Modelling insurance claims

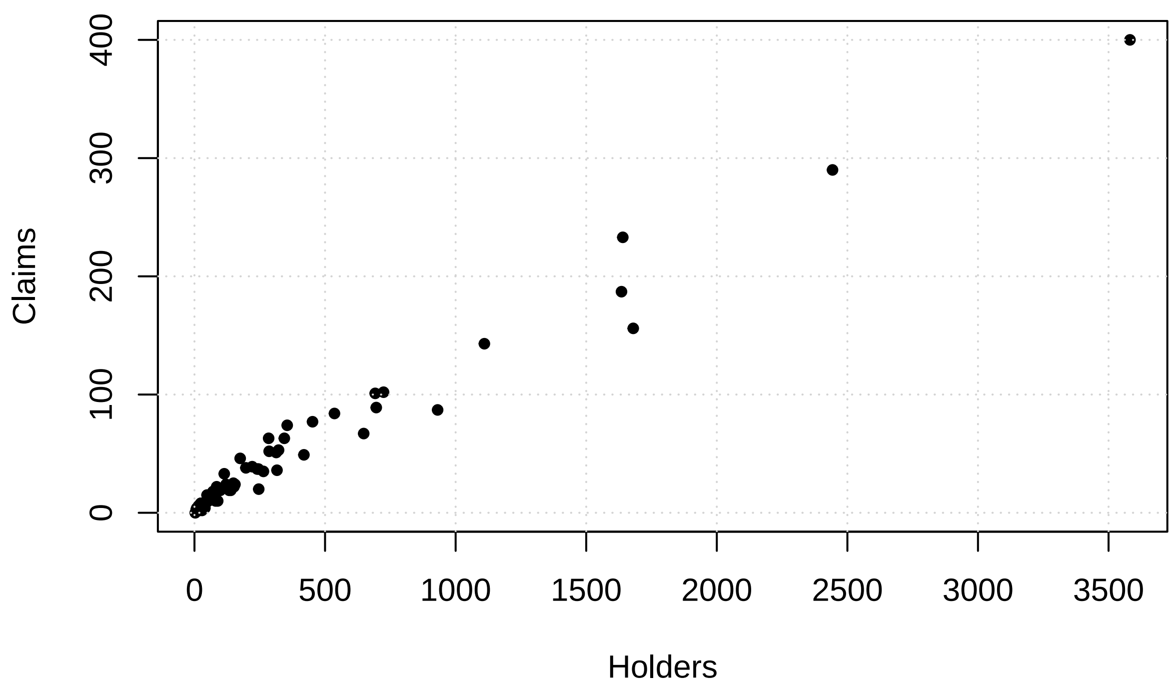
Problem 4

Problem solver: Rohan Karthikeyan

Reviewer: Adarsha Mondal

Let's load the dataset and plot a graph of the concerned variables:

```
library(MASS)
plot(Insurance$Holders, Insurance$Claims,
     xlab = 'Holders', ylab='Claims',
     pch=20)
grid()
```



Model 1: Linear regression with normally distributed errors

```
NLL_Model1 <- function(theta, y, X)
{
  beta_0 = theta[1]
  beta_1 = theta[2]
  sigma = exp(theta[3])  # variance of error terms

  n = length(data)
  # -ve log likelihood function of normal distribution
  l = -sum(dnorm(y, mean = beta_0 + beta_1*X,
                 sd = sigma, log=T))
  return(l)
}
```

Let's fit this model to the `Insurance` dataset.

```
theta_initial1 = c(4, 0.15, 0.35)
fit_1 = optim(theta_initial1, NLL_Model1,
              y=Insurance$Claims,
              X=Insurance$Holders,
              control=list(maxit=1500))
theta_hat = fit_1$par

beta0_hat = theta_hat[1]
beta1_hat = theta_hat[2]
sigma_hat = exp(theta_hat[3])

paste0("Estimated beta0: ", beta0_hat)
```

```
## [1] "Estimated beta0: 8.12308508107348"
```

```
paste0("Estimated beta1: ", beta1_hat)
```

```
## [1] "Estimated beta1: 0.112659436791941"
```

```
paste0("Estimated sigma: ", sigma_hat)
```

```
## [1] "Estimated sigma: 11.8684232473608"
```

Model 2: Linear regression with Laplace distributed errors

```
NLL_Model2 <- function(theta, y, X)
{
  beta_0 = theta[1]
  beta_1 = theta[2]
  sigma = exp(theta[3])  # variance of error terms
```

```
  n = length(data)
  # Log likelihood function of Laplace distribution
  l = -(n*log(2*sigma) + 1/sigma*sum(abs(y - beta_0 - beta_1*X)))
  return(-l)
}
```

Let's fit this model to the `Insurance` dataset.

```
theta_initial2 = c(4, 0.15, 0.35)
fit_2 = optim(theta_initial2, NLL_Model2,
              y=Insurance$Claims,
              X=Insurance$Holders,
              control=list(maxit=1500))
theta_hat = fit_2$par

beta0_hat = theta_hat[1]
beta1_hat = theta_hat[2]
# variance of Laplace model is 2 * sigma^2
sigma_hat = exp(theta_hat[3])/sqrt(2)

paste0("Estimated beta0: ", beta0_hat)
```

```
## [1] "Estimated beta0: 5.08201018006"
```

```
paste0("Estimated beta1: ", beta1_hat)
```

```
## [1] "Estimated beta1: 0.116625925908506"
```

```
paste0("Estimated sigma: ", sigma_hat)
```

```
## [1] "Estimated sigma: 371.708118099566"
```

Model 3: Linear regression for log-normally distributed data

```
NLL_Model3 <- function(theta, y, X)
{
  beta_0 = theta[1]
  beta_1 = theta[2]
  sigma = exp(theta[3])  # variance of error terms

  n = length(data)
  # -ve log likelihood function of log-normal distribution
  # Mean is given in question

  l = 0
  for (i in 1:n)
  {
    if (y[i] != 0)
    {
```

```r
      l = l + dlnorm(y[i], meanlog = beta_0 + beta_1*log(X[i]),
                     sdlog = sigma, log=T)
    }
  }
  return(-l)
}
```

Let's fit this model to the `Insurance` dataset.

```r
theta_initial3 = c(4.5, 0.15, 0.5)
fit_3 = optim(theta_initial3, NLL_Model3,
              y=Insurance$Claims,
              X=Insurance$Holders,
              control=list(maxit=1500))
theta_hat = fit_3$par

beta0_hat = theta_hat[1]
beta1_hat = theta_hat[2]
sigma_hat = exp(theta_hat[3])

paste0("Estimated beta0: ", beta0_hat)
```

```
## [1] "Estimated beta0: 9.28984611133503"
```

```r
paste0("Estimated beta1: ", beta1_hat)
```

```
## [1] "Estimated beta1: -1.06985462643872"
```

```r
paste0("Estimated sigma: ", sigma_hat)
```

```
## [1] "Estimated sigma: 1.50358387830253e-15"
```

Model 4: Gamma regression

Unattempted.

BIC analysis

The Bayesian information criterion (BIC) is a criterion for model selection among a finite set of models. It is based, in part, on the likelihood function and it is closely related to the Akaike information criterion (AIC).

Suppose that we have a statistical model of some data. Let k be the number of estimated parameters in the model. Let L be the maximized value of the likelihood function for the model and n be the total number of data points. Then the BIC value of the model is the following:

$$\text{BIC} = k \ln(n) - 2 \ln L$$

Given a set of candidate models for the data, models with lower BIC are generally preferred.

Let's define the BIC function:

```r
get_BIC <- function(optim_fit, data=Insurance) {
  log(nrow(data)) * length(optim_fit$par) + 2 * optim_fit$value
}
```

Let's calculate BIC:

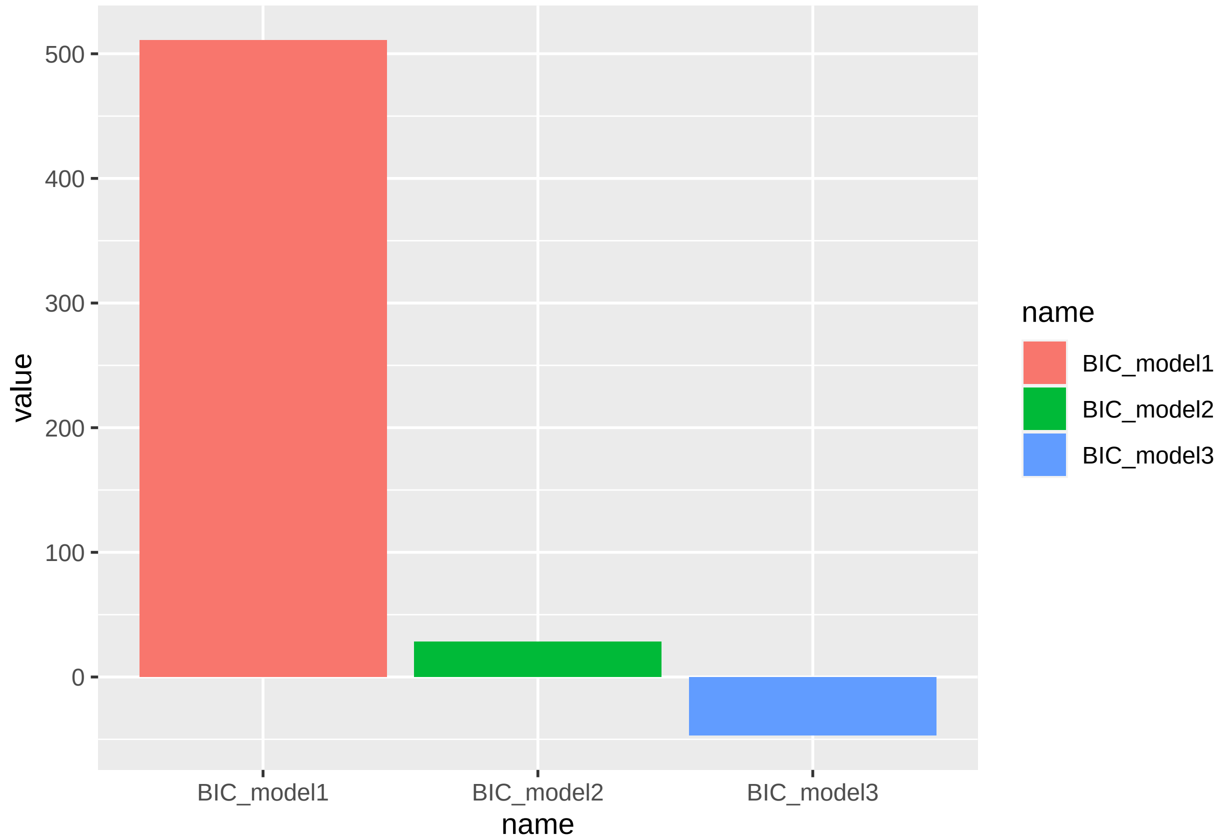
```r
library(dplyr)
BIC_scores <- tibble(
  BIC_model1 = get_BIC(fit_1),
  BIC_model2 = get_BIC(fit_2),
  BIC_model3 = get_BIC(fit_3),
)

BIC_scores
```

```
## # A tibble: 1 x 3
##   BIC_model1 BIC_model2 BIC_model3
##        <dbl>      <dbl>      <dbl>
## 1       511.       28.4      -46.7
```

A plot:

```r
library(tidyr)
library(ggplot2)
BIC_scores = pivot_longer(BIC_scores, cols = BIC_model1:BIC_model3)
ggplot(BIC_scores, aes(x=name, y=value, fill=name))+
  geom_bar(stat='identity')
```



From the BIC scores, we can see that Model 3 is better than the other models.