Modeling Count Data

Claire Boetticher, MSDS 410, SEC 56

November 22, 2020

Section 1: Introduction

This analysis of count regression models focuses on counts of medical care utilization by the elderly in the United States from a dataset consisting of a subsample of elderly individuals (66 and older) drawn from the National Medical Expenditure Survey, 1987. Beyond health-care data providing information on hospitalizations and physician office visits, these data include information on health status, employment, economic status, and sociodemographic characteristics. The objective of this analysis is to build a selection of count regression models for determining useful predictors of the number of physician office visits a patient will make. These models are developed in a predictive modeling framework, with cross-validation applied to assess models' performance in-sample and predictive accuracy out-of-sample, statistical model evaluation using selected accuracy- and fit-related metrics applicable to count regression, and assessment of models' predictive accuracy in classification of patients into segments that could serve as inputs to more effective planning of care.

Section 2: Sample Definition and Split

The original dataset includes 4406 observations of 22 variables. OFP, or number of physical office visits, serves as the response variable. Five variables (OFNP, OPP, OPNP, EMR, and HOSP) are dropped, leaving 16 predictor variables available for modeling and analysis (Table 1). All observations are kept in the sample for this analysis and no predictor variables are missing values.

Table 1: Variable Definitions

Variable	Definition
OFP (response)	# of physician office visits
EXCLHLTH	= 1 if self-perceived health is excellent
POORHLTH	= 1 if self-perceived health is poor
NUMCHRON	# of chronic conditions (cancer, heart attack, gall bladder problems, emphysema, arthritis, diabetes, other heart disease)
ADLDIFF	= 1 if the person has a condition that limits activities of daily living
NOREAST	= 1 if the person lives in northeastern US
MIDWEST	= 1 if the person lives in the midwestern US
WEST	= 1 if the person lives in the western US
AGE	Age in years (divided by 10)
BLACK	= 1 if the person is African American
MALE	= 1 if the person is male
MARRIED	= 1 if the person is married
SCHOOL	# of years of education
FAMINC	Family income in $10,000
EMPLOYED	= 1 if the person is employed
PRIVINS	= 1 if the person is covered by private health insurance
MEDICAID	= 1 if the person is covered by Medicaid

Section 2.1: Train/Test Split and Response Variable Distribution

In order to assess the count regression models' performance both in- and out-of-sample, basic cross-validation with a 50/50 train test split is applied to divide the sample. Approximately 50 percent of the 4406 observations are used for in-sample model development and 50 percent are used for out-of-sample model assessment. Table 2 shows resulting counts for subsequent model development and validation, reflecting the split.

Table 2: Train and Test Split

Dataset	Count	Percent of Total
Train	2193	49.77
Test	2213	50.23
Total	4406	100

Though traditional exploratory data analysis is not included in this analysis, it is worth noting the presence of observations with zero counts in the response variable, OFP, with percentages shown for the full, train, and test datasets in Table 3. This establishes the need for attention to data with excess zero features under consideration in this analysis.

Table 3: OFP Distribution by Count

	0 OFP	1 or more OFP
Original Dataset	0.1550	0.8500
Train	0.1674	0.8326
Test	0.1428	0.8572

Section 3: Model Identification

Five generalized linear models (GLMs) are identified and fit for evaluation of their fit of the data and predictive accuracy with respect to the count response variable, OFP (physician office visits, measured in integer counts). As GLMs also allow specification of different error distributions, this analysis explores Poisson and negative binomial error distributions. Where permissible, models are fit using backward variable selection, beginning with the full regression equation and successively eliminating variables one at a time based on contribution to Akaike's information criterion (AIC) . This selection process intends to simplify the model without undue impact on performance, with AIC as a proxy for model fit and simplicity. The preponderance of zeros in the response variable is explored through varying regression models designed to potentially mitigate the challenges that scenario presents. Each model's regression output is examined for model fit and complexity and as a basis for determining which variables may contribute most meaningfully to an observed patient's count of physician office visits.

Section 3.1: Baseline model – Poisson regression

A Poisson regression model is fitted first as a baseline, using all 16 available predictor models in backward variable selection. Table 4 shows regression results for the model.

Table 4: Poisson Regression

	Dependent variable:
	ofp
exclhlth	-0.34***
	(0.04)
poorhlth	0.36***
	(0.03)
numchron	0.14***
	(0.01)
adldiff	0.09***
	(0.02)
noreast	0.24***
	(0.02)
midwest	0.05**
	(0.02)
west	0.16***
	(0.03)
age	-0.08***
	(0.02)
black	-0.06**
	(0.03)
male	-0.08***
	(0.02)
school	0.02***
	(0.003)
faminc	0.005
	(0.003)
employed	-0.05
	(0.03)
privins	0.35***
	(0.03)
medicaid	0.28***
	(0.04)
Constant	1.40***
	(0.12)
Observations	2,193
Log Likelihood	-9,328.14
Akaike Inf. Crit.	18,688.28
Note:	*p**p***p<0.01

Backward variable selection retains all predictors except MARRIED (marital status, 1 if the person is married). Though the final model includes the majority of available predictors, many of which are statistically significant, this does not mean that they necessarily contribute to model fit and predictive capability on unseen data. Additionally, multicollinearity may be an issue (for example, a relationship between FAMINC, family income, and EMPLOYED), worth additional examination before determining predictor contribution to OFP. The two variables contributing most to reduction in AIC –

EXCLHLTH/POORHLTH (a patient's self-identification of being in excellent or poor health) and NUMCHRON (number of chronic conditions) – seem reasonable to associate with physician office visits: a patient's self-perception as having excellent health could be reasonably associated with lower physician office visits (lower OFP count). Similarly, a patient's self-perception of poor health may lead them to visit the physician more frequently (higher OFP count). One result meriting attention in this model is the residual deviance as compared to the residual degrees of freedom, since variance and mean should be roughly equivalent (given expected randomness) under Poisson errors. The residual deviance from this model is 12438 on 2177 degrees of freedom, a signal of overdispersion where the variance is larger than the mean. The log-likelihood of -9328.14 will serve as a point of comparison for later models.

Section 3.2: Poisson Regression with Dispersion

Ideally, the observed data should have a sample mean equal to the sample variance (or close to one another except for randomness) in order for the Poisson regression model to be appropriate for that data. The variance to mean ratio of the response variable for the training dataset is calculated at 8.8194. This strong signal of over-dispersion, where the variance is larger than the mean by a factor of almost nine, suggests that Poisson errors may not, in fact, be appropriate for this data. An outcome could be under-prediction of zeros in the response variable and general issues of accuracy. One potential approach to this scenario is quasi-likelihood, which theoretically enables a more appropriate variance function. AIC is not defined for this model, specified with quasi-Poisson errors, so simplification cannot be automated with backward variable selection and the stepAIC function. Instead, the variables identified for model 1 are used. Regression results are shown in Table 5.

Table 5: Poisson Regression with Dispersion

	Dependent variable:
	ofp
excllhlth	-0.34***
	(0.12)
poorhlth	0.36***
	(0.07)
numchron	0.14***
	(0.02)
adldiff	0.09
	(0.07)
noreast	0.24***
	(0.07)
midwest	0.05
	(0.07)
west	0.16**
	(0.07)
age	-0.08*
	(0.04)
black	-0.06
	(0.09)
male	-0.08
	(0.05)
school	0.02***
	(0.01)
faminc	0.005
	(0.01)
employed	-0.05
	(0.09)
privins	0.35***
	(0.08)
medicaid	0.28***
	(0.11)
Constant	1.40***
	(0.34)
Observations	2,193
Note:	*p**p***p<0.01

The coefficient results with quasi-likelihood are the same as the baseline model. Additionally, the residual deviance is the same and reflects a similar signal of over-dispersion (12438 on 2179 degrees of freedom), so it is possible this approach did not correct for that phenomenon. Since this approach is not a maximum likelihood method but a quasi-likelihood method), AIC and BIC values are not calculable and thus not available as a point of comparison for model fit and simplicity. Using this approach does not guarantee that other bias has not been introduced into the modeling process, but it serves as a useful point of comparison for modeling approach.

Section 3.3: Negative Binomial Regression

In situations where count data is over-dispersed, as in this analysis, Negative Binomial regression is an approach to consider for handling a preponderance of zero counts in the response variable. Given the large variance as compared to mean for this data's response variable, with many zero count values, the negative binomial model may be a suitable approach for managing the extra variance more effectively. The third model uses this distribution instead of Poisson for comparison to the baseline and second models. Backward variable selection is applied with this approach, with the full model as the upper model and variables eliminated based on contribution to lowered AIC. Table 6 shows regression results.

Table 6: Negative Binomial Regression

	Dependent variable:
	ofp
exclhlth	-0.35***
	(0.09)
poorhlth	0.40***
	(0.07)
numchron	0.16***
	(0.02)
noreast	0.21***
	(0.06)
west	0.15**
	(0.06)
male	-0.09*
	(0.05)
school	0.03***
	(0.01)
privins	0.37***
	(0.07)
medicaid	0.27***
	(0.09)
Constant	0.76***
	(0.09)
Observations	2,193
Log Likelihood	-6,041.97
theta	1.07*** (0.04)
Akaike Inf. Crit.	12,103.95
Note:	*p**p***p<0.01

Fewer variables are retained in this model (nine predictors total) than the baseline, with certain sociodemographic and economic status predictors removed. The topmost contributive predictor variables are the same (EXCLHLTH, POORHLTH, NUMCHRON). All predictor variables in this smaller set except for WEST and MALE have low p-values. The AIC metric for the negative binomial regression model (12103.95) is notably lower than that of the baseline model (18,688.28), suggesting better model fit. The residual deviance is much lower as well, 2501.1 on 2183 degrees of freedom.

Additionally, the log-likelihood is calculated at -6041.97, a higher value than the baseline model. These signs all point to potential improvement over the baseline Poisson regression model.

Section 3.4: Hurdle Regression

Hurdle count models have two components – one that models positive counts and one that models zero counts, with binary logistic regression. The count model component is only employed if the hurdle (a type of check) for modeling the occurrence of zeros is exceeded. This provides a possible alternative to managing the excess zero counts issue and the resulting over-dispersion posed with other models in the analysis. In health care data, this may be an effective approach given the regular occurrence of this scenario in datasets. Two hurdle models are compared – one with the Poisson distribution and one with negative binomial. The model with negative binomial errors performs better in terms of AIC and is selected as the final Hurdle model for this analysis. Results of that model are shown in Table 7.

Table 7: Hurdle Regression with Negative Binomial Distribution

	Dependent variable:
	ofp
exclhlth	-0.37***
	(0.10)
poorhlth	0.40***
	(0.07)
numchron	0.11***
	(0.02)
adldiff	0.15**
	(0.07)
noreast	0.22***
	(0.06)
west	0.10
	(0.06)
age	-0.11***
	(0.04)
male	0.01
	(0.05)
married	-0.08
	(0.06)
school	0.02***
	(0.01)
privins	0.27***
	(0.08)
medicaid	0.16
	(0.10)
Constant	1.82***
	(0.33)
Observations	2,193
Log Likelihood	-5,991.52
Note:	*p**p***p<0.01

The backward variable selection process retains 12 of the original 16 predictor variables, again with the same top contributing variables as EXCLHLTH, POORHLTH, and NUMCHRON. The Hurdle model retains the MARRIED predictor, though it is not statistically significant. The negative binomial distribution is selected as having a better fit based on AIC (16633.15 with Poisson versus 12037.04 with negative binomial), thus is retained for this portion of the modeling analysis. Log-likelihood for this model is calculated at -5992 on 27 degrees of freedom, slightly lower than Negative Binomial model and perhaps an indicator of slightly better fit.

Section 3.5: Zero-Inflated Regression

Zero-inflated count models are also two-component models, like hurdle regression. The zero-inflated regression approach fits two separate regression models simultaneously, one for counts using the Poisson distribution and one logistic regression model for the zero counts. A zero count can be estimated in either the zero part or in the count distribution part, allowing more flexibility for handling of zero counts. Table 8 contains results from this modeling approach.

Table 8: Zero-Inflated Regression

	Dependent variable:
	ofp
exclhlth	-0.30***
	(0.04)
poorhlth	0.35***
	(0.03)
numchron	0.09***
	(0.01)
adldiff	0.14***
	(0.02)
noreast	0.22***
	(0.03)
midwest	0.05**
	(0.02)
west	0.10***
	(0.03)
age	-0.10***
	(0.02)
male	-0.001
	(0.02)
married	-0.06***
	(0.02)
school	0.02***
	(0.003)
privins	0.23***
	(0.03)
medicaid	0.17***
	(0.04)
Constant	2.00***
	(0.13)
Observations	2,193
Log Likelihood	-8,288.85
Note:	*p**p***p<0.01

The backward variable selection for the zero-inflated regression retained one more variable than model 4 – MIDWEST – a regional variable. Log-likelihood is calculated at -8289 on 28 degrees of freedom, much higher than the Negative Binomial and Hurdle models, possibly a sign of inferior fit compared to the two.

Section 4: Model Comparison In-Sample

Table 9 show metrics calculated across all five models as a means of assessing performance in-sample. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are indicators of the absolute fit of each model to the data, giving more weight to larger differences via squaring. Mean Absolute Error (MAE) is calculated as a measure of the absolute average distance between observed and predicted data. Mean Absolute Percentage Error (MAPE), calculated as an

average of absolute percentage errors, produces infinite values in this analysis because both train and test datasets contain zeros, hence it is not a usable metric. Where applicable, AIC and Bayes' Information Criterion (BIC) are calculated as well, both measures assessing model fit and complexity based on deviance (and designed to prevent irrelevant predictors' inclusion). Neither metric is interpretable on its own but they do provide a means for model comparison and a way to assess model fit and simplicity. Finally, a +/- 2 Grade Percentage Rate is calculated as a measure of the percentage of observations that are Grade 2, i.e., within 2 of the actual number of physician office visits (OFP count).

Table 9: Model Predictive Accuracy Metrics (In-Sample)

Regression Model	AIC	BIC	MSE	RMSE	MAE	MAPE	+/- 2 Grade
Poisson	18688.28	18779.37	45.6057	6.7532	4.1927	INF	0.3051
Poisson with dispersion	NA	NA	45.6057	6.7532	4.1927	INF	0.3051
Negative binomial	12103.95	12166.57	46.0479	6.7859	4.2075	INF	0.3087
Hurdle	12037.04	12190.75	45.6225	6.7544	4.1876	INF	0.3087
Zero-Inflated	16633.69	16793.09	45.4901	6.7446	4.1837	INF	0.3096

Examining AIC and BIC, the Negative Binomial and Hurdle models fit best in-sample by a high margin; the Negative Binomial model, through automated variable selection, contains the smallest (possibly simplest) set of predictor variables (nine) needed to explain variability in OFP. The baseline Poisson regression model has the highest AIC and BIC values, supported by the over-dispersion present and the challenges that phenomenon poses for using Poisson errors. AIC and BIC are not defined for model 2 with quasi-likelihood applied. MSE (and thus RMSE) and MAE values are roughly similar across all five models, providing little point of comparison in predictive accuracy ability. The +/- 2 Grade Percentage metric is also quite close across all models, with the Zero-inflated model performing slightly better and the Poisson and Poisson with dispersion models performing poorest in-sample.

Section 5: Predictive Accuracy

To assess how well each model performs out-of-sample, thus potential utility for predictive modeling in future classification tasks, accuracy metrics are calculated using the test dataset. Table 10 results show MSE, RMSE, MAE, and the same +/- 2 Grade calculated in-sample. Again, MAPE produces infinite values in this scenario.

Table 10: Model Predictive Accuracy Metrics (Out-of-Sample)

Regression Model	MSE	RMSE	MAE	MAPE	+/- 2 Grade
Poisson	37.3729	6.1133	4.1041	INF	0.3199
Poisson with dispersion	37.3729	6.1133	4.1041	INF	0.3199
Negative binomial	37.2629	6.1043	4.0927	INF	0.3217
Hurdle	36.9456	6.0783	4.0833	INF	0.3222
Zero-Inflated	37.0439	6.0864	4.0849	INF	0.3186

Overall, MSE and RMSE values are lower for all models out-of-sample. MAE values are slightly lower as well. The percentage of observations falling within Grade 2 is slightly higher with test data. In general, all five models perform a bit better out-of-sample. The Hurdle model performs best across all four available metrics out-of-sample; the Zero-inflated model shows the next best performance in terms of MSE, RMSE, and MAE; the Negative Binomial model has the next best performance in terms of Grade 2 percentage rate. The baseline Poisson model performs worst in all metrics except Grade 2 percentage rate, where its performance is in the middle. It is a bit more challenging to decipher any of these models as demonstrably better-fitting out-of-sample given the proximity of the values of these calculated metrics, but the in-sample performance aligns with the results, even with the small differences. The AIC and BIC values in-sample help identify some of the issues worth exploring for further model refinement in terms of error distributions and variable selection.

Section 6: Patient Classification

In order to determine operational validity of the five models in concrete business application terms, with a scoring method and criteria for performance, a segmentation of patients by count of physician office visits is developed. Models' classification accuracy in assigning correct segments is calculated to determine their discrimination ability between segments, which could serve as a useful means for planning health care services and managing patient capacity more effectively at facilities. Table 11 contains segments and their respective criteria.

Table 11: Patient Segments

Segment	Count of physician office visits
1	0-5
2	5-10
3	11+

Confusion matrices are produced for each model to assess discrimination ability between classes, in-sample and out-of-sample. Results are shown in Table 12, with correctly-classified segments in bold for each model.

Table 12: Patient Segment Classification Accuracy

PREDICTED

ACTUAL		In-Sample			Out-of-Sample		
		0-5	6-10	11+	0-5	6-10	11+
Poisson	0-5	**0.5181**	0.4609	0.0209	**0.5352**	0.4397	0.0252
	6-10	0.3104	**0.6375**	0.0521	0.3089	**0.6606**	0.0305
	11+	0.2175	0.6707	**0.1118**	0.2216	0.6757	**0.1027**
Poisson with dispersion	0-5	**0.5181**	0.4609	0.0209	**0.5352**	0.4397	0.0252
	6-10	0.3104	**0.6375**	0.0521	0.3089	**0.6606**	0.0305
	11+	0.2175	0.6707	**0.1118**	0.2216	0.6757	**0.1027**
Negative binomial	0-5	**0.5412**	0.4313	0.0275	**0.5611**	0.4086	0.0303
	6-10	0.3021	**0.6271**	0.0708	0.3191	**0.6362**	0.0447
	11+	0.2508	0.6224	**0.1269**	0.2135	0.6568	**0.1297**
Hurdle	0-5	**0.5282**	0.4436	0.0282	**0.5366**	0.4367	0.0266
	6-10	0.2792	**0.6583**	0.0625	0.2927	**0.6646**	0.0427
	11+	0.2417	0.6435	**0.1148**	0.1919	0.7027	**0.1054**
Zero-inflated	0-5	**0.508**	0.4682	0.0239	**0.5307**	0.4449	0.0244
	6-10	0.2771	**0.6688**	0.0542	0.2907	**0.6748**	0.0346
	11+	0.2236	0.6677	**0.1088**	0.2	0.6973	**0.1027**

All models in this analysis exhibit the greatest classifying accuracy with the 6 to 10 physician office visits segment (all in the 60 percent range) with slightly worse discrimination for the 0 to 5 office visit segment, both in- and out-of-sample. The Negative Binomial model classifies most accurately for the 0 to 5 visit segment, in- and out-of-sample, possibly a reflection of the model's handling of zero count values and the possibility that that error distribution is more appropriate for this data than Poisson. The Zero-inflated model performs best with the 6 to 10 visit segment in- and out-of-sample. Across all models, in- and out-of-sample, classification is least accurate by a wide margin for the segment of 11-plus office visits; the Negative Binomial model classifies this segment more accurately but only marginally. For both train and test datasets, this segment represents the smallest proportion of observations: 331 of 2193 (15.09 percent) observations in-sample and 370 of 2213 (16.72 percent) observations out-of-sample, perhaps contributing to the challenge of discriminating this class accurately.

Healthcare settings for applying predictive models likely come with concrete human considerations and risks. Given a business strategy requiring 80 percent out-of-sample performance in order to be considered successful, none of these models meet this threshold for any segment of the classification task or in terms of overall classification accuracy across all classes. For individual segments, the closest any model comes to meeting the requirement is the Zero-inflated model, with 67.48 percent classification accuracy for the 6-10 physician office visit segment (table 12). Table 13 shows overall

classification accuracy for each model in- and out-of-sample, calculated as the correctly-classified observations from the diagonal of each confusion matrix divided by all observations in the respective dataset (train or test).

Table 13: Overall Classification Accuracy

Regression Model	Accuracy (In-Sample)	Accuracy (Out-of-Sample)
Poisson	0.4829	0.4907
Poisson with dispersion	0.4829	0.4907
Negative binomial	0.4975	0.5056
Hurdle	0.4943	0.4930
Zero-Inflated	0.4829	0.4912

All models' classification accuracy metrics, both in- and out-of-sample, fall close to 50 percent. The Negative Binomial model performs best by a slight margin in- and out-of-sample. The Poisson baseline, Poisson with dispersion, and Zero-inflated models perform worst in- and out-of-sample. The proximity of classification accuracy across all models suggests that this metric only serves as one part of determining which model may perform best in an operational setting. Overall, the Negative Binomial the high accuracy requirement and the potential risks associated with mis-classifying patient office visit behavior in terms of time and resources wasted (or possibly patient care mis-handled, models in this analysis would need revision and refinement to meet the threshold for application success.

Appendix 1: Metric Computation and Confusion Matrix Functions

```
# Function for in-sample metrics
gof.ins <- function(model,response){

  f.aic <- AIC(model)
  f.bic <- BIC(model)

  f.mse <- mean((response - model$fitted.values)^2)
  f.rmse <- sqrt(f.mse)
  f.mae <- mean(abs(response - model$fitted.values))
  f.mape <- mean(abs(response - model$fitted.values)/response)

  abs.residual <- abs(response - model$fitted.values)
  grade.2 <- mean(ifelse(abs.residual<=2,1,0))

  output <-
list(AIC=f.aic,BIC=f.bic,MSE=f.mse,RMSE=f.rmse,MAE=f.mae,MAPE=f.mape,GRADE=grade.
2)
  return(output)
}

# Function for out-of-sample metrics
gof.oos <- function(predictions,response){

  f.mse <- mean((response - predictions)^2)
  f.rmse <- sqrt(f.mse)
  f.mae <- mean(abs(response - predictions))
  f.mape <- mean(abs(response - predictions)/response)

  abs.residual <- abs(response - predictions)
  grade.2 <- mean(ifelse(abs.residual<=2,1,0))

  output <- list(MSE=f.mse,RMSE=f.rmse,MAE=f.mae,MAPE=f.mape,GRADE=grade.2)
  return(output)
}

# Confusion matrix function for in- and out-of-sample
conf_matrix <- function(actual, predict){

  actual_segment <- ifelse(actual<=5,'0-5',
                           ifelse(actual>5 & actual<=10,'6-10',
                                  ifelse(actual>10,'11+',
                                         'Other')))
  predict_segment <- ifelse(predict<=5,'0-5',
                            ifelse(predict>5 & predict<=10,'6-10',
                                   ifelse(predict>10,'11+',
                                          'Other')))
  t <- table(actual_segment, predict_segment)
  r <- apply(t,MARGIN=1,FUN=sum)
  return (t/r)
}
```

Appendix 2: R Script

```
# Load dependencies
library(stargazer)
library(MASS)
library(pscl)
library(MLmetrics)

########################################################################
# Section 1: Load data and rename columns
########################################################################

# Set working directory
path <- "/Users/clb/Documents/MSDS410/9-Poisson_ZIP/"
setwd(path)

# Read in csv file for banking data
path.name <- '/Users/clb/Documents/MSDS410/9-Poisson_ZIP/data/'
file.name <- paste(path.name,'medical_care.txt',sep='')
my.data <- read.table(file.name,header=FALSE,sep=" ")

# 4406 observations of 22 variables
# ofp is response variable
str(my.data)
head(my.data)
colnames(my.data)

# Rename columns
colnames(my.data) <-
c("ofp","ofnp","opp","opnp","emr","hosp","exclhlth","poorhlth","numchron",

"adldiff","noreast","midwest","west","age","black","male","married",
                        "school","faminc","employed","privins","medicaid")
head(my.data)

########################################################################
# Exploratory Data Analysis
########################################################################

# OFP distribution

# Summary stats for response variable
summary(my.data$ofp)

ofp_counts <- table(my.data$ofp)
ofp_counts/sum(ofp_counts)

# Proportions of zero and non-zero values in OFP
683/4406
3723/4406

########################################################################
# Section 2: Split the sample population
########################################################################
# Set the seed on the random number generator to get the same split every time
code is run
# 50/50 training/test split
```

```
set.seed(789)
my.data$u <- runif(n=dim(my.data)[1],min=0,max=1)

# Create train/test split
train.df <- subset(my.data, u<.50)
test.df <- subset(my.data, u>=.50)

# Check data split. Sum of parts should equal whole
# 4406 is sample size
# 2193 = ~50% of sample
# 2213 = ~50% of sample
dim(my.data)[1]
dim(train.df)[1]
dim(test.df)[1]
dim(train.df)[1]+dim(test.df)[1]
2193/4406
2213/4406

# Check for missing values - none missing
sapply(train.df, function(x) sum(is.na(x)))

# Drop selected columns from train and test df
train.df <- subset(train.df,select=-c(u,ofnp,opp,opnp,emr,hosp))
test.df <- subset(test.df,select=-c(u,ofnp,opp,opnp,emr,hosp))

# Final values by ofp counts
table(train.df$ofp)
2193-367
367/2193
1826/2193

table(test.df$ofp)
2213-316
316/2213
1897/2213

# Tabulate certain effect means

# Not excellent health had higher mean count of ofp
tapply(train.df$ofp,train.df$exclhlth,mean)

# Poor health had much higher mean count of ofp
tapply(train.df$ofp,train.df$poorhlth,mean)

# Mean count of ofp increases with number of chronic conditions up to 7
tapply(train.df$ofp,train.df$numchron,mean)

# Mean count of ofp goes up a bit with private insurance held
tapply(train.df$ofp,train.df$privins,mean)

# Evaluate for overdispersion - variance to mean ratio
# Variance exceeds mean by factor of 8-9 (full dataset and train set)
var(my.data$ofp)/mean(my.data$ofp)
var(train.df$ofp)/mean(train.df$ofp)

#################################################################
# Section 3: Model identification
```

```
#########################################################################

# ofp is response variable
# 16 predictor variables remaining

# Backward selection used

help(stepAIC)

#########################################################################
# Section 3.1: Model 1: Poisson Regression
#########################################################################

# Define upper model for stepAIC()
# AIC is penalized likelihood approach to variable selection, smallest AIC wins
# Emphasizes trade-off between model fit and complexity

# Compare ofp distribution with actual Poisson distribution
frequencies <- table(train.df$ofp)
frequencies

mean(train.df$ofp)

help(dpois)

par(mfrow=c(1,2))
barplot(frequencies,ylab="Frequency",xlab="OFP",col="darkgreen",main="Actual
OFP")
barplot(dpois(0:89,5.624259)*4406,names=as.character(0:89),ylab="Frequency",
        xlab="OFP",col="lightgreen",main="Poisson")

# Distributions are different: mode of observed data is zero, mode of Poisson is
much higher with
# same mean
# Observed data are highly aggregated, good candidate for negative binomial
distribution
# Var-mean ratio is another sign Poisson distribution is not appropriate

# Upper limit = full model
upper.glm <- glm(ofp ~ .,data=train.df,family=('poisson'))
summary(upper.glm)

# Poisson regression
model1.glm <- stepAIC(object=upper.glm,direction=c('backward'))
summary(model1.glm)

# Start:  AIC=18688.51
# ofp ~ exclhlth + poorhlth + numchron + adldiff + noreast + midwest +
#   west + age + black + male + married + school + faminc + employed +
#   privins + medicaid
#
# End: Step:  AIC=18688.28
# ofp ~ exclhlth + poorhlth + numchron + adldiff + noreast + midwest +
#   west + age + black + male + school + faminc + employed +
#   privins + medicaid

# married excluded, all others kept, minimal reduction in AIC
```

```
# 5 percent critical value for chi-squared with 12438 df is 2286.66
qchisq(0.95, df.residual(model1.glm))
deviance(model1.glm)

# Pearson's chi-squared is 17095.69
pr <- residuals(model1.glm,"pearson")
sum(pr^2)

# Estimate phi
# When phi is larger than 1 sign of over-dispersion = 7.8529
phi <- sum(pr^2)/df.residual(model1.glm)
round(c(phi,sqrt(phi)),4)

# Predict expected mean count
mu <- predict(model1.glm, type = "response")

# Sum the probabilities of a 0 count for each mean
exp <- sum(dpois(x = 0, lambda = mu))

# Model predicts 27, but there are 367 zeros

# Predicted number of 0's
round(exp)

# Observed number of 0's
sum(train.df$ofp < 1)

# Model output table
out.path <- '/Users/clb/Documents/MSDS410/9-Poisson_ZIP/report_outputs/'
file.name <- 'model1.html';
stargazer(model1.glm, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX: Poisson Regression'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

#########################################################################
# Section 3.2: Model 2: Poisson Regression with Dispersion
#########################################################################

# AIC is not defined for the model so simplification can't be automated with
stepAIC

# Model 1: residual deviance is much larger than residual df, use quasi-Poisson
approach
# Residual deviance: 12438  on 2177  degrees of freedom

# Regression model with quasi-Poisson errors, married excluded from model 1
model2.glm <- glm(ofp ~ exclhlth + poorhlth + numchron + adldiff +
                            noreast + midwest + west + age + black + male +
school +
                            faminc + employed + privins + medicaid,
family="quasipoisson", data=train.df)
summary(model2.glm)

# Won't work - NA
AIC(model2.glm)
BIC(model2.glm)
```

```
# Model output table
out.path <- '/Users/clb/Documents/MSDS410/9-Poisson_ZIP/report_outputs/'
file.name <- 'model2.html';
stargazer(model2.glm, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX: Poisson Regression with Dispersion'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

#####################################################################
# Section 3.3: Model 3: Negative Binomial Regression
#####################################################################

help(glm.nb)

# One distribution that helps with overdispersion is the negative binomial.
# We can specify that the positive-count process be fit with a negative binomial
model
# instead of a poisson by setting dist = "negbin".

# Upper limit = full model
upper.glm <- glm.nb(ofp ~ .,data=train.df)
summary(upper.glm)

# # Lower limit = intercept model
# lower.glm <- glm.nb(ofp ~ 1,data=train.df)
# summary(lower.glm)

# Give algorithm iterations to assist, still doesn't like large number of zeros
but
# can still see predicted values, convergence
model3.glm <- stepAIC(object=upper.glm,direction=c('backward'), maxit=100000)

# model3.glm <- stepAIC(object=upper.glm,direction=c('backward'))
summary(model3.glm)

# Likelihood ratio test for over-dispersion in count data
help("odTest")
odTest(model3.glm)

# Model output table
out.path <- '/Users/clb/Documents/MSDS410/9-Poisson_ZIP/report_outputs/'
file.name <- 'model3.html';
stargazer(model3.glm, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX: Negative Binomial Regression'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

#####################################################################
# Section 3.4: Model 4: Hurdle Regression
#####################################################################

help(hurdle)

# By default the zero-count process is “binomial” (ie, binary logistic
regression) and
# the positive-count process is “poisson”

# Poisson distribution
```

```
# Upper limit = full model
upper.glm <- hurdle(ofp ~ ., data=train.df, dist="poisson")
summary(upper.glm)

# # Lower limit = intercept model
# lower.glm <- hurdle(ofp ~ 1,data=train.df)
# summary(lower.glm)

model4a.glm <- stepAIC(object=upper.glm,direction=c('backward'))

# In our summary we get output for two different models.
# The first section of output is for the positive-count process. The second
section is
# for the zero-count process.
summary(model4a.glm)

# Negative binomial distribution

# Upper limit = full model
upper.glm <- hurdle(ofp ~ ., data=train.df, dist="negbin")
summary(upper.glm)

# # Lower limit = intercept model
# lower.glm <- hurdle(ofp ~ 1,data=train.df)
# summary(lower.glm)

model4b.glm <- stepAIC(object=upper.glm,direction=c('backward'))

summary(model4b.glm)

# Negative binomial version is fitting better
AIC(model4a.glm)
AIC(model4b.glm)

# Model output table
out.path <- '/Users/clb/Documents/MSDS410/9-Poisson_ZIP/report_outputs/'
file.name <- 'model4.html';
stargazer(model4b.glm, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX: Hurdle Regression with Negative Binomial
Distribution'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

######################################################################
# Section 3.5: Model 5: Zero-Inflated Regression
######################################################################

help(zeroinfl)

# The formula can be used to specify both components of the model: If a formula
of
# type y ~ x1 + x2 is supplied, then the same regressors are employed in both
components

# Upper limit = full model
upper.glm <- zeroinfl(ofp ~ .,data=train.df,dist="poisson")
summary(upper.glm)
```

```
# Lower limit = intercept model
# lower.glm <- zeroinfl(ofp ~ 1,data=train.df)
# summary(lower.glm)

model5.glm <- stepAIC(object=upper.glm,direction=c('backward'))
summary(model5.glm)

# Model output table
out.path <- '/Users/clb/Documents/MSDS410/9-Poisson_ZIP/report_outputs/'
file.name <- 'model5.html';
stargazer(model5.glm, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX: Zero-Inflated Regression'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

#####################################################################
# Section 4: Model Comparison and Predictive Accuracy
# In-Sample and Out-of-Sample
#####################################################################

# Report AIC, BIC, MSE, MAE, MAPE, and +/- 2 grade percentage rate in-sample

# Function for in-sample metrics
gof.ins <- function(model,response){

  f.aic <- AIC(model);
  f.bic <- BIC(model);

  f.mse <- mean((response - model$fitted.values)^2);
  f.rmse <- sqrt(f.mse);
  f.mae <- mean(abs(response - model$fitted.values));
  f.mape <- mean(abs(response - model$fitted.values)/response);

  abs.residual <- abs(response - model$fitted.values);
  grade.2 <- mean(ifelse(abs.residual<=2,1,0));

  output <-
list(AIC=f.aic,BIC=f.bic,MSE=f.mse,RMSE=f.rmse,MAE=f.mae,MAPE=f.mape,GRADE=grade.
2);
  return(output)
}

# Predictions out-of-sample
model1.test <- predict(model1.glm,newdata=test.df,type="response")
model2.test <- predict(model2.glm,newdata=test.df,type="response")
model3.test <- predict(model3.glm,newdata=test.df,type="response")
model4.test <- predict(model4b.glm,newdata=test.df,type="response")
model5.test <- predict(model5.glm,newdata=test.df,type="response")

# Function for out-of-sample metrics
gof.oos <- function(predictions,response){

  f.mse <- mean((response - predictions)^2);
  f.rmse <- sqrt(f.mse);
  f.mae <- mean(abs(response - predictions));
  f.mape <- mean(abs(response - predictions)/response);
```

```
  abs.residual <- abs(response - predictions);
  grade.2 <- mean(ifelse(abs.residual<=2,1,0));

  output <- list(MSE=f.mse,RMSE=f.rmse,MAE=f.mae,MAPE=f.mape,GRADE=grade.2);
  return(output)
}

#########################################################################
# Section 4.1: Model 1 - Poisson regression
#########################################################################

# AIC and BIC
AIC(model1.glm)
BIC(model1.glm)

# Metrics
gof.ins(model1.glm,train.df$ofp)
gof.oos(model1.test,test.df$ofp)

#########################################################################
# Section 4.2: Model 2 - Poisson regression with dispersion
#########################################################################

# AIC and BIC - will be NA
AIC(model2.glm)
BIC(model2.glm)

# Metrics
gof.ins(model2.glm,train.df$ofp)
gof.oos(model2.test,test.df$ofp)

#########################################################################
# Section 4.3: Model 3 - Negative Binomial regression
#########################################################################

# AIC and BIC
AIC(model3.glm)
BIC(model3.glm)

# Metrics
gof.ins(model3.glm,train.df$ofp)
gof.oos(model3.test,test.df$ofp)

#########################################################################
# Section 4.4: Model 4 - Hurdle regression
#########################################################################

# AIC and BIC
AIC(model4.glm.nb)
BIC(model4.glm.nb)

# Metrics
gof.ins(model4b.glm,train.df$ofp)
gof.oos(model4.test,test.df$ofp)

#########################################################################
# Section 4.5: Model 5 - Zero-Inflated regression
```

```
#########################################################################

# AIC and BIC
AIC(model5.glm)
BIC(model5.glm)

# Metrics
gof.ins(model5.glm,train.df$ofp)
gof.oos(model5.test,test.df$ofp)

#########################################################################
# Section 5: Patient Classification
#########################################################################

# Validate all 5 models

# Define PredictedSegment
# Segment 1: 0-5 physician office visits
# Segment 2: 6-10 physician office visits
# Segment 3: 11+ physician office visits

# Out-of-sample prediction
model1.test <- predict(model1.glm,newdata=test.df,type="response")
model2.test <- predict(model2.glm,newdata=test.df,type="response")
model3.test <- predict(model3.glm,newdata=test.df,type="response")
model4.test <- predict(model4b.glm,newdata=test.df,type="response")
model5.test <- predict(model5.glm,newdata=test.df,type="response")

#########################################################################
# Section 5.1: Model 1 Patient Classification
#########################################################################

# Segment frequencies - train
segment1 <- ifelse(train.df$ofp <=5,1,0)
segment2 <- ifelse((train.df$ofp>5 & train.df$ofp<=10),1,0)
segment3 <- ifelse(train.df$ofp >10,1,0)

table(segment1)
table(segment2)
table(segment3)

331/2193

# Segment frequencies - test
segment1 <- ifelse(test.df$ofp <=5,1,0)
segment2 <- ifelse((test.df$ofp>5 & test.df$ofp<=10),1,0)
segment3 <- ifelse(test.df$ofp >10,1,0)

table(segment1)
table(segment2)
table(segment3)

370/2213

# Confusion matrix function for in- and out-of-sample
conf_matrix <- function(actual, predict){
  actual_segment <- ifelse(actual<=5,'0-5',
```

```
                            ifelse(actual>5 & actual<=10,'6-10',
                                   ifelse(actual>10,'11+',
                                          'Other')))
  predict_segment <- ifelse(predict<=5,'0-5',
                            ifelse(predict>5 & predict<=10,'6-10',
                                   ifelse(predict>10,'11+',
                                          'Other')))
  t <- table(actual_segment, predict_segment)
  r <- apply(t,MARGIN=1,FUN=sum)
  return (t/r)
}

# Model 1 in-sample
conf_matrix(train.df$ofp,model1.glm$fitted.values)

# Model 1 out-of-sample
conf_matrix(test.df$ofp,model1.test)

# Model 2 in-sample
conf_matrix(train.df$ofp,model2.glm$fitted.values)

# Model 2 out-of-sample
conf_matrix(test.df$ofp,model2.test)

# Model 3 in-sample
conf_matrix(train.df$ofp,model3.glm$fitted.values)

# Model 3 out-of-sample
conf_matrix(test.df$ofp,model3.test)

# Model 4 in-sample
conf_matrix(train.df$ofp,model4b.glm$fitted.values)

# Model 4 out-of-sample
conf_matrix(test.df$ofp,model4.test)

# Model 5 in-sample
conf_matrix(train.df$ofp,model5.glm$fitted.values)

# Model 5 out-of-sample
conf_matrix(test.df$ofp,model5.test)
```