MKT 500T

Spring 2018 HW2

Sami Cheong

Feb 04 2018

Q1. Modeling count data

Data

We are given a sample of Internet visit data for a set of 2728 individuals. A quick first look at the data tells us this data set has a high proportion of 0 visitors:

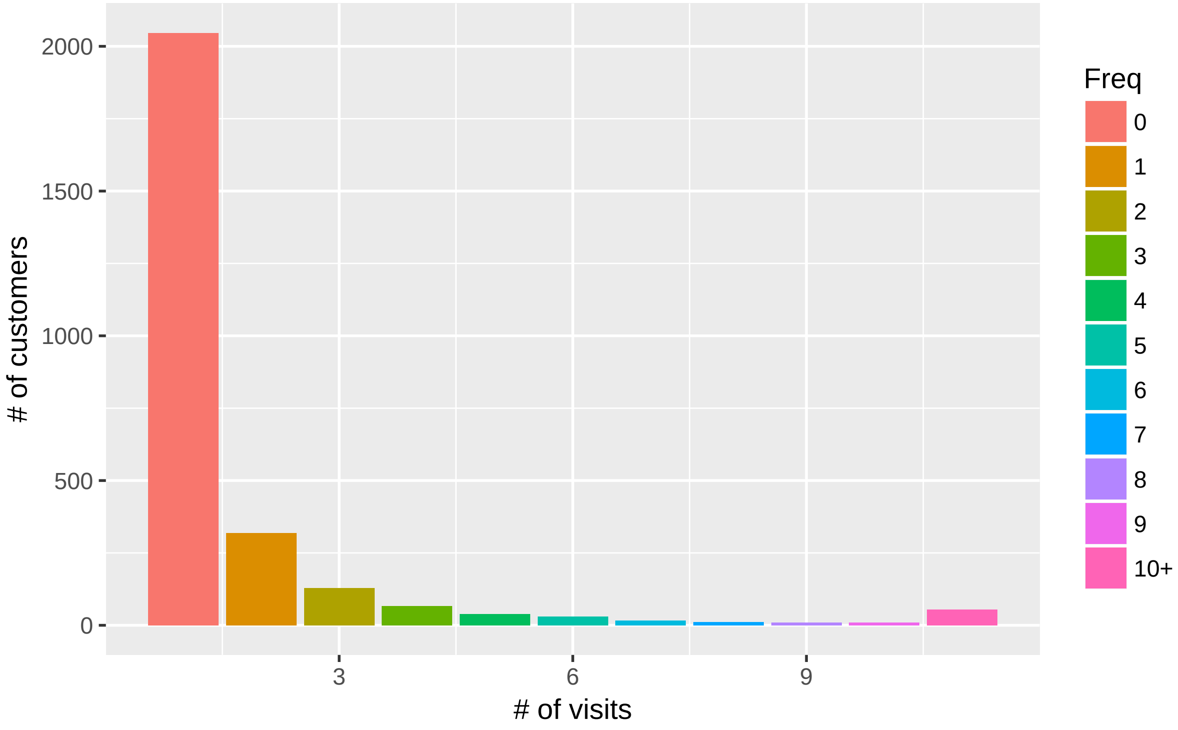
```
library(ggplot2)
library(knitr)
library(readxl)
library(dplyr)
data<-read_excel('data/khakichinos HW2 data.xlsx',sheet = 1)

data[,'Freq']<-ifelse(data$Visits <10, data$Visits,'10+')

data_grouped<-data%>%group_by(Freq)%>%
  summarise(total_visits=n())

data_grouped$Freq<-factor(data_grouped$Freq, levels = c(0:9,'10+'))

ggplot(data_grouped)+geom_col(aes(x=as.integer(Freq),y=total_visits,fill=Freq))+
  labs(x='# of visits',y='# of customers')
```



The goal of this question is to model the above data using the Poisson Distribution and Negative Binomial distribution, as well as their zero-inflated versions. In geneal, to estimate the parameters for each of the distribution, we start with a set of initial values, then use the `optim` function in R to solve for the optimal parameters with respect to the likelihood function that are computed for every combination of parameter values and the observations given. Below is the implementation of calculating log-likelihood function for a given pmf:

```r
getLL<-function(count.data,pmf,par,debug=F){

  pmf.val<-pmf(count.data,par)

  LL.val<-sum(log10(pmf.val))
  if(debug){
    plot(count.data,pmf.val,type = 'l')
    print(pmf.val)
    print(log10(pmf.val))
    print(LL.val)
  }

  return(LL.val)
}
```

1(a) Poisson Distribution

Our first task is to model the data using the Poisson distribution. The discrete distribution has a probability mass funciton definted as

$$p(X = x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \lambda > 0, x = 0, 1, 2, 3....$$

The code to generate the Poisson distribution is the implemented below:

```
POIS.pmf<-function(count.data,par){
  N<-length(count.data)
  lambda<-par[1]
  pmf.val = vector("numeric",N)
  for (n in 1:N){
    x<-count.data[n]
    pmf.val[n]<-ifelse(lambda>0, (lambda^x)*exp(-lambda)/(factorial(x)),0)
  }
  return(pmf.val)
}

Poisson.par.opt<-optim(par=2.5,fn=getLL,pmf=POIS.pmf,debug=F,
                       count.data =data$Visits, method = 'L-BFGS-B',
                       control=list(fnscale=-1),lower=1e-10,hessian = T)
print(paste('Optimal parameters:',Poisson.par.opt$par))
```

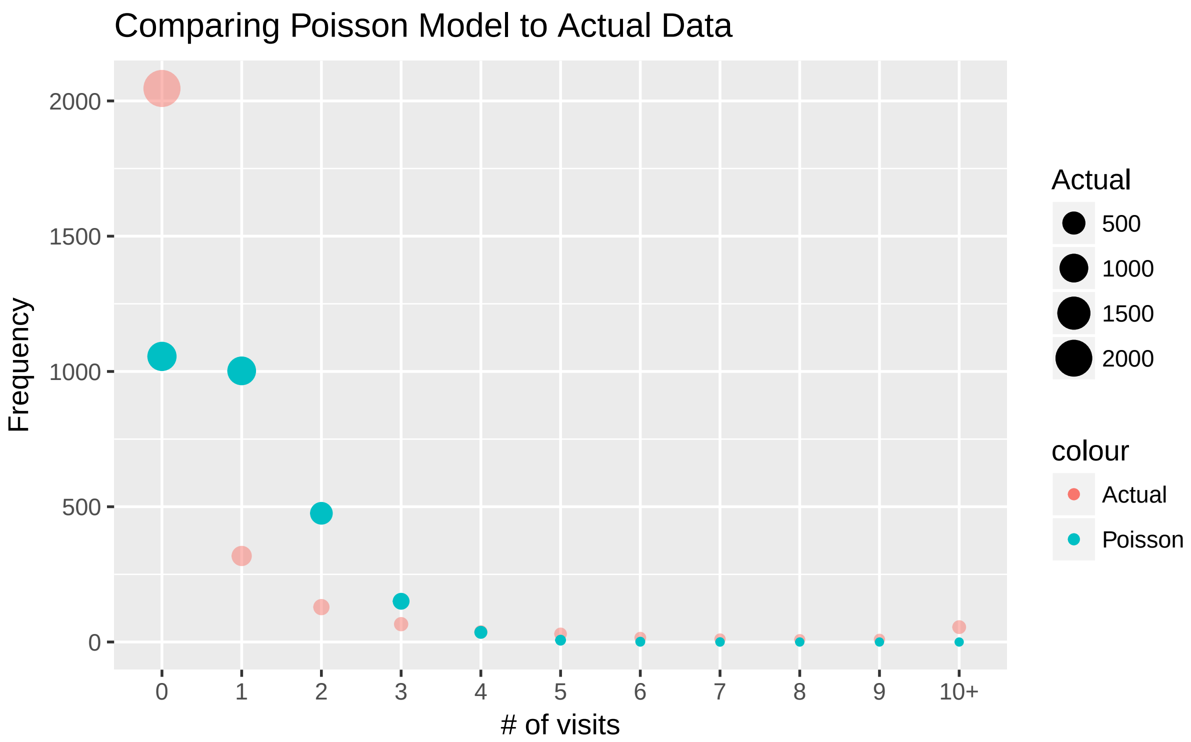
```
## [1] "Optimal parameters: 0.949413838463139"
```

```
print(paste('Log-likelihood:', Poisson.par.opt$value))
```

```
## [1] "Log-likelihood: -2770.19480079434"
```

After optimizing, the λ value for the best-fit Poisson distribution is 0.9494138, so we use that to compute the probability of visits at different frequencies:

```
df<-nrow(data)-length(Poisson.par.opt$par)-1
data[,'PoissonFit']<-POIS.pmf(data$Visits,par=Poisson.par.opt$par)
data[,'PoissonLL']<-log10(data$PoissonFit)
data[,'Chi2Pois']<-(data$PoissonFit-data$Visits)^2/(data$PoissonFit)
#kable(head(data[,c('Visits','PoissonFit','PoissonLL','Chi2Pois')]),align = 'c')
```



```
## [1] "ChiSquare  for  Poisson is  173169442.322283 p-val is 0  with  9  df"
```

Freq	Actual	Expected	Chi2
0	2046	1055.6481110	9.290945e+02
1	318	1002.2469251	4.671442e+02
2	129	475.7735501	2.527503e+02
3	66	150.5686642	4.749899e+01
4	38	35.7379933	1.431718e-01
5	30	6.7860291	7.941146e+01
6	16	1.0737917	2.074813e+02
7	11	0.1456390	8.089673e+02
8	9	0.0172840	4.668445e+03
9	10	0.0018233	5.482586e+04
10+	55	0.0000175	1.731072e+08

Overall, the Poisson distribution does not seem to be a very good fit. Visually, we can see that the model tends to over-estimate in ther smaller range and under estimate in the higher range. The not-so-good performance of the Poisson model is also shown in the high χ^2 value and low p-value, which rejects the probability that the model and the actual data comes from the same distribution.

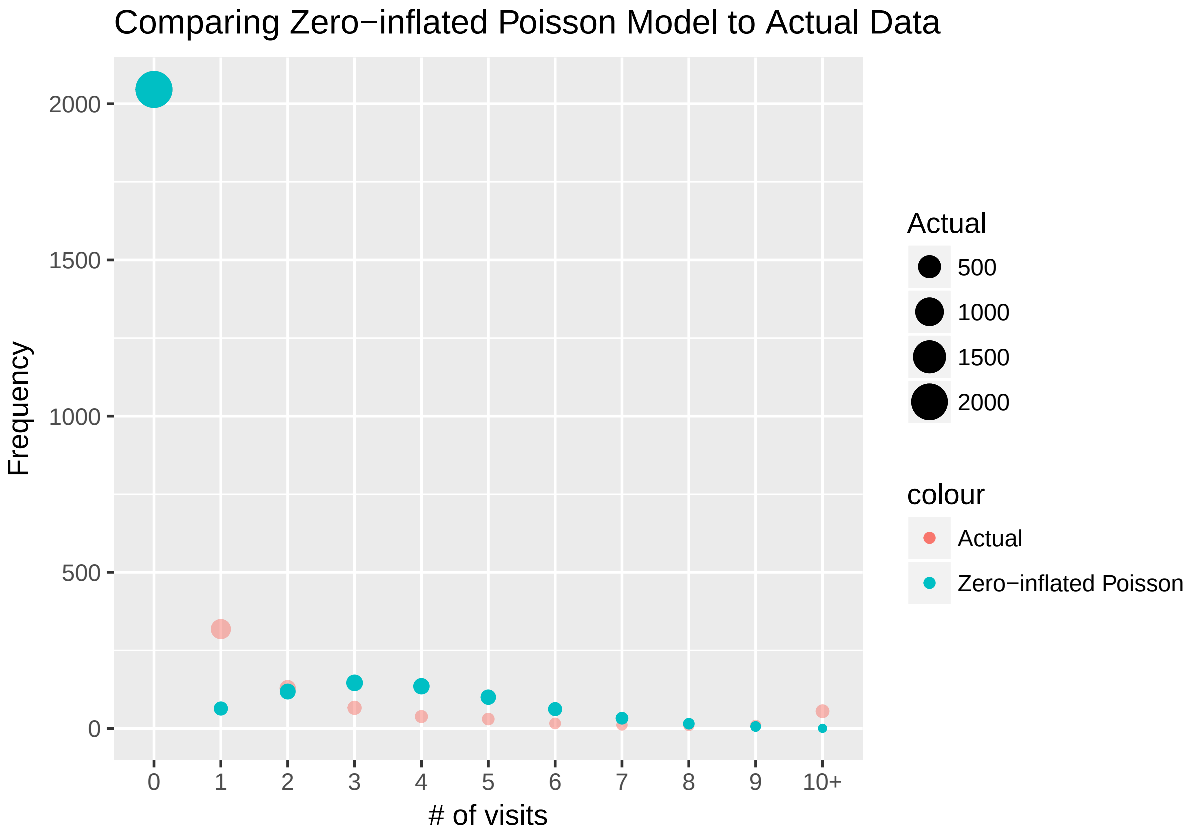
1(b) Poisson Distribution with Zero-Spikes

The zero-inflated Poisson distribution is defined as

$$P(X=x) = \begin{cases} p_0 + (1-p_0)Poi(\lambda) & \text{if x = 0} \\ (1-p_0)Poi(\lambda) & \text{else} \end{cases}.$$

```
## [1] "Optimal parameters for zero-inflated Poisson Distribution:"
```

```
## [1] "Log-likelihood is : -1870.05185645115"
```



Freq	Actual	Expected	Chi2
0	2046	2045.9978876	0.000000
1	318	63.7687060	1013.562213
2	129	118.1045657	1.005130
3	66	145.8258479	43.697096
4	38	135.0403632	69.733462
5	30	100.0420704	49.038286
6	16	61.7618788	33.906830
7	11	32.6822191	14.384538
8	9	15.1324981	2.485216
9	10	6.2281232	2.284324
10+	55	0.3179895	9403.210867

After adjusting for the spike at zero, the Zero-inflated Poisson seems to be doing a much better job at capturing the # of people that will never visit the site. However, it is still showing quite a bit of error in the mid-range, as evidenced by the χ^2 values.

1(c) Negative Binomial Distribution

The NBD is defined as

$$P(X = x|\alpha, r) = \frac{\Gamma(x+r)}{x!\Gamma r}(\frac{\alpha}{1+\alpha})^r(\frac{1}{1+\alpha})^x$$

. The parameters that result in the highest log-likelihood values are 0.13387 and 0.14101 respectively for r and α.

```
NBD.pmf<-function(count.data,par){
  N<-length(count.data)
  pmf.val = vector("numeric",N)
  for (n in 1:N){
    x<-count.data[n]
    gamma.top<-(gamma(x+par[1]))
    gamma.bot<-factorial(x)*gamma(par[1])
    p1<-par[2]/(1+par[2])
    p2<- 1-p1
    pmf.val[n]<-(gamma.top/gamma.bot)*(p1^par[1])*(p2^x)
  }
  return(pmf.val)
}



NBD.par.opt<-optim(par=c(0.1,0.1),fn=getLL,pmf=NBD.pmf,
                   count.data =data$Visits, method="L-BFGS-B",
                   control=list(fnscale=-1), lower=1e-10)

print('Optimal paramters for NBD:')
```

```
## [1] "Optimal paramters for NBD:"
```

```
print(NBD.par.opt$par)
```

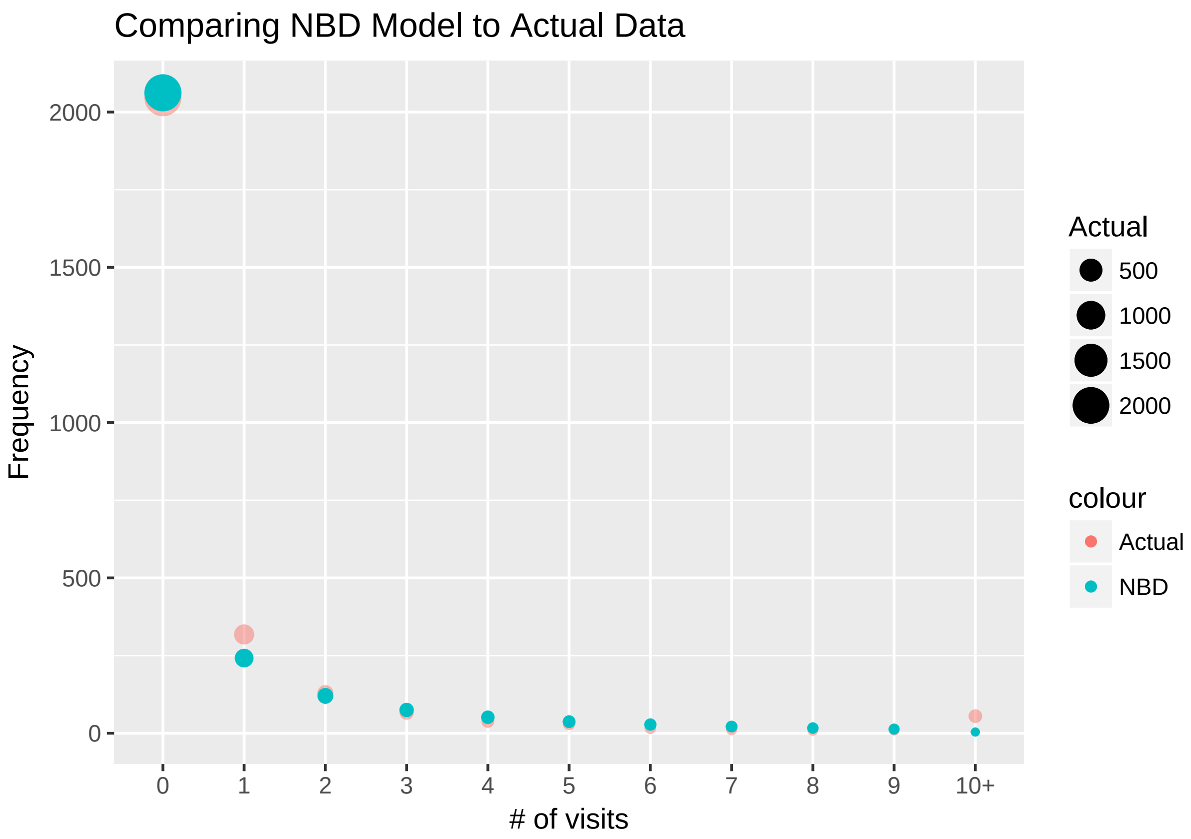
```
## [1] 0.1338703 0.1410047
```

```
print('Log-likelihood:')
```

```
## [1] "Log-likelihood:"
```

```
print(NBD.par.opt$value)
```

```
## [1] -1261.897
```



Freq	Actual	Expected	Chi2
0	2046	2061.972709	0.1237298
1	318	241.924466	23.9227019
2	129	120.205888	0.6433663
3	66	74.935061	1.0653933
4	38	51.453941	3.5178751
5	30	37.283618	1.4229061
6	16	27.959168	5.1153776
7	11	21.472042	5.1072770
8	9	16.781127	3.6079781
9	10	13.291941	0.8152969
10+	55	3.883654	672.7893227

Comparing to the Poisson models, the Negative Bionomial distribution does a much better job at fitting this set of data (as seen in the lower overall χ^2 value). However, we are still seeing a few parts of the data where the model is not predicting a good enough value, for example, at frequency 1, and 10+.

1(d) Zero-inflated NBD:

Similar as above, we impose an additional parameter in the zero-inflated NBD by defining

$$P(X = x) = p_0 + (1 - p_0) * NBD(x|r, \alpha).$$

This gives us almost exactly the same parameter values for r and α, with $p_0 = 1e^{-10}$, which is virtually zero.

```
NBD.pmf.zero<-function(count.data,par){
 p0 <-par[3]/(1+par[3])

 #p0 <-par[3]
 r<-par[1]
 alpha<-par[2]
 zero.case<-p0+(1-p0)*NBD.pmf(count.data,c(par[1],par[2]))
 nonzero.case<-(1-p0)*NBD.pmf(count.data,c(par[1],par[2]))
 pmf<-ifelse(count.data==0,zero.case,nonzero.case)
 #print(pmf)
 #print(pmf)
 return(pmf)
}

NBDZ.par.opt<-optim(par=c(0.2,0.2,5),fn=getLL,pmf=NBD.pmf.zero,
                    count.data =data$Visits, method="L-BFGS-B",
                    control=list(fnscale=-1), lower=1e-10)

print('Optimal paramters for zero-inflated NBD:')
```

```
## [1] "Optimal paramters for zero-inflated NBD:"
```

```
print(NBDZ.par.opt$par)
```

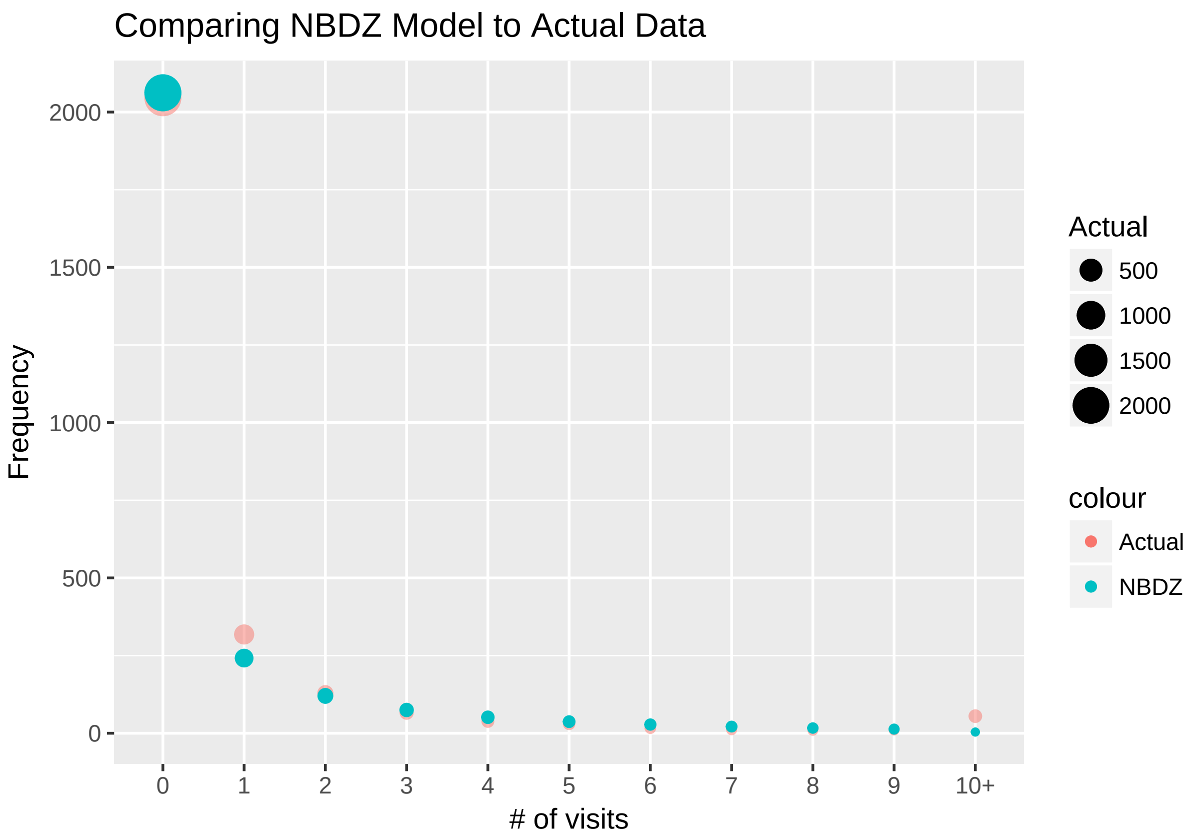
```
## [1] 0.1338703343 0.1410051955 0.0000000001
```

```
print('Log-likelihood:')
```

```
## [1] "Log-likelihood:"
```

```
print(NBDZ.par.opt$value)
```

```
## [1] -1261.897
```



Freq	Actual	Expected	Chi2
0	2046	2061.973471	0.1237415
1	318	241.924471	23.9226986
2	129	120.205842	0.6433732
3	66	74.935002	1.0653801
4	38	51.453880	3.5178474
5	30	37.283559	1.4228852
6	16	27.959112	5.1153401
7	11	21.471991	5.1072390
8	9	16.781080	3.6079447
9	10	13.291899	0.8152784
10+	55	3.883635	672.7930141

Comments on Question 1 results:

We've tried 4 different models, while it is clear that the Negative Binomial Distribution provides a better fit compared to the Poisson model, we are still seeing cases where the χ^2 is higher that we'd like. This could be due to the fact that we only have one feature from the data set to work with. Another interesting observation is that the zero-inflated NBD gives very similar result to the original NBD, this gives evidence that the NBD model by itself already does a fairly good job at handling the 0 visit caes, so adjustment in this case is not necessary.

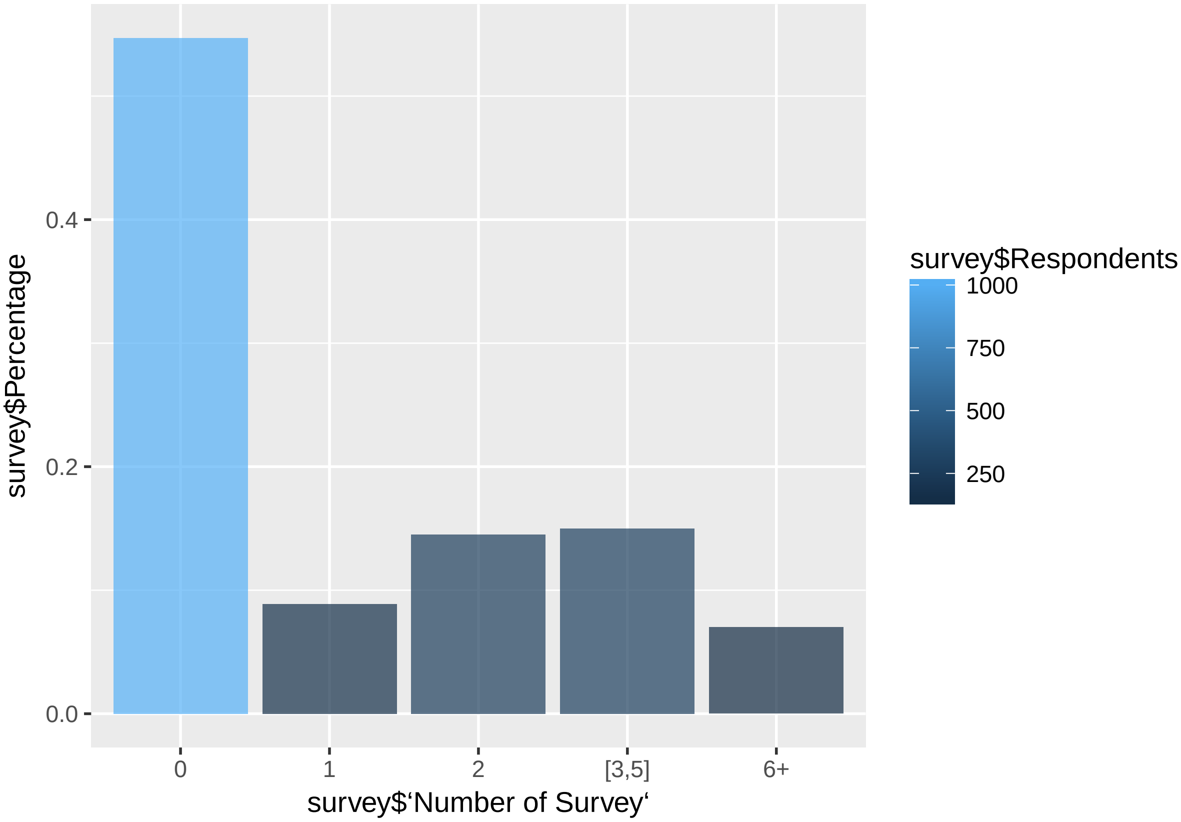
Question 2: Modeling Survey Data

Data Overview:

```
library(ggplot2)
survey<-read_excel('data/Survey HW2.xlsx',sheet = 'Raw Data')
kable(survey)
```

Number of Survey	Respondents	Percentage
0	1020	0.547
1	166	0.089
2	270	0.145
[3,5]	279	0.150
6+	130	0.070

```
survey$`Number of Survey`<-factor(survey$`Number of Survey`,levels=c(0,1,2,'[3,5]','6+'))
ggplot(survey)+geom_col(aes(x=survey$`Number of Survey`,y=survey$Percentage,fill=survey$Respondents),
```

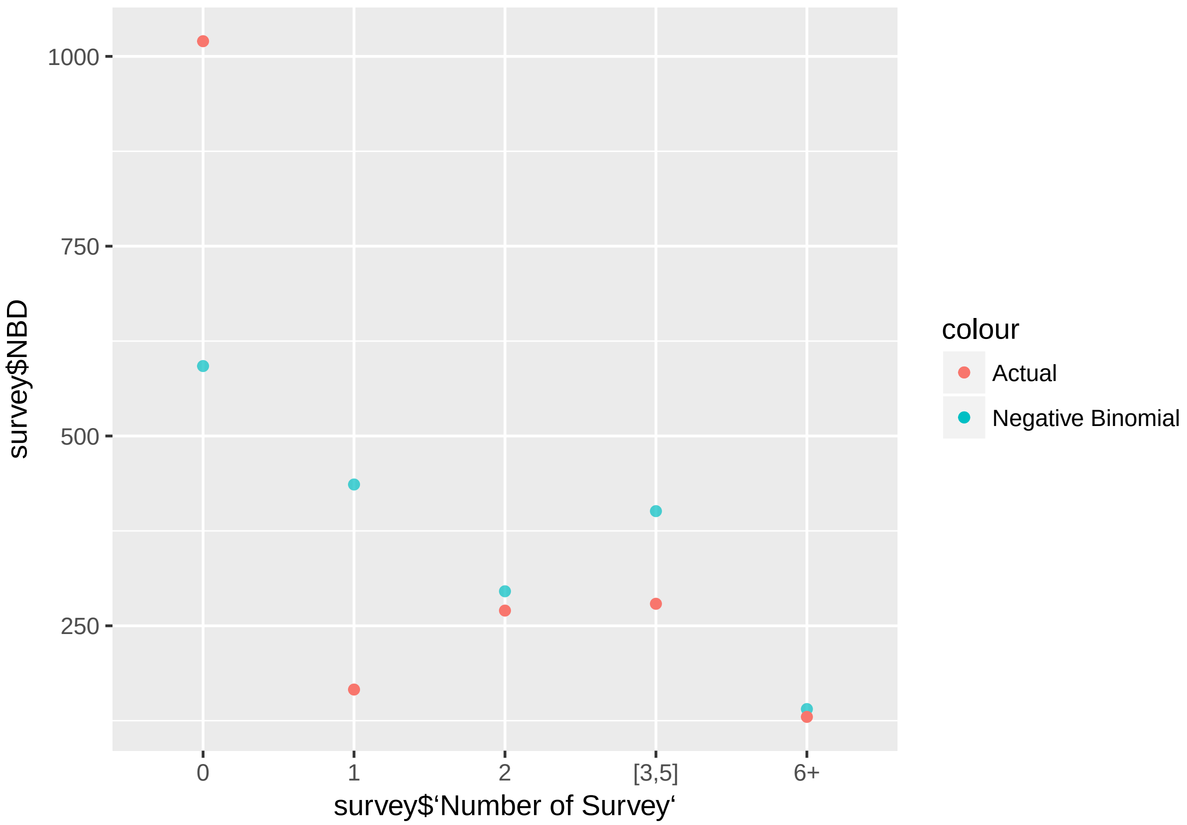


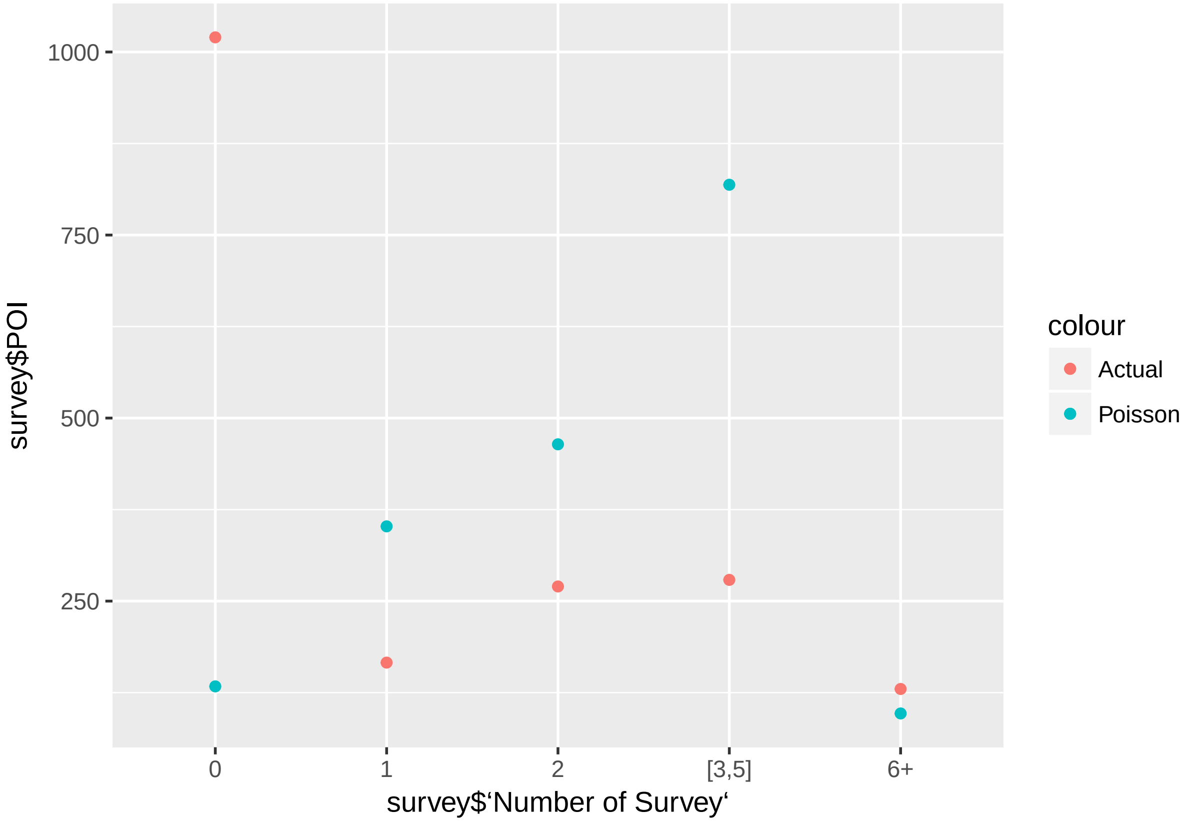
In this data set, we can use the recursive formula for NBD, which is defined as

$$P(X=x)=\begin{cases}(\frac{\alpha}{\alpha+1})^r & x=0\\ \frac{r+x-1}{x(\alpha+1)}P(X=x-1) & x=1,2,3..\end{cases}$$

Using the above formula, we can calculate the probability all the x values, and the log-likelihood is defined as:

$$LL=\sum_{k=1}^{2} n_k logP(X=k)+\log n_{3-5}\sum_{k=3}^{5}P(X=k)+\log n_{6+}(1-\sum_{k=0}^{5}P(X=k)$$


survey$NBD
1000
750
500
250
0
1
2
[3,5]
6+
survey$‘Number of Survey‘
colour
● Actual
● Negative Binomial


survey$POI
1000
750
500
250
0
1
2
[3,5]
6+
survey$‘Number of Survey‘
colour
● Actual
● Poisson