Untitled

2024-02-25

ZERO INFLATED NEGATIVE BINOMIAL

Traditional Negative Binomial regression extends Poisson regression to manage overdispersion in count data, but it fails when an unusually high number of zero counts is present. The ZINB model is combining the principles of NB regression with a mechanism to account for excess zeros. Specifically, it differentiates between two sources of zeros: those arising from the data's natural variability "sampling zeros" and those that are structurally inherent or "excess zeros."

$$P(Y_i = y_i) = \begin{cases} \pi_i + (1-\pi_i)\frac{\Gamma(r+y_i)}{\Gamma(r)y_i!}\left(\frac{r}{r+\mu_i}\right)^r\left(\frac{\mu_i}{r+\mu_i}\right)^{y_i} & \text{if } y_i = 0, \\ (1-\pi_i)\frac{\Gamma(r+y_i)}{\Gamma(r)y_i!}\left(\frac{r}{r+\mu_i}\right)^r\left(\frac{\mu_i}{r+\mu_i}\right)^{y_i} & \text{if } y_i > 0. \end{cases}$$

The ZINB model accounts for the excess of zeros through the component π_i, which represents the probability that an observation will have a count of zero not due to the process described by the Negative Binomial distribution but due to some other, external process. Another key feature is the dispersion parameter r of the Negative Binomial distribution, which is used to model overdispersion. Smaller values of r indicate greater overdispersion relative to the Poisson distribution.

We use the model 'zeroinfl' that has two parts:

- left of the | symbol: Specifies the variables for the count model part. This part models the actual count of doctor visits based on predictors such as illness, actdays, hscore, chcond1, age:chcond2, hospadmi, prescrib, and nonpresc.
- right of the | symbol: Specifies the variables for the zero-inflation model part. This part models the excess zeros, predicting which zeros are "true zeros". Here, predictors like levyplus, age:income:freepoor, freepera, and interactions are used.

```
ZINB_model <- zeroinfl(doctorco ~illness * actdays + hscore + chcond1 + age: chcond2
+ hospadmi  + prescrib + nonpresc|levyplus + age:income:freepoor + freepera
+ illness * actdays + prescrib, data = train_data, dist = "negbin")

AIC(ZINB_model)
```

```
## [1] 4908.415
```

```
summary(ZINB_model)
```

```
##
## Call:
## zeroinfl(formula = doctorco ~ illness * actdays + hscore + chcond1 +
##     age:chcond2 + hospadmi + prescrib + nonpresc | levyplus + age:income:freepoor +
##     freepera + illness * actdays + prescrib, data = train_data, dist = "negbin")
```

```
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.2568 -0.4370 -0.2498 -0.1729 11.1233
##
## Count model coefficients (negbin with log link):
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.958567   0.132317  -7.244 4.34e-13 ***
## illness          0.097356   0.035625   2.733 0.006280 **
## actdays          0.101887   0.013337   7.640 2.18e-14 ***
## hscore           0.024368   0.013309   1.831 0.067112 .
## chcond11        -0.149808   0.088641  -1.690 0.091017 .
## hospadmi         0.185576   0.044157   4.203 2.64e-05 ***
## prescrib         0.082195   0.024279   3.385 0.000711 ***
## nonpresc        -0.151691   0.048728  -3.113 0.001852 **
## illness:actdays -0.007148   0.004526  -1.579 0.114245
## age:chcond20    -0.209779   0.209290  -1.002 0.316183
## age:chcond21    -0.624787   0.240798  -2.595 0.009469 **
## Log(theta)       0.893182   0.165991   5.381 7.41e-08 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.8827     0.2604   7.231 4.79e-13 ***
## levyplus1             -0.4330     0.2189  -1.978 0.047921 *
## freepera1             -1.5382     0.3749  -4.103 4.07e-05 ***
## illness               -0.4317     0.1119  -3.856 0.000115 ***
## actdays               -2.3676     0.8194  -2.890 0.003857 **
## prescrib              -1.6775     0.2515  -6.669 2.57e-11 ***
## illness:actdays        0.4764     0.1719   2.772 0.005567 **
## age:income:freepoor0   0.2498     0.6597   0.379 0.704958
## age:income:freepoor1  20.5385     8.8031   2.333 0.019643 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Theta = 2.4429
## Number of iterations in BFGS optimization: 72
## Log-likelihood: -2433 on 21 Df
```

There are significant variables that influence the number of doctor visits. Notably, income factors (both middle and high income showing lower visit rates compared to low-income counterparts), levyplus, health-related variables like illness severity, active days, and hospital admissions directly correlate with increased doctor visits, emphasizing the link between health needs and healthcare demand. Prescription medication requirements further elevate visit frequencies, reflecting ongoing health management needs. Conversely, the use of non-prescription medications is associated with fewer visits, hinting at self-care practices for minor health concerns.

The interaction term age:income:freepoor1 and its significant positive coefficient suggest that older individuals with higher income who qualify for free healthcare are less likely to visit the doctor. This pattern may arise from various factors such as improved health status, access to alternative health resources, or specific policies that affect their healthcare utilization differently. Additionally, the interaction between illness and actdays demonstrates a significant positive effect, indicating that individuals who are ill and experience more days of activity restriction are more likely to seek medical attention, which aligns with expectations.

In the zero-inflation part, variables like levyplus, freepera, illness, actdays, and prescrib are significant, pointing to specific factors that influence the propensity to have zero visits.

Theta is a parameter of the Negative Binomial distribution part of the model it is inversely related to the variance; a smaller θ indicates more dispersion (more variability in count data than what a Poisson model would suggest). Theta of 2.4 suggests some level of overdispersion in the data, but not extremely high

```
predicted_counts_zinb <- round(predict(ZINB_model, newdata = test_data, type = "response"))
predicted_category_zinb <- ifelse(predicted_counts_zinb < 1, 0,predicted_counts_zinb)

true_counts <- test_data$doctorco
mae_zinb <- mean(abs(predicted_counts_zinb - true_counts))
cat("MAE:", mae_zinb, "\n")
```

```
## MAE: 0.2649326
```

```
rmse_zinb <- sqrt(mean((predicted_counts_zinb - true_counts)^2))
cat("RMSE:", rmse_zinb, "\n")
```

```
## RMSE: 0.6988843
```

The modelsts Mean Absolute Error (MAE) is indicating a relatively precise prediction capability given the context of count data; but still has room for improvement, particularly in accurately predicting higher counts of visits as seen from the Root Mean Squared Error (RMSE).

This section explores the examination of binary outcomes—specifically, the presence or absence of doctor visits. By utilizing confusion matrices and metrics such as balanced accuracy and AUC-ROC, we want to evaluate the model's ability to accurately predict actual visits against the backdrop of a skewed distribution

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 778 104
##          1  72  84
##
##                Accuracy : 0.8304
##                  95% CI : (0.8062, 0.8528)
##     No Information Rate : 0.8189
##     P-Value [Acc > NIR] : 0.17730
##
##                   Kappa : 0.3878
##
##  Mcnemar's Test P-Value : 0.01945
##
##             Sensitivity : 0.9153
##             Specificity : 0.4468
##          Pos Pred Value : 0.8821
##          Neg Pred Value : 0.5385
##              Prevalence : 0.8189
##          Detection Rate : 0.7495
##    Detection Prevalence : 0.8497
##       Balanced Accuracy : 0.6811
##
##        'Positive' Class : 0
##
```

```
## Balanced Accuracy: 0.5384615
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## AUC-ROC: 0.6810513
```

The high accuracy indicates that almost all predictions made by the model are correct. This is a relatively high overall accuracy rate. The sensitivity of reflects the model's strong performance in predicting non-visits accurately, a result of the data's inherent imbalance towards this outcome.

However, the model's balanced accuracy and an AUC-ROC score suggest that while the model is better than a dummy classifier, there is room for improvement, particularly in correctly identifying actual visits.

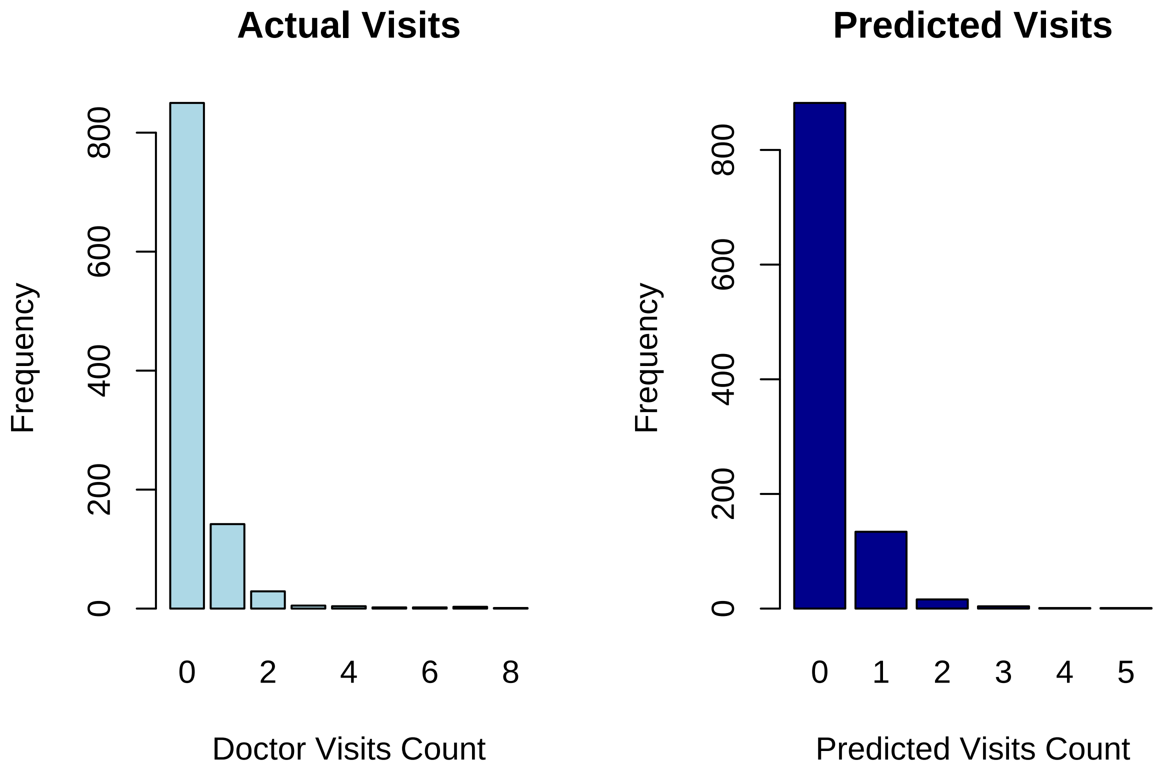
By plotting distribution of both actual and predicted visits, we gain insights into the model's performance in capturing the true distribution of healthcare utilization. Additionally, examining the specific actual versus predicted counts across visit frequencies enables us to identify where the model performs well.

```r
actual_freq <- table(true_counts)
predicted_freq_zinb <- table(predicted_counts_zinb)

par(mfrow=c(1,2))

# Bar plot for Actual Counts
barplot(actual_freq, main="Actual Visits", xlab="Doctor Visits Count",
        ylab="Frequency", col="lightblue")

# Bar plot for Predicted Counts
barplot(predicted_freq_zinb, main="Predicted Visits", xlab="Predicted Visits Count",
        ylab="Frequency", col="darkblue")
```



```r
par(mfrow=c(1,1))


for (i in 0:9) {
  actual_ <- true_counts == i
  predicted_ <- predicted_counts_zinb == i
  actual_count <- sum(actual_)
  predicted_count<- sum(predicted_)
  cat("Actual count for", i, "Visits:", actual_count, "\n")
  cat("Predicted count for", i, "Visits:", predicted_count, "\n\n")
}
```

```
## Actual count for 0 Visits: 850
## Predicted count for 0 Visits: 882
##
## Actual count for 1 Visits: 142
## Predicted count for 1 Visits: 134
##
## Actual count for 2 Visits: 29
## Predicted count for 2 Visits: 16
##
## Actual count for 3 Visits: 5
## Predicted count for 3 Visits: 4
##
## Actual count for 4 Visits: 4
## Predicted count for 4 Visits: 1
```

```
##
## Actual count for 5 Visits: 2
## Predicted count for 5 Visits: 1
##
## Actual count for 6 Visits: 2
## Predicted count for 6 Visits: 0
##
## Actual count for 7 Visits: 3
## Predicted count for 7 Visits: 0
##
## Actual count for 8 Visits: 1
## Predicted count for 8 Visits: 0
##
## Actual count for 9 Visits: 0
## Predicted count for 9 Visits: 0
```

The model does well at predicting when there are no doctor visits, although it predicts slightly more zeros than there actually are. However, as the number of visits goes up, the model does not do as well. It is close when predicting one visit but starts to fall short with two visits and struggles more as the visit numbers increase, not predicting any visits of five or more at all. This gap between the actual and predicted numbers, especially for higher counts of visits, suggests that the model might need some improvements or additional data to better predict these less common situations.

HURDLE NEGATIVE BINOMIAL

The hurdle model offers a distinct approach to modeling count data, unlike Zero-Inflated models, hurdle models decompose the prediction process into two sequential components: a binary process for distinguishing between zero and non-zero counts, followed by a truncated count distribution model exclusively for the positive counts. This structure creates a "hurdle" that separates zero predictions from positive ones, meaning observations must first cross this hurdle before they are considered for positive count predictions.

In the hurdle model framework, the probability of observing a zero count ($y_i = 0$) is denoted by p_i, while the distribution of positive counts ($y_i > 0$) follows a truncated distribution, adjusted to exclude the probability of zero counts. The mathematical representation of the HNB model is as follows:

$$P(Y_i = y_i) = \begin{cases} p_i & \text{if } y_i = 0, \\ (1 - p_i)\frac{p(y_i;\mu_i)}{1-p(y_i=0;\mu_i)} & \text{if } y_i > 0, \end{cases}$$

Here, p_i delineates the probability that an observation falls into the zero count category, while $p(y_i; \mu_i)$ denotes the probability mass function (PMF) for positive counts, parameterized by μ_i, within a Negative Binomial distribution managing the observed overdispersion in the data.

The HNB model separates the prediction of no visits from the prediction of one or more visits to better understand healthcare usage. It first decides if a visit happens at all and then predicts how many visits will happen if it does. The model uses different factors to predict both the chance of no visits and the expected number of visits, helping us understand what influences these outcomes.

The hurdle model implemented with the 'pscl' library in R is designed dividing the modeling process into two distinct parts separated by '|'.

```
hurdle_model <- hurdle(doctorco ~ illness + actdays + hospadmi|income:freepoor +
                          actdays *illness + sex*hscore + hospadmi + prescrib + nonpresc,
                        data = train_data, dist ="negbin")
```

```
AIC(hurdle_model)
```

```
## [1] 4980.611
```

```
summary(hurdle_model)
```

```
##
## Call:
## hurdle(formula = doctorco ~ illness + actdays + hospadmi | income:freepoor +
##     actdays * illness + sex * hscore + hospadmi + prescrib + nonpresc,
##     data = train_data, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.4122 -0.4015 -0.3132 -0.2595 10.4335
##
## Count model coefficients (truncated negbin with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.00125    1.11825  -2.684  0.00728 **
## illness      0.11847    0.05459   2.170  0.02998 *
## actdays      0.13667    0.01708   8.002 1.22e-15 ***
## hospadmi     0.30888    0.10419   2.965  0.00303 **
## Log(theta)  -1.89445    1.27138  -1.490  0.13620
## Zero hurdle model coefficients (binomial with logit link):
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.450556   0.131538 -18.630  < 2e-16 ***
## actdays           0.235126   0.024681   9.527  < 2e-16 ***
## illness           0.236380   0.035359   6.685 2.31e-11 ***
## sex1              0.318975   0.109305   2.918 0.003521 **
## hscore            0.105686   0.029848   3.541 0.000399 ***
## hospadmi          0.151833   0.079819   1.902 0.057144 .
## prescrib          0.341265   0.031415  10.863  < 2e-16 ***
## nonpresc         -0.159138   0.061956  -2.569 0.010212 *
## income:freepoor0 -0.054107   0.125726  -0.430 0.666934
## income:freepoor1 -4.064799   1.126978  -3.607 0.000310 ***
## actdays:illness  -0.040363   0.008885  -4.543 5.54e-06 ***
## sex1:hscore      -0.078716   0.036973  -2.129 0.033253 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Theta: count = 0.1504
## Number of iterations in BFGS optimization: 21
## Log-likelihood: -2473 on 17 Df
```

```
#hurdle with factors
hurdle_model2 <- hurdle(doctorco ~ income_factor+illness +  actdays+ hospadmi|
                          income:freepoor + actdays *illness + sex*hscore + hospadmi +
                          age_factor*prescrib + nonpresc,
                        data = train_data, dist = "negbin")

AIC(hurdle_model2)
```

```
## [1] 4921.703
```

Key findings from the model coefficients suggest that factors such as income level, illness severity, the number of activity days, hospital admissions, and prescription medication usage significantly influence both the likelihood of making any doctor visit and the frequency of those visits among patients who do.

Notably, the interaction terms, such sex with health score, underscore how the combined effect of these variables can either increase or decrease the likelihood of seeking medical care. For instance, the significant negative coefficient for the interaction between income and freepoor1 suggests that patients from lower-income brackets with access to free poor services are less likely to have zero visits, indicating targeted healthcare access among vulnerable populations.

The theta value, reported as 0.1933 in the count model, is indicative of the degree of overdispersion relative to what a Poisson distribution would predict. A theta value significantly lower than 1 points towards high overdispersion, validating the choice of a negative binomial distribution over a Poisson.

```
predicted_counts_hurdle <- round(predict(hurdle_model, newdata=test_data, type = "response"))
predicted_category_hnb <- ifelse(predicted_counts_hurdle< 1, 0, predicted_counts_hurdle)


true_counts <- test_data$doctorco
mae_hurdle <- mean(abs(predicted_counts_hurdle- true_counts))
cat("MAE:", mae_hurdle, "\n")
```

```
## MAE: 0.2736031
```

```
rmse_hurdle <- sqrt(mean((predicted_counts_hurdle - true_counts)^2))
cat("RMSE:", rmse_hurdle, "\n")
```

```
## RMSE: 0.731878
```

The MAE indicates a relatively small deviation, suggesting that the model's predictions are, on average, close to the true number of visits. The RMSE, which penalizes larger errors more heavily, is higher, suggesting that there are some instances of larger prediction errors, but overall, the model demonstrates a decent level of accuracy.

Also for the hurdle model, we evaluate alternative metrics for the binary outcome, focusing on the dichotomy of having or not having doctor visits. Employing confusion matrices, balanced accuracy, and AUC-ROC, this analysis aims to assess the model's precision in distinguishing actual visits in a dataset significantly skewed towards non-visits.

```
actual_binary <- ifelse(true_counts > 0, 1, 0)
predicted_binary <- ifelse(predicted_counts_hurdle > 0, 1, 0)
conf_matrix <- table(Actual = actual_binary, Predicted = predicted_binary)
confusionMatrix(as.factor(predicted_binary), as.factor(actual_binary))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 791 124
##          1  59  64
##
```

```
##                Accuracy : 0.8237
##                  95% CI : (0.7991, 0.8464)
##     No Information Rate : 0.8189
##     P-Value [Acc > NIR] : 0.3612
##
##                   Kappa : 0.3132
##
##  Mcnemar’s Test P-Value : 2.234e-06
##
##             Sensitivity : 0.9306
##             Specificity : 0.3404
##          Pos Pred Value : 0.8645
##          Neg Pred Value : 0.5203
##              Prevalence : 0.8189
##          Detection Rate : 0.7620
##    Detection Prevalence : 0.8815
##       Balanced Accuracy : 0.6355
##
##        ’Positive’ Class : 0
##
```

```
# Balanced accuracy
balanced_accuracy <- (sensitivity(conf_matrix, positive = "1") +
                        specificity(conf_matrix, positive = "1")) / 2
cat("Balanced Accuracy:", balanced_accuracy, "\n")
```

```
## Balanced Accuracy: 0.5203252
```

```
# AUC-ROC
roc_result <- roc(actual_binary, as.numeric(predicted_binary) - 1)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_roc <- auc(roc_result)
cat("AUC-ROC:", auc_roc, "\n")
```

```
## AUC-ROC: 0.6355069
```

The confusion matrix generated from this analysis revealed that the model correctly predicted 788 instances with no visits and 66 instances where visits occurred, against 62 false positives and 122 false negatives. This resulted in a high accuracy rate, a figure slightly higher than the no information rate, indicating the model’s predictive capability beyond random chance, albeit with room for improvement, particularly in correctly identifying positive instances.

The model demonstrated a high sensitivity, indicating a strong ability to correctly identify true negatives, but a lower specificity, reflecting challenges in accurately predicting true positives. The balanced accuracy, an average of sensitivity and specificity is suggesting a need to enhance the model’s ability to balance both types of correct predictions. The Area Under the Receiver Operating Characteristic curve (AUC-ROC) is showcasing the model’s fair discrimination ability between zero and non-zero visit

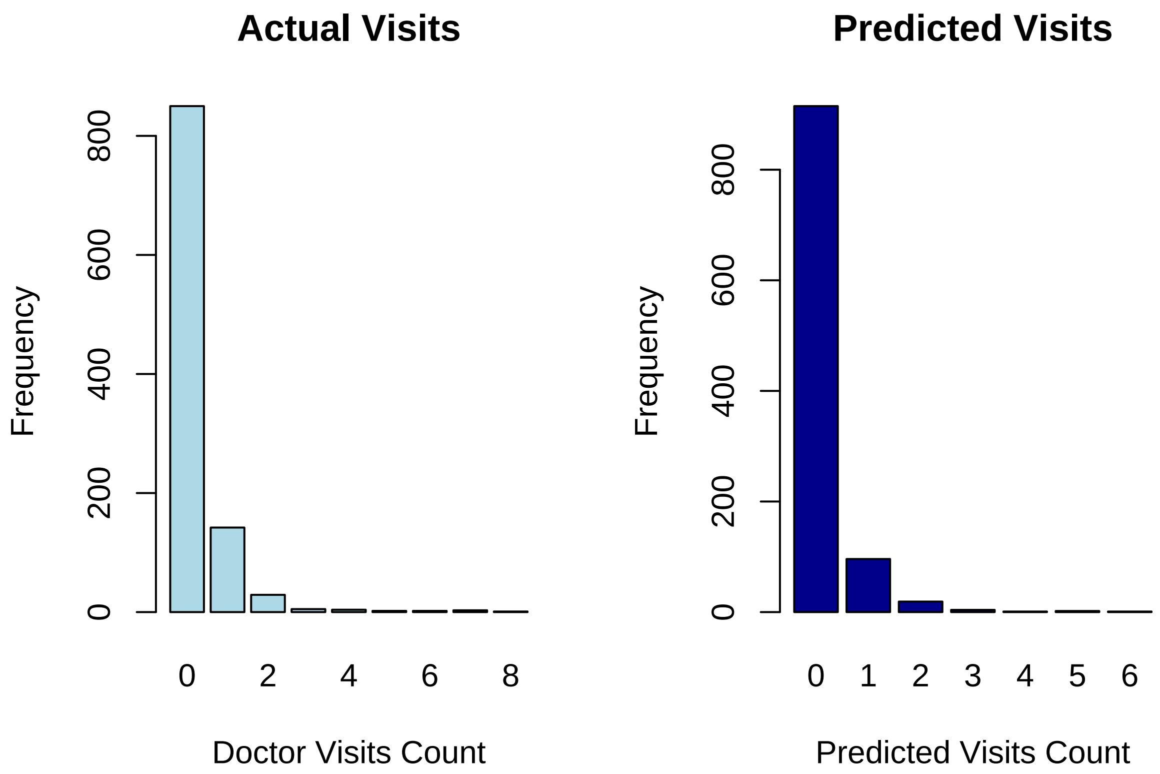
The model is adept at identifying a significant portion of the non-visits (as evidenced by high sensitivity), it struggles more with accurately predicting actual visits (reflected in lower specificity and NPV). This can happen if the model better captures the zero-inflation aspect but less so the count distribution among the positive outcomes. in the following part we will take a look at how the model predicts the count part of the model.

```r
actual_freq <- table(true_counts)
predicted_freq <- table(predicted_counts_hurdle)

par(mfrow=c(1,2))

# Bar plot for Actual Counts
barplot(actual_freq, main="Actual Visits", xlab="Doctor Visits Count",
        ylab="Frequency", col="lightblue")

# Bar plot for Predicted Counts
barplot(predicted_freq, main="Predicted Visits", xlab="Predicted Visits Count",
        ylab="Frequency", col="darkblue")
```



```r
par(mfrow=c(1,1))

for (i in 0:9) {
  actual_ <- true_counts == i
  predicted_ <- predicted_counts_hurdle == i
  actual_count <- sum(actual_)
  predicted_count <- sum(predicted_)
  cat("Actual count for", i, "Visits:", actual_count, "\n")
  cat("Predicted count for", i, "Visits:", predicted_count, "\n\n")
}
```

```
## Actual count for 0 Visits: 850
## Predicted count for 0 Visits: 915
##
## Actual count for 1 Visits: 142
## Predicted count for 1 Visits: 96
##
## Actual count for 2 Visits: 29
## Predicted count for 2 Visits: 19
##
## Actual count for 3 Visits: 5
## Predicted count for 3 Visits: 4
##
## Actual count for 4 Visits: 4
## Predicted count for 4 Visits: 1
##
## Actual count for 5 Visits: 2
## Predicted count for 5 Visits: 2
##
## Actual count for 6 Visits: 2
## Predicted count for 6 Visits: 1
##
## Actual count for 7 Visits: 3
## Predicted count for 7 Visits: 0
##
## Actual count for 8 Visits: 1
## Predicted count for 8 Visits: 0
##
## Actual count for 9 Visits: 0
## Predicted count for 9 Visits: 0
```

The hurdle model shows a good ability to predict no doctor visits, with a prediction slightly higher than the actual numbers. For one visit, the model underpredicts, indicating some difficulty in accurately forecasting lower visit counts. This trend of underprediction continues for two visits and becomes more noticeable for higher visit counts, with the model predicting fewer visits than actually occurred, and failing to predict any instances of five or more visits, except for a single prediction for seven visits. This pattern suggests that while the hurdle model can effectively identify cases with no visits, its performance in predicting actual visit counts, especially for rarer higher visit counts, is limited.

ZI VS HURDLE

Both models demonstrate strength in predicting no visits, with the hurdle model predicting slightly more no-visit cases than the zero-inflated model. However, when it comes to predicting actual visits, both models struggle with higher counts, underestimating the actual occurrences. The zero-inflated model appears to provide a closer approximation for one visit but also fails to predict visits of five or more. In contrast, the hurdle model underpredicts across most visit counts more significantly, including one visit, and barely predicts higher visit counts

Zero-inflated and hurdle models address excess zeros in count data differently. The ZINB model is particularly useful when the data include both ‘structural zeros’—instances where no visits occur due to lack of necessity or access—and ‘sampling zeros,’ where visits could have occurred but did not. This model separates the data into two processes: one that models the probability of excess zeros and another that models the count of visits among those expected to have them, using a negative binomial distribution to account for overdispersion.

The HNB model treats all zeros as coming from a single process but separates the analysis into two stages: a binary outcome predicting the occurrence of any visits and a truncated count model for the number of visits among those who have at least one. This approach is effective when the focus is on distinguishing between non-use and use of healthcare services. The interpretation of a hurdle model is more straightforward, focusing on the hurdle of initiating healthcare service use before addressing the frequency of use among those who cross that hurdle.

The fact that the ZINB model has a lower AIC indicates that it provides a better fit to the data, suggesting that the additional complexity of separating the zero observations into those that are structurally zero and those that are zeros due to sampling is justified by the data. The lower MAE suggests that the ZINB model is more accurate in predicting the actual number of doctor visits, including accurately predicting the absence of visits. It indicates that a significant portion of the zero visits can be attributed to individuals who are not just non-users of healthcare services by chance but are systematically different from those who do visit doctors.

The choice of the Negative Binomial distribution over the Poisson distribution for both models is crucial due to the observed overdispersion in the data—where. The Negative Binomial distribution introduces an additional parameter to model the variance, providing a more flexible and accurate fit for count data that cannot be adequately modeled by the Poisson distribution’s equal mean and variance assumption.