Modelling insurance claims

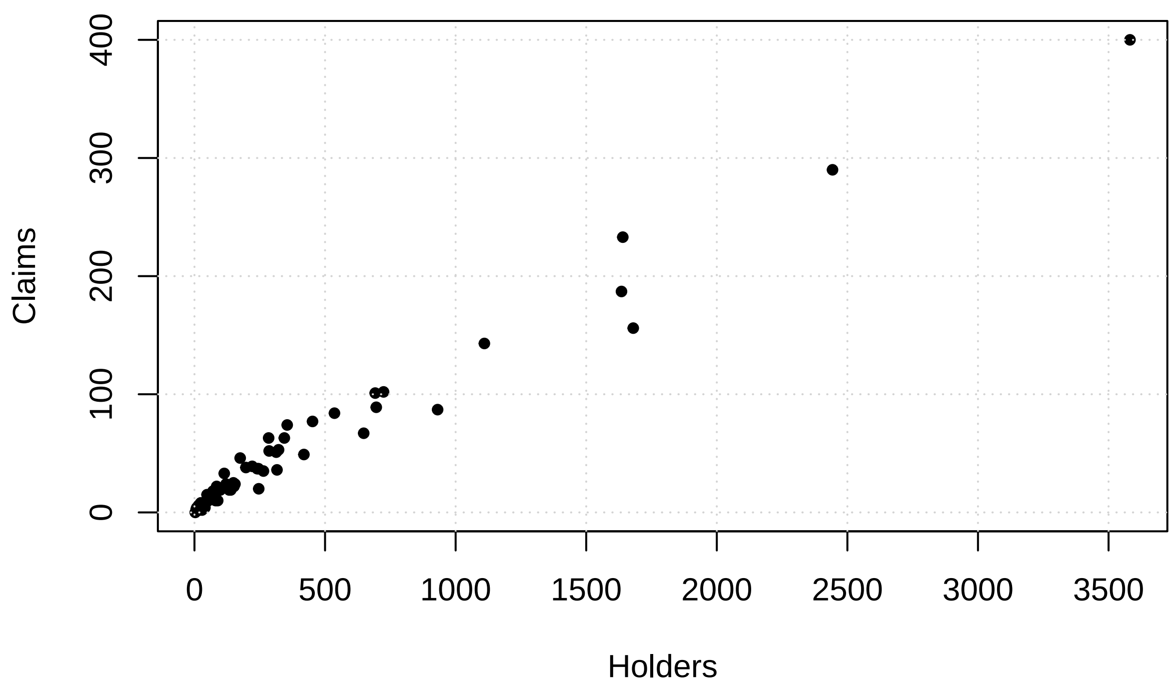
Problem 4

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

Here, we model

$$\texttt{Claims}_i = \beta_0 + \beta_1\ \texttt{Holders}_i + \varepsilon_i, \quad i = 1, 2, \cdots, n$$

assuming $\varepsilon_i \sim \mathrm{N}(0, \sigma^2)$.

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

  # -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

Here, we model

$$\texttt{Claims}_i = \beta_0 + \beta_1\ \texttt{Holders}_i + \varepsilon_i, \quad i = 1, 2, \cdots, n$$

assuming $\varepsilon_i \sim \mathrm{Laplace}(0, \sigma^2)$.

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

  n = length(y)
  # 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.

```r
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: 5.08449916545323"
```

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

```
## [1] "Estimated beta1: 0.116625247079814"
```

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

```
## [1] "Estimated sigma: 5.80442873531934"
```

Model 3: Linear regression for log-normally distributed data

Here, we model

$$\texttt{Claims}_i \sim \text{LogNormal}(\mu_i, \sigma^2)$$

where $\mu_i = \beta_0 + \beta_1 \log(\texttt{Holders}_i), \quad i = 1, 2, \cdots, n$.

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

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

  l = 0
  for (i in 1:n)
  {
    if (y[i] != 0)
    {
      l = l + dlnorm(y[i], meanlog = exp(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: -0.101897492703729"
```

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

```
## [1] "Estimated beta1: 0.242926365225336"
```

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

```
## [1] "Estimated sigma: 0.393898387518733"
```

Model 4: Gamma regression

We desire a model

$$\texttt{Claims}_i \sim \text{Gamma}(\alpha_i, \sigma)$$

where $\log(\alpha_i) = \beta_0 + \beta_1 \log(\texttt{Holders}_i), \quad i = 1, 2, \cdots, n.$

```
NLL_Model4 <- function(theta, y, X)
{
  beta_0 = theta[1]
  beta_1 = theta[2]
  sigma = exp(theta[3])  # variance of error terms

  n = length(y)
  l = 0
  for (i in 1:n)
  {
    if (y[i] != 0)
    {
      l = l + dgamma(y[i], shape = exp(beta_0 + beta_1*log(X[i])),
                     scale = sigma, log = T)
    }
  }
  return(-l)
}
```

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

```
theta_initial4 = c(1, 0.15, 0.5)
fit_4 = optim(theta_initial4, NLL_Model4,
              y=Insurance$Claims,
              X=Insurance$Holders,
              control=list(maxit=1500))
theta_hat = fit_4$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: -1.64121395888405"
```

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

```
## [1] "Estimated beta1: 0.837044680232921"
```

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

```
## [1] "Estimated sigma: 2.05526290680432"
```

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:

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

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

Let's define the BIC function:

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

Let's calculate BIC:

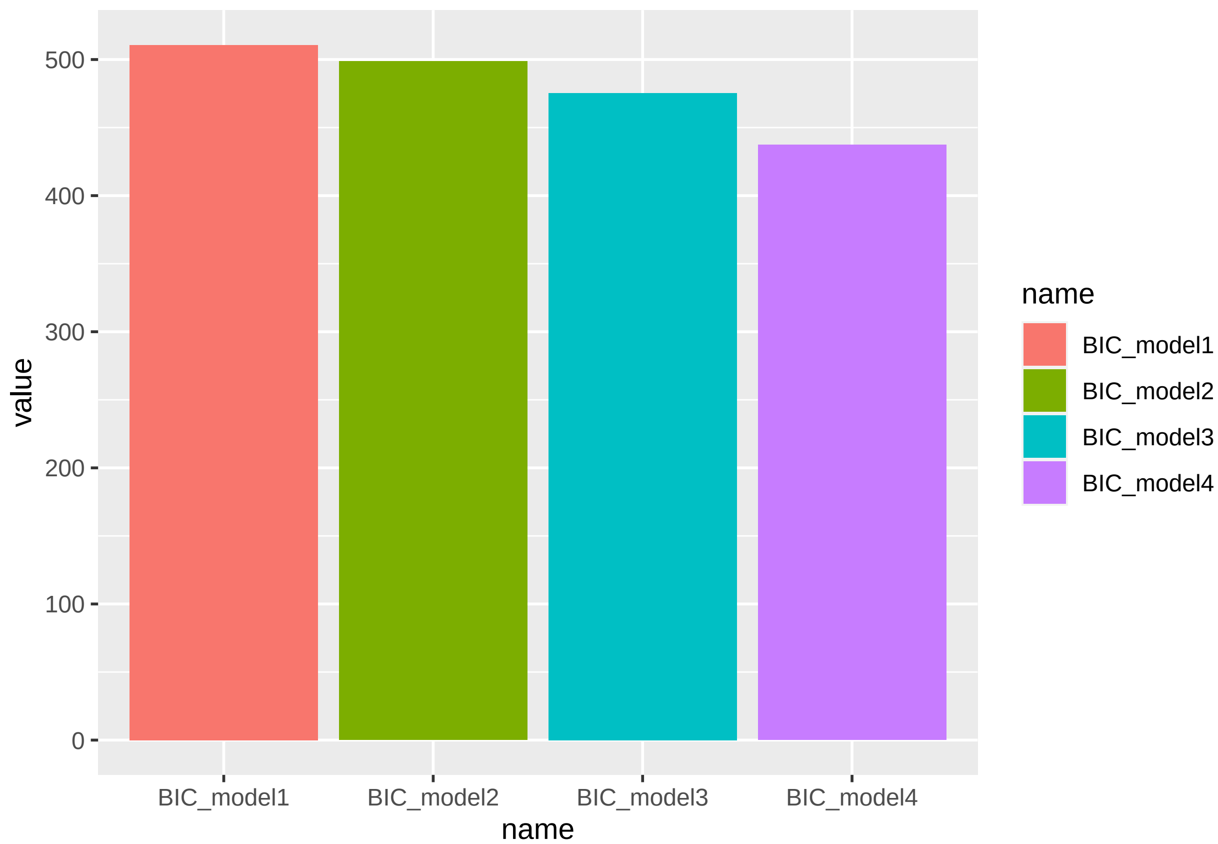
```
library(dplyr)
BIC_scores <- tibble(
  BIC_model1 = get_BIC(fit_1),
  BIC_model2 = get_BIC(fit_2),
  BIC_model3 = get_BIC(fit_3),
  BIC_model4 = get_BIC(fit_4),
)

BIC_scores
```

```
## # A tibble: 1 x 4
##   BIC_model1 BIC_model2 BIC_model3 BIC_model4
##        <dbl>      <dbl>      <dbl>      <dbl>
## 1       511.       499.       475.       437.
```

A plot:

```
library(tidyr)
library(ggplot2)
BIC_scores = pivot_longer(BIC_scores, cols = BIC_model1:BIC_model4)
ggplot(BIC_scores, aes(x=name, y=value, fill=name))+
  geom_bar(stat='identity')
```



From the BIC scores, we can see that Model 4 is better than the other models.