# poisson_reg

*meme girls*

*4/18/2019*

## introduction

Generalized linear model plays a critical role in categorical data analysis. And Poisson regression model is one special glm designed to deal with the count data. In our project, we are particularly interested in predicting the number of reviews for each Airbnb host, which can be regarded as a kind of count data and thus may be applied with the Poisson regression model.

### generalized linear model

As the name suggests, glm is one kind of generalization of the traditional linear regresion model. Or we can say the classic linear regression model is a special case of the glm when the random component is the normal distribution. More formally, we define the glm as follow, which consists of 3 parts:

1. The distribution of the response is from the exponential distribution family, which means it has the form: $f(y_i|\theta_i) = a(y_i)b(\theta_i)exp(y_i\theta_i)$
2. The systenatic component: $\eta_i = \Sigma_{i=1}^{p}\beta_i x_i$
3. A link fucntion which links the systematic component and the mean of the response.

### poisson regression model

As we have mentioned before, Poisson regresssion model is a specila glm. More speciffical y, when we assume the response has a poisson distribution and the link function is the canonical link. Mathematically, we write the pdf of the response as:

$f(y_i|\mu_i) = e^{-\mu_i}\frac{\mu_i^{y_i}}{y_i!}$

and the canonical link has the form :

$\eta_i = log(\mu_i)$

And this is our Poisson regresion model or log-linear model.

### quasi-poisson regression model

Usually, Poisson regression model is simple and thus it has several drawbacks. One common deficit of Poisson regression model is that it cna't deal with a phenomena called overdispersion. An overdispersion occurs when there the varaice of true response apperae to be larger than that of the predicted response. Thus, it is natura for us to come up with more complicated model to tackle this issue. For example, we can use the quasi-poisson regression model.

To introduce the quasi-likelihood regression, we will briefly talk about the likelihhod equation of the glm first. And more generally, we assume that our response is from the exponential dispersion family here, which has the form: $f(y_i|\theta_i,\phi) = exp(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi))$. Then, the log-likeihood function is:

$l = \Sigma_{i=1}^{n}\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + \Sigma_{i=1}^{n}c(y_i,\phi)$

By taking derivative w.r.t $\beta$, the likelihhod equation is:

$\Sigma_{i=1}^n \frac{(y_i-\mu_i)x_{ij}}{var(Y_i)} \frac{\partial\mu_i}{\partial\eta_i} = 0, j = 1, ..., p.$

Moreover, the likelihood equation for the POisson regression is:

$\Sigma_{i=1}^n \frac{(y_i-\mu_i)x_{ij}}{\mu_i} \frac{\partial\mu_i}{\partial\eta_i} = 0, j = 1, ..., p.$

For the quasi-Poisson regression, we change the above euqation slightly by introducing the parameter $\phi$:

$\Sigma_{i=1}^n \frac{(y_i-\mu_i)x_{ij}}{\phi\mu_i} \frac{\partial\mu_i}{\partial\eta_i} = 0, j = 1, ..., p.$

This is so-called quisi-Possion regression model.

### negative binomial regression model

Another option to deal with the overdispersion issue is the negative binomail regression model. That's to say, we will assume the response has a negative binomail distribution. More specifically, if we assume the parameter $\mu$ for the poisson distribution has a gamma idstribution $\Gamma(r, \frac{1-p}{p})$, we can show that the posterior distribution is exactly a negative binomail distribution with parameter $r$ and $p$, $Y \sim NegBinom(r, p)$.

### zero inflated poisson regresision model

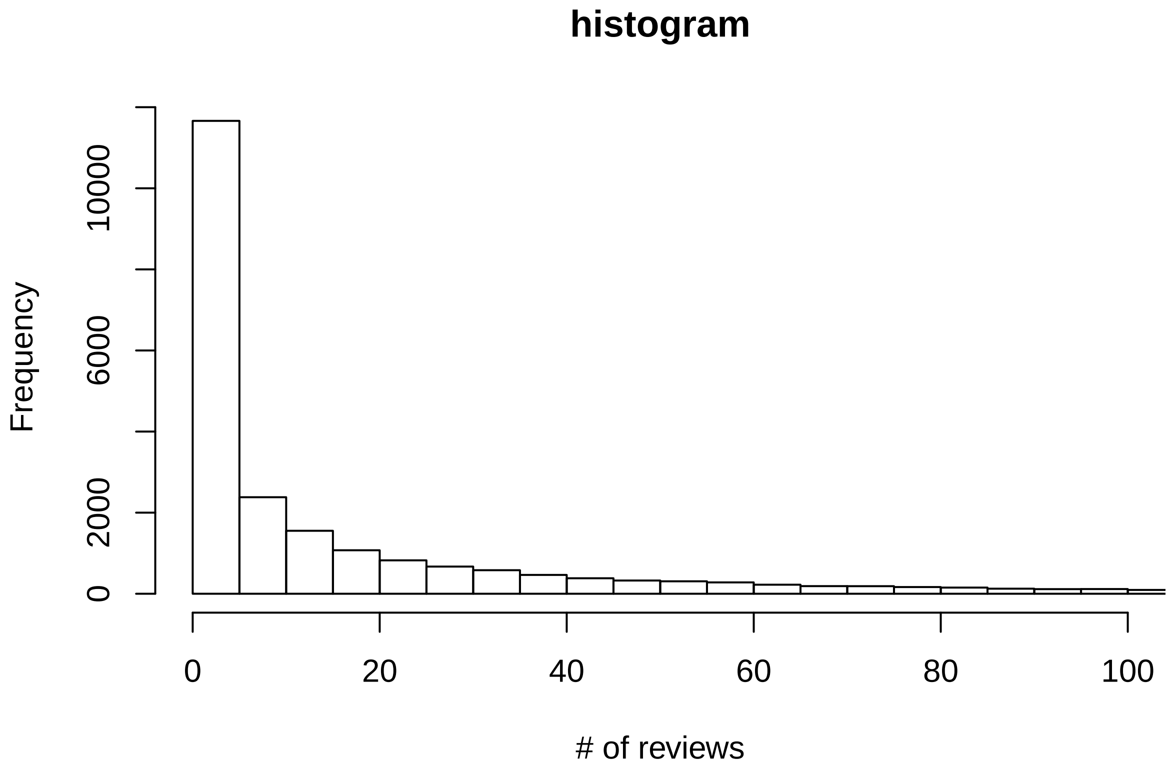
In real life, not all count data beahve exaclty like a poisson distribution. More often than not, it is common to see 0 takes a majority of the data which we call it zero inflation, like what we encountered in our project. To take this into consideration, people proposed teh zero inflated poisson regression or ZIP model. The idea is that we slightly change the response distribution by including an additional parameter $\pi_i$ called probability of extra zeros:

$\mathbb{P}(y_i = 0) = \pi_i + (1 - \pi_i)e^{-\mu_i}$,

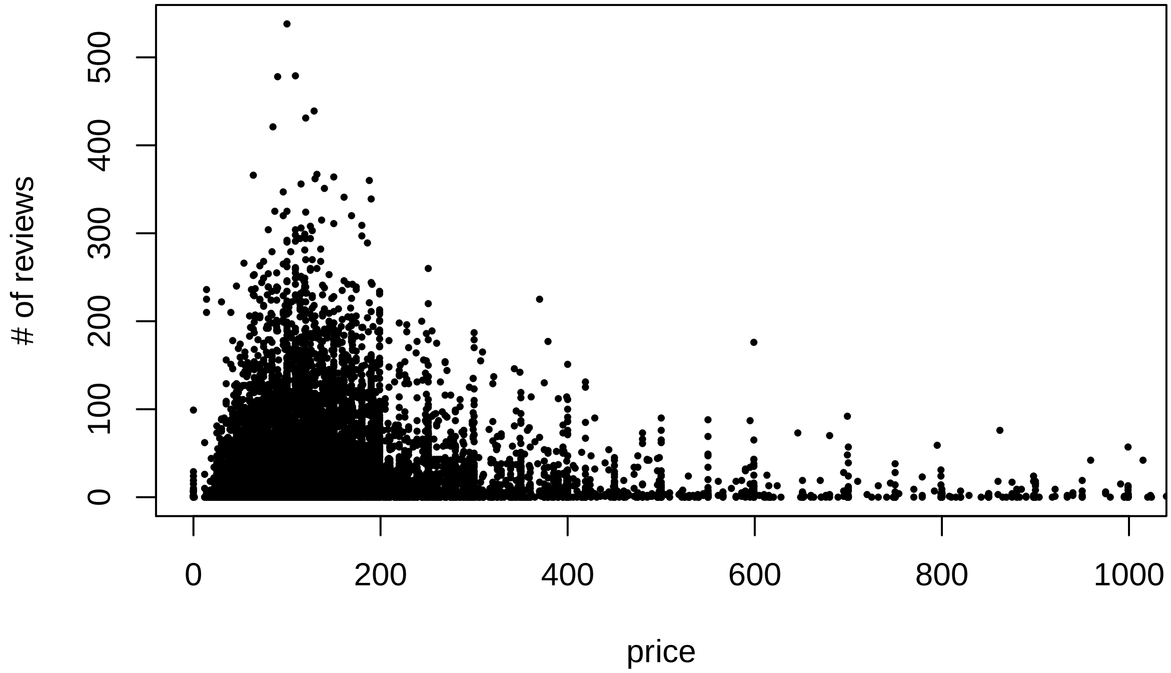
$\mathbb{P}(y_i = k) = (1 - \pi_i)e^{-\mu_i}\frac{\mu_i^k}{k!}$.

## exploratory data analysis

```r
# load data
data <- read.csv("data/listings_summary_dec18.csv")
data <- data[6:16]
data <- data[-8]
hist(data$number_of_reviews, breaks = 100, xlim = c(0, 100),
     xlab = "# of reviews", main = "histogram")
```



```
plot(data$price, data$number_of_reviews, xlim = c(0, 1000), pch=16,
     cex=0.5, xlab = "price", ylab="# of reviews")
```



# model building and analysis

```
# #summary(data$price)
# hist(data$price, breaks = 1000, xlim = c(0, 500))
#
# y <- data$number_of_reviews/20
# glm(formula = y ~ data$availability_365, family = poisson)
#
#
# poi <- glm(data$number_of_reviews ~ data$availability_365, family = poisson)
#
# glm(formula = data$number_of_reviews ~ data$room_type, family = poisson)
#
# x <- data$price/100
# glm(y ~ x, family = poisson)
#
# # missing data
# # glm(formula = data$reviews_per_month ~ data$availability_365, family = poisson)
```

## fitting poisson regression

We first try to fit the simplest model–the poisson regression model with number of reviews as the response and price as the predictor.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
library(glmnet)
```

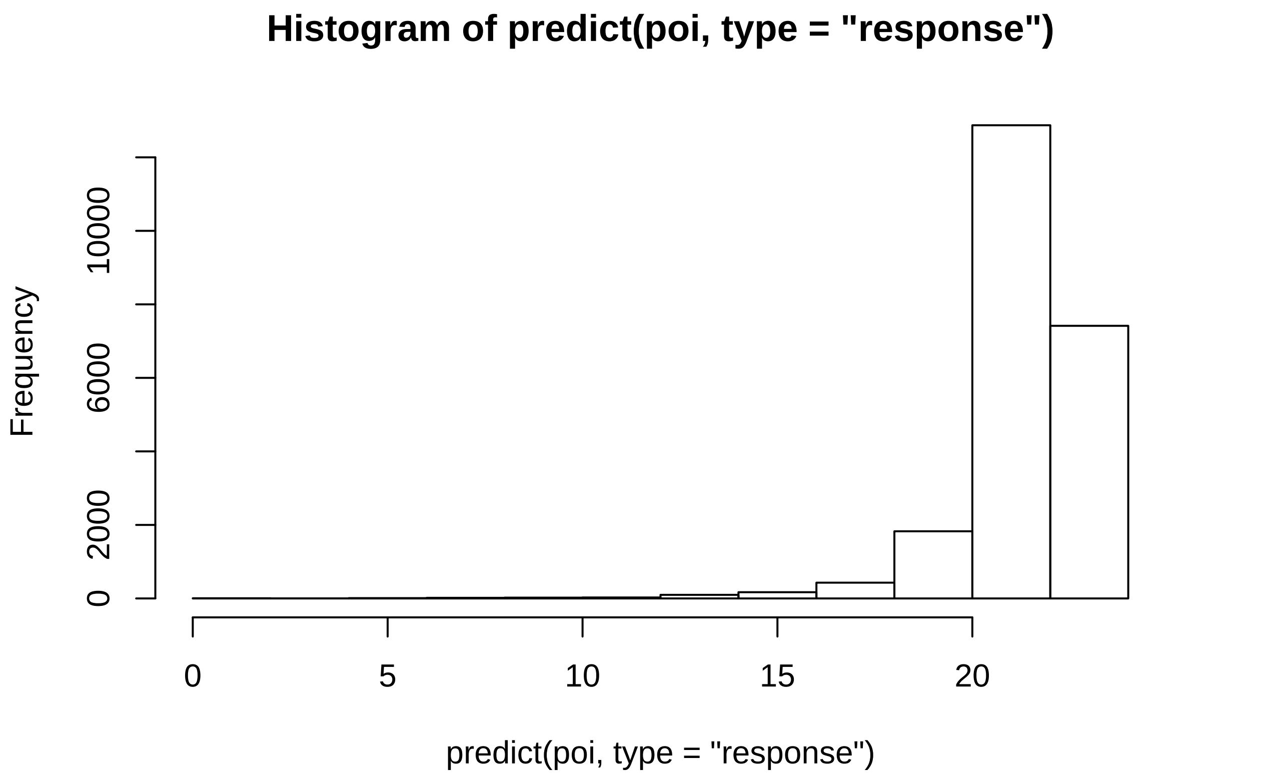
```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16
```

```
poi <- glm(number_of_reviews ~ price, family = poisson, data = data)
summary(poi)
```

```
##
## Call:
## glm(formula = number_of_reviews ~ price, family = poisson, data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -6.785  -5.990  -4.148   0.462  49.185
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.136e+00  2.130e-03 1472.38   <2e-16 ***
## price       -5.627e-04  1.164e-05  -48.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
##     Null deviance: 1016394  on 22894  degrees of freedom
## Residual deviance: 1013517  on 22893  degrees of freedom
## AIC: 1087884
##
## Number of Fisher Scoring iterations: 6
```

```
hist(predict(poi, type = "response"))
```



```
var(predict(poi, type = "response"))
```

```
## [1] 2.307561
```

The result above shows that the model is already significant. It can be written as:

$log(\mu) = 3.14 - 5.63 * 10^{-4} * x$.

It suggests there is a negative relationship between number of reviews and the price. We then try to compare the model with the saturated model using deviance.

```
poi_sat <- glm(number_of_reviews~1, family=poisson, data=data)
anova(poi_sat, poi, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: number_of_reviews ~ 1
## Model 2: number_of_reviews ~ price
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1     22894    1016394
## 2     22893    1013517  1   2876.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The analysis of deviance suggests that the p-value is small enough for us to accept the alternative that the number of reviews drops as the price increses.

We then test teh goodness-of-fit of pur model, which sugest tehre is a lack of fit.

```r
# goodness of fit test
1-pchisq(poi$deviance, poi$df.residual)
```

```
## [1] 0
```

## fitting quasi-poisson regression

We notice that there is more variabiity of the predicted response from our previous model than the actual response, which suggests the phenomena of overdisperison. And we then try to fit the quasi likelihood version poisson regression model.

```r
quasi_poi <- glm(number_of_reviews~price, family=quasipoisson, data=data)
summary(quasi_poi)
```

```
##
## Call:
## glm(formula = number_of_reviews ~ price, family = quasipoisson,
##     data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -6.785  -5.990  -4.148   0.462  49.185
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.136e+00  1.798e-02 174.454  < 2e-16 ***
## price       -5.627e-04  9.823e-05  -5.729 1.02e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 71.23225)
##
##     Null deviance: 1016394  on 22894  degrees of freedom
## Residual deviance: 1013517  on 22893  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

```r
1-pchisq(quasi_poi$deviance, quasi_poi$df.residual)
```

```
## [1] 0
```

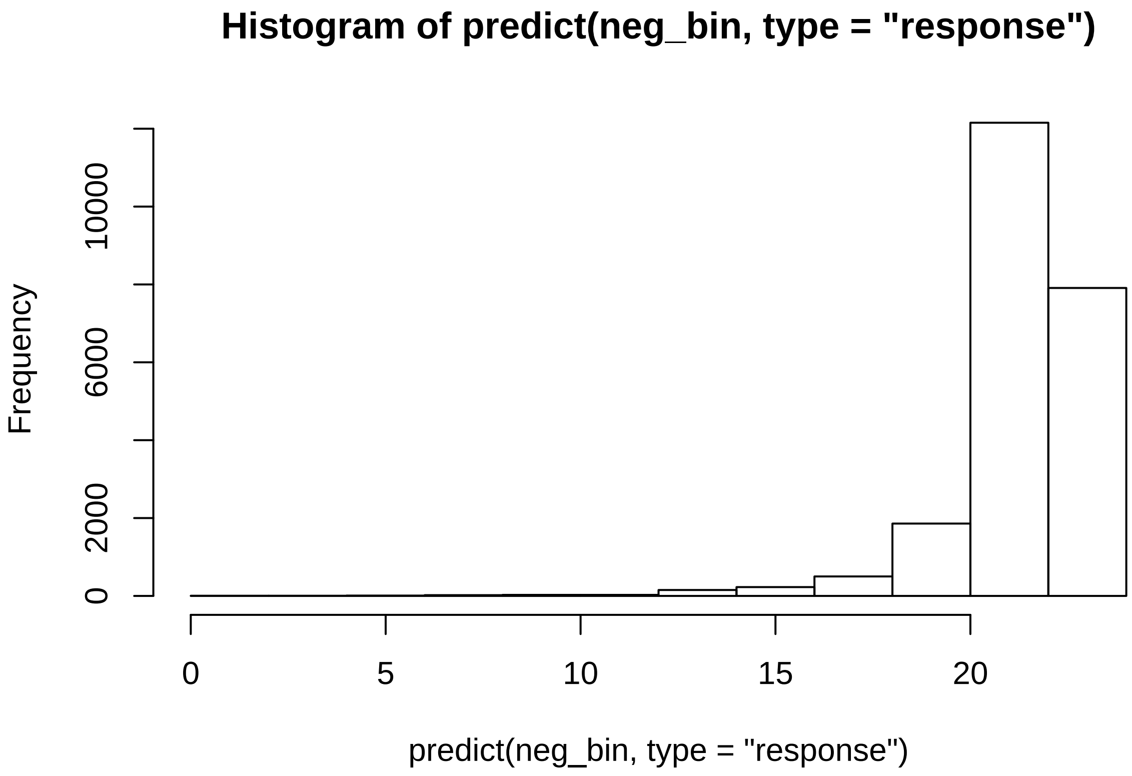
## fitting negative binomial regression

```r
library(MASS)
neg_bin <- glm.nb(number_of_reviews~price, data=data)
summary(neg_bin)
```

```
##
## Call:
```

```
## glm.nb(formula = number_of_reviews ~ price, data = data, init.theta = 0.3380764171,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6943  -1.1679  -0.6641   0.0560   3.7035
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.147e+00  1.547e-02 203.429   <2e-16 ***
## price       -6.433e-04  7.119e-05  -9.037   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3381) family taken to be 1)
##
##     Null deviance: 26506  on 22894  degrees of freedom
## Residual deviance: 26439  on 22893  degrees of freedom
## AIC: 169449
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.33808
##           Std. Err.:  0.00312
##
##  2 x log-likelihood:  -169443.12000
```

```
hist(predict(neg_bin, type="response"))
```



### fitting ZIP model

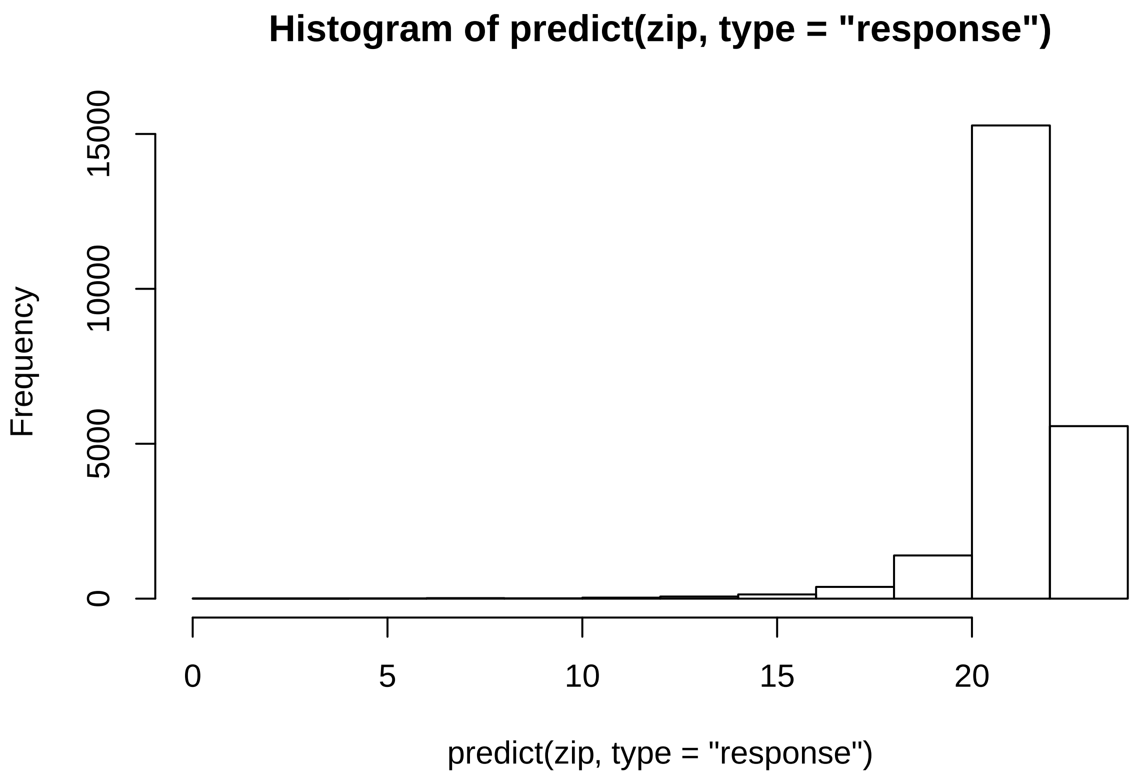
```r
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
zip <- zeroinfl(number_of_reviews ~ price, data = data)
summary(zip)
```

```
## 
## Call:
## zeroinfl(formula = number_of_reviews ~ price, data = data)
## 
## Pearson residuals:
##     Min      1Q  Median      3Q     Max 
## -1.8076 -1.6502 -1.2778  0.1692 58.4882 
## 
## Count model coefficients (poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  3.355e+00  1.450e-03  2313.4   <2e-16 ***
## price       -2.864e-04  5.973e-07  -479.4   <2e-16 ***
## 
## Zero-inflation model coefficients (binomial with logit link):
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.339e+00  2.099e-02  -63.83   <2e-16 ***
## price        7.754e-04  9.037e-05    8.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Number of iterations in BFGS optimization: 9 
## Log-likelihood: -4.315e+05 on 4 Df
```

```r
hist(predict(zip, type="response"))
```


Histogram of predict(zip, type = "response")
Frequency
0
5000
10000
15000
0
5
10
15
20
predict(zip, type = "response")

### conclusion