GLMs for Count Data Exercise

```
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
```

```
# install.packages("AER")
# install.packages("pscl")
library(AER)
library(pscl)
library(tidyverse)
```

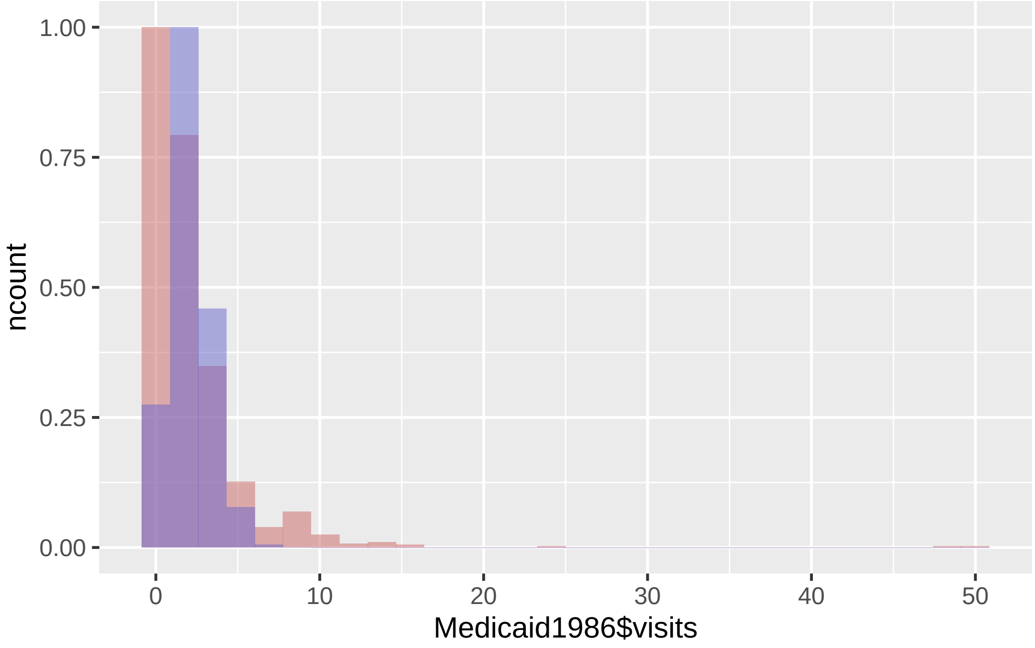
Use the `AER` package to load the `Medicaid1986` dataset. Information on the included data can be found using the `?Medicaid1986` command.

Suppose we are interested in modeling the number of doctor visits per year for each individual in the dataset, conditional on the information provided in all other given variables. What type of data is this response variable, and which distribution would be most applicable in this case?

This data is count data, as it describes the number of times an event occurs in a given timeframe. This type of data is frequently assumed to have the Poisson distribution. However, the histograms below compare the distribution of the number of visits with data generated from a Poisson distribution with the same mean. We see that the observed data has both a higher proportion of observations where the outcome is equal to 0, and a wider right tail, indicating that over-dispersion may be present.

```
data("Medicaid1986")
```

```
ggplot() +
  geom_histogram(
    aes(x = Medicaid1986$visits, y = after_stat(ncount)),
    fill = "#CC6666",
    alpha = 0.5
  ) +
  geom_histogram(
    aes(x = rpois(10000, 1.9307), y = after_stat(ncount)),
    fill = "#6666CC",
    alpha = 0.5
  )
```



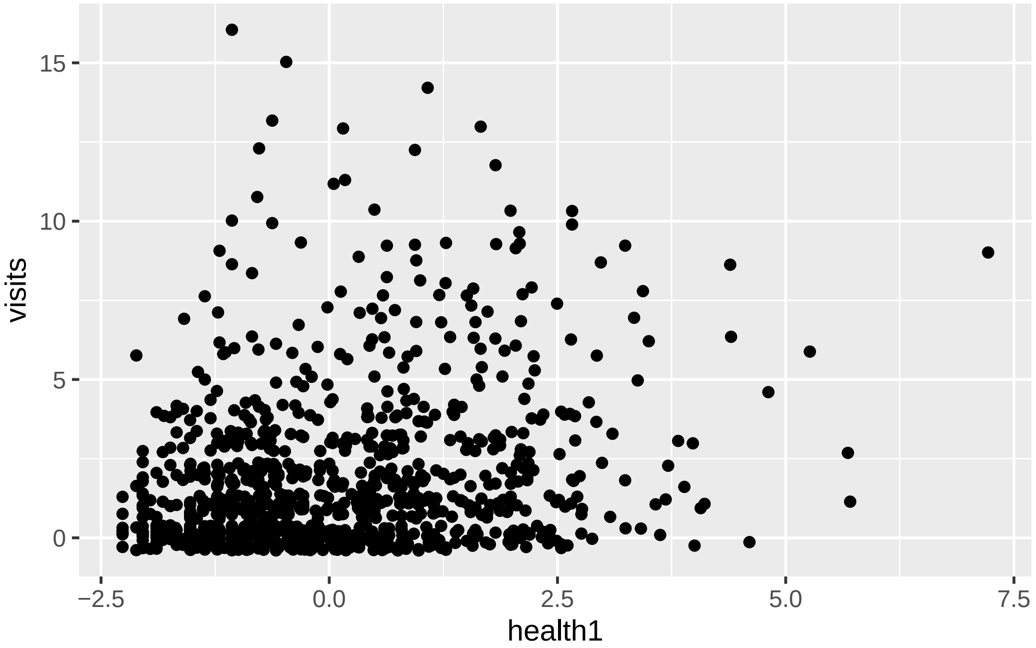
Part I

First, try using a log-linear model to fit the data. Use model diagnostics to determine whether the necessary assumptions hold, and provide an interpretation for the model coefficients. Do you think this model is a good fit?

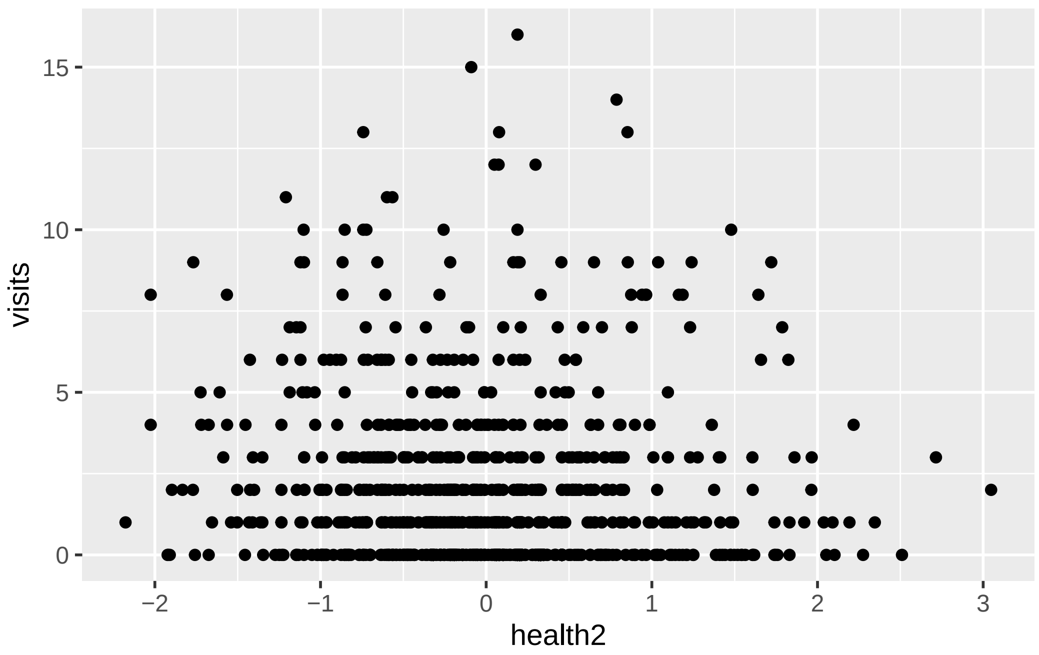
We begin with some data exploration. The variables that clearly have potential to be related to the number of visits attended are `health1`, `health2` and `access`, with the first two variables providing direct information about an individual's health status, while the third provides information on how easy or difficult it is to schedule and attend an appointment.

The presence of several outliers makes it difficult to assess the relationship between the variables, so we exclude them from the plots below. In the first plot, there appears to be a loose positive association between `health1` and the number of visits.

```
Medicaid1986 %>%
  filter(visits < 20) %>%
  ggplot() +
  geom_point(aes(x = health1, y = visits), position = "jitter")
```

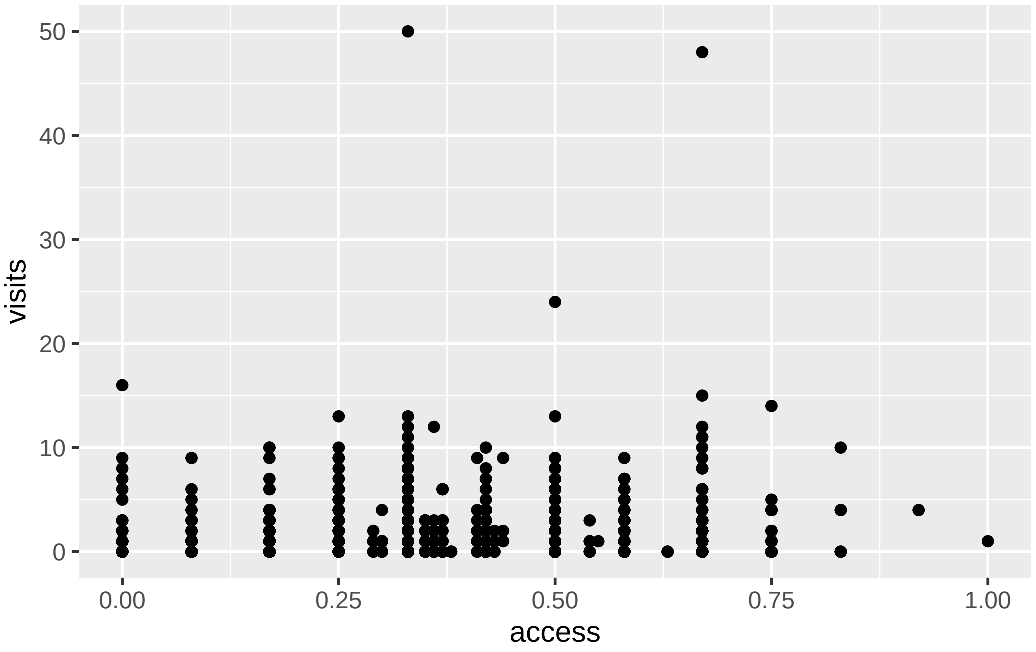


```
Medicaid1986 %>%
  filter(visits < 20) %>%
  ggplot() +
  geom_point(aes(x = health2, y = visits))
```



However, it is difficult to see any meaningful relationship between `health2` and the number of visits, and `access` and the number of visits.

```
ggplot(Medicaid1986) +
  geom_point(aes(x = access, y = visits))
```



We try fitting the Poisson regression model on all available predictive variables. The model summary indicates a large number of significant predictors, with `health1` and `access` in particular showing significant large positive coefficients.

```r
pois_mod <- glm(
  visits ~ .,
  data = Medicaid1986,
  family = poisson()
)

summary(pois_mod)
```

```
Call:
glm(formula = visits ~ ., family = poisson(), data = Medicaid1986)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.1806  -1.5863  -0.7866   0.5046  12.1999

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)        -0.8867054  0.3163991  -2.802 0.005071 ** 
exposure            0.0150185  0.0027020   5.558 2.73e-08 ***
children           -0.1291116  0.0248046  -5.205 1.94e-07 ***
age                -0.0102022  0.0030345  -3.362 0.000774 ***
income              0.0173267  0.0066191   2.618 0.008853 ** 
health1             0.2918326  0.0141811  20.579  < 2e-16 ***
health2             0.0003056  0.0289546   0.011 0.991580    
access              0.4335053  0.1274871   3.400 0.000673 ***
marriedyes         -0.1959003  0.0621391  -3.153 0.001618 ** 
gendermale          0.0495019  0.0673294   0.735 0.462205    
ethnicitycaucasian -0.0030264  0.0541794  -0.056 0.955454    
school              0.0075335  0.0059801   1.260 0.207754    
enrollyes          -0.1331639  0.0471751  -2.823 0.004761 ** 
programssi          0.5924984  0.1535290   3.859 0.000114 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3497.7  on 995  degrees of freedom
Residual deviance: 2971.5  on 982  degrees of freedom
AIC: 4627.8

Number of Fisher Scoring iterations: 6
```

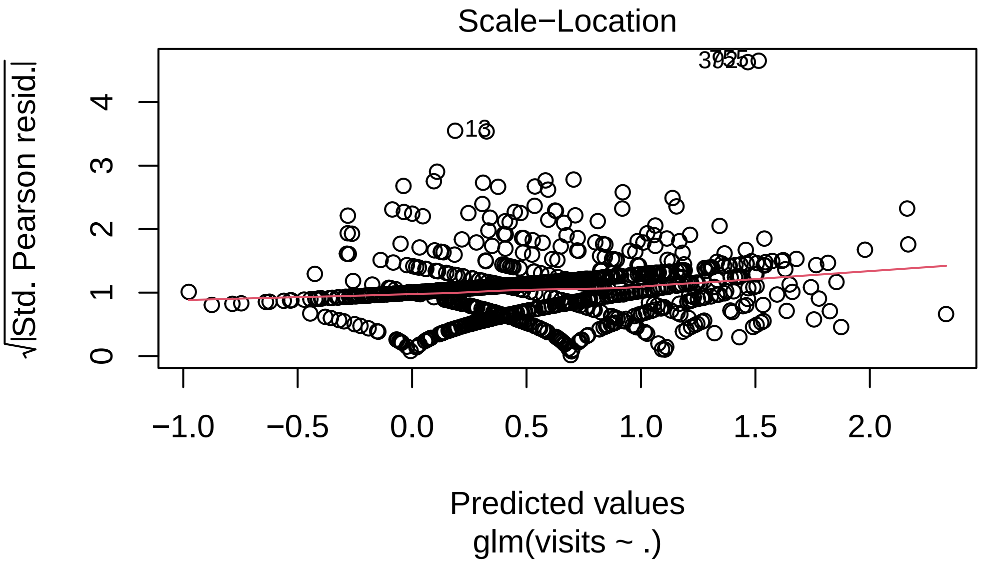
Exponentiating the coefficients allows us to clearly interpret them. For instance, we can say that, all else equal, for each unit increase in access to health care services, the predicted number of doctor's visits will be 154% that of the predicted number of visits for the lower access value. Interestingly, the model predicts that for each increase of one child living in the individual's household, the number of doctor's visits will be 88% that of the predicted number for someone with one fewer child.

```
exp(pois_mod$coefficients)
```

```
       (Intercept)           exposure           children                age 
         0.4120109          1.0151318          0.8788759          0.9898497 
            income            health1            health2             access 
         1.0174777          1.3388789          1.0003056          1.5426555 
        marriedyes         gendermale ethnicitycaucasian             school 
         0.8220942          1.0507476          0.9969782          1.0075620 
         enrollyes         programssi 
         0.8753216          1.8085011 
```

To assess whether over-dispersion is present, we use residual plots as well as well as estimating the dispersion term using the residual deviance. By dividing the residual deviance of 2971.5 by the number of degrees of freedom (982), we estimate the deviance term to be equal to 3.026, much higher than the value of 1 that represents equi-dispersion. The residual plot shown below also indicates the presence of over-dispersion, with the mean value of the square root of the Pearson residuals climbing above 1 at points in the plot (we want to see the red line flat and staying below 1, ideally around 0.8).

```
plot(pois_mod, which = 3)
```



Finally, the `AER` package provides a statistical test to confirm the presence of over-dispersion:

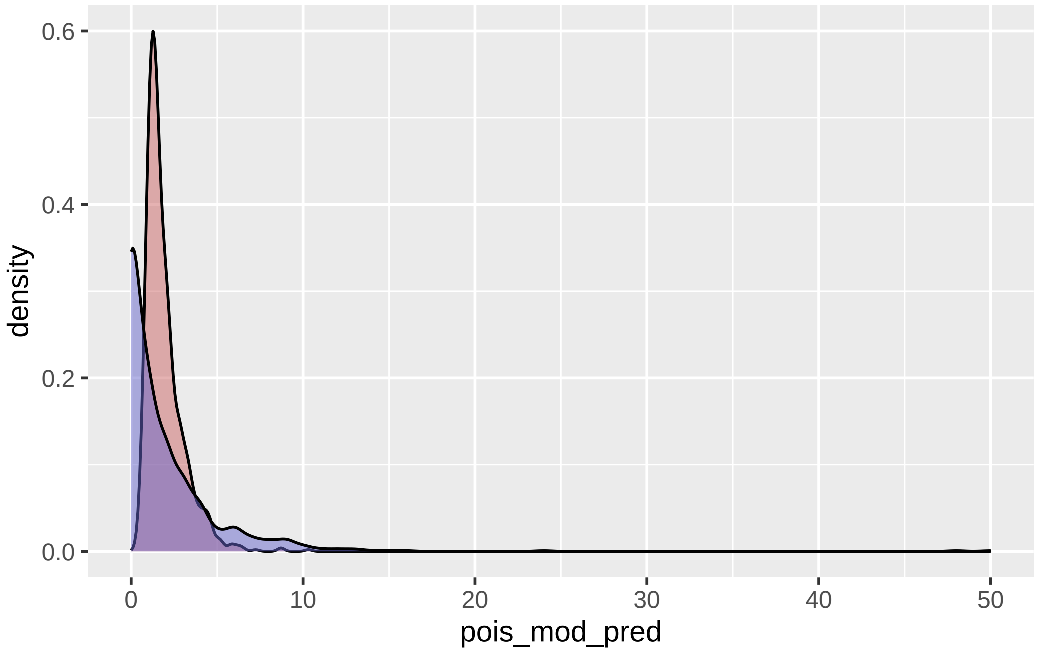
```
dispersiontest(pois_mod)
```

```
	Overdispersion test

data:  pois_mod
z = 4.4372, p-value = 4.556e-06
alternative hypothesis: true dispersion is greater than 1
sample estimates:
```

```
dispersion
  3.991026
```

Additionally, comparing the distribution of the predicted response values from the model and the observed response values shows that we are substantially underestimating the number of responses that equal 0.

```
pois_mod_pred <- predict(pois_mod, type = "response")
ggplot() +
  geom_density(aes(x = pois_mod_pred), fill = "#CC6666", alpha = 0.5) +
  geom_density(aes(x = Medicaid1986$visits), fill = "#6666CC", alpha = 0.5)
```



One way that we could possibly rectify this situation is to use the variance stabilizing transformation $f(Y) = \sqrt{Y}$.

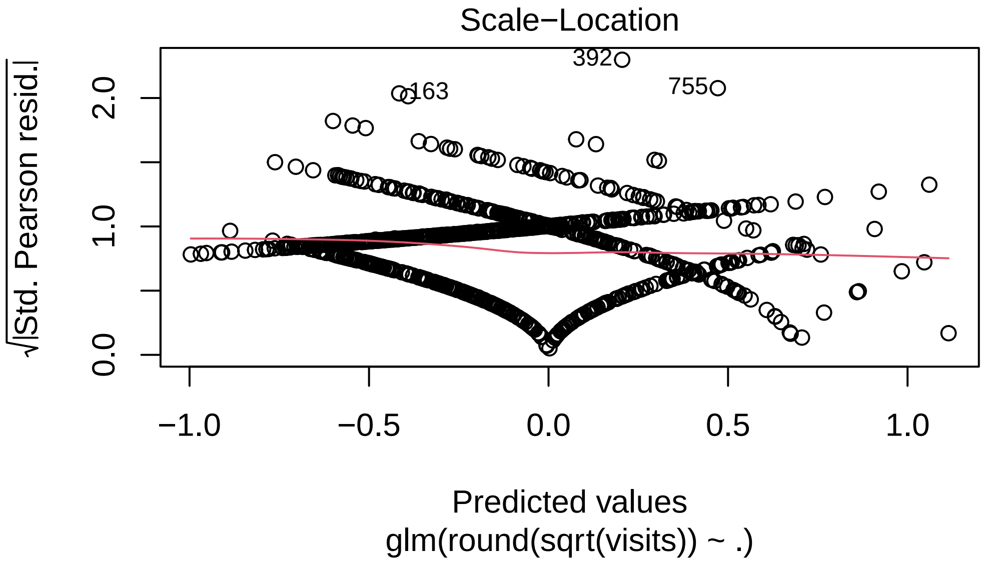
```
pois_mod2 <- glm(round(sqrt(visits)) ~ ., data = Medicaid1986, family = poisson())

summary(pois_mod2)
```

```
Call:
```

```
glm(formula = round(sqrt(visits)) ~ ., family = poisson(), data = Medicaid1986)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4033  -1.1473  -0.1158   0.4967   3.5814

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -0.801952   0.442589  -1.812   0.0700 .
exposure            0.005771   0.003785   1.524   0.1274
children           -0.057118   0.033508  -1.705   0.0883 .
age                -0.004138   0.004313  -0.959   0.3374
income              0.007029   0.009620   0.731   0.4650
health1             0.209518   0.021080   9.939   <2e-16 ***
health2            -0.054351   0.042701  -1.273   0.2031
access              0.171703   0.183385   0.936   0.3491
marriedyes         -0.085963   0.087790  -0.979   0.3275
gendermale         -0.029052   0.098130  -0.296   0.7672
ethnicitycaucasian  0.044403   0.078069   0.569   0.5695
school              0.003294   0.008638   0.381   0.7029
enrollyes          -0.052859   0.067641  -0.781   0.4345
programssi          0.448095   0.217540   2.060   0.0394 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1182.9  on 995  degrees of freedom
Residual deviance: 1050.1  on 982  degrees of freedom
AIC: 2439

Number of Fisher Scoring iterations: 5
```

From this second model, we see that the number of predictor variables with significant predictors is substantially reduced. Importantly, we see that the residual deviance indicates an estimate of the dispersion parameter close to 1, showing that over-dispersion is likely not a problem in this case.

```r
plot(pois_mod2, which = 3)
```

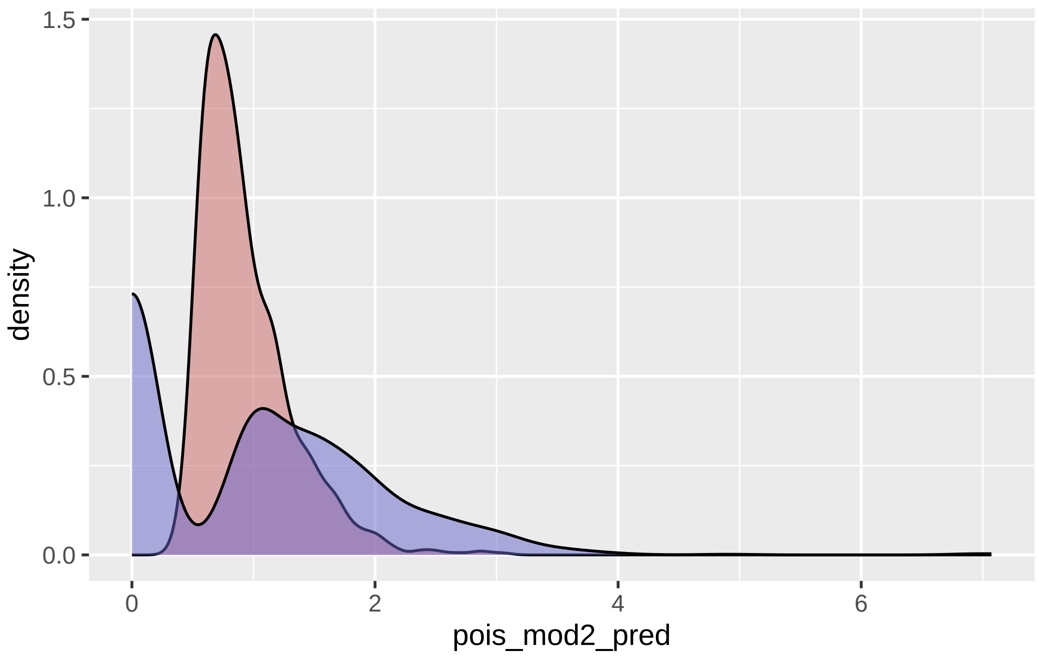


This is confirmed with a residual plot showing the Pearson residuals at more reasonable levels, as well as the dispersion test showing a high p-value.

```
dispersiontest(pois_mod2)
```

```


    Overdispersion test

data:  pois_mod2
z = -1.3895, p-value = 0.9177
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
 0.9384178
```

```
pois_mod2_pred <- predict(pois_mod2, type = "response")

ggplot() +
  geom_density(aes(x = pois_mod2_pred), fill = "#CC6666", alpha = 0.5) +
  geom_density(aes(x = sqrt(Medicaid1986$visits)), fill = "#6666CC", alpha = 0.5)
```



However, the density plots above still indicate that we are not properly capturing the distribution of the observed responses.

Part II

Next, try fitting a hurdle model for the same outcome. Why might this approach be useful in this case? How do the predictions differ between the two models?

From the previous section, we saw that the Poisson regression model did not properly capture the true distribution of the observed outcomes, particularly underestimating the number of outcomes equal to 0. A hurdle model may be better equipped to capture this.

```
hurdle_mod <- hurdle(visits ~ ., data = Medicaid1986)

summary(hurdle_mod)
```

```
Call:
hurdle(formula = visits ~ ., data = Medicaid1986)

Pearson residuals:
    Min      1Q  Median      3Q     Max
-2.3795 -0.8000 -0.5181  0.3775 14.1988

Count model coefficients (truncated poisson with log link):
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -0.808567   0.348541  -2.320 0.020348 *
exposure            0.018868   0.002991   6.307 2.84e-10 ***
children           -0.129444   0.028296  -4.575 4.77e-06 ***
age                -0.010718   0.003455  -3.102 0.001924 **
income              0.030114   0.007383   4.079 4.53e-05 ***
health1             0.200615   0.016087  12.470  < 2e-16 ***
health2             0.068033   0.030325   2.243 0.024869 *
access              0.485621   0.140629   3.453 0.000554 ***
marriedyes         -0.204584   0.068895  -2.970 0.002983 **
gendermale          0.064258   0.072953   0.881 0.378413
ethnicitycaucasian -0.012498   0.058488  -0.214 0.830791
school              0.008955   0.006540   1.369 0.170920
enrollyes          -0.165560   0.051057  -3.243 0.001184 **
programssi          0.382564   0.174958   2.187 0.028771 *
Zero hurdle model coefficients (binomial with logit link):
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)         0.265787   0.871609   0.305  0.76041
exposure            0.002439   0.007391   0.330  0.74143
children           -0.098075   0.061884  -1.585  0.11301
age                -0.007272   0.008804  -0.826  0.40882
income             -0.016952   0.020308  -0.835  0.40387
health1             0.433125   0.055473   7.808 5.82e-15 ***
health2            -0.253615   0.096338  -2.633  0.00847 **
access              0.165149   0.383701   0.430  0.66690
marriedyes         -0.120028   0.184436  -0.651  0.51519
gendermale         -0.082326   0.212973  -0.387  0.69908
ethnicitycaucasian  0.043048   0.157701   0.273  0.78487
school              0.004089   0.018667   0.219  0.82662
enrollyes          -0.036528   0.140425  -0.260  0.79477
programssi          0.954884   0.452864   2.109  0.03498 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 22
Log-likelihood: -2020 on 28 Df
```

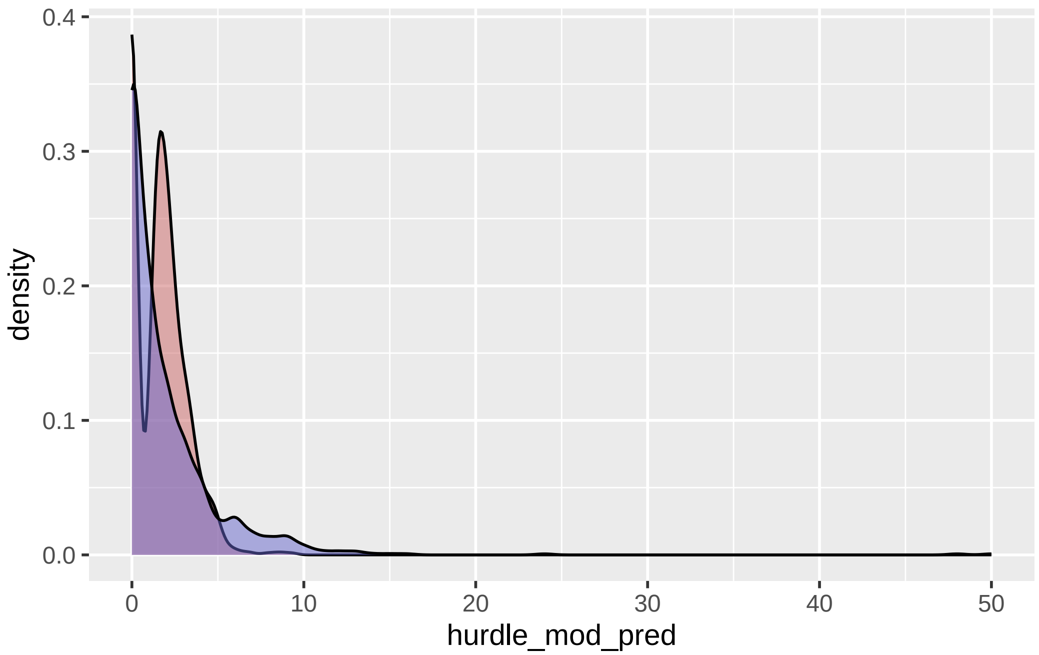
From the model summary, we see that the two health predictor variables are very important to modeling whether someone attends 0 doctor's visits in the given time frame or not. The count model at the top of the summary then considers a greater number of predictors. Generally, we can see that the coefficients are similar directionally to those seen in the Poisson regression model from the previous part.

To reach predictions from the hurdle model, we need to combine the results of the two models. To do this, we use two different `type` arguments in the `predict()` function: `type = "prob"` will give the probability of an output being equal to 0, while `type = "response"` gives the estimated outcome given that it is not equal to 0. To obtain the full predictions, we round the probability of predicting an outcome equal to 0 to the nearest integer, subtract this value from 1, then multiply by the response predictions. This will ensure that the full prediction is equal to 0 for any observation where the logistic regression model indicates that the outcome will be equal to 0, while not modifying the prediction from the count model when appropriate.

```
hurdle_mod_pred <- (1 - round(predict(hurdle_mod, type = "prob"))) *
  predict(hurdle_mod, type = "response")
```

From the density plots below, we can see that the hurdle model captures the true number of outcomes equal to 0 much more accurately than the Poisson regression model, indicating that the hurdle model likely has a better overall fit. We could confirm this by calculating the MSE for each model. Note that in practice, it will be more effective to split the data into train and test sets.

```
ggplot() +
  geom_density(aes(x = hurdle_mod_pred), fill = "#CC6666", alpha = 0.5) +
  geom_density(aes(x = Medicaid1986$visits), fill = "#6666CC", alpha = 0.5)
```


0.4
0.3
0.2
0.1
0.0
density
0
10
20
30
40
50
hurdle_mod_pred