Assignment Topics in Econometrics

Felix Joehnk Matrikel Nr.: 557993

23th December 2018

The Data

When working with count data, there are some special features within the response variable, that differ from a standard linear model. The dependent variable is by construction a discrete, non negative integer. The underlying distribution, and the estmating model have to take that into account. This assignment is coded in R, and created using a markdown document. The following libraries have to be installed and loaded to the workspace, if the results want to be reproduced:

```
library(readxl)
library(pscl)
library(ggplot2)
library(MASS)
library(sandwich)
library(pscl)
library(stargazer)
library(gridExtra)
```

Because the data has been provided in Excel format and has to be used as that, the package 'readxl' has to be loaded. 'Stargazer', 'gridExtra' and 'ggplot2' enables to provide a nicely formated output. The remaining packages provide a neat framework to use a negative binomial distribution, the hurdle and zero inflated models, as well as robust standard errors later on. The response variable of interest is doctoral visits per quarter and I will develop a suited model in order to predict that as good as possible. The histogram of its counts can be seen below and and the summary statistic is shown in Table 1 in the appendix. The variables 'welle' and 'persnr' will be excluded from any of the regressions, since they do not have any explanatory power. 'Doctor Visits per quarter' are a part of 'Doctor Visits per Year'. Therefore, the variable 'doctor visits per year' might cause endogeneity issues. Also during the Maximum Likelihood estimation of the hurdle and the zero inflated model this variable causes the algorithm to reach singularity. All this lead to the decision to exclude this variable from any of the regressions as well. Most of the remaining variables are dummys, so the mean and standard deviations do not provide meaningful information. Age and education are denoted in years / 100 and the income in income / 100,000. As can be seen from the descriptive statistics and the histogram, the doctor visits per quarter have a higher unconditional standard deviation than the unconditional mean. The counts are also skewed to the left side. The Counts for 0 - 3 account for 75% of the total counts.

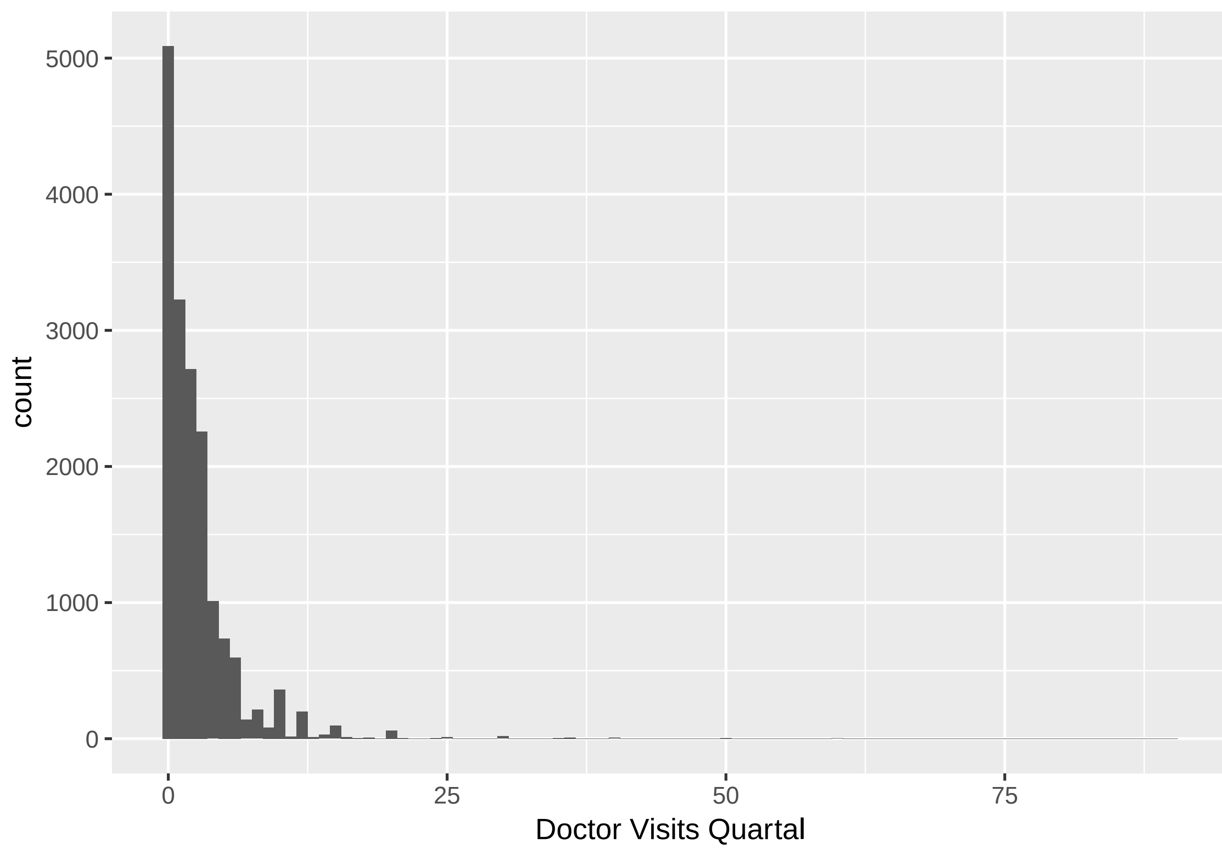
```
setwd("C:/Felix Ordner/Uni/Master/4. Fachsemester/Topics in Econometrics/Graded Assignment")
my_data = read_excel("Welle2005.xls")
my_data = as.data.frame(my_data)
colnames(my_data)[18] = "income"
table(my_data$DocVisits_Quartal)/length(my_data$DocVisits_Quartal)
```

```
##
##            0            1            2            3            4
## 3.002183e-01 1.903722e-01 1.602855e-01 1.332075e-01 5.958351e-02
##            5            6            7            8            9
## 4.341927e-02 3.521916e-02 8.200106e-03 1.256563e-02 4.896466e-03
##           10           11           12           13           14
## 2.135567e-02 1.002891e-03 1.179871e-02 7.669164e-04 1.710814e-03
```

```
##            15           16           17           18           19
## 5.604389e-03 7.079228e-04 2.949678e-04 4.719486e-04 5.899357e-05
##            20           21           22           23           24
## 3.480621e-03 2.359743e-04 1.179871e-04 5.899357e-05 3.539614e-04
##            25           26           28           30           33
## 6.489293e-04 1.179871e-04 5.899357e-05 1.061884e-03 5.899357e-05
##            35           36           40           45           50
## 2.359743e-04 5.309421e-04 4.129550e-04 5.899357e-05 3.539614e-04
##            56           60           70           80           90
## 5.899357e-05 1.769807e-04 1.179871e-04 5.899357e-05 5.899357e-05
```

```
#a =  table(my_data$DocVisits_Quartal)
#count_table = rbind(as.numeric(names(a)),a)
#rownames(count_table) = c("Doctor Visits per Quarter","Counts")
#count_table = as.data.frame(count_table)
#count_table
```

```
ggplot(my_data, aes(as.numeric(DocVisits_Quartal))) + geom_histogram(binwidth = 1) +
  xlab("Doctor Visits Quartal")
```



The Poisson Model

The standard distribution for a count model development is the Poisson Distribution, which is a discrete, non negative distribution with one parameter μ, that respresents all its moments. The probability mass function (pmf) looks as follows:

$$Pr[Y = y] = \frac{e^{-\mu}\mu^y}{y!}$$

The assumption that the mean as well as the variance are defined to be μ is very resricted. It assumes equidispersition. To Test for equidispersition I will regress the predicted $\hat{\mu}$ on y^*, where

$$y_i^* = \frac{(y_i - \hat{\mu}_i)^2 - y_i}{\hat{\mu}_i}$$

```
po_reg = glm(DocVisits_Quartal ~ .,
             data = subset(my_data, select=c(-persnr, -welle, -DocVisits_jährlich)), family="poisson")
mu_hat_Po = predict(po_reg, my_data, type = "response")
y_star = ((my_data$DocVisits_Quartal - mu_hat_Po)^2- my_data$DocVisits_Quartal) / mu_hat_Po
```

The regression output for the poisson regression can be seen in Table 2. The parameters can be interpreted straight forward: Being a foreigner increases the doctor visits per quarter on average by the factor of exp(0.085) ~ 1.09 keeping all other regressors constant. Being self employed (selbstständig) decreases the count on average by the factor of exp(0.12) ~ 1.13 keeping all other regressors fixed. The latter makes sence because when being self employed one is not covered under government insurance and a private insurance tends to be more expensive the more doctor visits are performed. The test for equidispersition can be seen below in Table 3. The Nullhypothesis of equidispersition can clearly be rejected. As long as the mean function is correctly specified, the ML Estimator will provide consistent estimators. I will therefore correct the standard errors as follows:

```
po_reg_robust = po_reg
cov_mat = vcovHC(po_reg_robust, type="HC0")
std_err = sqrt(diag(cov_mat))
po_reg_robust = cbind(Estimate= coef(po_reg_robust), "Robust SE" = std_err,
               "Pr(>|z|)" = round(2 * pnorm(abs(coef(po_reg_robust)/std_err),
                                            lower.tail=FALSE), digits = 3),
               LL = coef(po_reg_robust) - 1.96 * std_err,
               UL = coef(po_reg_robust) + 1.96 * std_err)
```

The Negative Binomial Model

To relax the very restricted poisson model, and allow for a more flexible model I will introduce the Negative Binomial distribution, in the following just called negbin. Its pmf is defined as follows:

$$h[y|\mu, \alpha] = \frac{\Gamma(\alpha^{-1} + y)}{\Gamma(\alpha^{-1})\Gamma(y+1)}(\frac{\alpha^{-1}}{\alpha^{-1} + \mu})^{\alpha^{-1}}(\frac{\mu}{\mu + \alpha^{-1}})^y$$

The mean is still μ but its variance is $\mu + \alpha\mu^2$. Because $\alpha > 0$ the variance is larger than the mean and hence, matches the summary statistics of the data intuitively better. The negbin regression will be estimated using ML as well. Its regression output can be seen in Table 2.

```
negbin_reg = glm.nb(DocVisits_Quartal ~ .,
                    data = subset(my_data, select=c(-persnr, -welle, -DocVisits_jährlich)))
```

Being married increases the count on average by the factor of exp(0.043) ~ 1.04. Being out of a job (arbeitslos) and income do not seem to be significant in the negbin model as well as the Poisson and the robust Poisson model. Consequently, I will test a restricted negbin model without those two regressors and the unrestricted negbin model and perform a Likelihood Ration (LR) test. This is the easiest statistic to compute for this purpose and possible because the restricted is nested within the unrestricted one. The result will be χ^2

distributed with 2 degrees of freedom.

```
restr_negbin_reg = glm.nb(DocVisits_Quartal ~ .,
                          data = subset(my_data, select=c(-persnr, -welle, -Arbeitslos,
                                                           -income, -DocVisits_jährlich)))
pchisq(2*(logLik(negbin_reg) - logLik(restr_negbin_reg)), df = 2)
```

```
## 'log Lik.' 0.177358 (df=17)
```

The null cannot be rejected and therefore there is no significant difference in explanatory power between the restricted and the unrestricted model. Since the restricted is more parsimonious, it will also be considered in the following models.

The Hurdle model

In order to account for the large number of zeros in the count (5089 zeros) a more flexible model approach might be usefull. Therefore, the so called hurdle model will be estimated. In the hurdle model it will be assumed, that the data generating process (gdp) does not need to be the same for the whole data, but split into a dgp for the zeros, and one for all observations greater than 0. Depending, if $f_2(0)$ and $f_1(0)$ are defined as the poisson or the negbin distribution, it will be a hurdle poisson or a hurdle negbin model.

$$g(y) = \begin{cases} f_1(0) & y = 0 \\ \frac{1-f_1(0)}{1-f_2(0)} f_2(0) & y \geq 1 \\ \end{cases}$$

```
hurdle_po_reg = hurdle(DocVisits_Quartal ~ Geschlecht+Alter+Alter_quadr+Bildung+Ausländer+
                         SP_Region+Vollzeit+Teilzeit+Gute_Gesundheit+Haushalt+Selbstständig+
                         Schlechte_Gesundheit+Verheiratet,
                       data = my_data, dist = "poisson", zero.dist = "binomial")
hurdle_negbin_reg = hurdle(DocVisits_Quartal ~ Geschlecht+Alter+Alter_quadr+Bildung+Ausländer+
                             SP_Region+Vollzeit+Teilzeit+Gute_Gesundheit+Haushalt+Selbstständig+
                             Schlechte_Gesundheit+Verheiratet,
                           data = my_data, dist = "negbin", zero.dist = "binomial")
```

Unfortunately, stargazer has trouble showing all the results of the regression in a nicely formated way. Because of that and because of space saving purposes the output is shown in the standard r summary style in the end of the appendix. Nevetheless, the results are the same. The Idea of splitting the data in a dgp for the zeros and a seperate one for the observations greater than zero and the sequential estimation of the two processes also provides a nice interpretation for the actual situation in germany in 2005. The so called "Praxisgebühr", or doctor fee that had to be paid for the first doctor visit in a new quarter indicates two differently motivated decision making processes according to doctor visits within a quarter. For the zeros one might not be willing to pay the 10 Euro. Once paid after the first visit, it doesnt matter anymore and one might be willing to go to the doctor more often since the quarter is paid already.

```
ZI_po_reg = zeroinfl(DocVisits_Quartal ~ Geschlecht+Alter+Alter_quadr+Bildung+Ausländer+
                         SP_Region+Vollzeit+Teilzeit+Gute_Gesundheit+Haushalt+Selbstständig+
                         Schlechte_Gesundheit+Verheiratet,
                       data = my_data, dist = "poisson", zero.dist = "binomial")
```

```
## Warning in optim(fn = loglikfun, gr = gradfun, par = c(start$count,
## start$zero, : unknown names in control: zero.dist
```

```r
ZI_NB_reg = zeroinfl(DocVisits_Quartal ~ Geschlecht+Alter+Alter_quadr+Bildung+Ausländer+
                              SP_Region+Vollzeit+Teilzeit+Gute_Gesundheit+Haushalt+Selbstständig+
                              Schlechte_Gesundheit+Verheiratet,
                            data = my_data, dist = "negbin", zero.dist = "binomial")
```

```
## Warning in optim(fn = loglikfun, gr = gradfun, par = c(start$count,
## start$zero, : unknown names in control: zero.dist
```

The Zero inflated model

The underlying motivation for this type of model specification is the same as for the hurdle type of model. Here $f_1(0)$ is defined as a binary model, more specific a logit model and $f_2(0)$ is defined as either the poisson or the negbin model. For the same reasons as for the hurdle model, the regression results are shown in the standard r summary style in the end of the appendix.

$$g(y) = \begin{cases} f_1(0) + (1 - f_0(0))f_2(0) & y = 0 \\ (1 - f_1(0))f_2(y) & y \geq 1 \\ \end{cases}$$

Model Selection

The model selection will be depending on the following comparison criteria:

- squared correclation
- Akaike information criteria
- mean and standard deviation of the pearson residuals
- predicted probabilities

To be able to start, the $\hat{\mu}_i$ have to be predicted. Using the same data that has been used for estimating the model to test the model might not be the best strategy, but methods like cross validation are beyond the scope of this assignment.

```r
mu_hat_Po = predict(po_reg, my_data, type = "response")
mu_hat_NB = predict(negbin_reg, my_data, type = "response")
mu_hat_hPo = predict(hurdle_po_reg, my_data, type = "response")
mu_hat_hNB = predict(hurdle_negbin_reg, my_data, type = "response")
mu_hat_ZIPo = predict(ZI_po_reg, my_data, type = "response")
mu_hat_ZINB = predict(ZI_NB_reg, my_data, type = "response")
```

Squared Correclation

This measure is in the fashion of a R-squared measure. The higher the squared correlation of the specific model, the better does the model explain the $\hat{\mu}_i$.

```r
cor(mu_hat_Po, my_data$DocVisits_Quartal)^2
```

```
## [1] 0.1347888
```

```r
cor(mu_hat_NB, my_data$DocVisits_Quartal)^2
```

```
## [1] 0.1332953
```

```r
cor(mu_hat_hPo, my_data$DocVisits_Quartal)^2
```

```
## [1] 0.1363375
```

```
cor(mu_hat_hNB, my_data$DocVisits_Quartal)^2
```

```
## [1] 0.1358048
```

```
cor(mu_hat_ZIPo, my_data$DocVisits_Quartal)^2
```

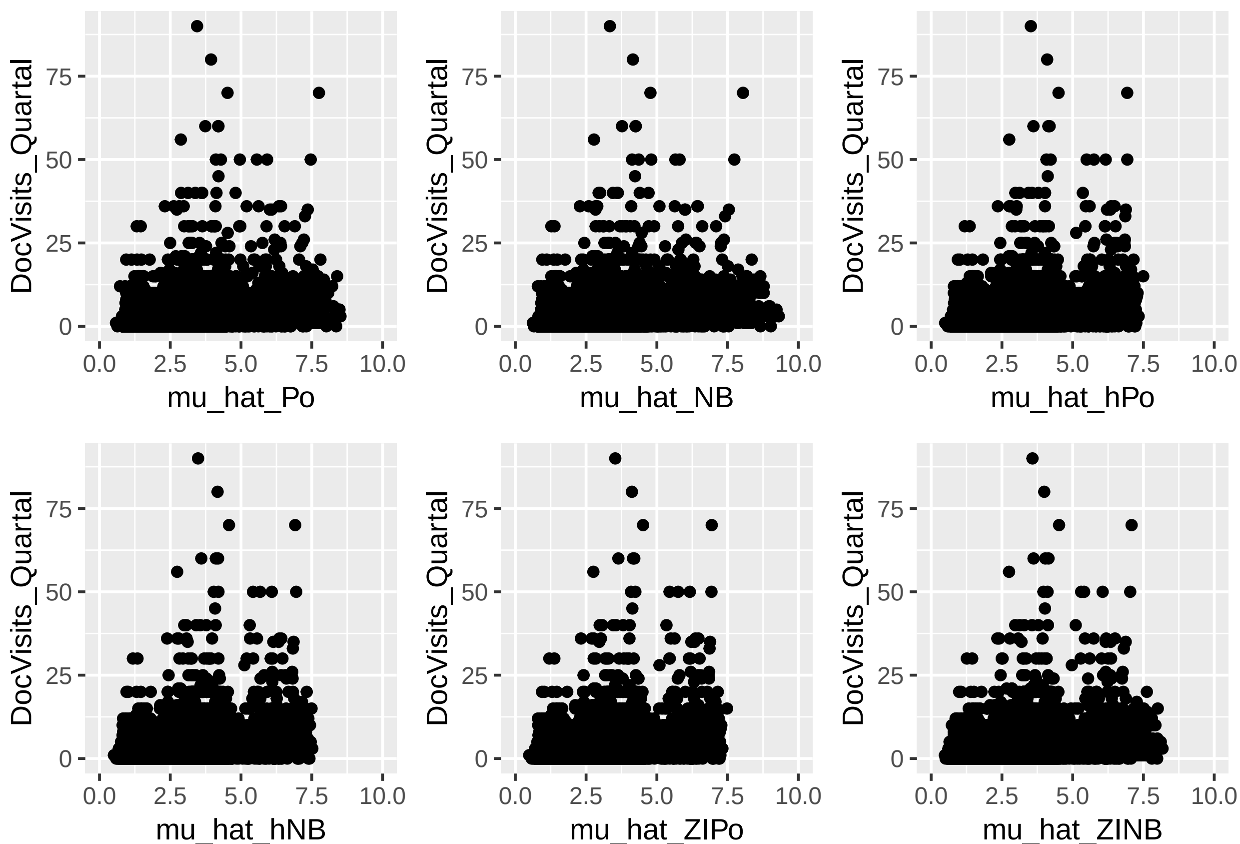
```
## [1] 0.1362667
```

```
cor(mu_hat_ZINB, my_data$DocVisits_Quartal)^2
```

```
## [1] 0.1351203
```

The squared correlations are all around the same magnitude, ~13%. The correlograms can be seen below for additional viual assessment:

```
p1 =ggplot(my_data, aes(x=mu_hat_Po, y=DocVisits_Quartal)) + geom_point() + xlim(0, 10) #+ geom_smooth(
p2 = ggplot(my_data, aes(x=mu_hat_NB, y=DocVisits_Quartal)) + geom_point() + xlim(0, 10) #+ geom_smooth
p3 = ggplot(my_data, aes(x=mu_hat_hPo, y=DocVisits_Quartal)) + geom_point() + xlim(0, 10) #+ geom_smoot
p4 = ggplot(my_data, aes(x=mu_hat_hNB, y=DocVisits_Quartal)) + geom_point() + xlim(0, 10) #+ geom_smoot
p5 = ggplot(my_data, aes(x=mu_hat_ZIPo, y=DocVisits_Quartal)) + geom_point() + xlim(0, 10) #+ geom_smoo
p6 = ggplot(my_data, aes(x=mu_hat_ZINB, y=DocVisits_Quartal)) + geom_point() + xlim(0, 10) #+ geom_smoo
grid.arrange(p1, p2, p3, p4, p5, p6, ncol=3)
```



Akaike Information Criteria

The AIC is a measure to find the most parsimonious model in terms of explanatory power, defined by the log likelihood of the regression and penalty term for each parameter used. The lower, the better the model fit.

```
po_reg$aic
```

```
## [1] 84877.32
```

```
negbin_reg$aic
```

```
## [1] 67621.64
```

```
AIC(hurdle_po_reg)
```

```
## [1] 79114.74
```

```
AIC(hurdle_negbin_reg)
```

```
## [1] 67131.45
```

```
AIC(ZI_po_reg)
```

```
## [1] 79112.67
```

```
AIC(ZI_NB_reg)
```

```
## [1] 67373.34
```

From the results above it can be concluded, that in general the negbin models perform better than the poisson models. The hurdle negbin model seems to have the best fit.

Residual Tests

The fit of a model can also be tested by having a look at the residuals, in this case the pearson residuals. These are standardized and shoud be, if the correct model have been applied, standard normal distributed with a mean of 0 and a standard deviation of 1.

```
residuals_Po = (mu_hat_Po - my_data$DocVisits_Quartal)/sqrt(mu_hat_Po)
residuals_NB2 = (mu_hat_NB - my_data$DocVisits_Quartal)/sqrt(mu_hat_NB)
residuals_hPo = (mu_hat_hPo  - my_data$DocVisits_Quartal)/sqrt(mu_hat_hPo)
residuals_hNB = (mu_hat_hNB  - my_data$DocVisits_Quartal)/sqrt(mu_hat_hNB)
residuals_ZIpo = (mu_hat_ZIPo  - my_data$DocVisits_Quartal)/sqrt(mu_hat_ZIPo)
residuals_ZINB = (mu_hat_ZINB  - my_data$DocVisits_Quartal)/sqrt(mu_hat_ZINB)

mean(residuals_Po)
```

```
## [1] 0.0009016293
```

```
sqrt(var(residuals_Po))
```

```
## [1] 2.065786
```

```
mean(residuals_NB2)
```

```
## [1] 0.001357413
```

```
sqrt(var(residuals_NB2))
```

```
## [1] 2.067487
```

```
mean(residuals_hPo)
```

```
## [1] -0.0002555015
```

```
sqrt(var(residuals_hPo))
```

```
## [1] 2.067235
```

```
mean(residuals_hNB)
```

```
## [1] -0.0002306947
```

```
sqrt(var(residuals_hNB))
```

```
## [1] 2.067774
```

```
mean(residuals_ZIpo)
```

```
## [1] -0.0001739079
```

```
sqrt(var(residuals_ZIpo))
```

```
## [1] 2.067249
```

```
mean(residuals_ZINB)
```

```
## [1] -0.0006455634
```

```
sqrt(var(residuals_ZINB))
```

```
## [1] 2.070684
```

From the pearson residuals no clear favourite can be chosen. All models seem to have similar filter abilities. What can be concluded is, that the mean is fitted quite well with values that all are approximately 0. The variance is not captured perfectly, but that is very difficult in general by the nature of the count data.

Predicted Probabilities

Maybe the most important criteria to chose the best model is the predicted probability, since this determines how well can the model forecast the count from the data. The predicted probability will be compared to the relative occurency of the counts in the beginning of the paper. The predicted counts are shown from 0 to the maximum count for the respective model. The predicted probability is calculated as the predicted count / total counts and printed below the count. The overlaid histogram's provide a nice visual assessment, how good the models predict the counts.

```
pred_prob_pois = rpois(n = length(mu_hat_Po), mu_hat_Po)
table(pred_prob_pois)/length(mu_hat_Po)
```

```
## pred_prob_pois
##            0            1            2            3            4
## 1.510825e-01 2.300159e-01 2.110790e-01 1.456551e-01 1.035927e-01
##            5            6            7            8            9
## 6.459796e-02 4.105952e-02 2.353843e-02 1.480739e-02 5.899357e-03
##           10           11           12           13           14
## 4.365524e-03 2.182762e-03 9.438971e-04 5.899357e-04 2.949678e-04
##           15           16           17
## 1.769807e-04 5.899357e-05 5.899357e-05
```

```
pred_prob_NB = rnegbin(n = length(mu_hat_NB), mu = mu_hat_NB, theta = negbin_reg$theta)
table(pred_prob_NB)/length(mu_hat_NB)
```

```
## pred_prob_NB
##            0            1            2            3            4
## 3.014571e-01 2.157395e-01 1.456551e-01 9.728040e-02 6.607280e-02
##            5            6            7            8            9
## 4.459914e-02 3.120760e-02 2.347944e-02 1.699015e-02 1.291959e-02
##           10           11           12           13           14
## 9.792933e-03 7.374196e-03 5.368415e-03 4.247537e-03 3.244646e-03
```

```
##           15           16           17           18           19
## 3.008672e-03 2.300749e-03 2.064775e-03 1.356852e-03 1.002891e-03
##           20           21           22           23           24
## 7.669164e-04 6.489293e-04 8.259100e-04 6.489293e-04 1.179871e-04
##           25           26           27           28           29
## 4.719486e-04 3.539614e-04 2.949678e-04 5.899357e-05 1.179871e-04
##           30           31           33           35           36
## 5.899357e-05 1.179871e-04 5.899357e-05 5.899357e-05 5.899357e-05
##           38           46           47
## 5.899357e-05 5.899357e-05 5.899357e-05
```

```
pred_prob_hPo = rpois(n = length(mu_hat_hPo), mu_hat_hPo)
table(pred_prob_hPo)/length(pred_prob_hPo)
```

```
## pred_prob_hPo
##            0            1            2            3            4
## 1.547401e-01 2.265353e-01 1.980414e-01 1.522624e-01 1.076043e-01
##            5            6            7            8            9
## 6.813757e-02 3.899475e-02 2.483629e-02 1.392248e-02 7.256209e-03
##           10           11           12           13           14
## 3.775588e-03 1.828801e-03 7.669164e-04 6.489293e-04 3.539614e-04
##           16           19
## 2.359743e-04 5.899357e-05
```

```
pred_prob_hNB = rnegbin(n = length(mu_hat_hNB), mu = mu_hat_hNB, theta = hurdle_negbin_reg$theta)
table(pred_prob_hNB)/length(mu_hat_hNB)
```

```
## pred_prob_hNB
##            0            1            2            3            4
## 3.241697e-01 2.095452e-01 1.323226e-01 9.002419e-02 6.105834e-02
##            5            6            7            8            9
## 4.418618e-02 2.961477e-02 2.436434e-02 1.905492e-02 1.297859e-02
##           10           11           12           13           14
## 1.091381e-02 9.320984e-03 6.961241e-03 4.601498e-03 3.716595e-03
##           15           16           17           18           19
## 3.421627e-03 3.126659e-03 2.182762e-03 1.238865e-03 1.120878e-03
##           20           21           22           23           24
## 1.179871e-03 1.179871e-03 6.489293e-04 3.539614e-04 4.129550e-04
##           25           26           27           28           29
## 3.539614e-04 4.129550e-04 2.949678e-04 2.949678e-04 1.179871e-04
##           30           31           32           34           36
## 2.949678e-04 1.179871e-04 5.899357e-05 5.899357e-05 1.179871e-04
##           38           39           43
## 5.899357e-05 5.899357e-05 5.899357e-05
```

```
pred_prob_ZIPo = rpois(n = length(mu_hat_ZIPo), mu_hat_ZIPo)
table(pred_prob_ZIPo)/length(pred_prob_ZIPo)
```

```
## pred_prob_ZIPo
##            0            1            2            3            4
## 0.1546221462 0.2248244941 0.1989853106 0.1542091912 0.1055984898
##            5            6            7            8            9
## 0.0653058817 0.0416494602 0.0256622028 0.0138044953 0.0072562091
##           10           11           12           13           14
## 0.0042475370 0.0017108135 0.0007669164 0.0005899357 0.0002949678
##           15           16           17
```

```
## 0.0002359743 0.0001179871 0.0001179871
```

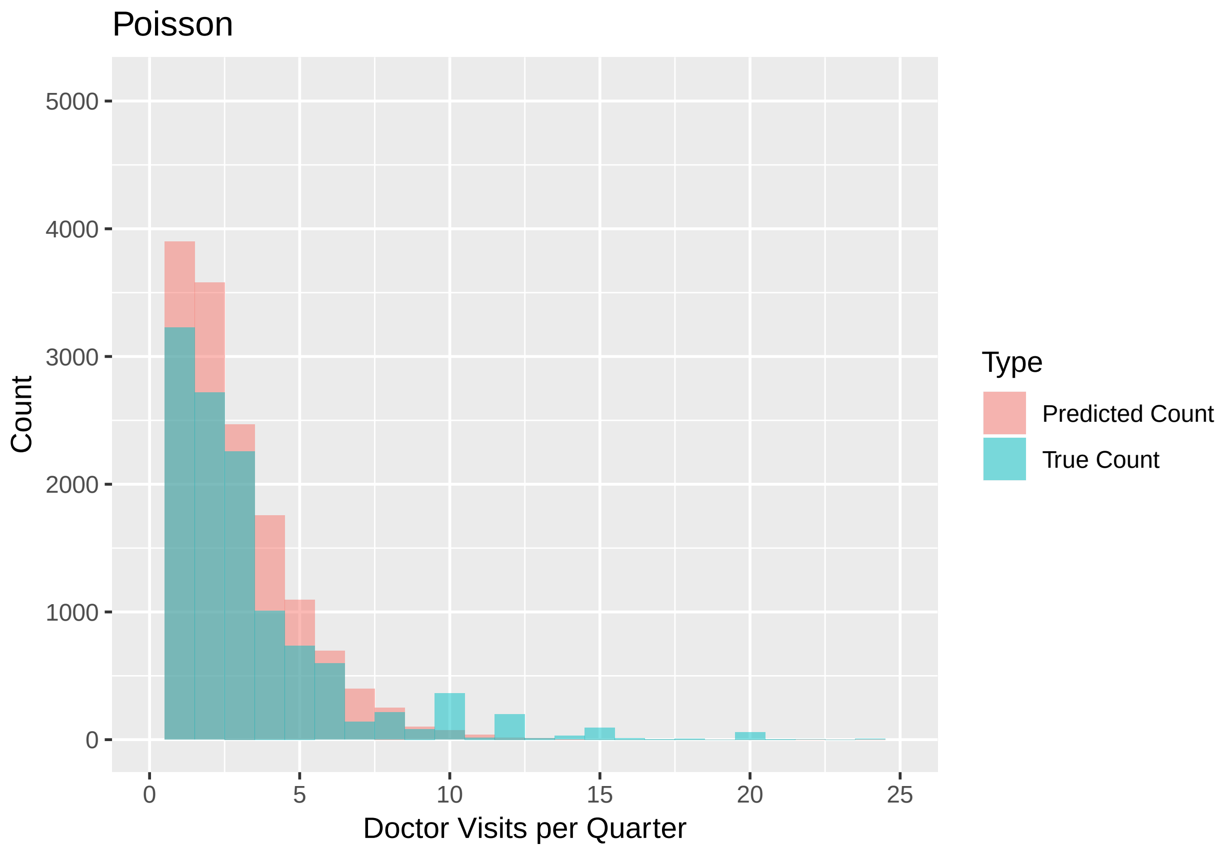
```r
pred_prob_ZINB = rnegbin(n = length(mu_hat_ZINB), mu = mu_hat_ZINB, theta = ZI_NB_reg$theta)
table(pred_prob_ZINB)/length(pred_prob_ZINB)
```

```
## pred_prob_ZINB
##            0            1            2            3            4
## 2.831691e-01 2.204590e-01 1.494307e-01 1.020589e-01 7.238511e-02
##            5            6            7            8            9
## 4.465813e-02 3.457023e-02 2.342045e-02 1.722612e-02 1.350953e-02
##           10           11           12           13           14
## 9.615952e-03 7.728158e-03 5.368415e-03 3.539614e-03 2.359743e-03
##           15           16           17           18           19
## 2.182762e-03 2.418736e-03 1.179871e-03 8.259100e-04 6.489293e-04
##           20           21           22           23           24
## 6.489293e-04 5.309421e-04 4.719486e-04 2.949678e-04 1.769807e-04
##           25           26           27           28           29
## 5.899357e-05 5.899357e-05 1.179871e-04 3.539614e-04 1.179871e-04
##           30           34           41           42
## 1.769807e-04 5.899357e-05 1.179871e-04 5.899357e-05
```

```r
datacount = as.data.frame(my_data$DocVisits_Quartal)
colnames(datacount)[1] = 'Count'
datacount$Type = 'True Count'
predcount_po = as.data.frame(pred_prob_pois)
colnames(predcount_po)[1] = 'Count'
predcount_po$Type = 'Predicted Count'
plotdata_po = as.data.frame(rbind(datacount, predcount_po))
predcount_NB = as.data.frame(pred_prob_NB)
colnames(predcount_NB)[1] = 'Count'
predcount_NB$Type = 'Predicted Count'
plotdata_NB = as.data.frame(rbind(datacount, predcount_NB))
predcount_hPo = as.data.frame(pred_prob_hPo)
colnames(predcount_hPo)[1] = 'Count'
predcount_hPo$Type = 'Predicted Count'
plotdata_hPo = as.data.frame(rbind(datacount, predcount_hPo))
predcount_hNB = as.data.frame(pred_prob_hNB)
colnames(predcount_hNB)[1] = 'Count'
predcount_hNB$Type = 'Predicted Count'
plotdata_hNB = as.data.frame(rbind(datacount, predcount_hNB))
predcount_ZIPo = as.data.frame(pred_prob_ZIPo)
colnames(predcount_ZIPo)[1] = 'Count'
predcount_ZIPo$Type = 'Predicted Count'
plotdata_ZIPo = as.data.frame(rbind(datacount, predcount_ZIPo))
predcount_ZINB = as.data.frame(pred_prob_ZINB)
colnames(predcount_ZINB)[1] = 'Count'
predcount_ZINB$Type = 'Predicted Count'
plotdata_ZINB = as.data.frame(rbind(datacount, predcount_ZINB))

ggplot(plotdata_po, aes(Count, fill = Type)) +
  geom_histogram(binwidth = 1,alpha = .5, position="identity") +
  xlim(0, 25) + ylab("Count") + ggtitle("Poisson") +
  xlab("Doctor Visits per Quarter")
```

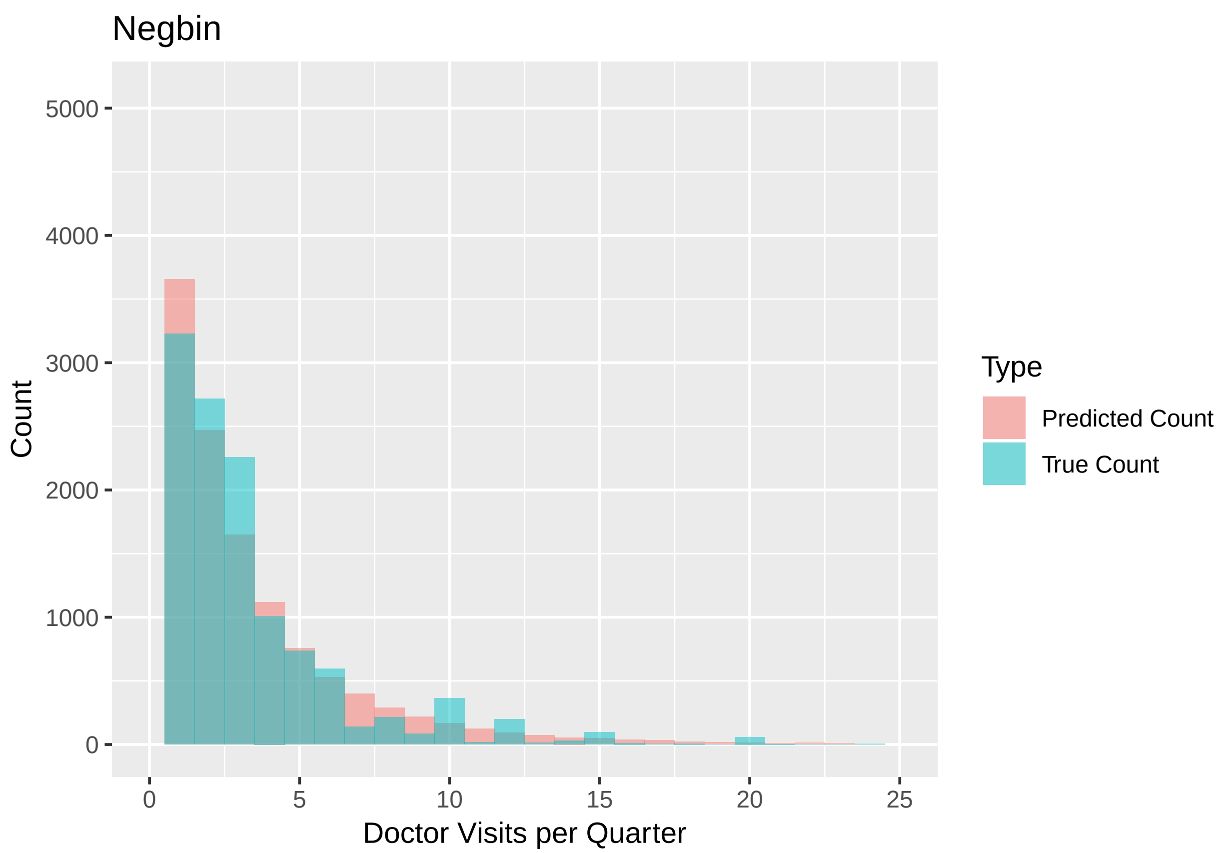
```
## Warning: Removed 57 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```



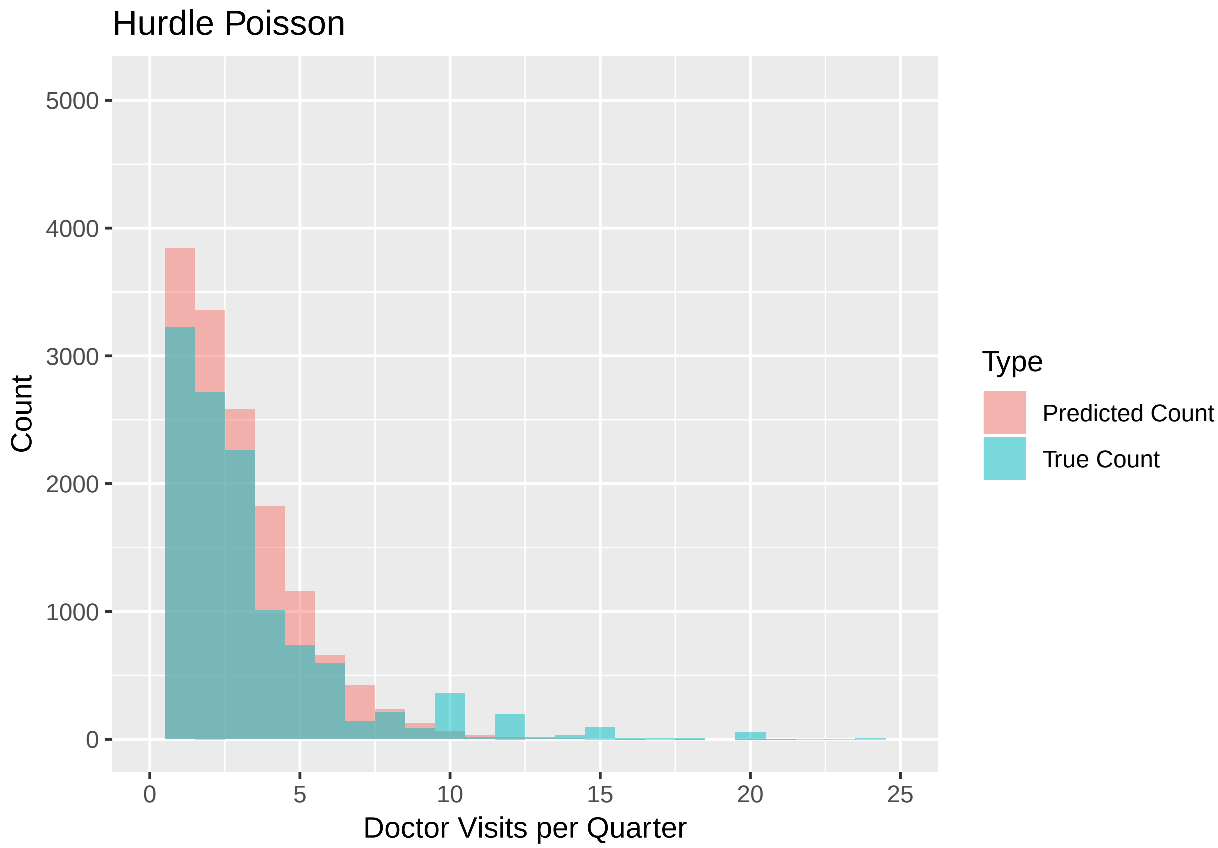

```r
ggplot(plotdata_NB, aes(Count, fill = Type)) +
  geom_histogram(binwidth = 1,alpha = .5, position="identity") +
  xlim(0, 25) + ylab("Count") + ggtitle("Negbin") +
  xlab("Doctor Visits per Quarter")
```

```
## Warning: Removed 80 rows containing non-finite values (stat_bin).

## Warning: Removed 4 rows containing missing values (geom_bar).
```



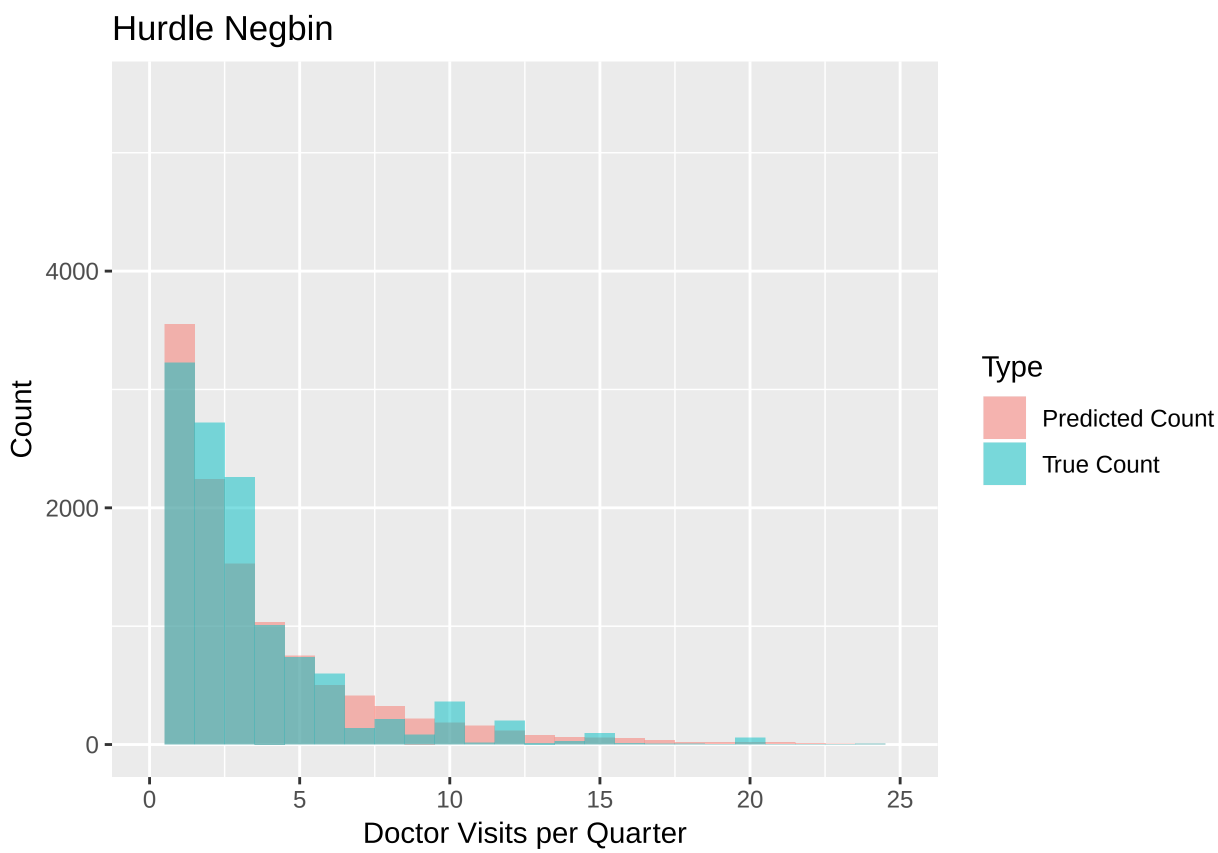

```
ggplot(plotdata_hPo, aes(Count, fill = Type)) +
  geom_histogram(binwidth = 1,alpha = .5, position="identity") +
  xlim(0, 25) + ylab("Count") + ggtitle("Hurdle Poisson") +
  xlab("Doctor Visits per Quarter")
```

```
## Warning: Removed 57 rows containing non-finite values (stat_bin).

## Warning: Removed 4 rows containing missing values (geom_bar).
```



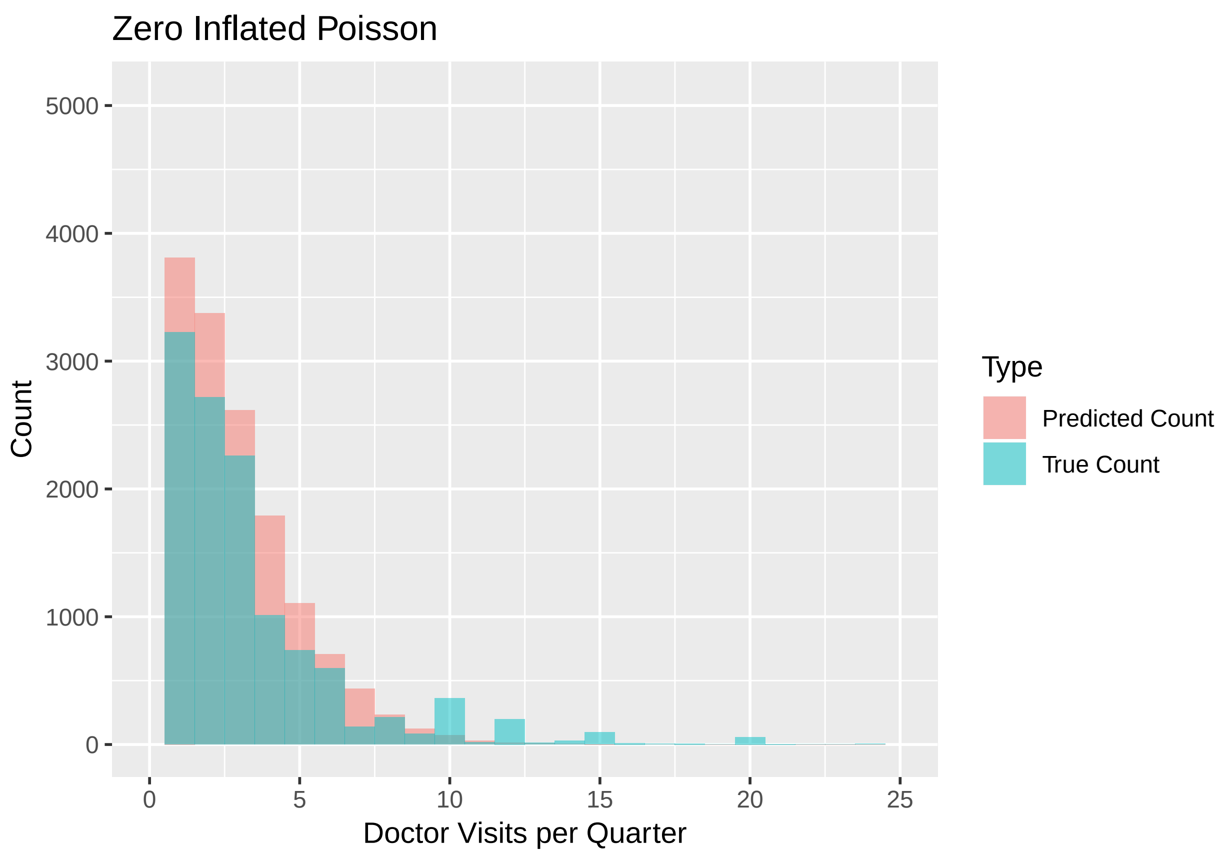

```r
ggplot(plotdata_hNB, aes(Count, fill = Type)) +
  geom_histogram(binwidth = 1,alpha = .5, position="identity") +
  xlim(0, 25) + ylab("Count") + ggtitle("Hurdle Negbin") +
  xlab("Doctor Visits per Quarter")
```

```
## Warning: Removed 90 rows containing non-finite values (stat_bin).

## Warning: Removed 4 rows containing missing values (geom_bar).
```



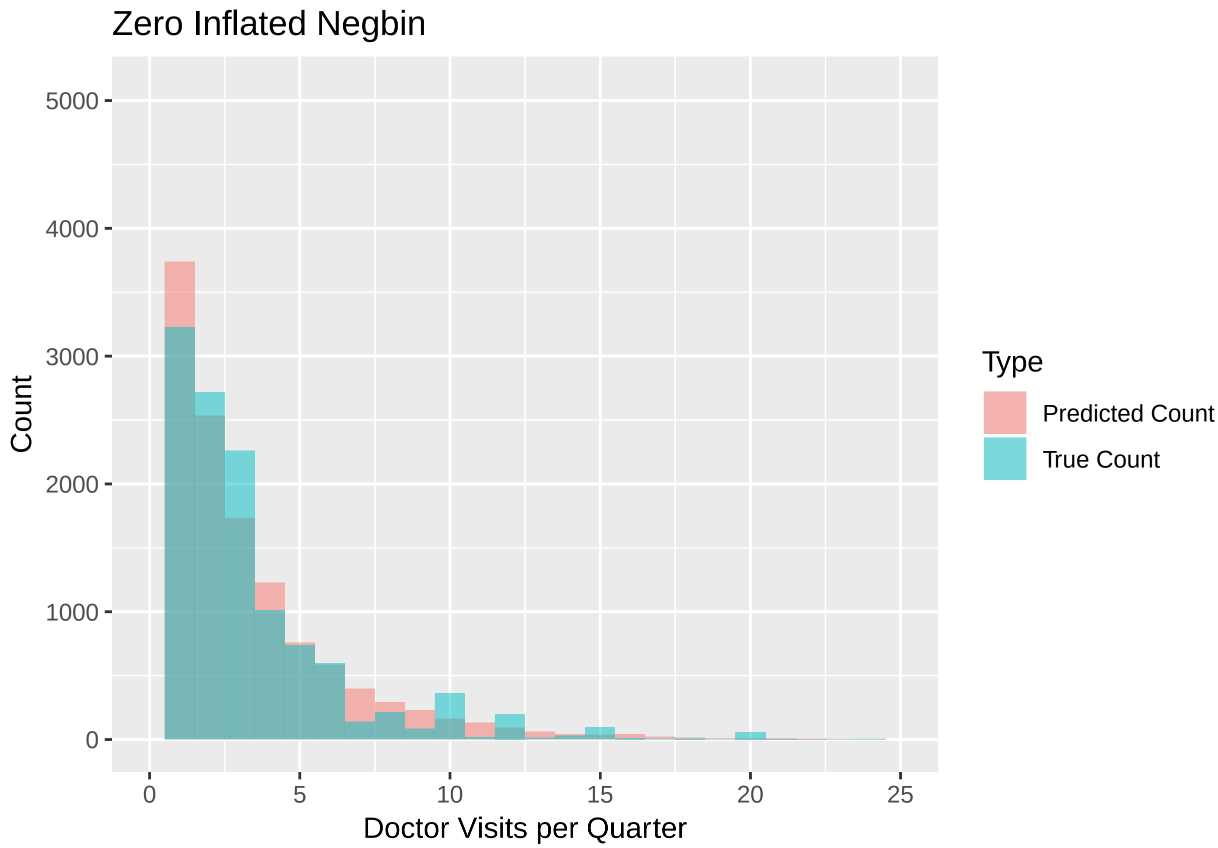

```
ggplot(plotdata_ZIPo, aes(Count, fill = Type)) +
  geom_histogram(binwidth = 1,alpha = .5, position="identity") +
  xlim(0, 25) + ylab("Count") + ggtitle("Zero Inflated Poisson") +
  xlab("Doctor Visits per Quarter")
```

```
## Warning: Removed 57 rows containing non-finite values (stat_bin).

## Warning: Removed 4 rows containing missing values (geom_bar).
```




```r
ggplot(plotdata_ZINB, aes(Count, fill = Type)) +
  geom_histogram(binwidth = 1,alpha = .5, position="identity") +
  xlim(0, 25) + ylab("Count") + ggtitle("Zero Inflated Negbin") +
  xlab("Doctor Visits per Quarter")
```

```
## Warning: Removed 75 rows containing non-finite values (stat_bin).

## Warning: Removed 4 rows containing missing values (geom_bar).
```



Again, the best models in general seem to be the negbin models. They predict especially the zeros much more precise than the poisson type of models. Among the negbin models, the Hurdle Negbin model predicts the zero's and the one's best. Although the Zero Inflated Negbin model predicts the two's a little better, there are much more one's than two's in the data. Hence, the lower the count, the more its predicted probability contributes to the overall predicted probability of the model.

Conclusion and Model Interpretation

I find the Hurdle Negbin model the best to fit among the models tested in this paper. The AIC as well as the predicted probability are in favour of this model. In terms of the residuals and the squared correlation all models seem to have an equal fit. I will calculate the marginal effect, more specific the Marginal Effect of an average person for the hurdle negbin model for the zeros.

```
## average regressor
x = c(1, mean(my_data$Geschlecht),
      mean(my_data$Alter),mean(my_data$Alter_quadr),mean(my_data$Bildung),
      mean(my_data$Ausländer),mean(my_data$SP_Region),mean(my_data$Vollzeit),
      mean(my_data$Teilzeit),mean(my_data$Gute_Gesundheit),mean(my_data$Haushalt),
      mean(my_data$Selbstständig), mean(my_data$Schlechte_Gesundheit),mean(my_data$Verheiratet))
## coefficients
b = coef(hurdle_negbin_reg, model = "zero")
## linear predictor x'b
xb = sum(x * b)
## marginal effects
dlogis(xb) * b["Geschlecht"]
```

```
## Geschlecht
```

```
## -0.1193157
```

```r
dlogis(xb) * b["Alter"]
```

```
##     Alter
## -1.026473
```

```r
dlogis(xb) * b["Alter_quadr"]
```

```
## Alter_quadr
##     1.37489
```

```r
dlogis(xb) * b["Bildung"]
```

```
##   Bildung
## 0.6815021
```

```r
dlogis(xb) * b["Ausländer"]
```

```
##  Ausländer
## 0.02404231
```

```r
dlogis(xb) * b["SP_Region"]
```

```
##   SP_Region
## -0.02077145
```

```r
dlogis(xb) * b["Vollzeit"]
```

```
##    Vollzeit
## -0.01840058
```

```r
dlogis(xb) * b["Teilzeit"]
```

```
##    Teilzeit
## -0.01523057
```

```r
dlogis(xb) * b["Gute_Gesundheit"]
```

```
## Gute_Gesundheit
##       0.1846582
```

```r
dlogis(xb) * b["Haushalt"]
```

```
##    Haushalt
## -0.01262889
```

```r
dlogis(xb) * b["Selbstständig"]
```

```
## Selbstständig
##   -0.04316658
```

```r
dlogis(xb) * b["Haushalt"]
```

```
##    Haushalt
## -0.01262889
```

```r
dlogis(xb) * b["Schlechte_Gesundheit"]
```

```
## Schlechte_Gesundheit
##            0.2831941
```

```r
dlogis(xb) * b["Verheiratet"]
```

```
## Verheiratet
##   0.0282933
```

The interpretation, however, is easy and straight forward. The first part can be interpreted according to the rules of binary classification models and the values from the average marginal effects calculated above. They hold true for a fictional person that represents an average of all 16951 persons in the sample and only for the increase of doctoral visits from 0 to 1. For all changes above one doctoral visit the values of the second part of the Hurdle and the ZI model have to be used and can be interpreted according to the interpretetaion in the Poisson/Negbin section of this paper. Either by interpreting the coefficients as semi-elasticities, calculating odds ratios or by calculating the avergage marginal effects.

References

Cameron & Trivedi Microeconometrics Methods and Applications 2005 (Chapter 20)

J.M. Woolrdige Introductory Econometrics 2015 (Chapter 17)

Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

Appendix

```r
stargazer(my_data, title = "Summary Statistics Welle2005 Data")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Fr, Jun 14, 2019 - 20:01:47

Table 1: Summary Statistics Welle2005 Data

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
persnr	16,951	2,859,469.000	2,244,341.000	201	589,254	5,010,802	8,263,202
DocVisits_Quartal	16,951	2.528	3.862	0	0	3	90
DocVisits_jährlich	16,951	10.111	15.446	0	0	12	360
Geschlecht	16,951	0.457	0.498	0	0	1	1
Alter	16,951	0.482	0.174	0.180	0.350	0.630	0.960
Alter_quadr	16,951	0.263	0.177	0.032	0.122	0.397	0.922
Bildung	16,951	0.116	0.028	0.000	0.105	0.130	0.180
Ausländer	16,951	0.078	0.268	0	0	0	1
SP_Region	16,951	0.738	0.440	0	0	1	1
Vollzeit	16,951	0.366	0.482	0	0	1	1
Teilzeit	16,951	0.193	0.395	0	0	0	1
Selbstständig	16,951	0.036	0.185	0	0	0	1
Arbeitslos	16,951	0.084	0.277	0	0	0	1
Verheiratet	16,951	0.608	0.488	0	0	1	1
Gute_Gesundheit	16,951	0.525	0.499	0	0	1	1
Schlechte_Gesundheit	16,951	0.064	0.244	0	0	0	1
Haushalt	16,951	2.712	1.276	1	2	4	13
income	16,951	0.198	0.118	0.000	0.128	0.239	3.132
welle	16,951	2,005.000	0.000	2,005	2,005	2,005	2,005

```r
stargazer(po_reg,negbin_reg,restr_negbin_reg, title = "Regression Output")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Fr, Jun 14, 2019 - 20:01:48

```r
stargazer(lm(y_star ~ mu_hat_Po -1), title = "Regression Test for Equidispersition")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Fr, Jun 14, 2019 - 20:01:48

```r
stargazer(po_reg_robust, title = "QMLE Poisson Regression")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Fr, Jun 14, 2019 - 20:01:48

```r
# Hurdle Poisson Regression Output
summary(hurdle_po_reg)
```

```
##
## Call:
## hurdle(formula = DocVisits_Quartal ~ Geschlecht + Alter + Alter_quadr +
##     Bildung + Ausländer + SP_Region + Vollzeit + Teilzeit + Gute_Gesundheit +
##     Haushalt + Selbstständig + Schlechte_Gesundheit + Verheiratet,
##     data = my_data, dist = "poisson", zero.dist = "binomial")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -2.6109 -0.9060 -0.3830  0.3821 35.5289
##
## Count model coefficients (truncated poisson with log link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.842171   0.055354  15.214  < 2e-16 ***
## Geschlecht           -0.082800   0.011047  -7.495 6.62e-14 ***
## Alter                -0.174438   0.194243  -0.898  0.36916
## Alter_quadr           0.148510   0.179533   0.827  0.40812
## Bildung               0.052540   0.212097   0.248  0.80435
## Ausländer             0.048470   0.019856   2.441  0.01464 *
## SP_Region             0.110613   0.012200   9.067  < 2e-16 ***
## Vollzeit             -0.151386   0.015449  -9.799  < 2e-16 ***
## Teilzeit             -0.061936   0.016405  -3.775  0.00016 ***
## Gute_Gesundheit       0.662873   0.014028  47.253  < 2e-16 ***
## Haushalt             -0.032541   0.005645  -5.764 8.20e-09 ***
## Selbstständig        -0.068082   0.035536  -1.916  0.05538 .
## Schlechte_Gesundheit  0.431842   0.014642  29.494  < 2e-16 ***
## Verheiratet           0.012483   0.013191   0.946  0.34398
## Zero hurdle model coefficients (binomial with logit link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.21504    0.16825   7.222 5.13e-13 ***
## Geschlecht           -0.61521    0.03826 -16.078  < 2e-16 ***
## Alter                -5.29267    0.76239  -6.942 3.86e-12 ***
## Alter_quadr           7.08916    0.80043   8.857  < 2e-16 ***
## Bildung               3.51394    0.66320   5.298 1.17e-07 ***
## Ausländer             0.12397    0.06722   1.844  0.06516 .
## SP_Region            -0.10710    0.04204  -2.548  0.01084 *
## Vollzeit             -0.09488    0.04976  -1.907  0.05658 .
## Teilzeit             -0.07853    0.05421  -1.449  0.14743
## Gute_Gesundheit       0.95213    0.03890  24.476  < 2e-16 ***
## Haushalt             -0.06512    0.01614  -4.036 5.45e-05 ***
```

Table 2: Regression Output

	Dependent variable:		
	DocVisits_Quartal		
	Poisson	*negative binomial*	
	(1)	(2)	(3)
Geschlecht	-0.199^{***}	-0.238^{***}	-0.240^{***}
	(0.010)	(0.019)	(0.019)
Alter	-0.354^{*}	-0.635^{*}	-0.662^{**}
	(0.182)	(0.335)	(0.332)
Alter_quadr	0.594^{***}	0.937^{***}	0.973^{***}
	(0.170)	(0.326)	(0.321)
Bildung	0.921^{***}	0.980^{***}	0.994^{***}
	(0.201)	(0.349)	(0.343)
Ausländer	0.085^{***}	0.056	0.054
	(0.019)	(0.034)	(0.034)
SP_Region	0.071^{***}	0.055^{***}	0.057^{***}
	(0.012)	(0.021)	(0.020)
Vollzeit	-0.198^{***}	-0.176^{***}	-0.172^{***}
	(0.015)	(0.026)	(0.025)
Teilzeit	-0.089^{***}	-0.070^{***}	-0.069^{**}
	(0.015)	(0.027)	(0.027)
Selbstständig	-0.119^{***}	-0.148^{***}	-0.147^{***}
	(0.032)	(0.052)	(0.052)
Arbeitslos	-0.029	-0.020	
	(0.019)	(0.034)	
Verheiratet	0.056^{***}	0.043^{*}	0.045^{**}
	(0.012)	(0.023)	(0.023)
Gute_Gesundheit	0.826^{***}	0.829^{***}	0.828^{***}
	(0.012)	(0.020)	(0.020)
Schlechte_Gesundheit	0.546^{***}	0.559^{***}	0.560^{***}
	(0.014)	(0.034)	(0.034)
Haushalt	-0.053^{***}	-0.047^{***}	-0.047^{***}
	(0.005)	(0.009)	(0.009)
income	-0.005	0.014	
	(0.047)	(0.083)	
Constant	0.456^{***}	0.498^{***}	0.498^{***}
	(0.051)	(0.086)	(0.086)
Observations	16,951	16,951	16,951
Log Likelihood	−42,422.660	−33,794.820	−33,795.010
θ		1.205^{***} (0.022)	1.205^{***} (0.022)

Table 3: Regression Test for Equidispersition

	Dependent variable:
	y_star
mu_hat_Po	1.229***
	(0.081)
Observations	16,951
R^2	0.013
Adjusted R^2	0.013
Residual Std. Error	30.776 (df = 16950)
F Statistic	228.238*** (df = 1; 16950)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 4: QMLE Poisson Regression

	Estimate	Robust SE	Pr(>|z|)	LL	UL
(Intercept)	0.456	0.107	0	0.246	0.665
Geschlecht	-0.199	0.024	0	-0.246	-0.152
Alter	-0.354	0.399	0.375	-1.135	0.428
Alter_quadr	0.594	0.382	0.120	-0.154	1.342
Bildung	0.921	0.422	0.029	0.094	1.749
Ausländer	0.085	0.040	0.031	0.008	0.163
SP_Region	0.071	0.024	0.003	0.024	0.118
Vollzeit	-0.198	0.034	0	-0.264	-0.132
Teilzeit	-0.089	0.033	0.008	-0.155	-0.024
Selbstständig	-0.119	0.062	0.053	-0.240	0.001
Arbeitslos	-0.029	0.043	0.500	-0.114	0.056
Verheiratet	0.056	0.029	0.050	-0.0001	0.112
Gute_Gesundheit	0.826	0.023	0	0.781	0.870
Schlechte_Gesundheit	0.546	0.038	0	0.472	0.619
Haushalt	-0.053	0.012	0	-0.076	-0.031
income	-0.005	0.109	0.960	-0.219	0.208

```
## Selbstständig        -0.22257    0.09036  -2.463  0.01377 *
## Schlechte_Gesundheit  1.46020    0.14600  10.001  < 2e-16 ***
## Verheiratet           0.14589    0.04672   3.123  0.00179 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 20
## Log-likelihood: -3.953e+04 on 28 Df
```

```
# Hurdle Negbin Regression Output
summary(hurdle_negbin_reg)
```

```
##
## Call:
## hurdle(formula = DocVisits_Quartal ~ Geschlecht + Alter + Alter_quadr +
##     Bildung + Ausländer + SP_Region + Vollzeit + Teilzeit + Gute_Gesundheit +
##     Haushalt + Selbstständig + Schlechte_Gesundheit + Verheiratet,
##     data = my_data, dist = "negbin", zero.dist = "binomial")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.1266 -0.6787 -0.2778  0.2950 23.2760
##
## Count model coefficients (truncated negbin with log link):
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.5051769  0.1121767   4.503 6.69e-06 ***
## Geschlecht           -0.1062997  0.0236507  -4.495 6.97e-06 ***
## Alter                -0.4512224  0.4143228  -1.089  0.27613
## Alter_quadr           0.4170351  0.3943459   1.058  0.29027
## Bildung               0.0923292  0.4399346   0.210  0.83377
## Ausländer             0.0317009  0.0431130   0.735  0.46216
## SP_Region             0.1254981  0.0257197   4.879 1.06e-06 ***
## Vollzeit             -0.1797758  0.0318782  -5.639 1.71e-08 ***
## Teilzeit             -0.0593508  0.0344887  -1.721  0.08527 .
## Gute_Gesundheit       0.8160767  0.0264326  30.874  < 2e-16 ***
## Haushalt             -0.0348177  0.0112068  -3.107  0.00189 **
## Selbstständig        -0.1150172  0.0690022  -1.667  0.09554 .
## Schlechte_Gesundheit  0.5185942  0.0381145  13.606  < 2e-16 ***
## Verheiratet          -0.0003311  0.0280757  -0.012  0.99059
## Log(theta)           -0.0070237  0.0375531  -0.187  0.85163
## Zero hurdle model coefficients (binomial with logit link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.21504    0.16825   7.222 5.13e-13 ***
## Geschlecht           -0.61521    0.03826 -16.078  < 2e-16 ***
## Alter                -5.29267    0.76239  -6.942 3.86e-12 ***
## Alter_quadr           7.08916    0.80043   8.857  < 2e-16 ***
## Bildung               3.51394    0.66320   5.298 1.17e-07 ***
## Ausländer             0.12397    0.06722   1.844  0.06516 .
## SP_Region            -0.10710    0.04204  -2.548  0.01084 *
## Vollzeit             -0.09488    0.04976  -1.907  0.05658 .
## Teilzeit             -0.07853    0.05421  -1.449  0.14743
## Gute_Gesundheit       0.95213    0.03890  24.476  < 2e-16 ***
## Haushalt             -0.06512    0.01614  -4.036 5.45e-05 ***
## Selbstständig        -0.22257    0.09036  -2.463  0.01377 *
## Schlechte_Gesundheit  1.46020    0.14600  10.001  < 2e-16 ***
```

```
## Verheiratet           0.14589    0.04672   3.123  0.00179 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 0.993
## Number of iterations in BFGS optimization: 21
## Log-likelihood: -3.354e+04 on 29 Df
```

```r
# Zero Inflated Poisson Regression Output
summary(ZI_po_reg)
```

```
##
## Call:
## zeroinfl(formula = DocVisits_Quartal ~ Geschlecht + Alter + Alter_quadr +
##     Bildung + Ausländer + SP_Region + Vollzeit + Teilzeit + Gute_Gesundheit +
##     Haushalt + Selbstständig + Schlechte_Gesundheit + Verheiratet,
##     data = my_data, dist = "poisson", zero.dist = "binomial")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -2.6284 -0.9098 -0.3846  0.3837 35.6641
##
## Count model coefficients (poisson with log link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.845938   0.055086  15.357  < 2e-16 ***
## Geschlecht           -0.082625   0.011016  -7.501 6.34e-14 ***
## Alter                -0.182319   0.193763  -0.941 0.346739
## Alter_quadr           0.150795   0.179313   0.841 0.400369
## Bildung               0.070226   0.210370   0.334 0.738513
## Ausländer             0.046461   0.019860   2.339 0.019314 *
## SP_Region             0.107113   0.012129   8.831  < 2e-16 ***
## Vollzeit             -0.151834   0.015430  -9.840  < 2e-16 ***
## Teilzeit             -0.060822   0.016370  -3.715 0.000203 ***
## Gute_Gesundheit       0.663670   0.014003  47.396  < 2e-16 ***
## Haushalt             -0.032438   0.005633  -5.759 8.47e-09 ***
## Selbstständig        -0.066615   0.035277  -1.888 0.058985 .
## Schlechte_Gesundheit  0.432055   0.014638  29.516  < 2e-16 ***
## Verheiratet           0.011477   0.013160   0.872 0.383140
##
## Zero-inflation model coefficients (binomial with logit link):
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.6833678  0.2060178  -8.171 3.06e-16 ***
## Geschlecht            0.6760328  0.0465713  14.516  < 2e-16 ***
## Alter                 6.2240187  0.9450425   6.586 4.52e-11 ***
## Alter_quadr          -8.4599686  1.0030487  -8.434  < 2e-16 ***
## Bildung              -4.0589140  0.7982216  -5.085 3.68e-07 ***
## Ausländer            -0.1205853  0.0792246  -1.522 0.127992
## SP_Region             0.1990535  0.0527794   3.771 0.000162 ***
## Vollzeit              0.0006697  0.0604605   0.011 0.991162
## Teilzeit              0.0546676  0.0649396   0.842 0.399888
## Gute_Gesundheit      -0.6443533  0.0475409 -13.554  < 2e-16 ***
## Haushalt              0.0544795  0.0194596   2.800 0.005116 **
## Selbstständig         0.2247441  0.1109786   2.025 0.042856 *
## Schlechte_Gesundheit -1.4183211  0.1614672  -8.784  < 2e-16 ***
## Verheiratet          -0.1666232  0.0562053  -2.965 0.003031 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 34
## Log-likelihood: -3.953e+04 on 28 Df
```

```r
# Zero Inflated Negbin Regression Output
summary(ZI_NB_reg)
```

```
##
## Call:
## zeroinfl(formula = DocVisits_Quartal ~ Geschlecht + Alter + Alter_quadr +
##     Bildung + Ausländer + SP_Region + Vollzeit + Teilzeit + Gute_Gesundheit +
##     Haushalt + Selbstständig + Schlechte_Gesundheit + Verheiratet,
##     data = my_data, dist = "negbin", zero.dist = "binomial")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.0915 -0.7019 -0.2905  0.2916 24.0563
##
## Count model coefficients (negbin with log link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.713342   0.092950   7.674 1.66e-14 ***
## Geschlecht           -0.112266   0.020169  -5.566 2.60e-08 ***
## Alter                -0.873863   0.346157  -2.524  0.01159 *
## Alter_quadr           0.946059   0.330158   2.865  0.00416 **
## Bildung               0.527749   0.357381   1.477  0.13975
## Ausländer             0.063623   0.035915   1.771  0.07648 .
## SP_Region             0.069754   0.021207   3.289  0.00100 **
## Vollzeit             -0.182177   0.026437  -6.891 5.54e-12 ***
## Teilzeit             -0.064385   0.028439  -2.264  0.02358 *
## Gute_Gesundheit       0.781380   0.021306  36.674  < 2e-16 ***
## Haushalt             -0.044377   0.009385  -4.728 2.26e-06 ***
## Selbstständig        -0.092490   0.056552  -1.635  0.10195
## Schlechte_Gesundheit  0.522389   0.031842  16.406  < 2e-16 ***
## Verheiratet           0.016866   0.023366   0.722  0.47039
## Log(theta)            0.336670   0.023612  14.258  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -5.90979    0.78791  -7.501 6.35e-14 ***
## Geschlecht             2.30935    0.33401   6.914 4.71e-12 ***
## Alter                 21.93179    4.08683   5.366 8.03e-08 ***
## Alter_quadr          -31.57248    5.14969  -6.131 8.74e-10 ***
## Bildung               -6.85615    2.09328  -3.275  0.00106 **
## Ausländer              0.03758    0.21053   0.179  0.85832
## SP_Region              0.20641    0.16477   1.253  0.21030
## Vollzeit              -0.49797    0.18864  -2.640  0.00829 **
## Teilzeit              -0.03531    0.20743  -0.170  0.86485
## Gute_Gesundheit       -0.65384    0.14548  -4.494 6.98e-06 ***
## Haushalt               0.03480    0.05410   0.643  0.52009
## Selbstständig          0.63405    0.27233   2.328  0.01990 *
## Schlechte_Gesundheit -11.77556   90.19151  -0.131  0.89612
## Verheiratet           -0.24785    0.17824  -1.391  0.16437
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 1.4003
## Number of iterations in BFGS optimization: 60
## Log-likelihood: -3.366e+04 on 29 Df
```