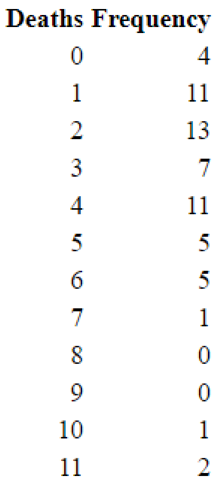
Stochastic Models - Assignment 1

Eveline Surbakti - 19200629

Topic: Stochastic Models for the Number of Deaths in Game of Thrones Episodes

Game of Thrones was a popular fantasy drama TV series known for its level of violence.
The provided dataset recording the number of deaths per episode for 60 episodes.
The data recorded are as follows:



Deaths	Frequency
0	4
1	11
2	13
3	7
4	11
5	5
6	5
7	1
8	0
9	0
10	1
11	2

1(a) Fit the Poisson model to the data using maximum likelihood.

x<-scan()

0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 6 6 6 6 6 7 10 11 11

```
#fitting the Poisson model to the data using maximum likelihood
loglikelihoodpoisson<-function(lambda,x)
{
  sum(dpois(x,lambda,log=TRUE))}

mean(x)

## [1] 3.266667

#set initial lambda into 3 to reach converge faster
lambda0<-3
fit0<-optim(par=lambda0,loglikelihoodpoisson,method="BFGS",x=x,control=list(fnscale=-
1),hessian=TRUE)

fit0

## $par
## [1] 3.266648
##
```

```
## $value
## [1] -135.9608
## 
## $counts
## function gradient 
##        9        4 
## 
## $convergence
## [1] 0
## 
## $message
## NULL
## 
## $hessian
##           [,1]
## [1,] -18.36756
```

(b) Fit the Poisson-Gamma model using maximum likelihood.

```
# Get log likelihood of the possion-gamma distribution
dPoissonGamma <- function(x,alpha,beta,log=TRUE)
{ if (log)
  {
    lognumer <- lgamma(x+alpha)+x*log(beta)
    logdenom <- lgamma(alpha)+(alpha+x)*log(1+beta)+lgamma(x+1)
    res <- lognumer-logdenom
  }else
  {
    numer <- gamma(x+alpha)*beta^x
    denom <- gamma(alpha)*(1+beta)^(alpha+x)*gamma(x+1)
    res <- numer/denom
  }
  res}

loglikelihoodpoissonGamma<-function(theta,x)
{ alpha<-theta[1]
  beta<-theta[2]
  sum(dPoissonGamma(x,alpha,beta,log=TRUE))}

alpha0<- 1
beta0<- mean(x)
theta0<-c(alpha0,beta0)

fit<-optim(par=theta0,loglikelihoodpoissonGamma,method="BFGS",x=x,control=list(fnscal
e=-1),hessian=TRUE)

## Warning in log(beta): NaNs produced
## Warning in log(beta): NaNs produced
## Warning in log(1 + beta): NaNs produced

#ignore the "NaNs producedNaNs producedNaNs produced$par"
fit

## $par
## [1] 4.3981351 0.7428105
## 
```

```
## $value
## [1] -130.1835
##
## $counts
## function gradient
##       41       21
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##            [,1]       [,2]
## [1,]  -6.084126  -34.42715
## [2,] -34.427154 -203.81422
```

(c) Actually both distributions are giving a good presentation of the data. However, Poisson-Gamma is a better model for the data compares to the Poisson. Here, we will see how well the data meets the assumptions of the model, by graphing the data.

```
hist(x, probability = TRUE, xlab = "Number of Deaths", ylim= c(0,0.30), col="darkoliv
egreen2", xlim = c(0,12), main = "Number of Deaths in Game of Thrones Episodes", brea
ks = 12)
points(x, dpois(x,fit0$par), type="l", col="black")
points(x, dPoissonGamma(x,fit$par[1],fit$par[2],F), type="l", col="red")
legend("right", legend=c("Poisson","Poisson-Gamma"),col= c("black","red"),lty = c(1,1
),cex=0.7,bty="n")
```

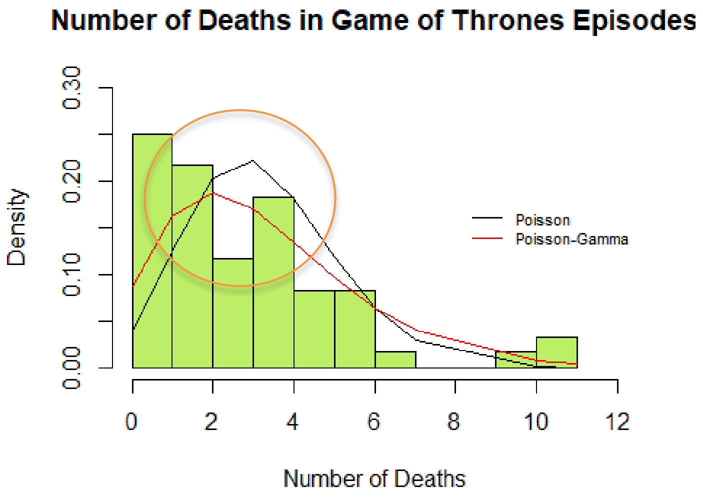


Figure 1 Comparing the Proposed Stochastic Models with the dataset

Let us focus to the visualization inside the circle shape; at the beginning, the probability is quite high, fitted model Poisson can detect this and get a higher position of line at the start, but it seems higher and shifts from the data. Meanwhile Poisson-Gamma adapts well with the data and takes the lower start while moving smoother rather than Poisson.

While we are doing the stochastic modeling, we will deal with the random events. With Poisson model we have the assumption that all of the observations have a constant expected rate of events. But that would not be the case in this modeling. We can expect that a few episodes would have a much higher death rate due to the plot of the story. The Poisson-gamma model allows for this difference in rates.

(d) In general, we can add more parameters to improve the model, but as the parsimony is still the main idea of fitting the model, AIC and BIC are the best way to assess the goodness of fit since they penalized-likelihood criteria.

Since we had fitted the models with maximum likelihood, the models can be assessed by comparing the results and the number of parameters. We need to balance the quality of fit and model complexity to find the simplest model which gives a good fit, to void overfitting the data.

From the table below, **Poisson-Gamma** model has the highest likelihood.

```
n=60
AIC0<-(2*fit0$value)-(2*1)
BIC0<-(2*fit0$value)-(log(n)*1)
AIC1<-(2*fit$value)-(2*2)
BIC1<-(2*fit$value)-(log(n)*2)

d <- cbind(Model = c("Poisson","Poisson-Gamma"),
           AIC = c(round(AIC0,2),round(AIC1,2)),
           BIC=c(round(BIC0,2),round(BIC1,2)))

knitr::kable(d)
```

Model	AIC	BIC
Poisson	-273.92	-276.02
Poisson-Gamma	-264.37	-268.56

```
inf0<- -fit0$hessian
SE0 = sqrt(diag(solve(inf0))); SE0
## [1] 0.233332
inf1<- -fit$hessian
SE1= sqrt(diag(solve(inf1))); SE1
## [1] 1.9284872 0.3331951 #note that the SE for Alpha is very large

CI for Poisson Lambda
fit0$par[1] + 1.96*SE0 ## [1] 3.723979
fit0$par[1] - 1.96*SE0 ## [1] 2.809317

CI for Poisson- Gamma Alpha
fit$par[1] + 1.96*SE1[1] ## [1] 8.17797
fit$par[1] - 1.96*SE1[1] ## [1] 0.6183002

CI for Poisson- Gamma Beta
fit$par[2] + 1.96*SE1[2] ## [1] 1.395873
fit$par[2] - 1.96*SE1[2] ## [1] 0.0897481
```

From calculation above, even though **Poisson-Gamma** model is the better model, it has a large standard error; this might be because of the small sample of data or the range of data.