Investigating Models for Overdispersed Count Data and its Applications

PRISCILLA SERWAA OMARI-BAAH

Department of Mathematical Sciences
Montana State University

May 3, 2018

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

Priscilla Serwaa Omari-Baah

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date	Laura Hildreth Writing Project Advisor
Date	Mark C. Greenwood Writing Projects Coordinator

Contents

Abstract

Count data often exhibit more variation than what is assumed by Poisson distribution. This is known as over-dispersion.[1] When over-dispersion is present, standard errors of estimated parameters are underestimated. Several models and standard error estimators have been developed to accommodate this phenomenon. This paper seeks to discuss four of them; the quasi-Poisson regression model, negative binomial regression model, Poisson regression model with robust standard errors and Poisson regression model with bootstrap standard errors. A simulation study is conducted to compare prediction errors and standard errors obtained from implementing the models and standard error estimators discussed. Results showed similar prediction errors and standard errors for all methods discussed. We then present an example of modeling diamond prices and provide estimates of parameters and standard errors using models and standard error estimators discussed. Results indicated similar parameter estimates for the quasi-Poisson regression model, Poisson regression model with robust standard errors and, Poisson regression model with bootstrap standard errors but not the negative binomial regression and similar standard errors for the quasi-Poisson regression model, negative binomial regression model, and Poisson regression model with bootstrap standard errors but not the Poisson regression model with robust standard errors.

1 Introduction

The Poisson regression model is often used to model count data due to its simplicity in interpretations. This model, like all parametric regression models, is based on an underlying probability distribution function for the response variable. To no surprise, the underlying probability distribution function of the response is the Poisson distribution. One fundamental feature of this distribution is that the mean and variance of the random variable is the same. This is known as equidispersion [2]. Practically, we may never see this happen but large deviations from equidispersion leads to underestimated standard errors of estimated covariates and thereby leading to p-values that are too small, confidence intervals of estimates that are too narrow and an inflated type I error rate. Over-dispersion is the phenomenon where the variance observed for the response variable is larger than what is assumed for the variable.

Consider an example of 54,000 diamond prices. These prices come from a dataset in ggplot2 package called diamonds (Wickham 2009) along with attributes like cut, color, carat, clarity of the diamonds. The prices are treated as a count variable because it is made up of discrete values (only dollars and no cents). Figure 1 shows a histogram of prices of the diamonds. From the histogram, we can see that the distribution of the prices are highly skewed to the right. We can also see that the sample mean price is \$3933 where as the sample variance of the prices is well over \$ squared 15,915,629. Fitting a Poisson regression regression to the data will lead to issues with over-dispersion since will be assuming an equal mean and variance while the observed variance is about 4000 times of the mean. When over-dispersion is present, a basic Poisson regression model no longer remains suitable. As a result, several count models and methods have been developed which relax this assumption. These models or methods may re-define the variance structure by including an additional parameter to account for over-dispersion in the response variable, change the underlying probability distribution or use other ways to obtain the correct standard errors. In this paper, we will consider the quasi-Poisson regression model, negative binomial regression model, Poisson regression model with robust standard errors and Poisson regression with bootstrap standard errors. The goal of this paper is to discuss and assess whether these models or methods do provide the same results or are different under certain conditions.

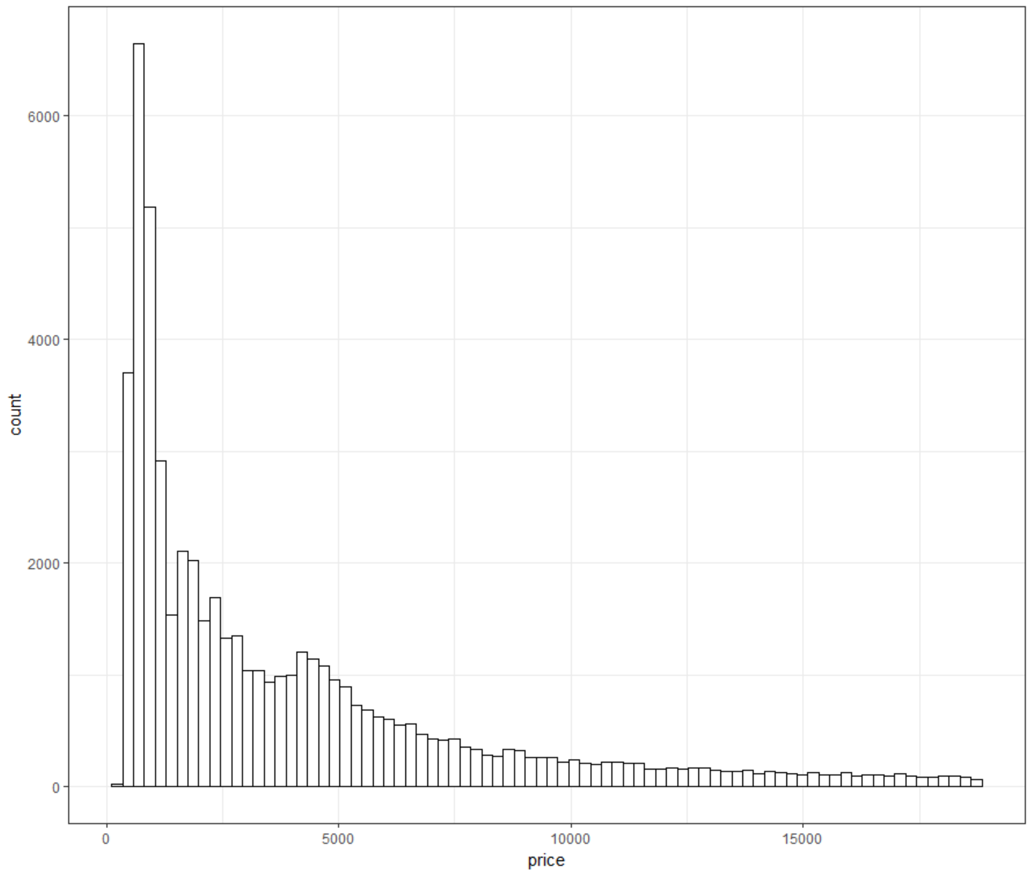

Figure 1: A histogram showing the frequency of the prices of the diamonds

2 Methods for Handling Overdispersed Count Data

Before we discuss the methods that accommodate over-dispersion, it is worthwhile to review the Poisson regression model since these methods of interest are extensions and or modifications of the Poisson model.

2.1 Poisson Regression Model

Poisson regression assumes that the response variable Y, follows a Poisson distribution with mean say, μ. The probability distribution of Y is given by:

$$f(y;\mu) = \frac{e^{-\mu}\mu^{y_i}}{y_i!}, \quad y = 0, 1, \ldots, n; \mu > 0$$

where y is the response - the number of occurrences of an event with mean and variance equal to μ. The regression model is built on the assumption that the log of the mean of these counts μ, can be modeled by a function of some unknown parameters β and fixed predictors X. That is,

$$\log(\mu) = X^T\beta$$

$x^T\beta$ is usually called the linear predictor since it is linear in the unknown parameters. The log in the equation is the canonical link function which is used by most generalized linear models. This link function guarantees that we realize positive values for the mean at all times.

2.2 Quasi-Poisson Regression Model

In many cases where we observe a greater variance for the response than the mean, the quasi-Poisson regression model is used. This regression model uses a quasi-likelihood method to obtain parameter estimates where the likelihood function is not based on any probability distribution from the exponential family of distributions but only characterized by its mean and variance. The mean ad variance is given by:

$$E(Y) = \mu, \quad \mu > 0$$

$$Var(Y) = \theta\mu, \quad \mu > 0; \theta > 0$$

We can see that the mean is exactly like that of a Poisson distribution but its variance has additional parameter. This additional parameter is the over-dispersion parameter. It accommodates extra variation in the response such that the variance function is now a linear function of the mean.

What makes the quasi-Poisson model more comparable to the Poisson is the use of the same log canonical link function. Thus, a quasi-Poisson regression is known to produce similar predictions as we would have obtained with Poisson regression. The only difference are in standard errors values; they are modified to result in ones that would been obtained if there was no over-dispersion.

2.3 Negative Binomial Regression Model

The negative binomial distribution is another distribution used for count data. Consider a random variable, Y that follows a Poisson-gamma mixture distribution with a mean and variance of μ where μ follows a gamma distribution with mean of 1. Then the marginal distribution of y results in a negative binomial distribution. The probability mass function is given by:

$$f(y;\mu,\alpha) = \frac{\Gamma(y_i + 1/\alpha)}{y_i!\Gamma(1/\alpha)} \left(\frac{1}{1+\alpha\mu_i}\right)^{1/\alpha} \left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right)^{y_i} \tag{1}$$

with mean and variance respectively as:

$$E(Y) = \mu \quad Var(Y) = \mu + \alpha\mu^2 \tag{2}$$

While the mean does not change from the mean of a Poisson distribution, its variance has additional parameter, α. This additional parameter is the over-dispersion parameter. It accommodates extra variation in the response such that the variance function is now a quadratic function of the mean which is different from the way the quasi-Poisson handles over-dispersion. However, they both use the canonical link when modeling the mean making both models comparable. The population regression model is given by:

$$\log(\mu) = X^T\beta \tag{3}$$

$X^T\beta$ is the linear predictor and log is the canonical link function.

2.4 Poisson Regression Model with Robust Standard Errors

Robust standard error estimators were first introduced by Huber (1967) and then later independently by White (1980), thus they are also known as Huber standard errors or White standard errors. They are popularly used in econometrics to obtain unbiased standard errors for the estimates of the coefficients in ordinary least squares regression when there is heteroscedasticity[2]. That is, when the error variance is not constant.

The most popular type of robust standard error estimators are the sandwich estimators. The name emanates from the fact that the estimator is the first derivative of the model likelihood function (score) enclosed in the second derivative (Hessian). White (1980) showed that this estimator does not need the underlining distribution of the response to be correctly specified. Thus, they have been proven to also work well in cases like over-dispersion.

Robust estimation of standard errors is carried post-estimation of the model. As a result, we do not expect the estimates of the coefficients to be change; we only correct the estimates of the standard errors. Robust standard error estimation for a Poisson regression model is implemented by the following steps [3]:

1. Estimate model parameters using the Poisson regression model

2. Determine the linear predictor matrix, $X\beta$

3. Determine the score vector which is the first derivative of the likelihood function
$$g' = \frac{X\partial L(X\beta)}{\partial X\beta}$$

4. Adjust the degrees of freedom to be $n(n-1)$

5. Calculate the new variance-covariance matrix as
$$H^{-1}\{n/(n-1)\sum(gg')\}H^{-1}$$
where H represent the Hessian matrix of the likelihood function.

6. Exchange the old variance-covariance matrix with the new robust one.

2.5 Poisson Regression Model with Bootstrap Standard Errors

Bootstrapping involves creating samples from an original sample by method of sampling with replacement. Estimates obtained from the bootstrapped samples are only based on the data and do not require any parametric distributional assumptions. Therefore, we are able to find estimates whether or not the distribution estimators have a close form, are misspecified or are even not stated. For a Poisson regression model, bootstrap standard errors are obtained from the following steps [4]:

1. Randomly sample r independent bootstrap samples of size equal to the size of the original sample $\mathbf{B}_1^*, \mathbf{B}_2^* \ldots \mathbf{B}_r^*$ with replacement from the data.

2. For each bootstrap sample, fit the Poisson regression model to obtain estimates of the parameter $\beta_{b0}^*, \ldots \beta_{bp}^*$ for $b = 1, \ldots, r$.

3. For each parameter, compute the standard error of the r estimates obtained.

$$\widehat{se(\beta)} = \left[\frac{1}{r-1}\sum_{b=1}^{r}\{\hat{\beta_b^*} - \hat{\beta^*}.\}\right]$$

where $\beta^*.$ is the average of all r estimates of the parameter:

$$\beta^*. = r^{-1}\sum_{b=1}^{B}\beta_b^*$$

3 Simulation Study

One of the main goals when fitting a regression model is to make predictions. A well-fitted regression model will produce predictions that are similar to the observed data leading to smaller prediction errors. Parameter estimates obtained from the regression model are also of interest since they are used in predictions. Ideally, we want these estimates to be as precise as possible. Standard errors of the estimates provide information on how precise our estimates are. Smaller standard errors provide more precision. Thus, a simulation study is conducted to compare prediction errors and standard errors of the parameter estimates for the methods discussed in the presence of over-dispersed data. The simulation process was carried out under the following conditions:

1. One predictor variable, x is sampled from a $N(0,1)$ distribution.

2. Samples of size $n \in \{100, 200, \ldots, 1000\}$ were randomly selected from a Negative Binomial distribution with overdispersion parameters, $\alpha \in 0.01, 0.1, 0.5, 1$ and a mean function such that:

$$\log(\mu) = 8 + 0.2x$$

3. Each 10×4 combination of n and α is sampled 1000 times to be our observed responses.

3.1 Comparing Prediction Errors

For each of the 1000 random samples from each n $\times$ α combination, the root mean square error is computed and the mean of the 1000 root mean square errors will is found. The root mean square error (RMSE) measures the difference between the observed values and the values predicted using the model, standardized by the sample size. Larger values of the RMSE indicate more discrepancies between what is observed and what is predicted by the model. The RMSE is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}}$$

where $\hat{y}_i$ is predicted response value obtained from fitting the model.

The following questions of interest to us:

1. How do prediction errors compare for a given sample size and over-dispersion parameter?
2. How does sample size impact prediction errors?
3. How does the over-dispersion parameter impact prediction errors?

Figure 2 shows line graphs of the mean RMSE against sample size for each of the four over-dispersion parameters. First, it is evident from the graph that prediction errors are relatively not different for each for the four different methods as the lines on the plot are very close to each other. This is actually not surprising. Remember that the mean function does not change across the four methods. This means that $\exp(x^T\beta)$ stays fairly the same so prediction errors will also stay fairly equal. Secondly, we can see that for each line graph in Figure 2, the mean RMSE increases as sample size increases. This can be associated with increasing number of deviations being added as the sample size is increasing. If we increase sample size then the number of deviations to be added will increase which in result will increase the mean RMSE.

Lastly, the graph suggests an overall increase in mean RMSE when the over-dispersion parameter increases. We can see that the mean RMSEs for over-dispersion parameter of 0.01 ranges from about 300 to 318, however that of over-dispersion parameter of 1 ranges from about 3000 to 3200. This makes sense because as the over-dispersion parameter increases, the variability in the response variable increases. More variability in the response means less precise prediction leading to larger prediction errors.

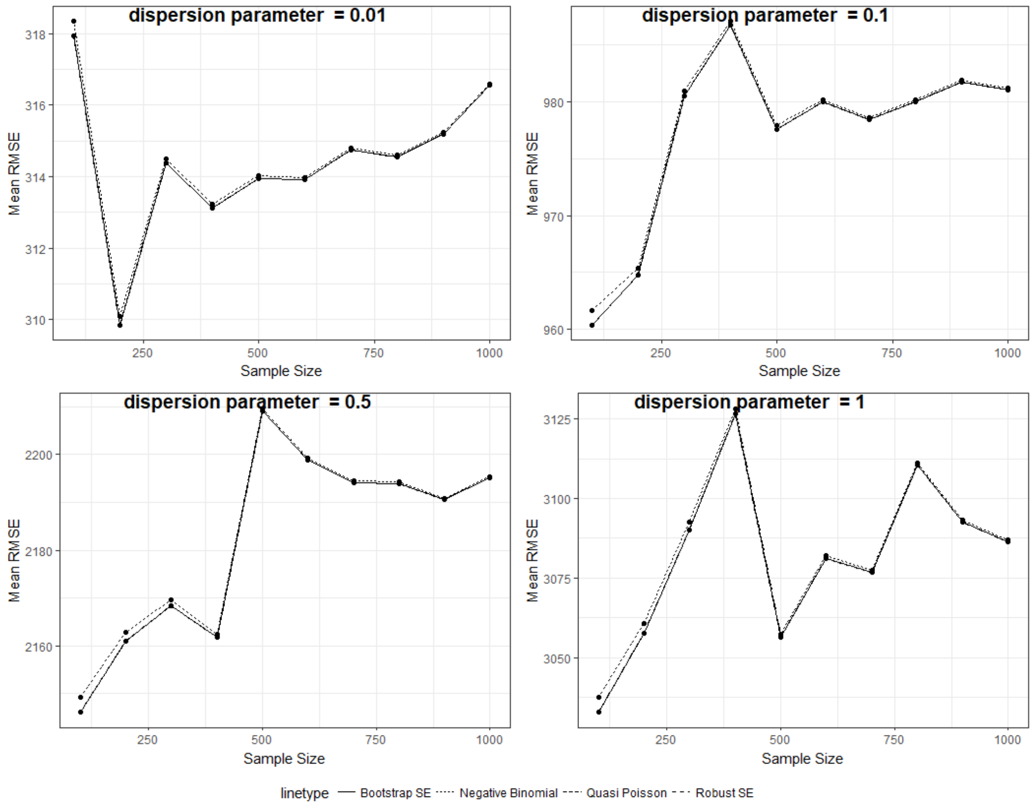

Figure 2: Line graphs showing Mean RMSE against sample size for each of the four over-dispersion parameter.

3.2 Comparing Standard Errors

As we have seen the mean RMSE for a given sample size and over-dispersion parameter are similar for all four methods. The next question one may ask is "how do the estimates vary across the four methods for a given sample size and over-dispersion parameter?" To do this, we can compare the distribution of standard errors of the coefficient estimates. The questions of interest are:

1. How does the distribution of standard errors of the estimated slope compare for a given sample size and over-dispersion parameter?

2. How does sample size impact the distribution of the standard errors of the estimated slope?

3. How does the over-dispersion parameter impact standard error of the estimated slope?

Figure 3 shows beanplots of the distributions of the estimated standard errors of the slope coefficients of the four methods at a over-dispersion parameter of 0.1. From the plots, it is evident that there is no substantial difference among the distributions of the standard errors at an over-dispersion parameter of 0.1. Actually, an over-dispersion parameter of 0.1 can be considered fairly low. Thus with a small over-dispersion in the response, we will expect the distribution of the standard errors to be similar across the methods. The four plots in the figure represent the distribution at different samples sizes. It is not too surprising that as the sample size increases the distribution gets narrower. Figure 4 shows beanplots of the distributions of the estimated standard errors of the coefficients of the four methods at an over-dispersion parameter, α of 1. The data used in these plots are considered to have more variability in the response than those used in Figure 3. Overall, we can see that the distributions are wider than when we have an over-dispersion parameter of 0.1. Also, the distribution of bootstrap standard errors seems to be slightly wider across sample sizes which can be attributed to taking only 100 bootstrap samples.

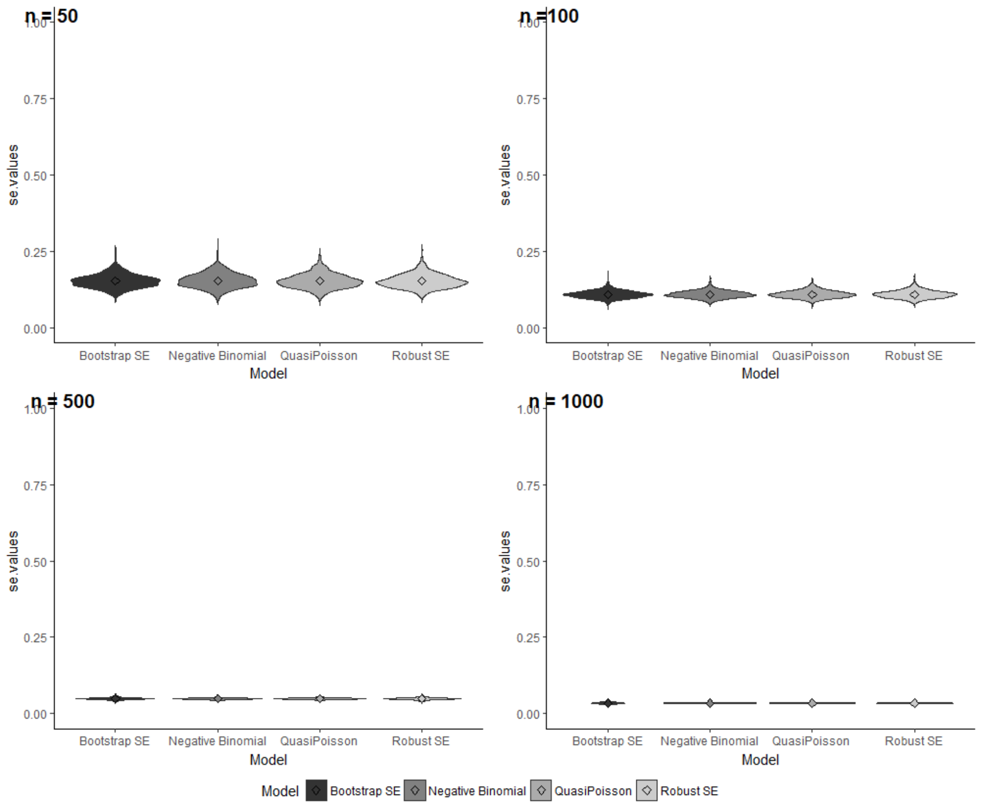

Figure 3: Beanplots showing the distributions of the estimated standard errors of the coefficients of the four methods at $\alpha = 0.1$

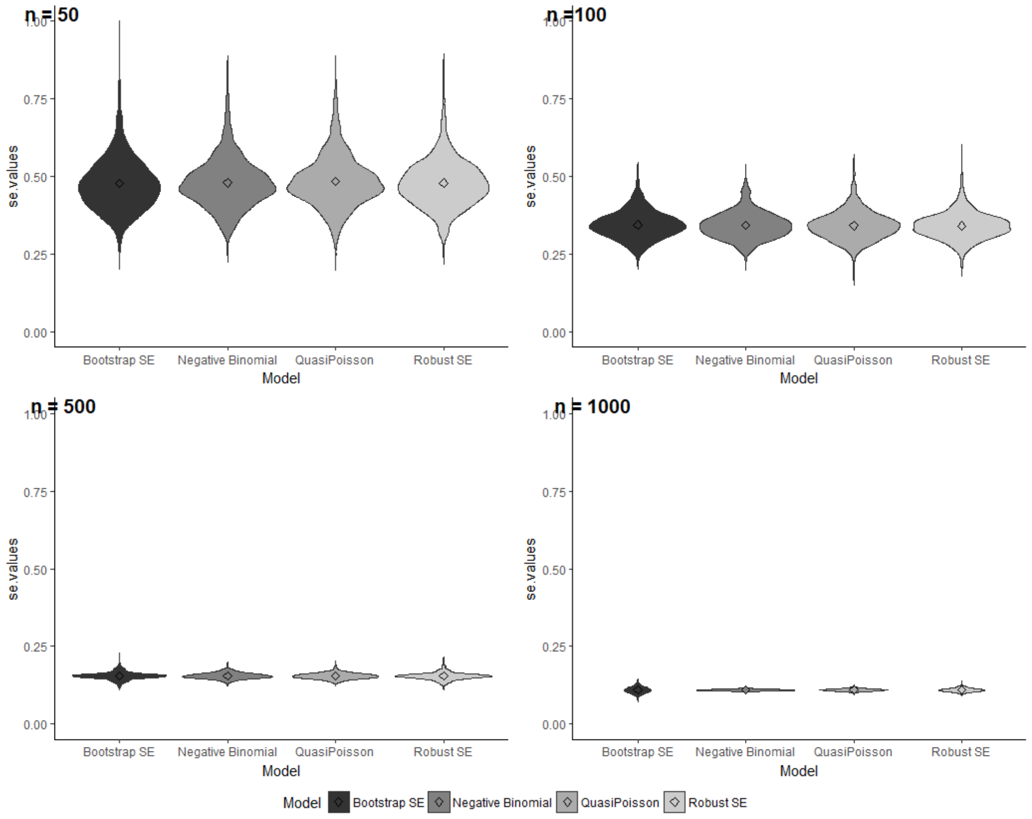

Figure 4: Beanplots showing the distributions of standard errors of the slope coefficient four methods at $\alpha = 1$

4 Diamond Price Example

4.1 The data

The dataset “diamonds” can be found in the ggplot2 package (Wickham 2009) and it contains information about the characteristics of 53940 diamonds. Table 2 describes the variables in the dataset. Our research goal is to describe the relationship between the price of a diamond and the other attributes of the diamond.

Table 1: Descriptin of variables in the diamond dataset

Variable	Description
Price	Price of the diamond in USD
Carat	The weight of the diamond
Cut	The quality of the cut of the diamond (Fair, Good, Ideal, Premium, Very Good)
Color	Diamond color (D, E, F, G, H, I, J)
Clarity	A measurement of clarity of diamond (I1, IF, SI1, SI2, VS1, VS2, VVS1, VVS2)
Table	Width of the top of the diamond relative to the widest point
x	Length in mm
y	Width in mm
z	Depth in mm
Depth	Total depth percentage = $\left(\frac{2z}{x+y}\right)$

4.2 Analysis

Figure 5 is a correlation/scatterplot matrix of the variables in the diamonds dataset. From this plot we can see that variables x, y and z are all highly correlated with carat. This indicates that if we fit a model including all four variables in the model, we may have issues with multicollinearity. Based on that, we choose to exclude x, y and z as covariates and use carat instead in any model that will be fit.

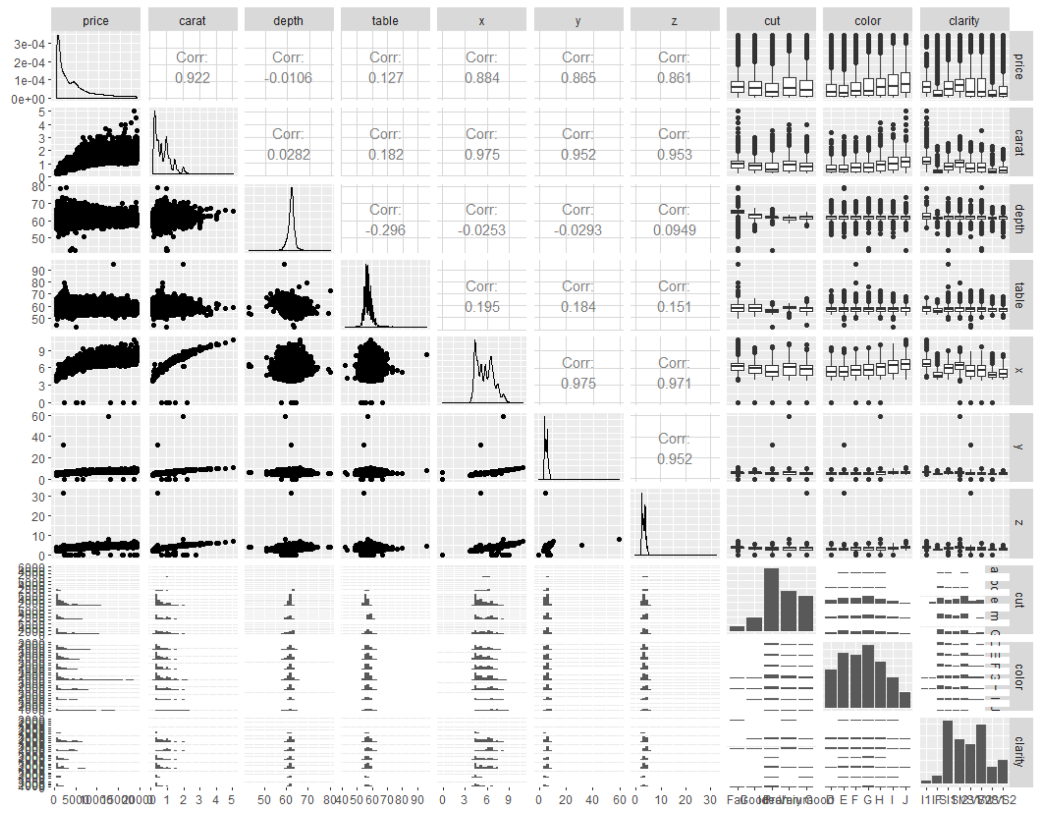

Figure 5: A correlation/scatterplot matrix of the variables

We now a fit a Poisson regression model to assess if there is overdispersion in the data. The over-dispersion parameter is estimated to be 483.317 which we expected based on the histogram in Figure 1. This suggests that the data is over-dispersed and it is appropriate to use any of the models or methods discussed.

For each of the methods being fit, the covariates considered are carat, depth, table, cut with fair as the baseline group, color with D as the baseline group and clarity with I1 as the baseline group.

Table 2: Parameter estimates and standard errors of the quasi-Poisson, negative binomial, Poisson with robust standard errors and Poisson with bootstrap standard errors regression models

	Estimates				Standard Errors			
Variables	Quasi-Poisson	Neg.Bin.	Robust SE	Bootstrap SE	Quasi-Poisson	Neg.Bin.	Robust SE	Bootstrap SE
Intercept	5.12	5.21	5.12	5.12	0.107	0.102	0.198	0.105
Carat	1.66	2.29	1.66	1.66	0.003	0.003	0.014	0.025
Depth	0.00	0.00	0.00	0.00	0.001	0.001	0.002	0.010
Table	0.00	0.01	0.00	0.00	0.001	0.001	0.001	0.001
CutGood	0.17	0.04	0.17	0.17	0.010	0.009	0.040	0.080
CutIdeal	0.22	0.10	0.22	0.22	0.010	0.009	0.039	0.011
CutPremium	0.18	0.04	0.18	0.18	0.009	0.009	0.040	0.009
CutVeryGood	0.21	0.01	0.21	0.21	0.009	0.009	0.039	0.009
ColorE	-0.06	-0.06	-0.06	-0.06	0.006	0.005	0.008	0.007
ColorF	-0.03	-0.07	-0.03	-0.03	0.006	0.005	0.008	0.006
ColorG	-0.10	-0.15	-0.10	-0.10	0.006	0.005	0.009	0.060
ColorH	-0.25	-0.27	-0.25	-0.25	0.006	0.005	0.011	0.005
ColorI	-0.42	-0.43	-0.42	-0.42	0.006	0.006	0.014	0.006
ColorJ	-0.64	-0.58	-0.64	-0.64	0.008	0.007	0.018	0.070
ClarityIF	1.68	0.95	1.68	1.68	0.017	0.014	0.123	0.018
ClaritySI1	1.35	0.61	1.35	1.35	0.014	0.012	0.127	0.015
ClaritySI2	1.11	0.44	1.11	1.11	0.014	0.012	0.131	0.015
ClarityVS1	1.55	0.78	1.56	1.56	0.014	0.012	0.125	0.014
ClarityVS2	1.47	0.71	1.47	1.47	0.014	0.012	0.126	0.014
ClarityVVS1	1.57	0.86	1.57	1.57	0.016	0.013	0.122	0.015
ClarityVVS2	1.63	0.85	1.63	1.63	0.015	0.013	0.123	0.014

Table 3 shows the parameter estimates and their standard errors of the quasi-Poisson, negative binomial, Poisson with robust standard errors and Poisson with bootstrap standard errors regression models. We can see we have that the parameter estimates are very similar for all methods except the negative binomial model which has slightly lower estimates. The standard errors of the parameter estimates are also very similar for the model except for the robust standard errors which has slightly higher values.

Table 3: RMSE of the quasi-Poisson, negative binomial, Poisson with robust standard errors and Poisson with bootstrap standard errors regression models

Model	RMSE
Quasi-Poisson	2747.653
Negative Binomial	59113.17
Robust SE	2747.653
Bootstrap SE	2747.653

Table 4 shows the RMSEs for the four methods discussed. From the table we can see that the RMSE for the quasi-Poisson, Poisson with robust standard errors and Poisson with bootstrap standard errors regression models are all the same and the RMSE for the negative binomial model is higher.

5 Conclusions and Discussions

In this paper, we presented the problem of over-dispersion and extensions of the Poisson regression model that deal with it. A simulation study was then carried out to compare the prediction errors and standard errors of estimated parameters. As an example, we fit regression models to determine the relationship between a price and certain other attributes of a diamond and also compare parameter estimates and their standard errors across the models and methods discussed.

In the simulation study, we observed similar prediction errors and standard errors of the estimated paramters across the models and methods discussed. We also observed that higher prediction errors were associated with larger sample sizes and larger over-dispersion parameters. It was not too surprising to see the variability of the distribution of standard errors decrease as sample size increase but we also saw that with an increase in the over-dispersion parameter, the variability of the standard error distribution decreased.

In our analysis of the example data, we observed similar parameter estimates for all but the negative binomial regression. We also saw that the prediction error for the negative binomial was higher than the rest of the methods. We may attribute this to the prices not coming from Poisson-Gamma mixture distribution that results in a negative binomial rather some other type of Poisson mixture distribution like Poisson-Lognormal or Poisson-Inverse Gaussian distribution.

Some recommendations for further research will be to consider other models that allow for over-dispersion in the data like a quasi-negative binomial model, Poisson mixture model with a heterogeneous shape parameter that follows say an inverse gaussian or a lognormal distribution, or non parameteric regression models.

6 References

1. Hoef, J. M., & Boveng, P. L. (2007). Quasi-Poisson Vs. *Negative Binomial Regression: How Should We Model Overdispersed Count Data?* Ecology, 88(11), 2766-2772. doi:10.1890/07-0043.1

2. Hilbe, J. M. (n.d.). *Overdispersion. Negative Binomial Regression*, 51-76. doi:10.1017/cbo9780511811852.006

3. Cruyff, M. J., & Peter G. M. Van Der Heijden. (2008). *Point and Interval Estimation of the Population Size Using a Zero-Truncated Negative Binomial Regression Model.* Biometrical Journal, 50(6), 1035-1050. doi:10.1002/bimj.200810455

4. Bonafede, E., Picard, F., Robin, S., & Viroli, C. (2015). *Modeling overdispersion heterogeneity in differential expression analysis using mixtures.* Biometrics, 72(3), 804-814. doi:10.1111/biom.12458

5. Hall, D. B., & Shen, J. (2009). *Robust Estimation for Zero-Inflated Poisson Regression.* Scandinavian Journal of Statistics, 37(2), 237-252. doi:10.1111/j.1467-9469.2009.00657.x

6. Wickham, H. "*ggplot2.*" 2009, doi:10.1007/978-0-387-98141-3.

7. Rodriguez, G. (2013). *Models for Count Data With Overdispersion.* Unpublished manuscript.

8. Roy, Tapon. "*Bootstrap Accuracy for Non-Linear Regression Models.*" Journal of Chemometrics, vol. 8, no. 1, 1994, pp. 37–44., doi:10.1002/cem.1180080105.

9. White, H. " *A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity.*" Econometrica, vol. 48, no. 4, 1980, p. 817., doi:10.2307/1912934.

10. Fox, J. Applied *Regression Analysis and Generalized Linear Models.* Sage, 2016.

11. Agresti, A. *Categorical Data Analysis.* Wiley-Interscience, 2002.

12. Huber, P. J. (1967). T*he behavior of maximum likelihood estimates under nonstandard conditions, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA: University of California Press, pp. 221– 233.

13. Pan, W. (2001b). *On the robust variance estimator in generalized estimating equations*, Biometrika 88(3): 901–906.

7 Appendix

7.1 Rcode

```
require(MASS)
require(car)
require(sandwich)
require(ggplot2)
require(ggpubr)
require(DescTools)
require(boot)
require(tidyverse)

##Importing Dataset##
dia<-read.csv("~/diamonds.csv")
var(dia$price)
mean(dia$price)

##Exploratory Data Analysis##
#Histogram of price#
p<-ggplot(dia, aes(price)) +
geom_histogram(bins = 80, fill="#ffffff", colour="black")+theme_bw()
p+annotate("text", x = 11000, y = 4300, label = "Mean = $3,932.8",
size = 7)+annotate("text", x = 11000,
y = 3700, label = "Variance = $15,915,629", size = 7)+
labs(x="Prices (in dollars)", y = "Count")

##Simulations###
```

```
##RMSE##
alpha<-c(0.01,0.1,0.5,1)
n<-seq(100,1000,100)
num.sims<-100

plot<-list()

for(k in 1:length(alpha)){

mean.rmse.qp<-c()
mean.rmse.nb<-c()
mean.rmse.rv<-c()
mean.rmse.bs<-c()


for(j in 1: length(n)){




r.qp<-c()
r.nb<-c()
r.rv<-c()
r.bs<-c()

for( i in 1:num.sims){

x<-matrix(c(rep(1,n[j]),rnorm(n[j])),n[j],2)
b<-matrix(c(8,0.2),2,1)
mu<-exp(x%*%b)
mu
y<-rnegbin(n=n[j], mu = mu,theta =( 1/(alpha[k])))
x<-x[,-1]
sim.data<-data.frame(y,x)

y.qp<-glm(y~.,family = quasipoisson(link = "log"), data = sim.data)
r.qp[i]<-sqrt(sum((y-fitted(y.qp))^2)/n[j])

y.nb<-glm.nb(y~.,init.theta = 1, data = sim.data)
r.nb[i]<-sqrt(sum((y-fitted(y.nb))^2)/n[j])

y.rv<-glm(y~.,family = poisson(link = "log"), data = sim.data)
r.rv[i]<-sqrt(sum((y-fitted(y.rv))^2)/n[j])

y.bs<-glm(y~.,family = poisson(link = "log"), data = sim.data)
r.bs[i]<-sqrt(sum((y-fitted(y.bs))^2)/n[j])
}

mean.rmse.qp[j]<-mean(r.qp)
mean.rmse.nb[j]<-mean(r.nb)
mean.rmse.rv[j]<-mean(r.rv)
mean.rmse.bs[j]<-mean(r.bs)
}
data.plot<-data.frame(n,mean.rmse.qp,mean.rmse.nb,mean.rmse.rv,mean.rmse.bs)

plot[[k]]<- ggplot(data=data.plot, aes(x = n)) +theme_bw()+
ylab(label = "Mean RMSE")+
xlab(label = "Sample Size")+
theme(legend.position="none")+
geom_line(aes(y = mean.rmse.qp, linetype = "Quasi Poisson"))+
geom_point(aes(y = mean.rmse.qp))+
geom_line(aes(y = mean.rmse.nb, linetype = "Negative Binomial"))+
geom_point(aes(y = mean.rmse.nb))+
geom_line(aes(y = mean.rmse.rv, linetype = "Robust SE"))+
geom_point(aes(y = mean.rmse.rv))+
geom_line(aes(y = mean.rmse.bs, linetype = "Bootstrap SE"))+
geom_point(aes(y = mean.rmse.bs))

}
ggarrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],
labels = c("dispersion parameter  = 0.01", "dispersion parameter  = 0.1",
"dispersion parameter  = 0.5","dispersion parameter  = 1"),
legend  = "bottom", common.legend = T,ncol = 2, nrow = 2)

##Standard Errors##

num.sims<-1000
```

```
n<-c(50,100,500,1000)
plot<-list()

for (j in 1:length(n)){

s.qp<-c()
s.nb<-c()
s.rv<-c()
s.bs<-c()

for (i in 1:num.sims){

x<-matrix(c(rep(1,n[j]),runif(n[j])),n[j],2)
b<-matrix(c(8,0.2),2,1)
mu<-exp(x%*%b)
y<-rnegbin(n=n[j], mu = mu,theta =1)
x<-x[,-1]
sim.data<-data.frame(y,x)

##quasipoisson##
y.qp<-glm(y~.,family = quasipoisson(link = "log"), data = sim.data)
s.qp[i]<-summary(y.qp)$coefficients[2,2]
summary(y.qp)
##neg bin###
y.nb<-glm.nb(y~.,init.theta = 1, data = sim.data)
s.nb[i]<-summary(y.nb)$coefficients[2,2]

##robust se##
y.rv<-glm(y~.,family = poisson(link = "log"), data = sim.data)
s.rv[i]<-sqrt(diag(vcovHC(y.rv, type="HC0")))[2]
summary(y.rv)
##bootstrap se##
se <- function(d,indices) {
d <- d[indices,]
fit <- glm(y~., family = poisson(link = "log"), data = d)
return(coef(fit))
}

bse <- boot(
data = sim.data,
statistic = se,
R = 100
)
s.bs[i]<-summary(bse)$bootSE[2]
}


se.values<-c(s.qp,s.nb,s.rv,s.bs)
Model<-as.factor(c(rep("QuasiPoisson",n[j]), rep("Negative Binomial",n[j]),
rep("Robust SE",n[j]),rep("Bootstrap SE",n[j])))
data.se<-data.frame(Model,se.values)

plot[[j]]<-ggplot(data.se, aes(x=Model, y=se.values, fill = Model))+
scale_fill_grey() + theme_classic()+
geom_violin(trim=FALSE)+stat_summary(fun.y=mean, geom="point",
shape=23, size=2)+theme(legend.position="none")+ylim(0,1)

}




ggarrange(plot[[1]],plot[[2]],plot[[3]], plot[[4]],
labels = c("n = 50", "n =100",
"n = 500","n = 1000"),legend  = "bottom", common.legend = T,ncol = 2, nrow = 2)


###Power###

num.sims<-1000


n<-c(50,100,500,1000)
```

```
p.qp<-c()
p.nb<-c()
p.rv<-c()
p.bs<-c()

for (j in 1:length(n)){

qp<-c()
nb<-c()
rv<-c()
bs<-c()

for (i in 1:num.sims){

x<-matrix(c(rep(1,n[j]),runif(n[j])),n[j],2)
b<-matrix(c(8,0.2),2,1)
mu<-exp(x%*%b)
y<-rnegbin(n=n[j], mu = mu,theta =0.1)
x<-x[,-1]
sim.data<-data.frame(y,x)

##quasipoisson##
y.qp<-glm(y~.,family = quasipoisson(link = "log"), data = sim.data)
if (summary(y.qp)$coefficients[2,4]<0.05) {
qp[i]<-1
} else {
qp[i]<-0
}

##neg bin###
y.nb<-glm.nb(y~.,init.theta = 1, data = sim.data)
if (summary(y.nb)$coefficients[2,4]<0.05) {
nb[i]<-1
} else {
nb[i]<-0
}

##robust se##
y.rv<-glm(y~.,family = poisson(link = "log"), data = sim.data)
z.stat<-(summary(y.rv)$coefficients[2,1])/
(sqrt(diag(vcovHC(y.rv, type="HC0")))[2])
pval<-2*(1-pnorm(z.stat))
if (pval<0.05) {
rv[i]<-1
} else {
rv[i]<-0
}

##bootstrap se##
se <- function(d,indices) {
d <- d[indices,]
fit <- glm(y~., family = poisson(link = "log"), data = d)
return(coef(fit))
}

bse <- boot(
data = sim.data,
statistic = se,
R = 100
)
z.stat<-(summary(y.rv)$coefficients[2,1])/(summary(bse)$bootSE[2])
pval<-2*(1-pnorm(z.stat))
if (pval<0.05) {
bs[i]<-1
} else {
bs[i]<-0
}

}

p.qp[j]<-sum(qp == 1)/1000
p.nb[j]<-sum(nb == 1)/1000
p.rv[j]<-sum(rv == 1)/1000
p.bs[j]<-sum(bs == 1)/1000
```

```
}

power<-t(cbind(t(p.qp),t(p.nb),t(p.rv),t(p.bs)))
model<-c(rep("qp",4),rep("nb",4),rep("rv",4),rep("bs",4))
size<-rep(c(5,100,500,1000), 4)
alpha<-rep(0.1)

###Data Analysis

data.power<-cbind(data.power,data.frame(alpha,model,size,power))
View(data.power)

dia2<-dia[c(-5,-6,-7)]
View(dia2)

##qp##
fit<-glm(price~.,family = quasipoisson(link = "log"), data = dia2)
summary(fit)

##nb##
fit<-glm.nb(price~.,init.theta = 1, data = dia2)
summary(fit)


##rv##
fit<-glm(price~.,family = poisson(link = "log"), data = dia2)
summary(fit)

data.frame(as.vector(sqrt(diag(vcovHC(fit, type="HC0")))
))
##bs##
se <- function(dia2,indices) {
d <- d[indices,]
fit <- glm(price~., family = poisson(link = "log"), data = dia2)
return(coef(fit))
}
bse <- boot(
data = dia2,
statistic = se,
R = 100
)
bse
```