on 646 degrees of freedom
## Multiple R-squared:  0.07154,    Adjusted R-squared:  0.06579
## F-statistic: 12.44 on 4 and 646 DF,  p-value: 0.0000000009333
```

```
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: STRESSPLUSONELN
##            Df  Sum Sq Mean Sq F value        Pr(>F)
## COHES       1  13.679 13.6795 33.5059 0.00000001109 ***
## ESTEEM      1   2.672  2.6725  6.5458       0.01074 *
## GRADES      1   1.957  1.9565  4.7923       0.02894 *
## SATTACH     1   2.013  2.0127  4.9299       0.02674 *
## Residuals 646 263.742  0.4083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(model2)
```

```
## [1] 1271.255
```

```
# subtract 1 to compensate for the +1 transformation for predictions and residuals
model2Pred <- (predict(model2,newdata=mydata)) - 1
mydata$model2Pred <- model2Pred
```

**Discuss how well this model fits.**

The linear equation from the linear regression model is:

$$ln(STRESS) - 1 = 2.073322 - 0.007947 * COHES - 0.010915 * ESTEEM - 0.014336 * GRADES - 0.011283 * SATTACH$$

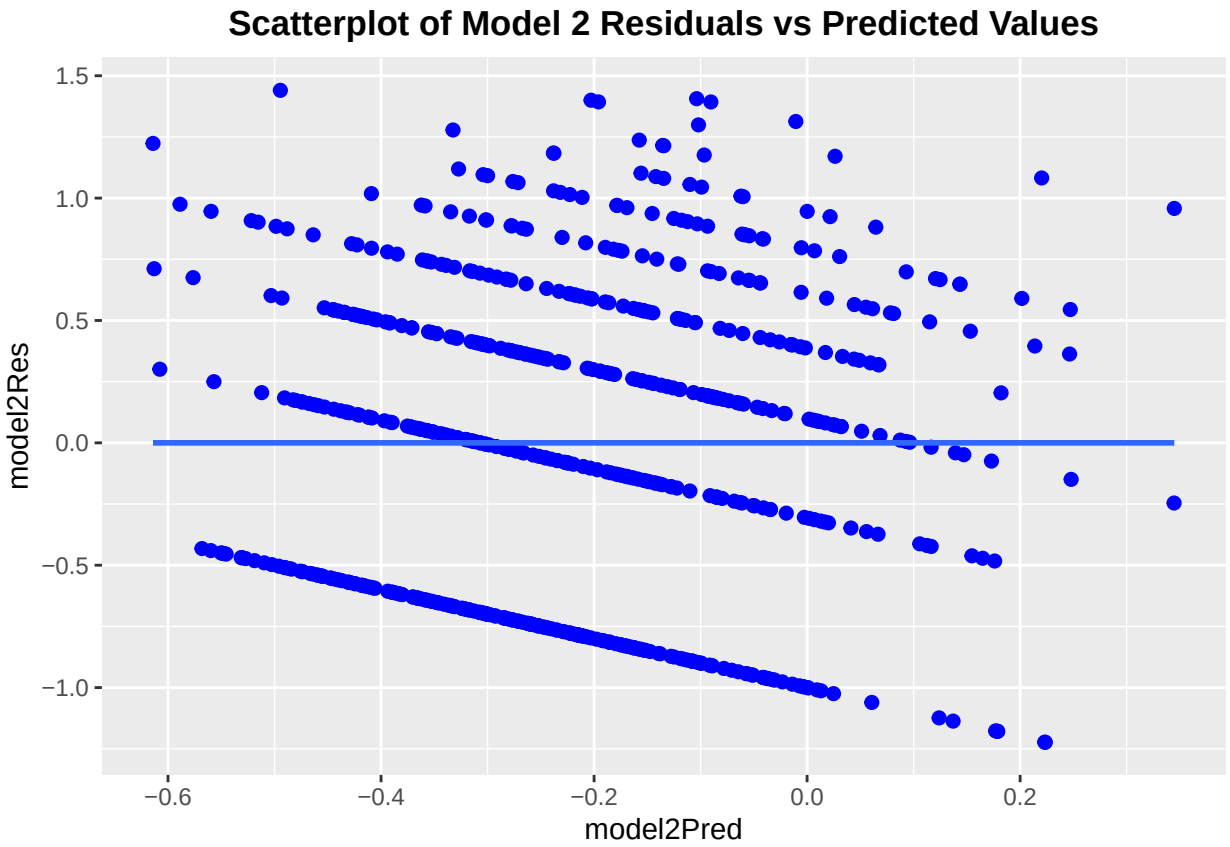
For every unit of COHES, ln(STRESS) goes down 0.007947 units; for every unit of ESTEEM, ln(STRESS) goes down 0.010915 units; for every unit of GRADES, ln(STRESS) goes down 0.014336 units; and for every unit of SATTACH, ln(STRESS) goes down 0.011283 units. The R-Squared value for the model is even lower (0.07154), indicating that the model doesn't explain much of the variability around its mean.

The ln(STRESS) predicted value needs to have 1 subtracted from its results to compensate for the added 1 constant to STRESS for the ln(STRESS) transformation.

As a linear regression model, its residuals vs predictor plot below also doesn't show a random output, indicating a not-so-great fit of this model over the data.

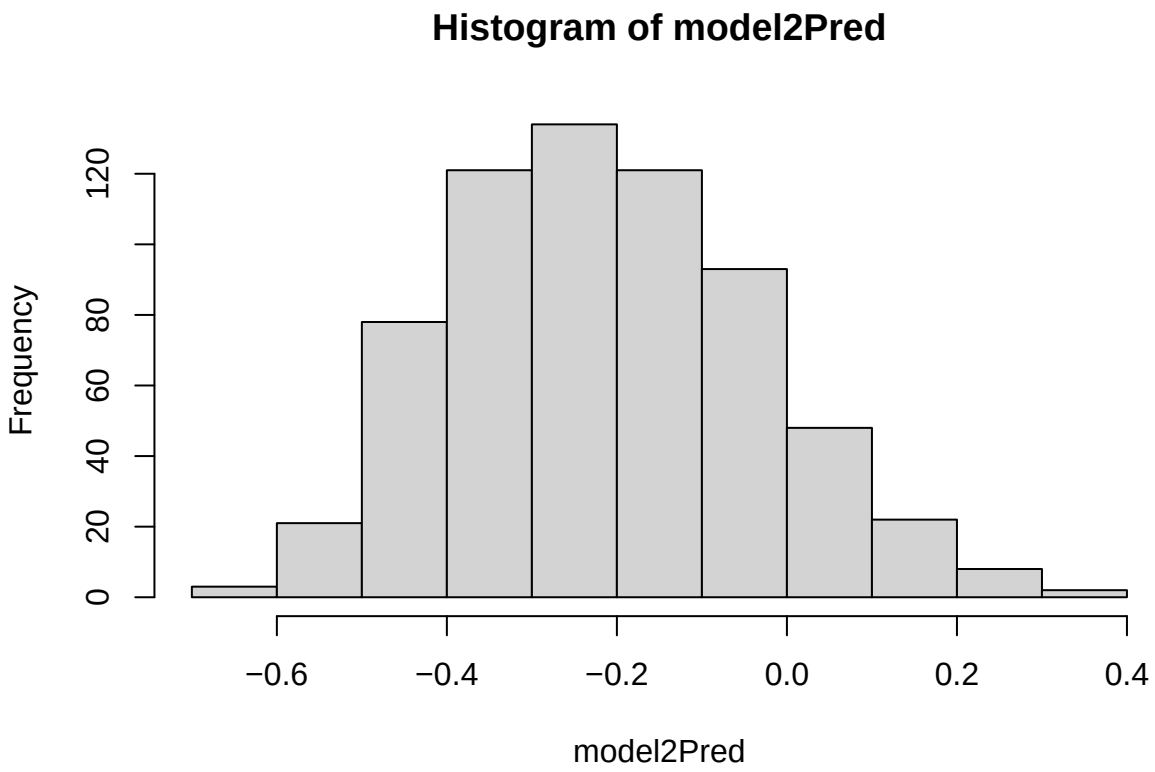
```
model2Res <- resid(model2)
ggplot(mydata, aes(x=model2Pred, y=model2Res)) +
  geom_point(color="blue", size=2) +
  ggtitle("Scatterplot of Model 2 Residuals vs Predicted Values") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5)) +
  geom_smooth(method=lm,se=FALSE)
```

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




**Obtain predicted values (LN(Y)_hat) and plot them in a histogram.**

```
hist(model2Pred)
```




**What issues do you see?**

```
describe(model2Pred)
```

```
##    vars   n  mean   sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 651 -0.21 0.18  -0.22   -0.22 0.19 -0.61 0.34  0.96 0.28    -0.25 0.01
```

We still see a normal distribution of the predicted values, but in this case we needed to compensate for the large number of 0 STRESS events in the dataset by adding a constant of 1 to all STRESS levels in order to make the LN(STRESS) computation, then subtract that same constant from the predicted values afterwards. Doing so results in having a negative mean (-0.21), with the majority of predicted values lying in the negative range, which is not what we're looking for. The model doesn't work well with finding countable discrete predictions.

**Does this correct the issue?**

No, as described above, fitting a linear regression model over the data based on the log of the dependent variable STRESS is a not a great fit.

**Part 4**

**Use the glm() function to fit a Poisson Regression for STRESS (Y) using COHES, ESTEEM, GRADES, SATTACH as explanatory variables (X).**

```
model3 <- glm(STRESS ~ COHES + ESTEEM + GRADES + SATTACH, family = "poisson", data = mydata)
```

**Interpret the model's coefficients and discuss how this model's results compare to your answer for part 3).**

```
summary(model3)
```

```
##
## Call:
## glm(formula = STRESS ~ COHES + ESTEEM + GRADES + SATTACH, family = "poisson",
##     data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7111  -1.5989  -0.2914   0.7107   3.6424
##
## Coefficients:
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)  2.734513   0.234066  11.683 < 0.0000000000000002 ***
## COHES       -0.012918   0.002893  -4.466          0.00000798 ***
## ESTEEM      -0.023692   0.008039  -2.947             0.00321 **
## GRADES      -0.023471   0.009865  -2.379             0.01735 *
## SATTACH     -0.016481   0.005783  -2.850             0.00437 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1349.8  on 650  degrees of freedom
## Residual deviance: 1245.4  on 646  degrees of freedom
## AIC: 2417.2
##
## Number of Fisher Scoring iterations: 5
```

```
anova(model3)
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: STRESS
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev
## NULL                      650     1349.8
## COHES    1   68.798       649     1281.0
## ESTEEM   1   18.002       648     1263.0
## GRADES   1    9.497       647     1253.5
## SATTACH  1    8.051       646     1245.4
```

$$LN(STRESS) = 2.734513 - 0.012918 * COHES - 0.023692 * ESTEEM - 0.023471 * GRADES - 0.016481 * SATTACH$$

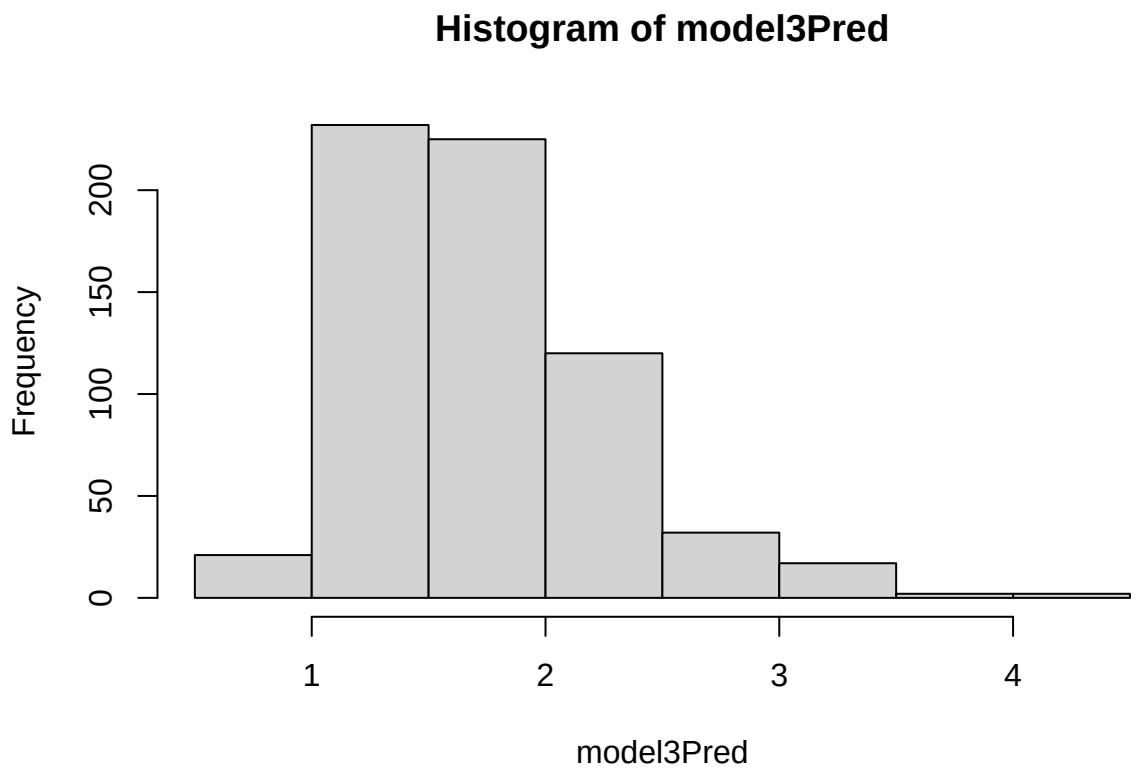
Based on the Poisson model, each unit of COHES decreases LN(STRESS) in an adolescent by 1.29%; each unit of ESTEEM decreases LN(STRESS) in an adolescent by 2.37%; each unit of GRADES decreases LN(STRESS) in an adolescent by 2.35%; and each unit of SATTACH decreases LN(STRESS) in an adolescent by 1.65%.

The equation above is based on LN(STRESS) and needs to be transformed with the following to get a prediction of STRESS:

$$PREDSTRESS = exp(LN(STRESS))$$

We display a histogram of the predicted STRESS values of the Poisson model after exp() transformation:

```
model3Pred <- fitted(model3)
mydata$model3Pred <- model3Pred
hist(model3Pred)
```




Compared to the linear regression model based on log(STRESS) created in part 3, this model more follows the Poisson distribution of STRESS in the dataset, but still does not account for the large number of 0 STRESS levels in the dataset.

**Similarly, fit an over-dispersed Poisson regression model using the same set of variables.**

```r
# Over-Dispersed Poisson Regression Model -> Negative Binomial Regression Model
model4 <- glm.nb(STRESS ~ COHES + ESTEEM + GRADES + SATTACH, data = mydata)
summary(model4)
```

```
##
## Call:
## glm.nb(formula = STRESS ~ COHES + ESTEEM + GRADES + SATTACH,
##     data = mydata, init.theta = 1.865329467, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0179  -1.3900  -0.2214   0.4882   2.3199
##
## Coefficients:
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)  2.759032   0.341531   8.078 0.000000000000000656 ***
## COHES       -0.013391   0.004136  -3.238              0.00121 **
## ESTEEM      -0.023058   0.011477  -2.009              0.04453 *
## GRADES      -0.024360   0.013969  -1.744              0.08118 .
## SATTACH     -0.016750   0.008296  -2.019              0.04349 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.8653) family taken to be 1)
##
##     Null deviance: 792.47  on 650  degrees of freedom
## Residual deviance: 738.53  on 646  degrees of freedom
## AIC: 2283.6
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.865
##           Std. Err.:  0.257
##
##  2 x log-likelihood:  -2271.590
```

```r
anova(model4)
```

```
## Analysis of Deviance Table
##
## Model: Negative Binomial(1.8653), link: log
##
## Response: STRESS
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev       Pr(>Chi)
## NULL                     650     792.47
## COHES    1   36.228       649     756.24 0.000000001756 ***
## ESTEEM   1    8.823       648     747.42       0.002975 **
## GRADES   1    4.710       647     742.71       0.029983 *
```

```
## SATTACH  1    4.171       646     738.53       0.041122 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$$LN(STRESS) = 2.759032 - 0.013391 * COHES - 0.023058 * ESTEEM - 0.024360 * GRADES - 0.016750 * SATTACH$$

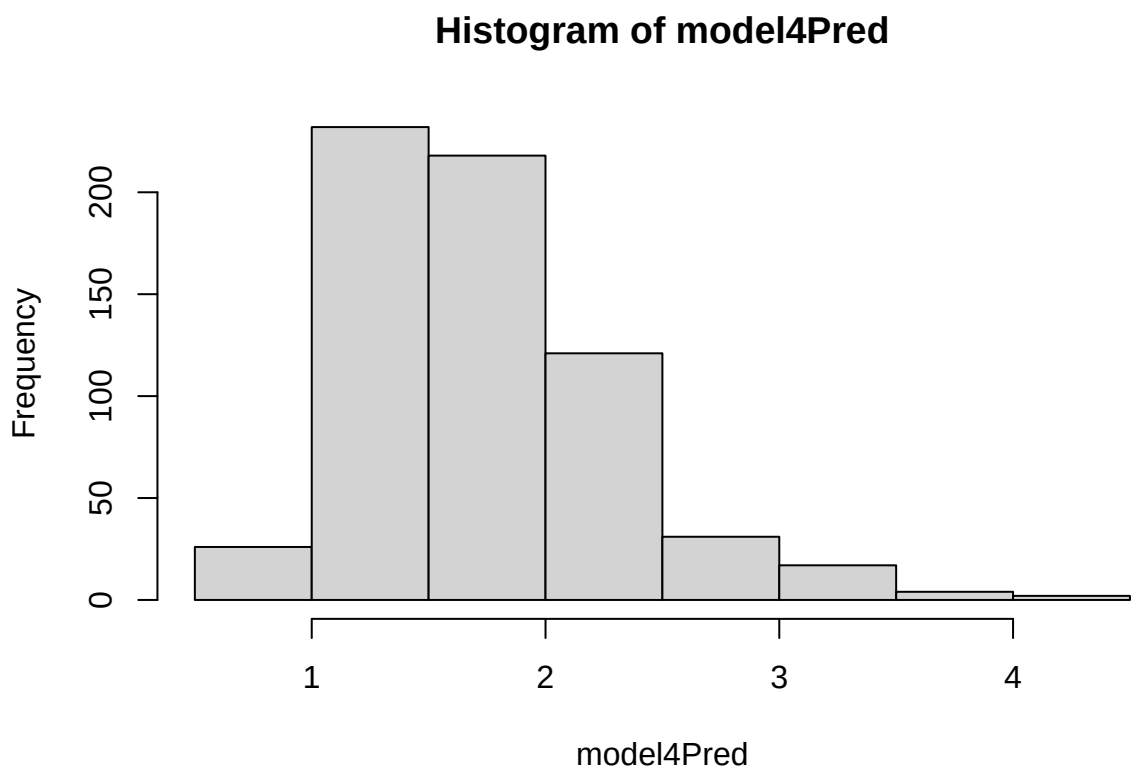
Based on the Negative Binomial model, each unit of COHES decreases LN(STRESS) in an adolescent by 1.34%; each unit of ESTEEM decreases LN(STRESS) in an adolescent by 2.31%; each unit of GRADES decreases LN(STRESS) in an adolescent by 2.44%; and each unit of SATTACH decreases LN(STRESS) in an adolescent by 1.68%.

The equation above is based on LN(STRESS) and need to be transformed with the following to get a prediction of STRESS:

$$PREDSTRESS = exp(LN(STRESS))$$

We display a histogram of the predicted STRESS values of the Negative Binomial model after exp() transformation:

```
model4Pred<-fitted(model4)
mydata$model4Pred <- model4Pred
hist(model4Pred)
```




**How do these models compare?**

```
AIC(model3)
```

```
## [1] 2417.219
```

```
BIC(model3)
```

```
## [1] 2439.612
```

```
AIC(model4)
```

```
## [1] 2283.59
```

```
BIC(model4)
```

```
## [1] 2310.461
```

The Poisson regression model and the Over-Dispersed Poisson/Negative Binomial regression model are very similar to each other. That being said AIC and BIC scores are lower for model 4, the Over-Dispersed Poisson/Negative Binomial model, suggesting that it is a better-fitting model. Also the residual deviance of model 4 at 738.53 is closer to its 646 degrees of freedom while model 3's (the Poisson regression model) residual deviance is 1245.4, which is almost twice that of its 646 degrees of freedom, also suggesting that model 4 is the better-fitting model.

**Part 5**

**Based on the Poisson model in part 4), compute the predicted count of STRESS for those whose levels of family cohesion are less than one standard deviation below the mean (call this the low group), between one standard deviation below and one standard deviation above the mean (call this the middle group), and more than one standard deviation above the mean (high).**

We will use a simplified Poisson model using only the COHES explanatory variable:

```
model3Alt <- glm(STRESS ~ COHES, family = "poisson", data = mydata)
```

We will create our low/mid/high COHES values based on the mean and standard deviation:

```
describe(mydata$COHES)
```

```
##    vars   n mean    sd median trimmed   mad min max range  skew kurtosis   se
## X1    1 651   53 11.38     54   53.58 11.86  18  75    57 -0.43    -0.29 0.45
```

```
# mean: 53, sd: 11.38
COHESLowerBound <- 53 - 11.38
COHESLowerBound # 41.62
```

```
## [1] 41.62
```

```
COHESUpperBound <- 53 + 11.38
COHESUpperBound # 64.38
```

```
## [1] 64.38
```

```
COHESgroup <- data.frame('COHES'=c(COHESLowerBound,mean(mydata$COHES), COHESUpperBound),
                         COHESLEVEL=c("low","mid","high"))
COHESgroup
```

```
##      COHES COHESLEVEL
## 1 41.62000        low
## 2 53.00426        mid
## 3 64.38000       high
```

Given our low/med/nigh COHES values, we can use the simplified Poisson model to predict the number of STRESS events:

```
# model3 predicted values are in LN(STRESS), need to transform results with exp()
model3PredCOHES <- exp(predict(model3Alt, newdata=COHESgroup))
COHESgroup$model3PredCOHES <- model3PredCOHES
COHESgroup
```

```
##      COHES COHESLEVEL model3PredCOHES
## 1 41.62000        low        2.134273
## 2 53.00426        mid        1.679093
## 3 64.38000       high        1.321227
```

**What is the expected percent difference in the number of stressful events for those at high and low levels of family cohesion?**

The formula for expected percent difference is:

$$PercentDifference = (\frac{COHESHIGHPRED}{COHESLOWPRED} - 1) * 100$$

The expected percent difference between those with high and low levels of family cohesion is:

```
pct_diff_alt2 <- ((model3PredCOHES[1]/model3PredCOHES[3]) - 1) * 100
round(pct_diff_alt2,2)
```

```
##     1
## 61.54
```

The expected number of adolescents with lover cohesion to have more STRESS events is 61.54% higher than those with higher levels of family cohesion. The model supports that more family cohesion in an adolescent can result in a lower number of STRESS events compared to those with less family cohesion. Rounding out the number of STRESS events to the nearest integer in their respective groups, those with low family cohesion are more likely to have two STRESS events, while those with high family cohesion are more likely to have 1 STRESS event.

**Part 6**

**Compute the AICs and BICs from the Poisson Regression and the over-dispersed Poisson regression models from part 4).**

```
AIC(model3)
```

```
## [1] 2417.219
```

```
BIC(model3)
```

```
## [1] 2439.612
```

```
AIC(model4)
```

```
## [1] 2283.59
```

```
BIC(model4)
```

```
## [1] 2310.461
```

**Is one better than the other?**

Based on the AIC and BIC scores, model4 (the over-dispersed Piosson/negative binomial regression model) is favored more than model3 (the Poisson regression model), indicated by lower AIC and BIC scores.

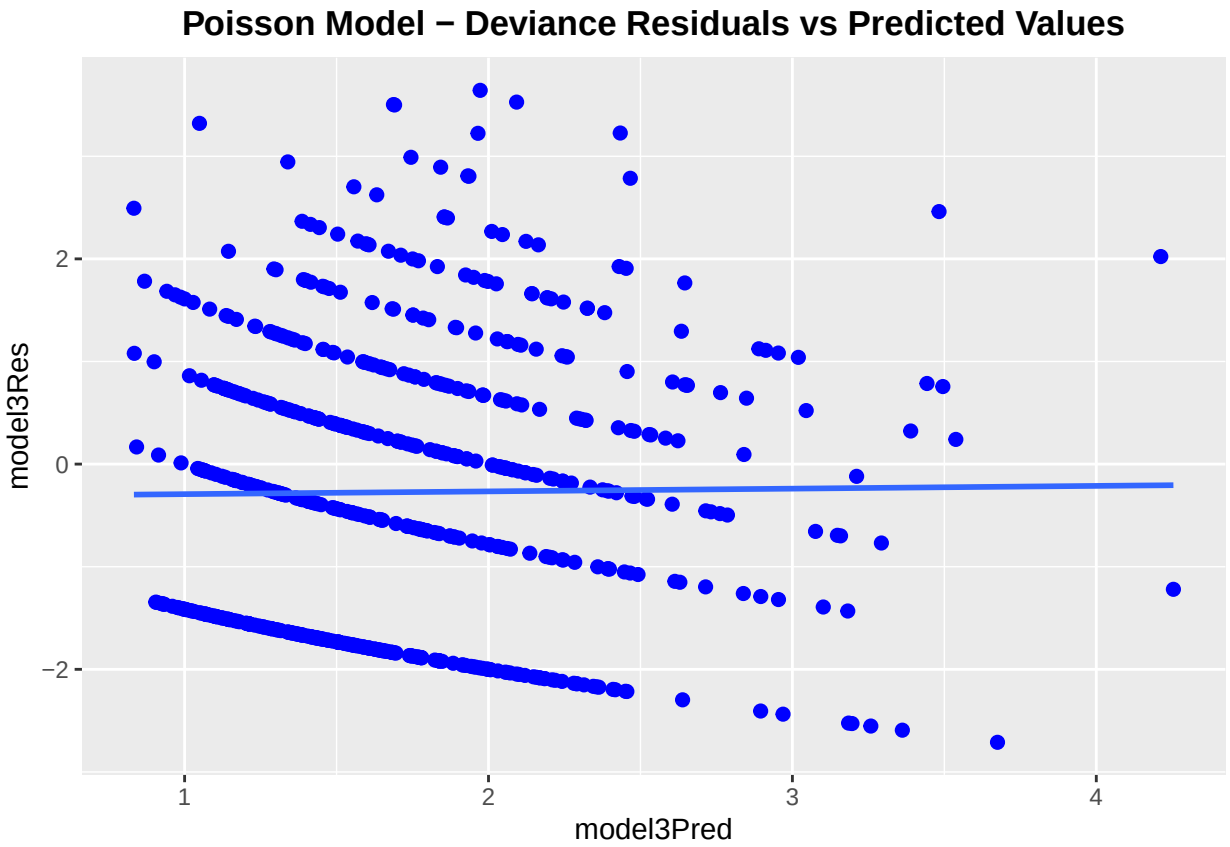
**Part 7**

**Using the Poisson regression model from part 4), plot the deviance residuals by the predicted values.**

Deviance Residuals vs Predicted Values: Poisson model

```
model3Res <- resid(model3, type="deviance")
ggplot(mydata, aes(x=model3Pred, y=model3Res)) +
  geom_point(color="blue", size=2) +
  ggtitle("Poisson Model - Deviance Residuals vs Predicted Values") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5)) +
  geom_smooth(method=lm,se=FALSE)
```

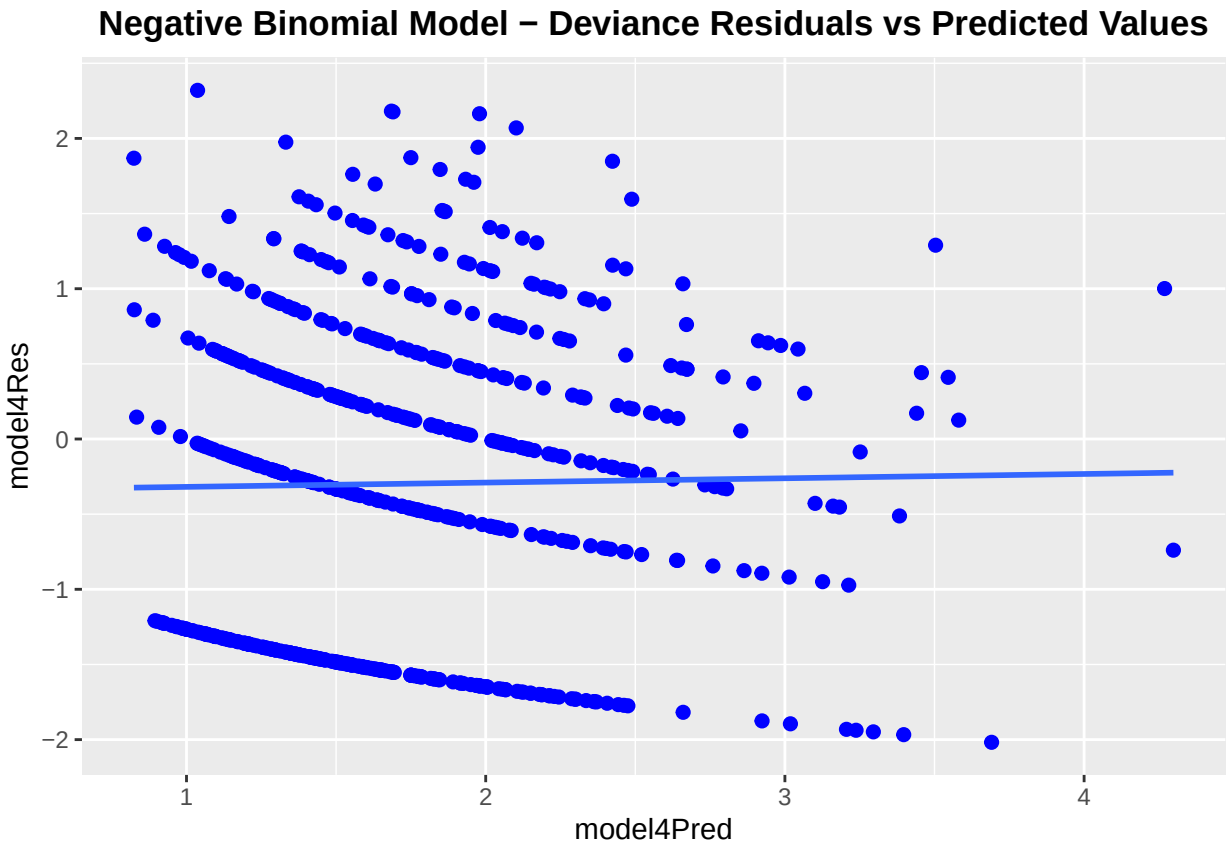
```
## `geom_smooth()` using formula 'y ~ x'
```




Deviance Residuals vs Predicted Values: Negative Binomial model

```
model4Res <- resid(model4, type="deviance")
ggplot(mydata, aes(x=model4Pred, y=model4Res)) +
  geom_point(color="blue", size=2) +
  ggtitle("Negative Binomial Model - Deviance Residuals vs Predicted Values") +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5)) +
  geom_smooth(method=lm,se=FALSE)
```

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




**Discuss what this plot indicates about the regression model.**

Comparing the Poisson model to the Negative Binomial model, we see the a smaller, tighter range of residuals between -2 to 2 in the Negative Binomial model compared to Poisson model, indicating a better goodness-of-fit of the Negative Binomial model over the Poisson model.

**Part 8**

**Create a new indicator variable (HASSTRESS renamed from Y_IND) of STRESS that takes on a value of 0 if STRESS=0 and 1 if STRESS>0. This variable essentially measures is stress present, yes or no.**

```
# instead of Y_IND, already made similar HASSTRESS variable
# mydata$NUMSTRESS <- ifelse(mydata$HASSTRESS==1,mydata$STRESS,NA)
table(mydata$HASSTRESS)
```

```
##
##   0   1
## 221 430
```

**Fit a logistic regression model to predict HASSTRESS using the variables using COHES, ESTEEM, GRADES, SATTACH as explanatory variables (X).**

```
model5 <- glm(HASSTRESS ~ COHES + ESTEEM + GRADES + SATTACH, family = "binomial", data = mydata)
summary(model5)
```

```
##
## Call:
## glm(formula = HASSTRESS ~ COHES + ESTEEM + GRADES + SATTACH,
##     family = "binomial", data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9069  -1.3283   0.7829   0.9366   1.2693
##
## Coefficients:
##              Estimate Std. Error z value   Pr(>|z|)
## (Intercept)  3.516735   0.737131   4.771 0.00000183 ***
## COHES       -0.020733   0.008751  -2.369     0.0178 *
## ESTEEM      -0.018867   0.023741  -0.795     0.4268
## GRADES      -0.025492   0.028701  -0.888     0.3744
## SATTACH     -0.027730   0.017525  -1.582     0.1136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 834.18  on 650  degrees of freedom
## Residual deviance: 811.79  on 646  degrees of freedom
## AIC: 821.79
##
## Number of Fisher Scoring iterations: 4
```

**Report the model, interpret the coefficients, obtain statistical information on goodness of fit, and discuss how well this model fits.**

$HASSTRESS = 3.516735 - 0.020733 * COHES - 0.018867 * ESTEEM - 0.025492 * GRADES - 0.027730 * SATTACH$

```
model5COHESPct <- exp(-0.020733) - 1
model5COHESPct
```

```
## [1] -0.02051955
```

```
model5ESTEEMPct <- exp(-0.018867) - 1
model5ESTEEMPct
```

```
## [1] -0.01869013
```

```
model5GRADESPct <- exp(-0.025492) - 1
model5GRADESPct
```

```
## [1] -0.02516982
```

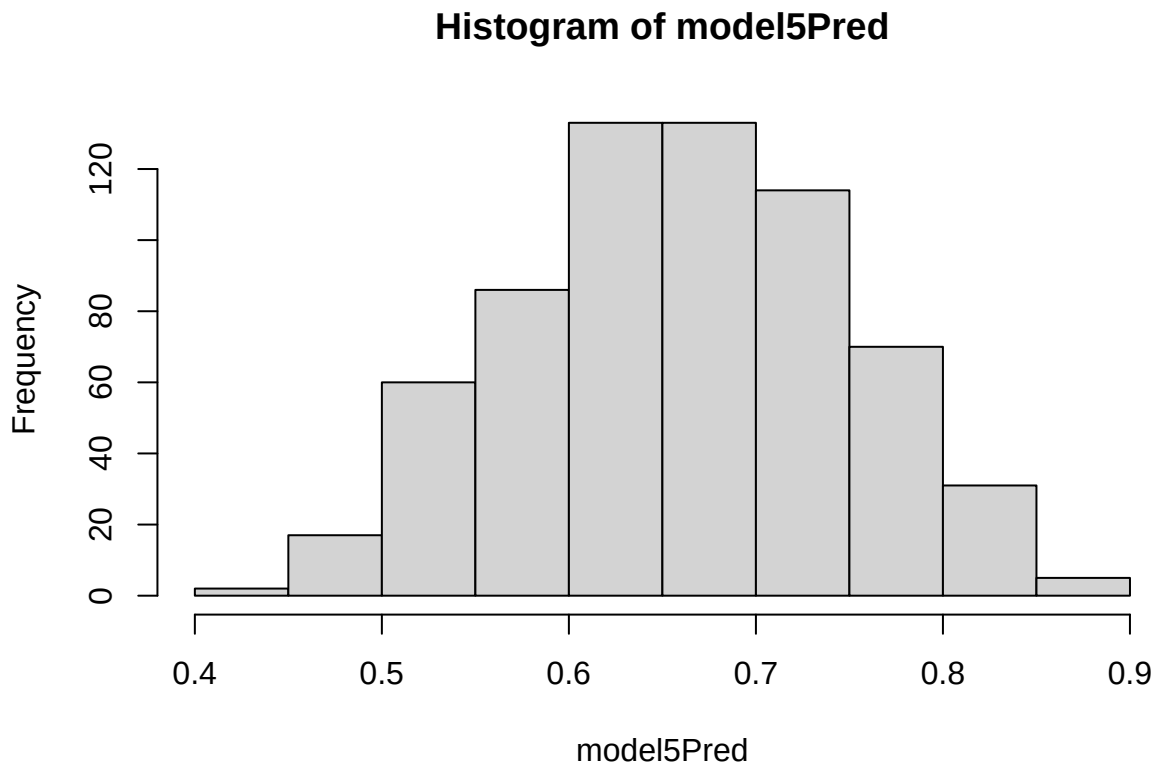
```
model5SATTACHPct <- exp(-0.027730) - 1
model5SATTACHPct
```

```
## [1] -0.02734905
```

Converting each of the coefficients of this model into percentages, each unit of COHES increases the odds of having no STRESS (Y_IND = 0 or HASSSTRESS = 0) in an adolescent by 2.05%; each unit of ESTEEM increases the odds by 1.87%; each unit of GRADES increases by 2.52%; and each unit of SATTACH increases the odds by 2.73%.

Histogram of the predicted values is as follows:

```
model5Pred <- fitted(model5)
mydata$model5Pred <- model5Pred
hist(model5Pred)
```




**Should you rerun the logistic regression analysis? If so, what should you do next?**

Given that ESTEEM, GRADES, and SATTACH have high p-values, they are not statistically significant to the model against the Y_IND/HASSTRESS dependent variable. We could simplify the model by removing them and using the only statistically significant variable COHES:

```
model6 <- glm(HASSTRESS ~ COHES, family = "binomial", data = mydata)
summary(model6)
```

```
##
## Call:
## glm(formula = HASSTRESS ~ COHES, family = "binomial", data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.9543  -1.3432   0.8055   0.9375   1.1703
##
## Coefficients:
##              Estimate Std. Error z value    Pr(>|z|)
## (Intercept)  2.296371   0.427310   5.374 0.000000077 ***
## COHES       -0.030393   0.007715  -3.939 0.000081681 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 834.18  on 650  degrees of freedom
## Residual deviance: 817.86  on 649  degrees of freedom
## AIC: 821.86
##
## Number of Fisher Scoring iterations: 4
```

$$HASSTRESS = 3.516735 - 0.030393 * COHES$$

```
model6COHESPct <- exp(-0.030393) - 1
model6COHESPct
```

```
## [1] -0.02993578
```

Converting the COHES coefficient of this model into a percentage, each unit of COHES increases the odds of having no STRESS (Y_IND = 0 or HASSTRESS = 0) in an adolescent by 2.99%, an almost 1% improvement from the full model.

```
AIC(model5)
```

```
## [1] 821.7858
```

```
BIC(model5)
```

```
## [1] 844.1784
```

```
AIC(model6)
```

```
## [1] 821.8574
```

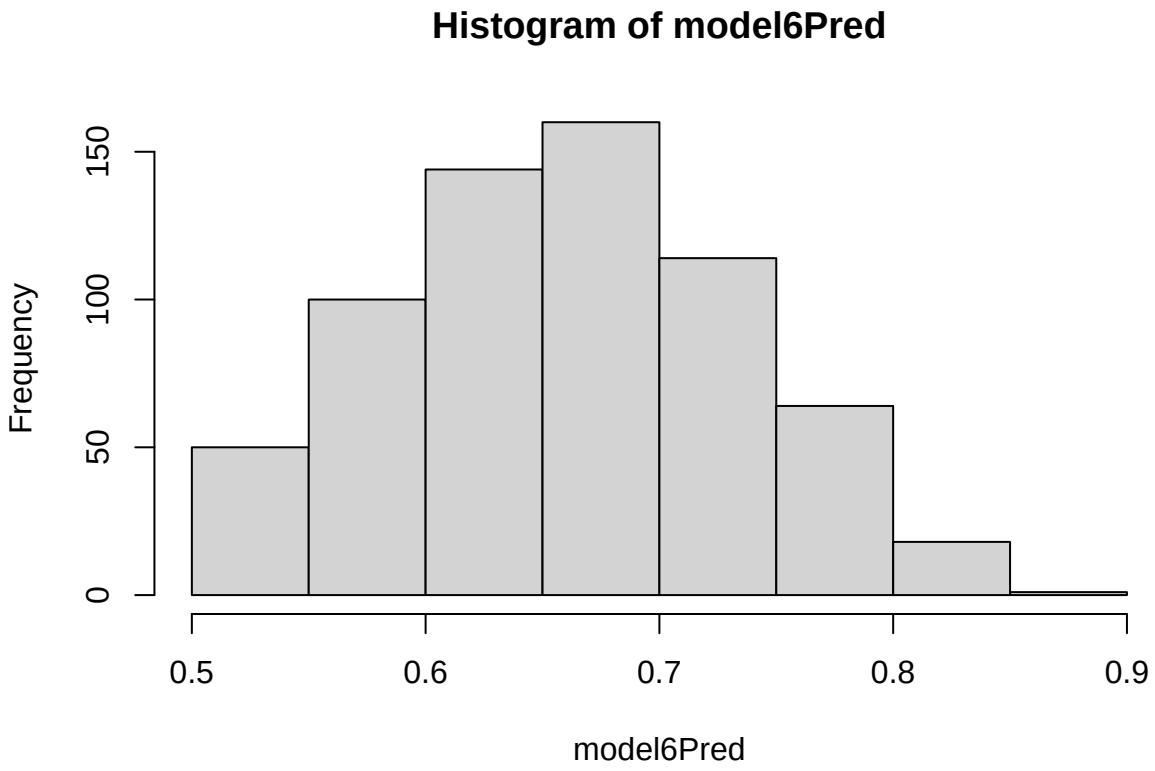
```
BIC(model6)
```

```
## [1] 830.8144
```

AIC, by nature, tends to favor the more complex logistic regression model5 with a lower score over the more simplified model6, but just barely. BIC, by nature, tends to favor the more simplistic model6 with a lower score over the more complex model5, and the model6 BIC score is lower than the model5 score by just over 13 points. With AIC scores very similar and BIC scores showing a more distinct difference, I would choose the simplified model6 logistic regression model.

Let's create a histogram of predicted values for Model 6.

```
model6Pred <- fitted(model6)
mydata$model6Pred <- model6Pred
hist(model6Pred)
```




Using the drop-in deviance test:

```
anova(model6, model5, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: HASSTRESS ~ COHES
## Model 2: HASSTRESS ~ COHES + ESTEEM + GRADES + SATTACH
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       649     817.86
## 2       646     811.79  3   6.0716   0.1082
```

The difference in deviance between the full model (model5) and reduced model (model6) is 6.0716. But with a p-value of 0.1082, the difference is not large enough (or statistically significant) to favor the full model over the reduced model.

Though the simplified model6 would be favored over the more complex model5, we notice that predictions for model6 fall over 0.5, in which the default threshold of 0.5 would predict all observations in the dataset with HASSTRESS = 1.

**Part 9**

**It may be that there are two (or more) process at work that are overlapped and generating the distributions of STRESS(Y). What do you think those processes might be?**

Given the large number of 0 STRESS events in the dataset and are not well-represented in the model, we should create a hybrid ZIP model which handles the predicting 0 STRESS events using logistic regression with a Poisson regression model which handles predicting one or more STRESS events.

**To conduct a ZIP regression model by hand, fit a Logistic Regression model to predict if stress is present (HASTRESS), and then use a Poisson Regression model to predict the number of stressful events (STRESS) conditioning on stress being present.**

Logistic regression model:

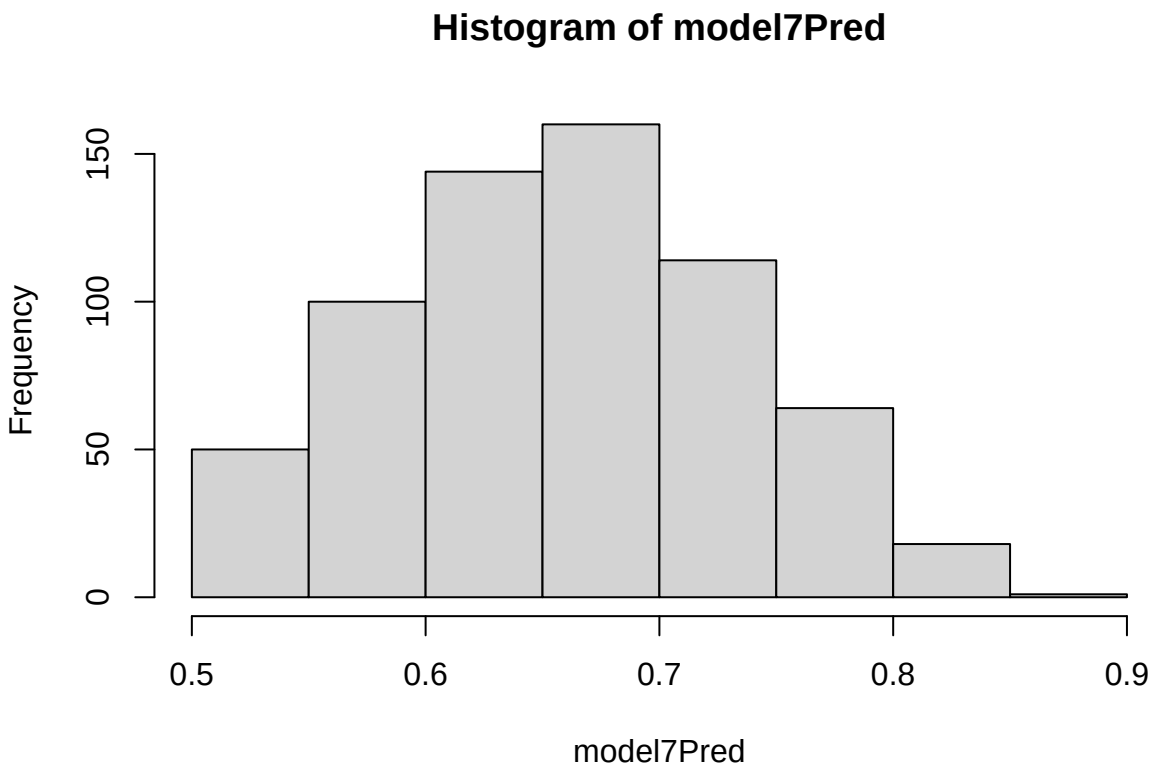
```
# using same model6 from part 8
model7 <- glm(HASSTRESS ~ COHES , family = "binomial", data = mydata)
summary(model7)
```

```
##
## Call:
## glm(formula = HASSTRESS ~ COHES, family = "binomial", data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9543  -1.3432   0.8055   0.9375   1.1703
##
## Coefficients:
##              Estimate Std. Error z value    Pr(>|z|)
## (Intercept)  2.296371   0.427310   5.374 0.000000077 ***
## COHES       -0.030393   0.007715  -3.939 0.000081681 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 834.18  on 650  degrees of freedom
## Residual deviance: 817.86  on 649  degrees of freedom
## AIC: 821.86
##
## Number of Fisher Scoring iterations: 4
```

```
model7Pred <- fitted(model7)
range(model7Pred)
```

```
## [1] 0.5042167 0.8518633
```

```
hist(model7Pred)
```



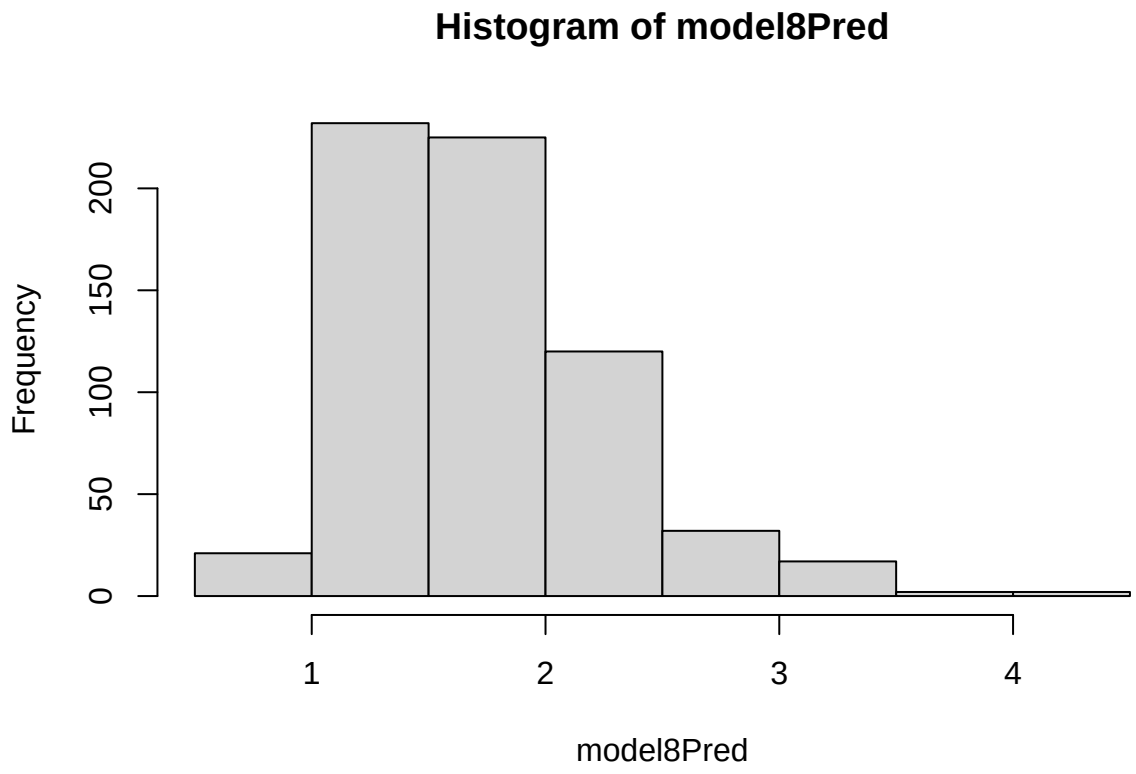

Poisson Regression model:

```
# same as model3 from part 4 - Poisson Regression
model8 <- glm(STRESS ~ COHES + ESTEEM + GRADES + SATTACH, family = "poisson", data = mydata)
summary(model8)
```

```
##
## Call:
## glm(formula = STRESS ~ COHES + ESTEEM + GRADES + SATTACH, family = "poisson",
##     data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7111  -1.5989  -0.2914   0.7107   3.6424
##
## Coefficients:
##             Estimate Std. Error z value             Pr(>|z|)
## (Intercept)  2.734513   0.234066  11.683 < 0.0000000000000002 ***
## COHES       -0.012918   0.002893  -4.466           0.00000798 ***
## ESTEEM      -0.023692   0.008039  -2.947              0.00321 **
## GRADES      -0.023471   0.009865  -2.379              0.01735 *
## SATTACH     -0.016481   0.005783  -2.850              0.00437 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
##     Null deviance: 1349.8  on 650  degrees of freedom
## Residual deviance: 1245.4  on 646  degrees of freedom
## AIC: 2417.2
##
## Number of Fisher Scoring iterations: 5
```

```
model8Pred <- fitted(model8)
hist(model8Pred)
```




**Is it reasonable to use such a model?**

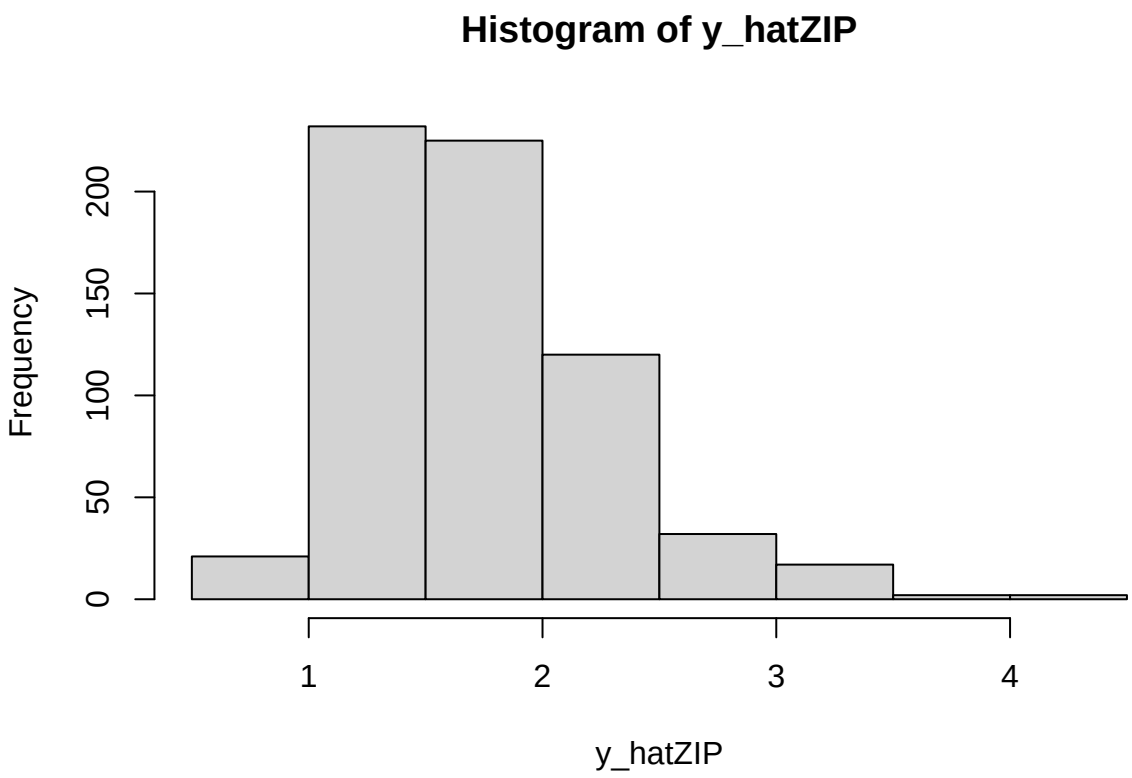
Using this manual hybrid ZIP model should help take into account the probability of calculating 0 STRESS events in adolescents as the dataset has a very large portion of observations that have 0 STRESS and should be represented, while able to also predict a number of STRESS events around the overall mean.

**Combine the two fitted model to predict STRESS (Y).**

```
y_hatZIP <- ifelse(model7Pred < 0.50, 0, model8Pred)
mydata$manZIPModelPred <- y_hatZIP
```

**Obtained predicted values and residuals.**

```
hist(y_hatZIP, breaks=10)
```




**How well does this model fit? HINT:You have to be thoughtful about this. It is not as straight forward as plug and chug!**

While this manual hybrid model can fit better than the other models made so far, it still doesn't represent the large number of 0 STRESS events from the dataset. Two things I would suggest in improving the model would be determining a better threshold for the predicted values of the logistic regression model. The range of the predicted/fitted values of the logistic regression model is:

```
range(model7Pred)
```

```
## [1] 0.5042167 0.8518633
```

Given the minimum value of 0.504, with the threshold set at 0.5 to use the logistic regression model, none of the observations in the dataset would be able to use it. The threshold needs to be raised, but to what threshold would require additional research into it.

For example, if the threshold was changed to 0.62:

```
y_hatZIPAlt <- ifelse(model7Pred < 0.62, 0, model8Pred)
hist(y_hatZIPAlt, breaks=10)
```


We would see a Poisson distribution more similar to that of STRESS in the dataset.

Another suggestion would be to gather more data, as I feel 650 observations is a somewhat limited sample size. As we collect more data, though, the Poisson regression model will start to look similar to a normal distribution, and could see where 0 STRESS events actually play in a larger sample size in respect to the other number of STRESS events.

**Part 10**

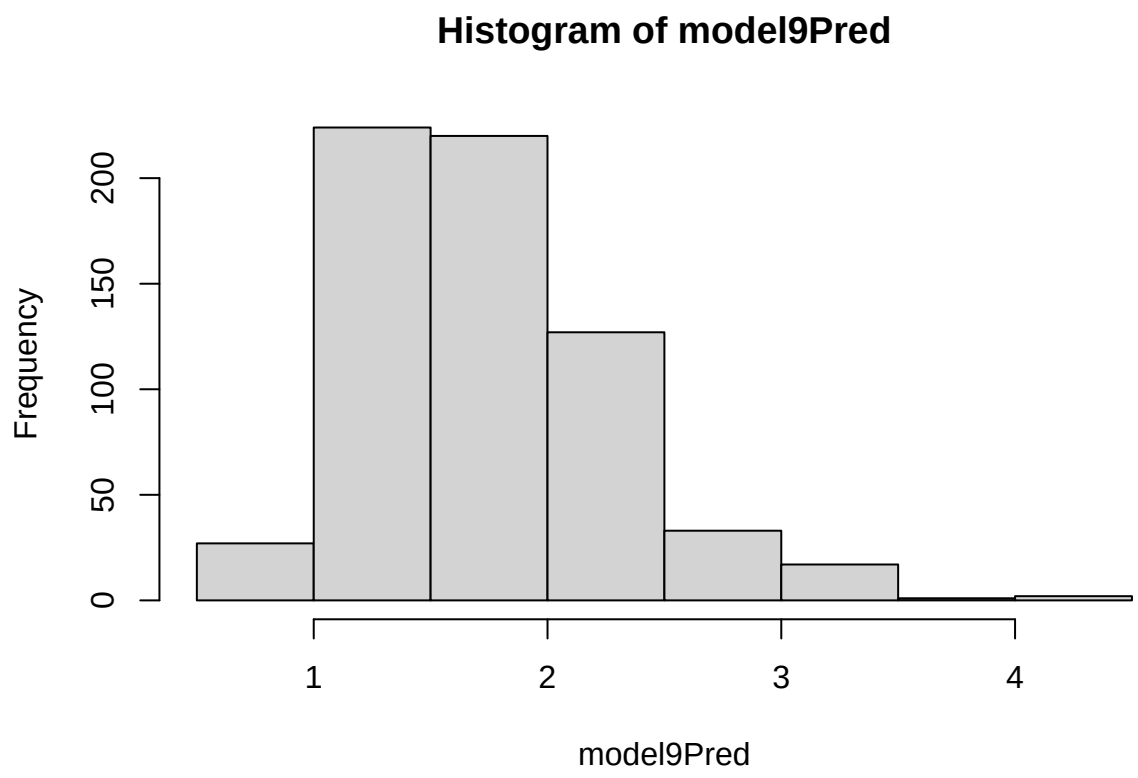
**Use the pscl package and the zeroinfl() function to Fit a ZIP model to predict STRESS(Y). You should do this twice, first using the same predictor variables for both parts of the ZIP model.**

```
model9 <- zeroinfl(STRESS ~ COHES + ESTEEM + GRADES + SATTACH | COHES + ESTEEM + GRADES + SATTACH, data
summary(model9)
```

```
##
## Call:
## zeroinfl(formula = STRESS ~ COHES + ESTEEM + GRADES + SATTACH | COHES +
##     ESTEEM + GRADES + SATTACH, data = mydata)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.4534 -0.9136 -0.2166  0.6257  3.9954
##
## Count model coefficients (poisson with log link):
```

```
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)  2.641690   0.272348   9.700 < 0.0000000000000002 ***
## COHES       -0.008258   0.003416  -2.418              0.01561 *
## ESTEEM      -0.026068   0.009206  -2.832              0.00463 **
## GRADES      -0.019553   0.010914  -1.792              0.07320 .
## SATTACH     -0.010485   0.006673  -1.571              0.11611
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.835429   0.983249  -2.884  0.00393 **
## COHES        0.018917   0.012124   1.560  0.11869
## ESTEEM      -0.004328   0.032777  -0.132  0.89495
## GRADES       0.014330   0.037731   0.380  0.70409
## SATTACH      0.024838   0.024083   1.031  0.30238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 17
## Log-likelihood: -1134 on 10 Df
```

```
model9Pred <- fitted(model9)
mydata$model9Pred <- model9Pred
hist(model9Pred)
```

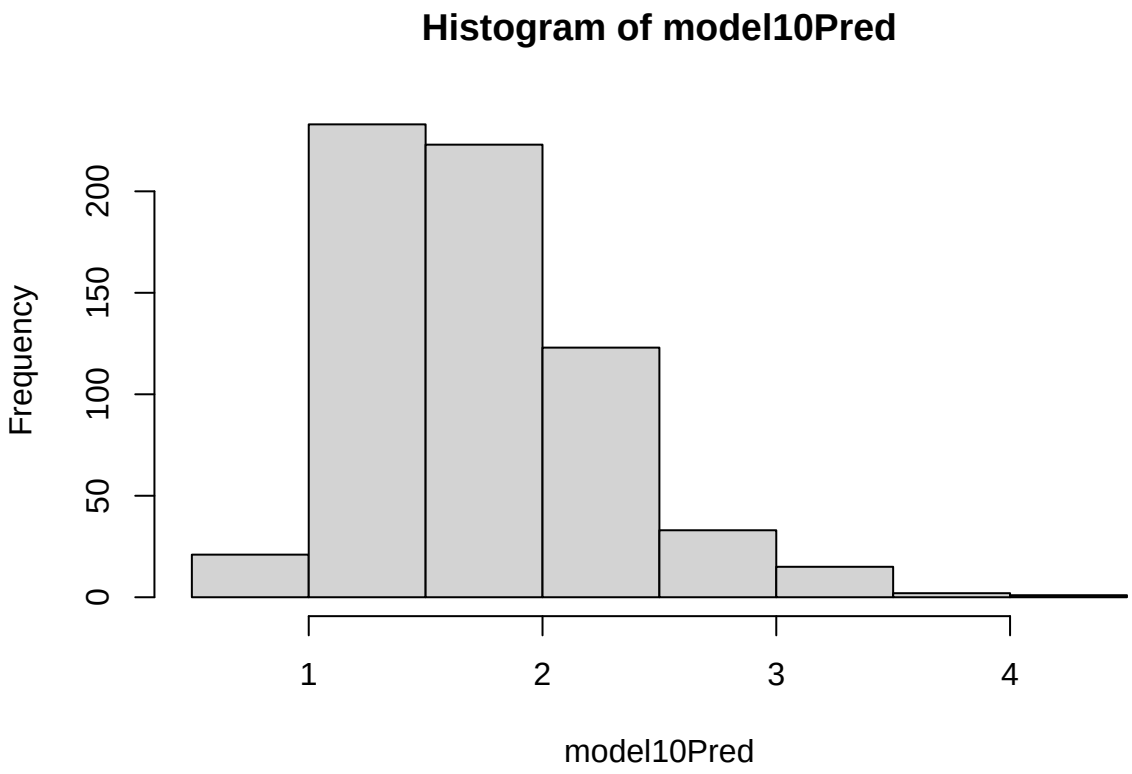



**Second, finding the best fitting model.**

```r
# AIC 2284.279
model10 <- zeroinfl(STRESS ~ COHES + ESTEEM + GRADES + SATTACH | COHES, data = mydata)
summary(model10)
```

```
##
## Call:
## zeroinfl(formula = STRESS ~ COHES + ESTEEM + GRADES + SATTACH | COHES,
##     data = mydata)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.4987 -0.9186 -0.2347  0.6176  4.0011
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)  2.677581   0.261613  10.235 < 0.0000000000000002 ***
## COHES       -0.007637   0.003363  -2.271              0.02317 *
## ESTEEM      -0.026172   0.008785  -2.979              0.00289 **
## GRADES      -0.020506   0.010540  -1.945              0.05172 .
## SATTACH     -0.012557   0.006362  -1.974              0.04841 *
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept) -2.39920    0.57802  -4.151 0.0000331 ***
## COHES        0.02444    0.01060   2.306    0.0211 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 13
## Log-likelihood: -1135 on 7 Df
```

```r
model10Pred <- fitted(model10)
mydata$model10Pred <- model10Pred
hist(model10Pred)
```




**Report the results and goodness of fit measures.**

```
AIC(model9)
```

```
## [1] 2288.802
```

```
BIC(model9)
```

```
## [1] 2333.587
```

```
AIC(model10)
```

```
## [1] 2284.279
```

```
BIC(model10)
```

```
## [1] 2315.628
```

The AIC and BIC scores are lower for the simplified 'best-fitting' model which suggests to use that model instead. That said, the models are very similar to that of the manual hybrid ZIP model we created in part 9, as it looks like the logistic regression part of the model was not used and defaulted to the Poisson model.

Like part 9, the logistic regression portion of these ZIP models are supposed to support the 0 STRESS counts (thus the 'Zero-Inflated' in ZIP) but it still has its limitations if the default lower threshold is not met with its predictions or fitted values from its dataset.

**Synthesize your findings across all of these models, to reflect on what you think would be a good modeling approach for this data.**

If we're trying to predict a discrete countable quantity of something as our dependent variable, the best bet would be to use some form of Poisson regression model: either a regular Poisson, Negative Binomial, manual hybrid ZIP, or an auto-ZIP model. There is not too much effort involved to attempt all these models in order to find the best-fitting one. I appreciate these exercises of developing these kinds of models along with comparing them to linear regression models to get an understanding of how fit to the data.