Advanced Statistical Modelling Coursework 1

087074

12/03/2021

Question 2

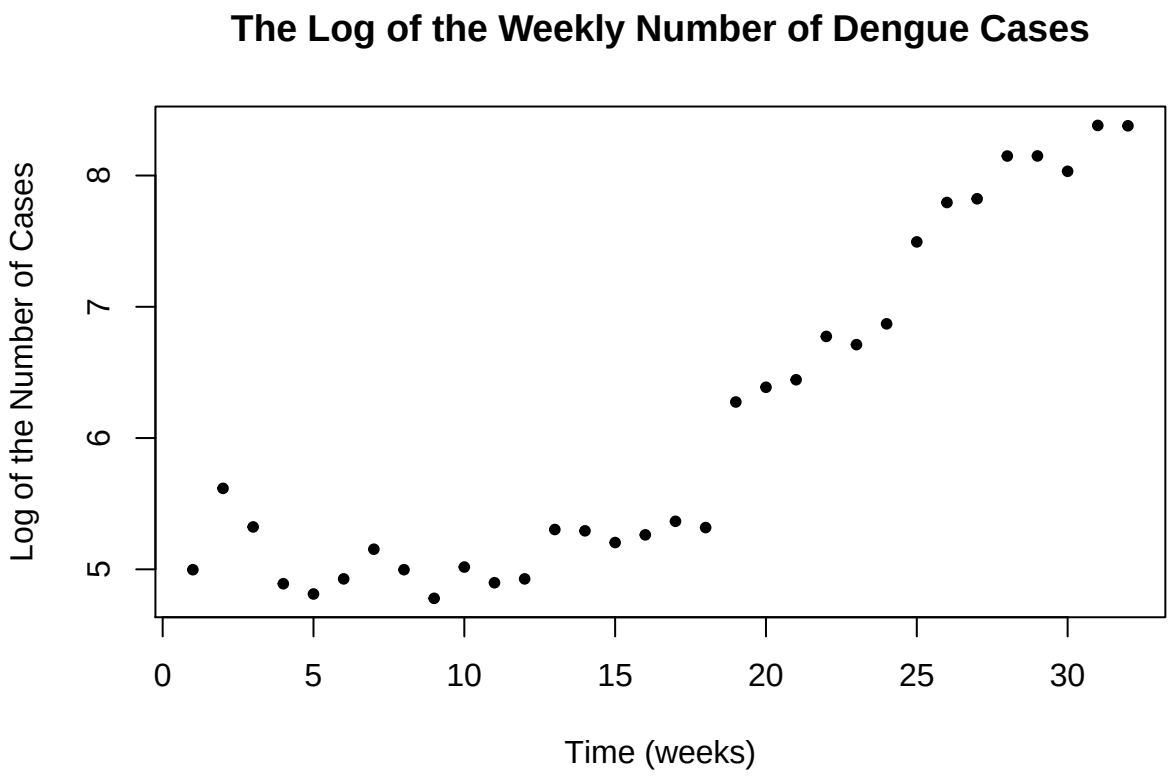
The dengue dataset contains time series data on the weekly dengue fever cases in Rio de Janeiro. The weekly dengue cases, y can be modeled using a General Linear Model using the week, x as the predictive variable. My GLM will incorporate the Negative Binomial distribution with a log link function.

$$Y_i \sim NegBin(\mu_i, \theta)$$
$$log(\mu_i) = \beta_0 + \beta_1 x_i$$

2a

The log link function assumes that there is a linear relationship between the log of the mean number of cases and the week. I have plotted the data with a log scale in the y axis to see if this relationship is visible.

```
plot(dengue$Time,log(dengue$Cases),xlab="Time (weeks)", ylab="Log of the Number of Cases",
     main="The Log of the Weekly Number of Dengue Cases", pch=20)
```



The graph shows that the link function misses some structure in the data. The weekly number of cases remains fairly constant until around week 15, after which the number of cases rise in a linear fashion.

2b

The probability mass function (pmf) of the Negative Binomial distribution is

$$p(y; \mu_i, \theta) = \binom{y + \theta - 1}{y} \left(\frac{\theta}{\theta + \mu}\right)^{\theta} \left(\frac{\mu}{\theta + \mu}\right)^{y}$$

The samples from the Negative Binomial distribution are assumed to be independent. Therefore we can calculate the likelihood of the distribution by taking the product of the pmf.

$$L(\mu_i, \theta; y) = \prod_{i=1}^{n} \binom{y_i + \theta - 1}{y_i} \left(\frac{\theta}{\theta + \mu_i}\right)^{\theta} \left(\frac{\mu_i}{\theta + \mu_i}\right)^{y_i}$$

Take the log of the likelihood to obtain the log likelihood.

$$\mathcal{L}(\mu_i, \theta; y) = n\theta \log(\theta) + \sum_{i=1}^{n} \left[log\binom{y_i + \theta - 1}{y_i} - (\theta + y_i) log(\theta + e^{\beta_0 + \beta_1 x_i}) + y_i(\beta_0 + \beta_1 x_i) \right]$$

2c

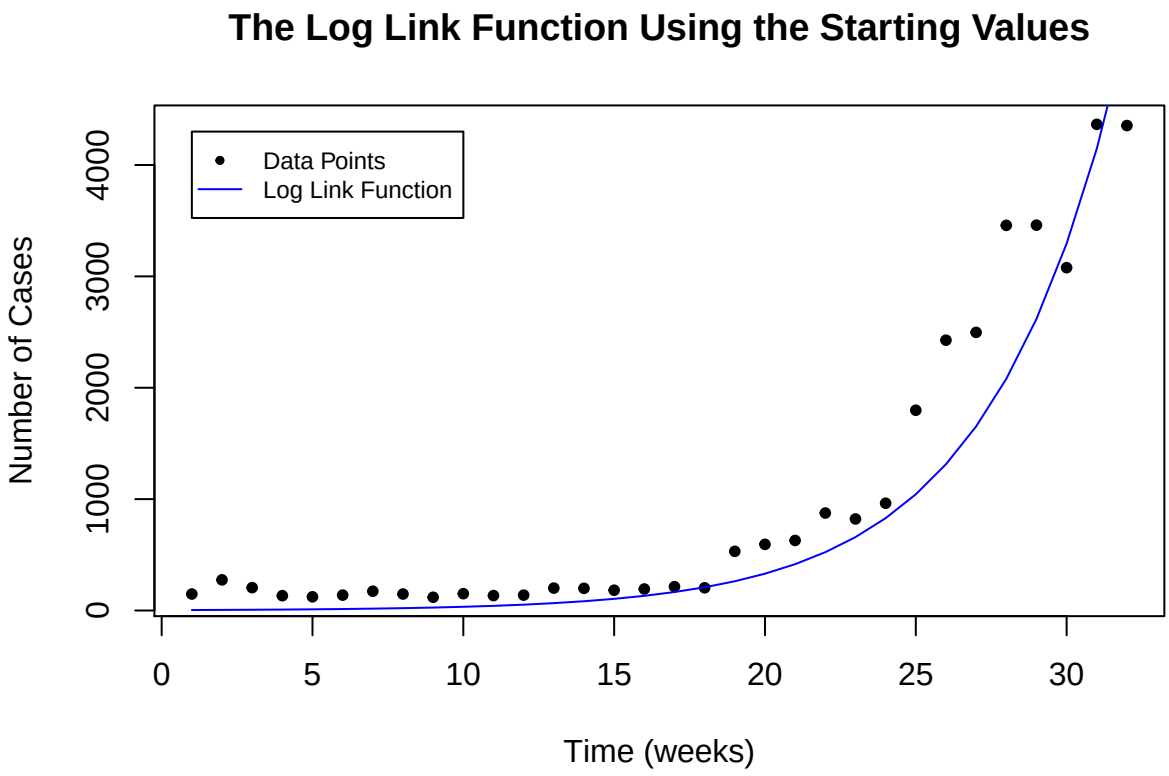
I have created a function, mylike(), that evaluates the negative of the log likelihood based on different values of β.

```
mylike <- function(beta){
  result <- prod(beta[3]*log(beta[3])) + sum(lchoose(dengue$Cases+beta[3]-1, dengue$Cases)-
    (beta[3]+dengue$Cases)*log(beta[3]+exp(beta[1]+beta[2]*dengue$Time)) +
    dengue$Cases*(beta[1]+beta[2]*dengue$Time))
  return(-result)
}
```

2d

By tweaking the values of β and judging the fit by eye, I have obtained a starting value of $\beta = 1.2, 0.23, 1$. I plotted the log link function using these starting values to show that they provide a sensible starting point.

```
# plot the data
plot(dengue$Time,dengue$Cases,xlab="Time (weeks)", ylab="Number of Cases",
     main="The Log Link Function Using the Starting Values", pch=20)

beta_init <- c(1.2, 0.23, 1)  # starting values of beta

yfit <- exp(beta_init[1]+beta_init[2]*dengue$Time) # log link function
lines(sort(dengue$Time), yfit[order(dengue$Time)],lwd=1,lty=1,col="blue")

# add a legend
legend(x=1,y=4300, legend=c("Data Points", "Log Link Function"), pch=c(20,-1),
       lty=c(-1,1), col=c('black','blue'), lwd=c(-1,1), cex = 0.75)
```



The R function nlm() finds values of β that numerically optimize the mylike() function.

```
model.NegBin <- nlm(mylike, p = beta_init, hessian = T, gradtol = 1e-10, iterlim = 1000)
```

2e

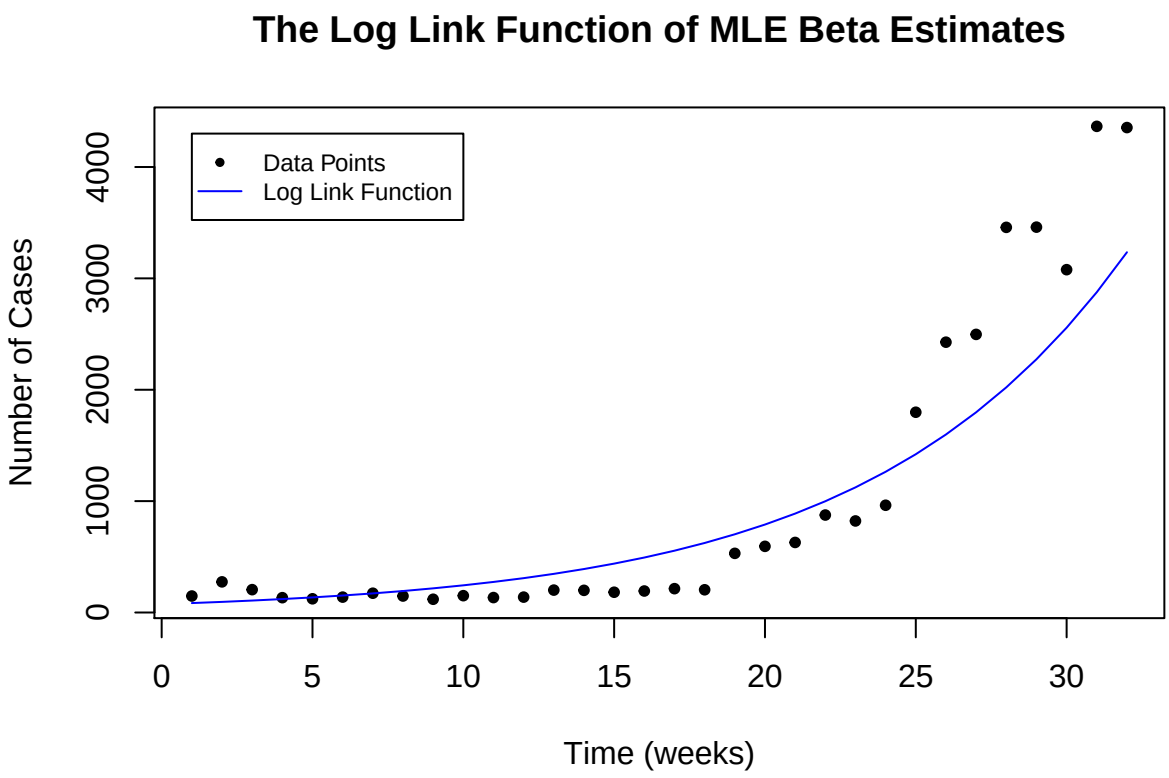
The maximum likelihood estimates for β are

```
betas <- model.NegBin$estimate
print(round(betas,3))
```

```
## [1] 4.321 0.118 0.736
```

I have plotted the log link function using the maximum likelihood estimates.

```
# plot the data
plot(dengue$Time,dengue$Cases,xlab="Time (weeks)", ylab="Number of Cases",
     main="The Log Link Function of MLE Beta Estimates", pch=20)
yfit <- exp(betas[1]+betas[2]*dengue$Time) # log link function
lines(sort(dengue$Time), yfit[order(dengue$Time)],lwd=1,lty=1,col="blue")
# add a legend
legend(x=1,y=4300, legend=c("Data Points", "Log Link Function"), pch=c(20,-1),
       lty=c(-1,1), col=c('black','blue'), lwd=c(-1,1), cex = 0.75)
```



2f

The standard errors for β_0 and β_1 are

```
OIM <- solve(model.NegBin$hessian) # solve the Hessian matrix to obtain covariance matrix
VarianceBeta <- diag(OIM)
stand_error <- sqrt(VarianceBeta) # square root to obtain std. error
stand_error[c(1,2)]
```

```
## [1] 0.37400171 0.01888456
```

Likelihood theory tells us that β_0 and β_1 are approximately Gaussian with standard errors as calculated above. Therefore, testing the null hypothesis that $\beta_1 = 0$ at the 5% significance level is done via a z-test. The Z score is very high.

```
z_test <- betas/stand_error # z-test equation
z_test[2]
```

```
## [1] 6.222256
```

The p-value for the test is below 0.05.

```r
2*(1-pnorm(z_test[2],0,1)) ## beta_1 p-value
```

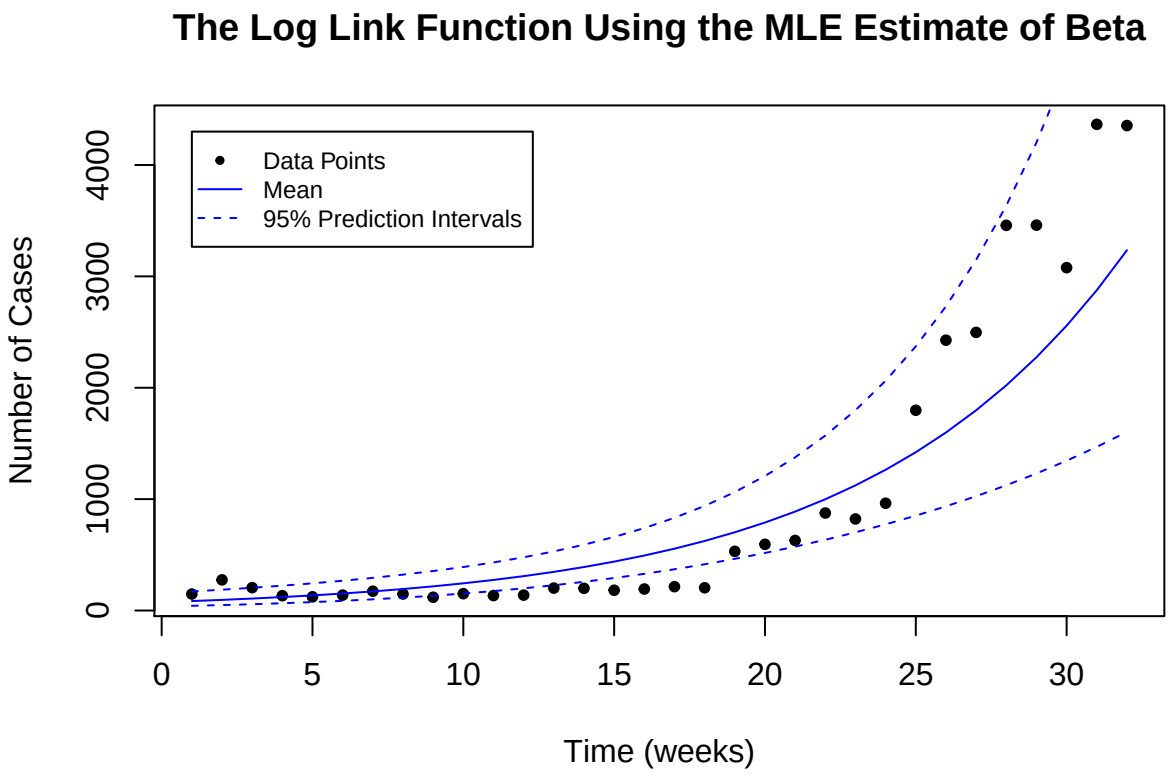
```
## [1] 4.90056e-10
```

Therefore, we can reject the null hypothesis that $\beta_1 = 0$ at a 5% significance level and accept the alternative hypothesis that $\beta_1 \neq 0$.

2h

By using plug in prediction, the 95% prediction intervals of the model can be calculated and plotted.

```r
# plot the data
plot(dengue$Time,dengue$Cases,xlab="Time (weeks)", ylab="Number of Cases",
     main="The Log Link Function Using the MLE Estimate of Beta", pch=20)

xx <- dengue$Time # x-coordinates
linf <- betas[1] + betas[2]*xx # obtain mean from log link function
mu <- exp(linf)

# use standard errors to calculate prediction intervals
stdErr <- sqrt(VarianceBeta[1] + xx^2*VarianceBeta[2] + 2*xx*OIM[1,2])
lower <- exp(linf - 1.96*stdErr)
upper <- exp(linf + 1.96*stdErr)

lines(xx,mu,lwd=1,col="blue") # plot mean and prediction interval
lines(xx,upper,lwd=1,col="blue",lty=2)
lines(xx,lower,lwd=1,col="blue",lty=2)

legend(x=1,y=4300, legend=c("Data Points", "Mean", "95% Prediction Intervals"),
       pch=c(20,-1,-1), lty=c(-1,1,2), col=c('black','blue', 'blue'), lwd=c(-1,1, 1),
       cex = 0.75)
```



2i

At the lower and upper tail, the model appears to predict the data well, with the data points falling within the 95% prediction interval. However, the model does not predict the data well for weeks 10 to 20 as there are a number of data points that lie outside the 95% prediction interval. Therefore, higher order terms should be added to the model to incorporate more structure from the data.

Question 5

5a

The Inverse Gaussian distribution has probability mass function.

$$p(y;\mu,\lambda) = \sqrt{\frac{\lambda}{2\pi y^3}} exp\left(-\frac{\lambda(y-\mu)^2}{2\mu^2 y}\right)$$

This can be rearranged to show that the Inverse Gaussian distribution is a member of the exponential family.

$$\begin{aligned}
p(y;\mu,\lambda) &= \sqrt{\frac{\lambda}{2\pi y^3}} exp\left(-\frac{\lambda(y-\mu)^2}{2\mu^2 y}\right) \\
&= exp\left(log\left(\sqrt{\frac{\lambda}{2\pi y^3}} exp\left(-\frac{\lambda(y-\mu)^2}{2\mu^2 y}\right)\right)\right) \\
&= exp\left(log\left(\sqrt{\frac{\lambda}{2\pi y^3}}\right) - \frac{\lambda(y-\mu)^2}{2\mu^2 y}\right) \\
&= exp\left(\frac{1}{2} log\left(\frac{\lambda}{2\pi y^3}\right) - \frac{\lambda(y^2 - 2\mu y + \mu^2)}{2\mu^2 y}\right) \\
&= exp\left(\frac{1}{2} log\left(\frac{\lambda}{2\pi y^3}\right) - \frac{\lambda}{2\mu^2} y + \frac{\lambda}{\mu} - \frac{\lambda}{2y}\right) \\
&= exp\left(-\frac{\lambda}{2\mu^2} y + \frac{\lambda}{\mu} + \frac{1}{2} log\left(\frac{\lambda}{2\pi y^3}\right) - \frac{\lambda}{2y}\right) \\
&= exp\left(\frac{-\frac{1}{2\mu^2} y + \frac{1}{\mu}}{\frac{1}{\lambda}} + \frac{1}{2} log\left(\frac{\lambda}{2\pi y^3}\right) - \frac{\lambda}{2y}\right)
\end{aligned}$$

By comparing this to the distribution of the exponential family,

$$p(y;\mu,\lambda) = exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y;\phi)\right)$$

It is evident that:

$$\theta = -\frac{1}{2\mu^2} \quad \text{and} \quad \phi = \lambda$$

I now display $a(\phi)$, $b(\theta)$ and $c(y,\phi)$ in terms of λ and μ and also θ and ϕ.

$$\begin{aligned}
a(\phi) &= \frac{1}{\lambda} \\
&= \frac{1}{\phi} \\
b(\theta) &= -\frac{1}{\mu} \\
&= -\sqrt{-2\theta} \\
c(y,\phi) &= \frac{1}{2} log\left(\frac{\lambda}{2\pi y}\right) - \frac{\lambda}{2y} \\
&= \frac{1}{2} log\left(\frac{\phi}{2\pi y}\right) - \frac{\phi}{2y}
\end{aligned}$$

5b

The mean and variance of distributions in the exponential family can be calculated using $a(\phi)$ and the derivatives of $b(\theta)$.

$$\text{mean} = b'(\theta) \quad \text{and} \quad \text{variance} = a(\phi)b''(\theta)$$

For the negative binomial distribution, we have shown that

$$b(\theta) = -\sqrt{-2\theta}$$

Therefore,

$$\begin{aligned} b'(\theta) &= -\frac{1}{2}(-2)\sqrt{(-2\theta)} \\ &= \frac{1}{\sqrt{(-2\theta)}} \end{aligned}$$

Therefore, the mean is equal to,

$$\begin{aligned} \text{mean} &= \frac{1}{\sqrt{(-2\theta)}} \\ &= \mu \end{aligned}$$

The second derivative of $b(\theta)$ can be calculated,

$$\begin{aligned} b''(\theta) &= (-\frac{1}{2})(-2)(-2)(-\frac{1}{2})(-2\theta)^{-\frac{3}{2}} \\ &= (-2\theta)^{-\frac{3}{2}} \\ &= \left(\frac{1}{\sqrt{(-2\theta)}}\right)^3 \\ &= \mu^3 \end{aligned}$$

Therefore, the variance is equal to,

$$\begin{aligned} \text{variance} &= a(\phi)b''(\theta) \\ &= \frac{1}{\phi}\mu^3 \\ &= \frac{\mu^3}{\lambda} \end{aligned}$$

Question 7

The dataset aids contains the number of quarterly aids cases in the UK from January 1983 to March 1994. We consider two competing models to describe the trend in the cases of aids over time.

Model 1

$$Y_i \sim Pois(\lambda_i)$$
$$log(\lambda_i) = \beta_0 + \beta_1 x_i$$

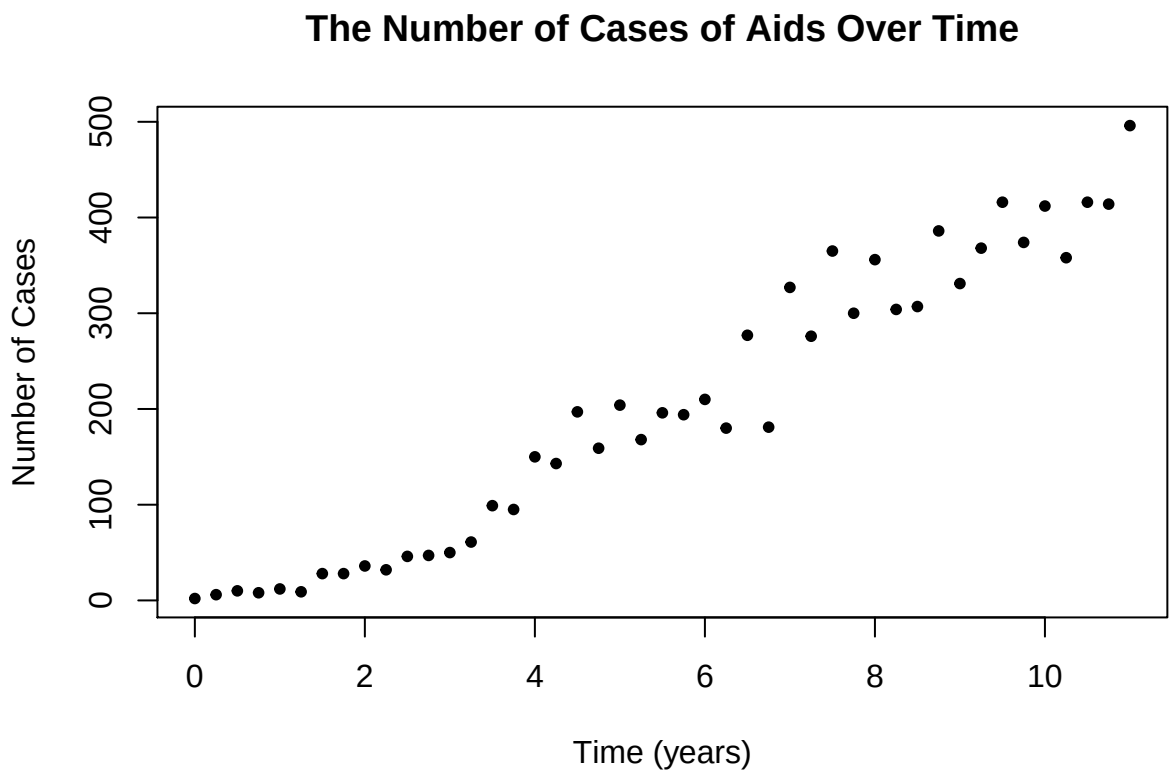
Model 2

$$Y_i \sim N(\mu_i, \sigma^2)$$
$$log(\mu_i) = \gamma_0 + \gamma_1 x_i$$

7a

I have created a new variable time that is the same as date in the original dataset, however it starts at 0 rather than 83. The graph shows how the number of cases evolves over time.

```
aids$time = aids$date - 83 # create a new dataset from t=0
# plot the data
plot(aids$time, aids$cases, xlab="Time (years)", ylab="Number of Cases",
     main='The Number of Cases of Aids Over Time',pch=20)
```



The plot shows that the variance of the number of cases increases as the mean number of cases increases. The Poisson distribution respects this property, however the Normal distribution does not. Therefore, preliminary data analysis suggests that the Poisson model would be better suited to the dataset than a Normal model.

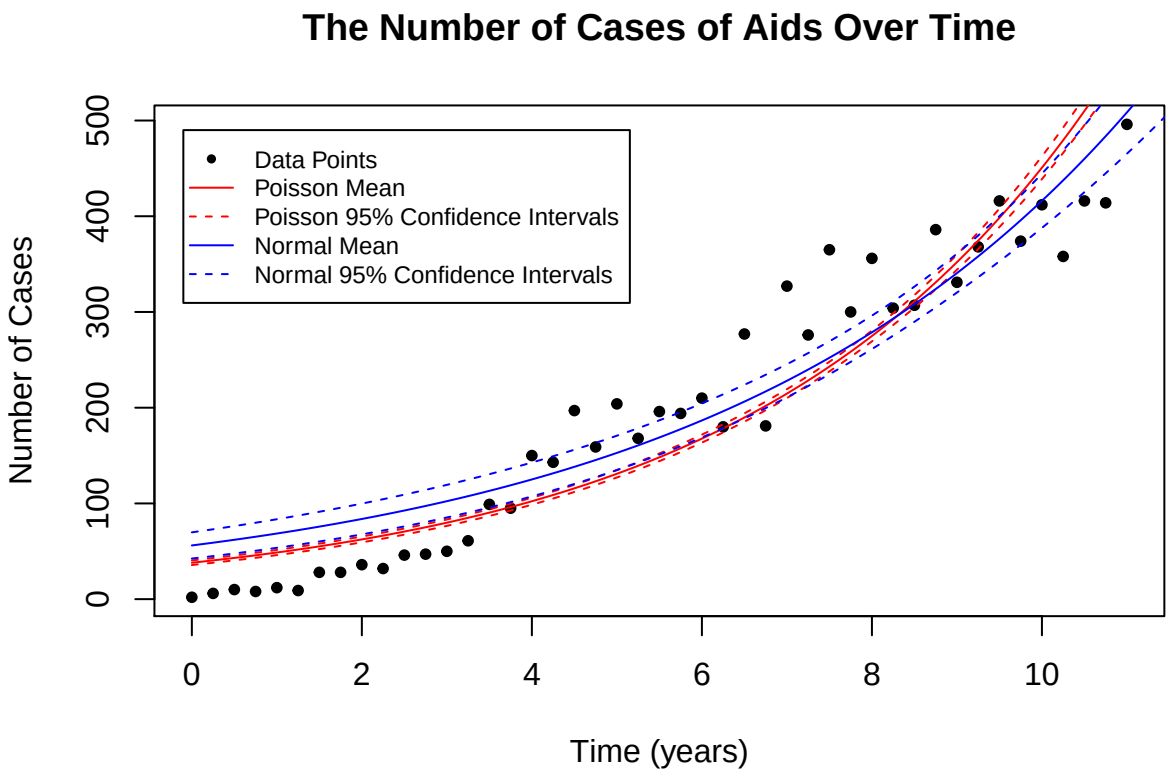
7b

Fit the models to the dataset.

```
model_pois <- glm(cases~time, data=aids, family=poisson(link="log")) # model 1
model_norm <- glm(cases~time, data=aids, family=gaussian(link="log")) # model 2
```

We can use the models to predict estimates of the mean number of cases over the range of time. Likelihood theory tells us that the mean has a Normal Distribution with a standard error, therefore we can construct a 95% confidence interval of the mean.

```
plot(aids$time, aids$cases, xlab="Time (years)", ylab="Number of Cases",
     main='The Number of Cases of Aids Over Time',pch=20)

dates_list <- seq(81,94, length=200) # date range
times_list <- seq(0,12, length=200) # date range
# obtain the mean of the Poisson model over the range of dates
preds_pois <- predict(model_pois, newdata=data.frame(time=times_list),type="response",
                      se.fit=T)
lines(times_list, preds_pois$fit,lwd=1,col="red") # plot in red
# Calculate and plot the 95% confidence interval of the mean (approx. Gaussian)
upper_pois <- preds_pois$fit+1.96*preds_pois$se.fit
lines(times_list, upper_pois,lty=2,lwd=1,col="red")
lower_pois <- preds_pois$fit-1.96*preds_pois$se.fit
lines(times_list,lower_pois,lty=2,lwd=1,col="red")

# obtain the mean of the normal model over the range of dates
preds_norm <- predict(model_norm, newdata=data.frame(time=times_list),type="response",
                      se.fit=T)
lines(times_list, preds_norm$fit,lwd=1,col="blue") # plot in blue
# Calculate and plot the 95% confidence interval of the mean (approx. Gaussian)
upper_norm <- preds_norm$fit+1.96*preds_norm$se.fit
lines(times_list, upper_norm,lty=2,lwd=1,col="blue")
lower_norm <- preds_norm$fit-1.96*preds_norm$se.fit
lines(times_list, lower_norm,lty=2,lwd=1,col="blue")

legend(x=-0.1,y=490, legend=c("Data Points", "Poisson Mean",
                              "Poisson 95% Confidence Intervals", "Normal Mean",
                              "Normal 95% Confidence Intervals"),
       pch=c(20,-1,-1,-1,-1), lty=c(-1,1,2,1,2), col=c('black','Red', 'Red','Blue','Blue'),
       lwd=c(-1,1,1,1,1), cex = 0.75)
```



The Akaike Information Criterion (AIC) of a model is an estimate of model quality, with a lower AIC meaning a better model.

```
model_pois$aic  # AIC of model 1
```

```
## [1] 1153.873
```

```
model_norm$aic # AIC of model 2
```

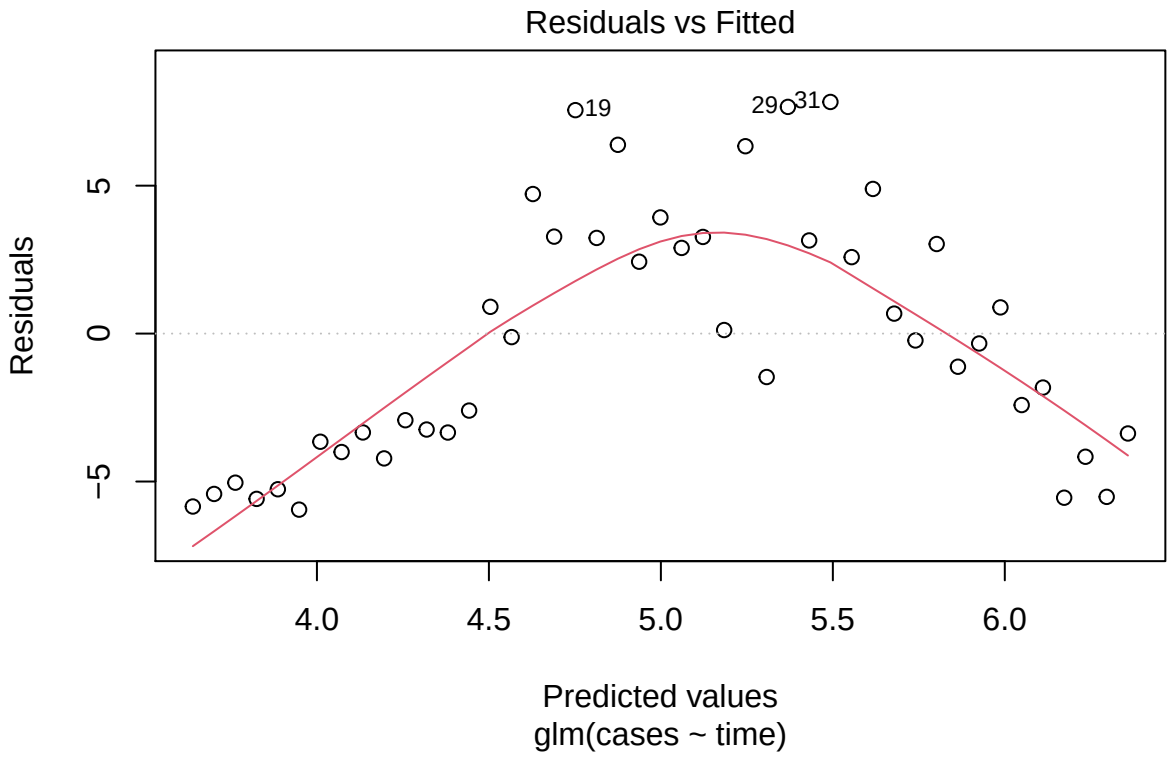
```
## [1] 482.8128
```

The model that incorporates the Normal distribution has a lower AIC value and is therefore the better model for our dataset.

7c

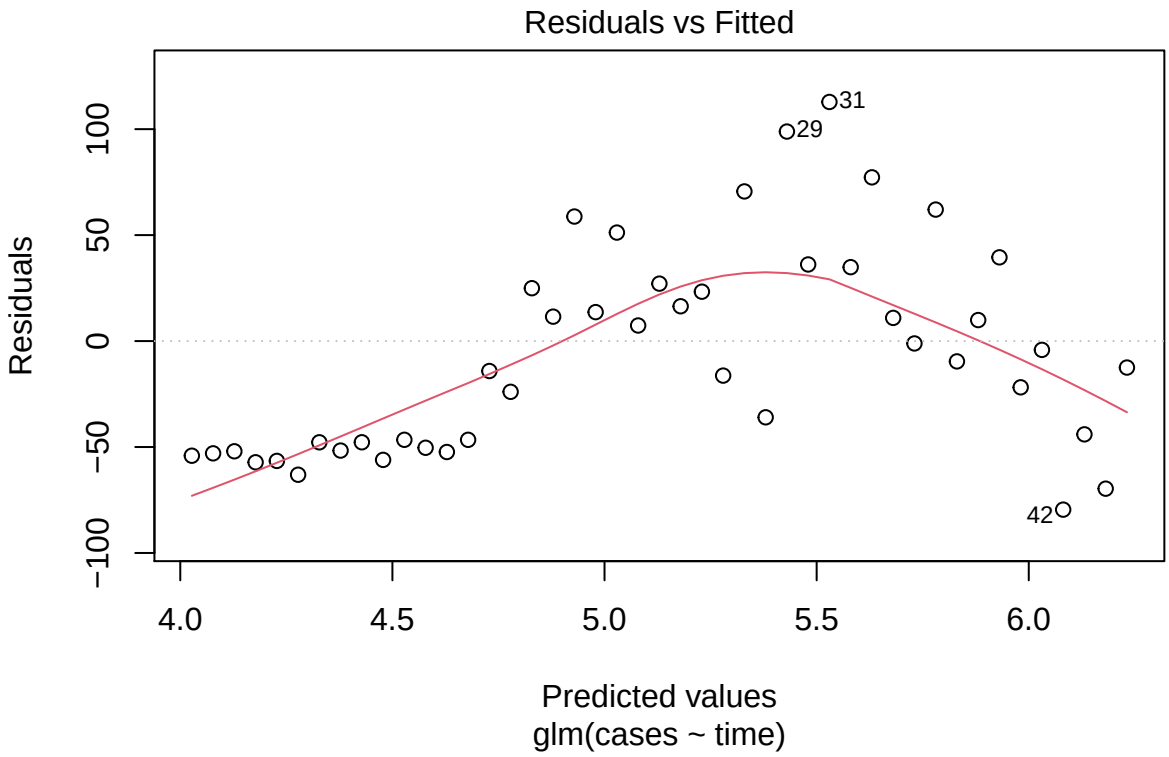
Residual plot of the Poisson model.

```
# plotting the residuals of the Poisson model.
plot(model_pois,1)
```



Residual plot of the Normal model.

```
# plotting the residuals of the Normal model.
plot(model_norm,1)
```



Both of the plots indicate that there is structure in the dataset that the models are missing. Therefore, higher order terms should be added to the models.

7d

I add higher order terms to the Poisson model (model 1) to incorporate more data structure into the model.

```
model_pois2 <- glm(cases~date+I(time^2)+I(time^3)+I(time^4), data=aids,
                   family=poisson(link="log"))
```

Perform a Chai-squared test to see if removing a term will improve the AIC score of the model.

```
drop1(model_pois2,test="Chi")
```

```
## Single term deletions
##
## Model:
## cases ~ date + I(time^2) + I(time^3) + I(time^4)
##           Df Deviance    AIC    LRT  Pr(>Chi)
## <none>         166.16 472.01
## date       1   252.22 556.07 86.054 < 2.2e-16 ***
## I(time^2)  1   179.86 483.71 13.698 0.0002147 ***
## I(time^3)  1   169.07 472.92  2.902 0.0884725 .
## I(time^4)  1   166.68 470.53  0.515 0.4731649
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that the time^4 term should be dropped. Let us fit the model without this term.

```
model_pois2 <- glm(cases~time+I(time^2)+I(time^3), data=aids, family=poisson(link="log"))
```

Let us see if any other terms should be dropped.

```
drop1(model_pois2,test="Chi")
```

```
## Single term deletions
##
## Model:
## cases ~ time + I(time^2) + I(time^3)
##           Df Deviance    AIC    LRT  Pr(>Chi)
## <none>         166.68 470.53
## time       1   595.02 896.87 428.34 < 2.2e-16 ***
## I(time^2)  1   319.77 621.62 153.09 < 2.2e-16 ***
## I(time^3)  1   250.14 551.99  83.46 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Removing any other terms would increase the AIC of the model. Therefore, we conclude that we have found the optimal Poisson model.

```
summary(model_pois2)$coefficients
```

```
##                 Estimate   Std. Error    z value     Pr(>|z|)
## (Intercept)  1.286749374 0.1457263793   8.829900 1.047692e-18
## time         1.347376814 0.0754895625  17.848518 2.968088e-71
## I(time^2)   -0.141428911 0.0121556103 -11.634867 2.740165e-31
## I(time^3)    0.005362342 0.0006062301   8.845391 9.120818e-19
```

I repeat the process of finding higher order terms that optimize the AIC score for the Normal model (model 2).

```
model_norm2 <- glm(cases~time+I(time^2)+I(time^3)+I(time^4), data=aids,
                   family=gaussian(link="log"))
```

Perform a Chai-squared test to see if removing a term will improve the AIC score of the model.

```
drop1(model_norm2,test="Chi")
```

```
## Single term deletions
##
## Model:
## cases ~ time + I(time^2) + I(time^3) + I(time^4)
##           Df Deviance    AIC scaled dev. Pr(>Chi)
## <none>         37947 442.88
```

```
## time       1    41956 445.40      4.5189  0.03352 *
## I(time^2)  1    38523 441.56      0.6776  0.41040
## I(time^3)  1    38041 440.99      0.1111  0.73895
## I(time^4)  1    37951 440.89      0.0039  0.95011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that the time^4 term should be dropped. Let us fit the model without this term.

```
model_norm2 <- glm(cases~time+I(time^2)+I(time^3), data=aids, family=gaussian(link="log"))
summary(model_norm2)$coefficients # coefficients of the model summary
```

```
##                 Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)  1.501911228 0.644571461  2.330093 2.480310e-02
## time         1.247136182 0.281972725  4.422897 7.016438e-05
## I(time^2)   -0.127134687 0.039459051 -3.221940 2.496116e-03
## I(time^3)    0.004723959 0.001765666  2.675455 1.067466e-02
```

Let us see if any other terms should be dropped.

```
drop1(model_norm2,test="Chi")
```

```
## Single term deletions
##
## Model:
## cases ~ time + I(time^2) + I(time^3)
##           Df Deviance    AIC scaled dev.  Pr(>Chi)
## <none>          37951 440.89
## time       1    74792 469.42     30.5291 3.289e-08 ***
## I(time^2)  1    51830 452.91     14.0257 0.0001803 ***
## I(time^3)  1    46327 447.86      8.9749 0.0027372 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Removing any other terms would increase the AIC of the model. Therefore, we conclude that we have found the optimal Normal model.

```
summary(model_norm2)$coefficients
```

```
##                 Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)  1.501911228 0.644571461  2.330093 2.480310e-02
## time         1.247136182 0.281972725  4.422897 7.016438e-05
## I(time^2)   -0.127134687 0.039459051 -3.221940 2.496116e-03
## I(time^3)    0.004723959 0.001765666  2.675455 1.067466e-02
```

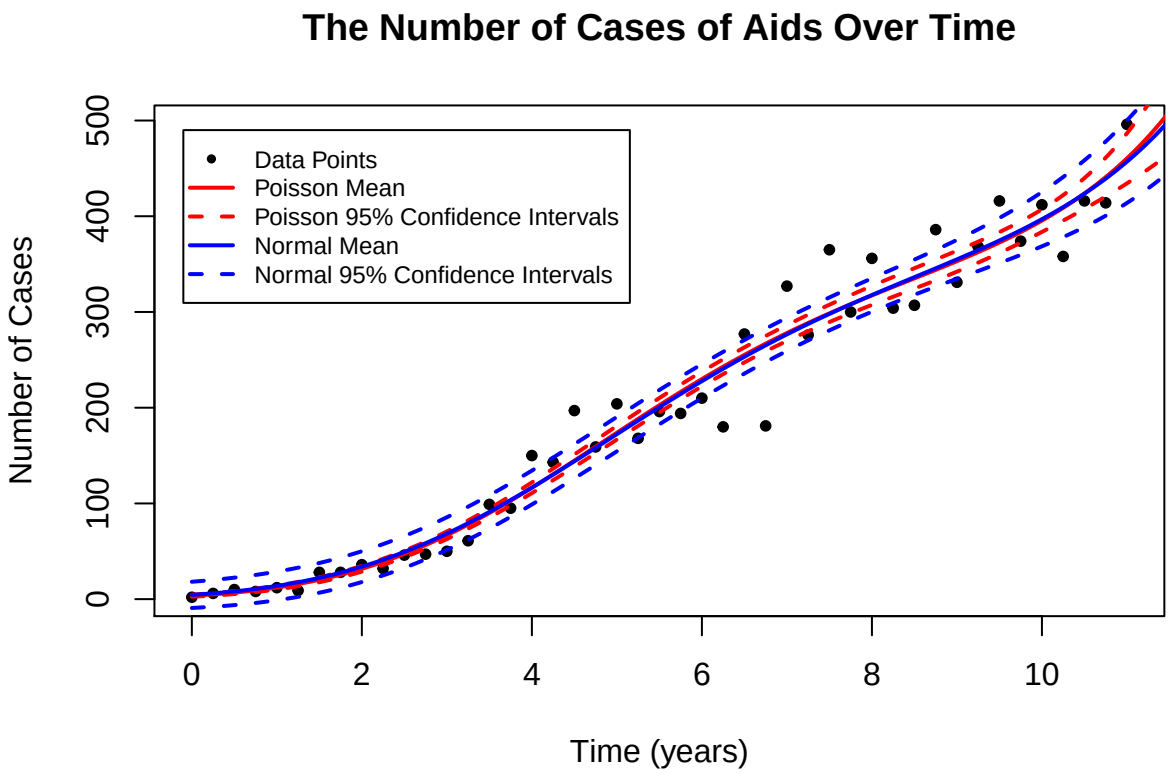
7e

The response variable in the dataset is the number of cases which is a form of count data and hence takes positive integer values. The plot in part (a) shows that the variance of the number of cases increases as the mean number of cases increases The Poisson distribution respects both of these properties, however the

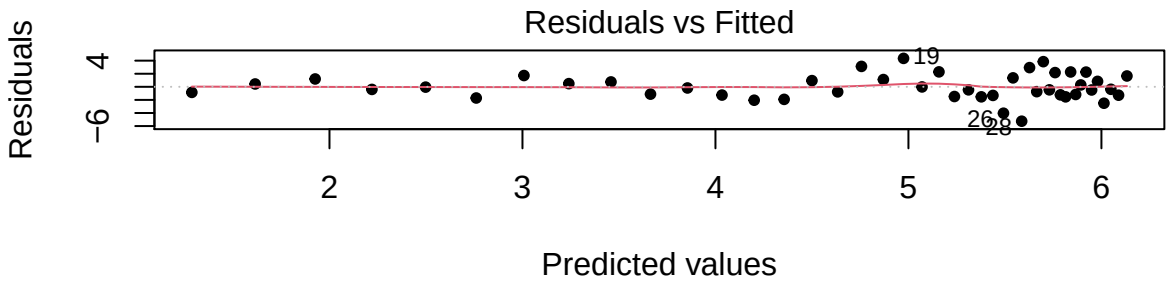
Normal distribution does not. Therefore, preliminary data analysis suggests that the Poisson model would be better suited to the dataset than a Normal model.

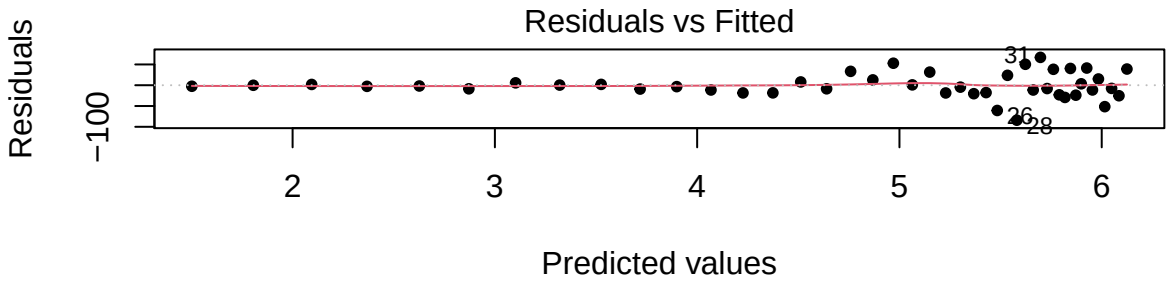
The plot below shows that the means of the model follow the trend in the data well. The 95% confidence interval of the means increase as the mean number of cases increase, highlighting that the level of uncertainty also increases due to the increase in variance.

```
plot(aids$time, aids$cases, xlab="Time (years)", ylab="Number of Cases",
     main='The Number of Cases of Aids Over Time',pch=20)

dates_list <- seq(81,94, length=200) # date range
times_list <- seq(0,12, length=200) # date range
# obtain the mean of the Poisson model over the range of dates
preds_pois2 <- predict(model_pois2, newdata=data.frame(time=times_list),type="response",
                       se.fit=T)
lines(times_list, preds_pois2$fit,lwd=2,col="red") # plot in red
# Calculate and plot the 95% confidence interval of the mean (approx. Gaussian)
upper_pois <- preds_pois2$fit+1.96*preds_pois2$se.fit
lines(times_list, upper_pois,lty=2,lwd=2,col="red")
lower_pois <- preds_pois2$fit-1.96*preds_pois2$se.fit
lines(times_list,lower_pois,lty=2,lwd=2,col="red")

# obtain the mean of the normal model over the range of dates
preds_norm2 <- predict(model_norm2, newdata=data.frame(time=times_list), type="response",
                       se.fit=T)
lines(times_list, preds_norm2$fit,lwd=2,col="blue") # plot in blue
# Calculate and plot the 95% confidence interval of the mean (approx. Gaussian)
upper_norm <- preds_norm2$fit+1.96*preds_norm$se.fit
lines(times_list, upper_norm,lty=2,lwd=2,col="blue")
lower_norm <- preds_norm2$fit-1.96*preds_norm$se.fit
lines(times_list, lower_norm,lty=2,lwd=2,col="blue")

legend(x=-0.1,y=490, legend=c("Data Points", "Poisson Mean",
                              "Poisson 95% Confidence Intervals","Normal Mean",
                              "Normal 95% Confidence Intervals"), pch=c(20,-1,-1,-1,-1),
       lty=c(-1,1,2,1,2), col=c('black','Red', 'Red','Blue','Blue'), lwd=c(-1,2, 2,2,2),
       cex = 0.75)
```



This figure below shows the residual error plots of the predictive values for the Poisson and Normal model respectively. It is important to note the difference in scales of the errors between the models. The Normal model's x-axis is an order of magnitude larger than that of the Poisson model, suggesting that the Poisson model is by far the most accurate model.





The data appears to be overdispersed which causes both of the models to break down at the upper tail of the dataset. This results in the models obtaining high deviance values. The deviance of the Poisson model and Normal model respectively are shown below.

```
model_pois2$deviance
```

```
## [1] 166.6792
```

```
model_norm2$deviance
```

```
## [1] 37950.73
```

The AIC of the Poisson and Normal model respectively are shown below. The Normal model has a lower AIC value than the Poisson model, suggesting that the Normal model is of better quality.

```
model_pois2$aic
```

```
## [1] 470.5294
```

```
model_norm2$aic
```

```
## [1] 440.8866
```

Upon reviewing the properties of both of the models, I do not believe that there is enough evidence to suggest that one model is better than the other. The properties of the Poisson model is better suited to the data and the deviance is far lower than the Normal model. However, the Normal model has a slightly lower AIC value, suggesting that it is the better model. Due to the inconclusive evidence, I have no reason to believe that one model is preferable.

7f

I have fit a Negative Binomial model to the data, including interactive terms up to time^4.

```
library(MASS) # contains the function glm.nb
model_nb <- glm.nb(cases~time+I(time^2)+I(time^3)+I(time^4), data=aids)
```

Let us see if removing a term will improve the AIC score of the model.

```
drop1(model_nb,test="Chi")
```

```
## Single term deletions
##
## Model:
## cases ~ time + I(time^2) + I(time^3) + I(time^4)
##           Df Deviance    AIC    LRT Pr(>Chi)
## <none>         45.022 405.91
## time       1   82.126 441.02 37.104 1.12e-09 ***
## I(time^2)  1   48.872 407.76  3.850  0.04974 *
## I(time^3)  1   45.410 404.30  0.389  0.53292
## I(time^4)  1   45.023 403.92  0.001  0.97384
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The AIC will be improved by removing the time^4 term. Therefore I remove this term and refit the model.

```
model_nb <- glm.nb(cases~time+I(time^2)+I(time^3), data=aids)
```

Let us see if any other terms should be removed.

```
drop1(model_nb,test="Chi")
```

```
## Single term deletions
##
## Model:
## cases ~ time + I(time^2) + I(time^3)
##           Df Deviance    AIC     LRT  Pr(>Chi)
## <none>         45.007 403.92
## time       1  228.417 585.33 183.410 < 2.2e-16 ***
## I(time^2)  1  100.697 457.60  55.689 8.488e-14 ***
## I(time^3)  1   71.502 428.41  26.495 2.643e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

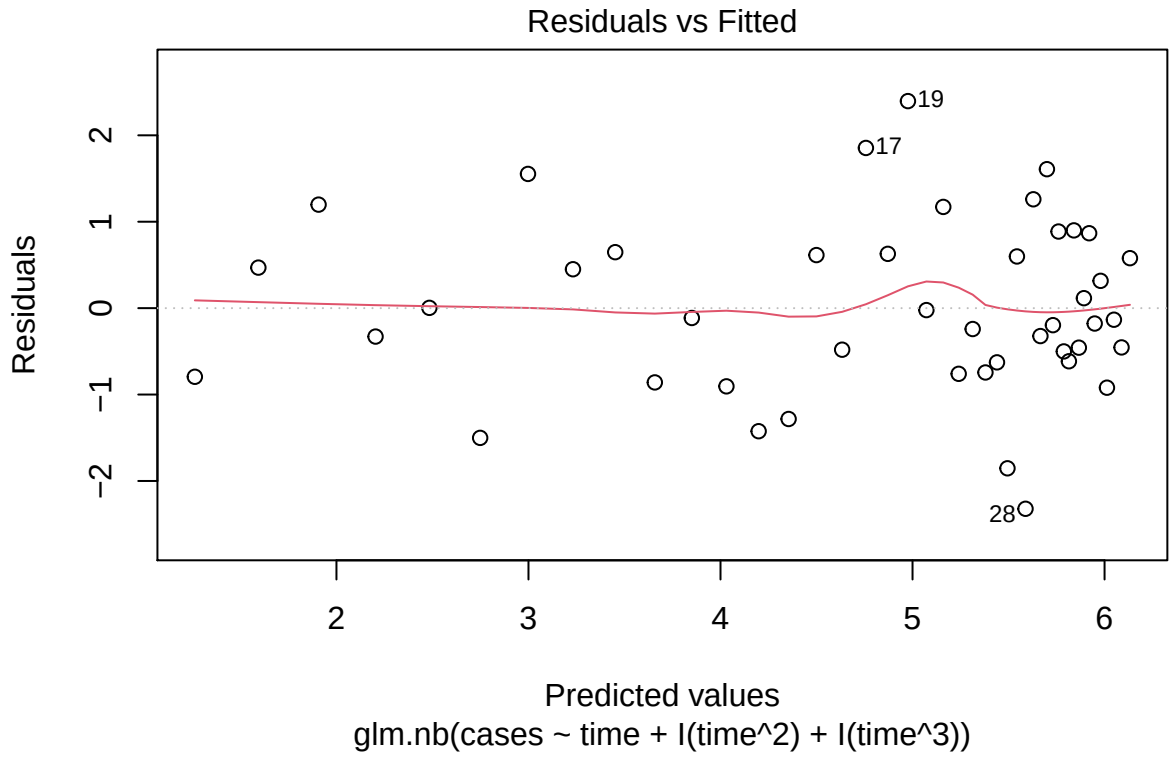
This shows that no other terms should be removed. We have therefore obtained the optimal coefficients for the Negative Binomial model and should therefore start to analyse the performance of the finalized model.

```
summary(model_nb)$coefficients
```

```
##                 Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)  1.262516824 0.18634280  6.775238 1.242016e-11
## time         1.358989791 0.11219373 12.112885 9.026272e-34
## I(time^2)   -0.143054229 0.02017434 -7.090898 1.332442e-12
## I(time^3)    0.005432359 0.00108955  4.985876 6.168184e-07
```

The graph shows that the Negative Binomial model obtains consistently low residual values throughout the dataset. The deviance of the model is also lower than the other two models at 45.

```
plot(model_nb,1)
```




```
model_nb$deviance
```

```
## [1] 45.00748
```

The AIC of the Negative Binomial model is lower than the other two models.

```
model_nb$aic
```

```
## [1] 405.9155
```

The analysis of the two models in part (d) indicated that our dataset contains over-dispersed count data. The Negative Binomial is a suitable model for such data as it has the same mean structure as Poisson regression and it has an extra parameter to model the over-dispersion.

The negative Binomial model obtained a a lower deviance and AIC compared to the Poisson and Normal model. Therefore, we can conclude that the Negative Binomial model is the preferable model.

Question 9

9a

The dataset titanic contains information about passengers on the titanic and whether or not they survived. I have fit a Bernoulli model to this data by setting $N = 1$ in the Binomial distribution. The covariates of the model are age, class, gender and their two-way interactions.

```
model.Bern <- glm(cbind(survived,1-survived)~age+pclass+gender+age:gender+pclass:gender,
                  data=titanic, family=binomial(link="logit"))
```

Let us use the Chai-squared test to see if removing any of the covariates increases the AIC of the model.

```
drop1(model.Bern,test="Chi")
```

```
## Single term deletions
##
## Model:
## cbind(survived, 1 - survived) ~ age + pclass + gender + age:gender +
##     pclass:gender
##               Df Deviance    AIC     LRT  Pr(>Chi)
## <none>             1126.6 1142.6
## age:gender     1   1127.2 1141.2  0.5613    0.4537
## pclass:gender  2   1157.2 1169.2 30.5885 2.279e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that removing the age:gender interaction term will reduce the AIC of the model. I fit the model again without this term and check if any other terms should be removed.

```
model.Bern <- glm(cbind(survived,1-survived)~age+pclass+gender +pclass:gender,
                  data=titanic, family=binomial(link="logit"))
drop1(model.Bern,test="Chi")
```

```
## Single term deletions
##
## Model:
## cbind(survived, 1 - survived) ~ age + pclass + gender + pclass:gender
##               Df Deviance    AIC    LRT  Pr(>Chi)
## <none>             1127.2 1141.2
## age            1   1189.5 1201.5 62.281 2.978e-15 ***
## pclass:gender  2   1172.7 1182.7 45.529 1.299e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that removing any other covariates will increase the AIC.

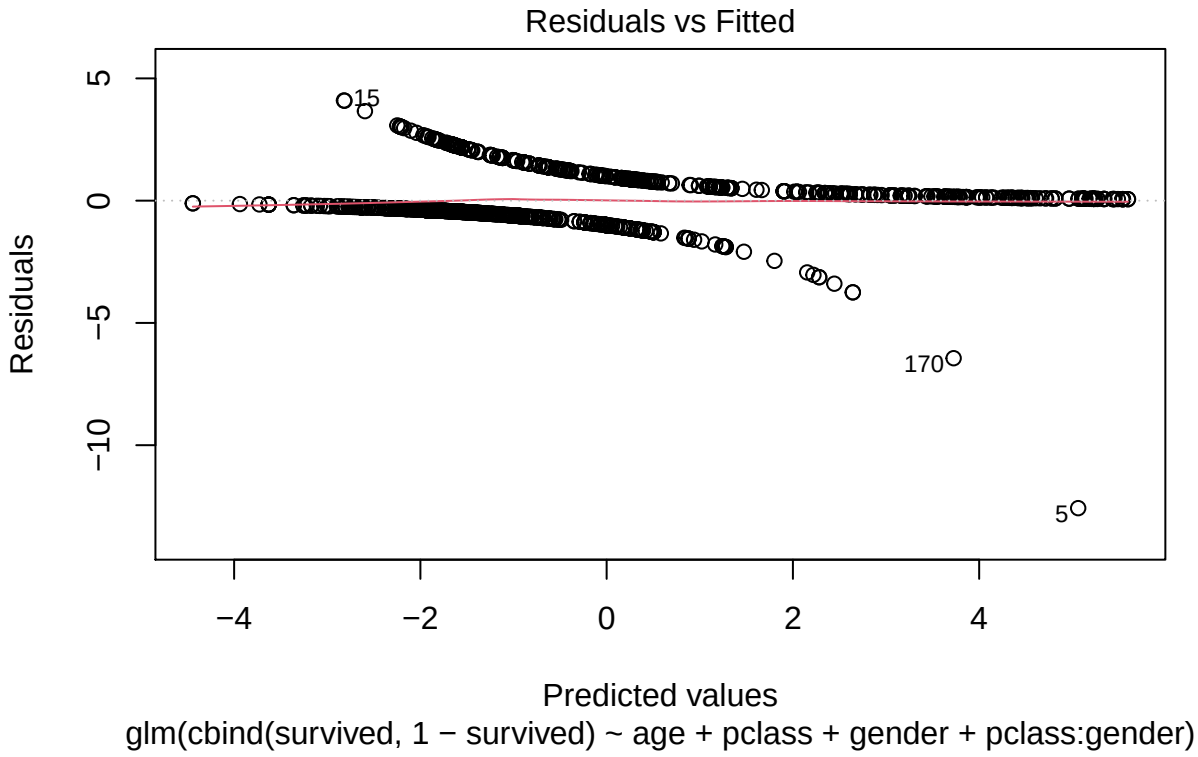
A Chai-squared goodness of fit test shows that the model fits the data better than the saturated model.

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

```
## [1] 0.9998278
```

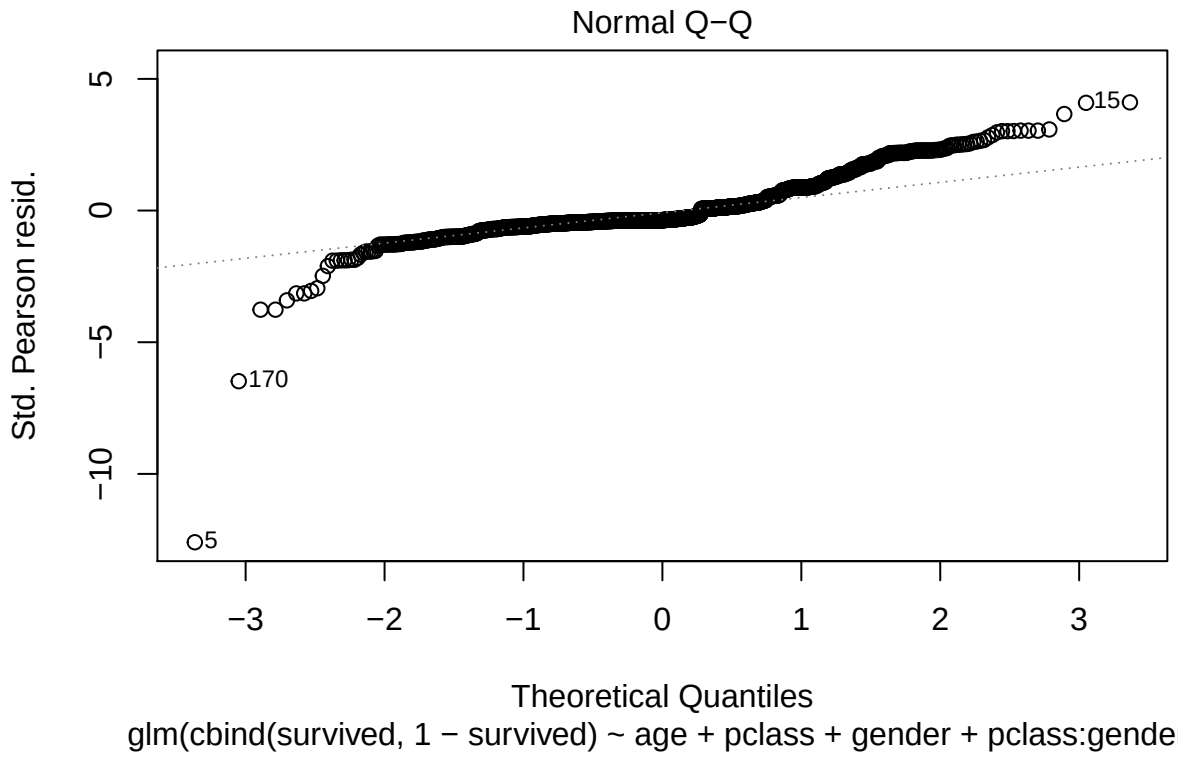
In a classification task with two outcomes, there are two types of errors: false negatives and false positives. The residual plot shows that our model obtains larger errors if a data point is more similar to it's opposite class.

```
plot(model.Bern,1)
```



The Q-Q plot shows that the model performs well at classifying the body of the data but experiences some difficulty at classifying the tails, particularly the upper tail.

```
plot(model.Bern,2)
```



The model summary shows the relationship between the covariates of the mean of the model. The intercept is the mean (at logit scale) probability of survival for a female of 0.08 years in passenger class 1. The rest of the coefficients express differences from this baseline.

For the effect of a covariate to be deemed significant, it must have a p-value of less than 0.05. Therefore, all of the coefficients are deemed significant.

```
summary(model.Bern)$coefficients
```

```
##                      Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept)         6.37453825 0.782345165  8.147987 3.700321e-16
## age                -0.05275265 0.007056516 -7.475735 7.677354e-14
## pclass2            -2.55268066 0.795669461 -3.208217 1.335605e-03
## pclass3            -5.02853346 0.746152561 -6.739283 1.591698e-11
## gendermale         -4.97534967 0.734919525 -6.769924 1.288496e-11
## pclass2:gendermale  1.13697705 0.832229300  1.366182 1.718817e-01
## pclass3:gendermale  3.10222963 0.760084050  4.081430 4.475951e-05
```

It is useful to use plots to interpret the effects of 2-way interactions in a model.

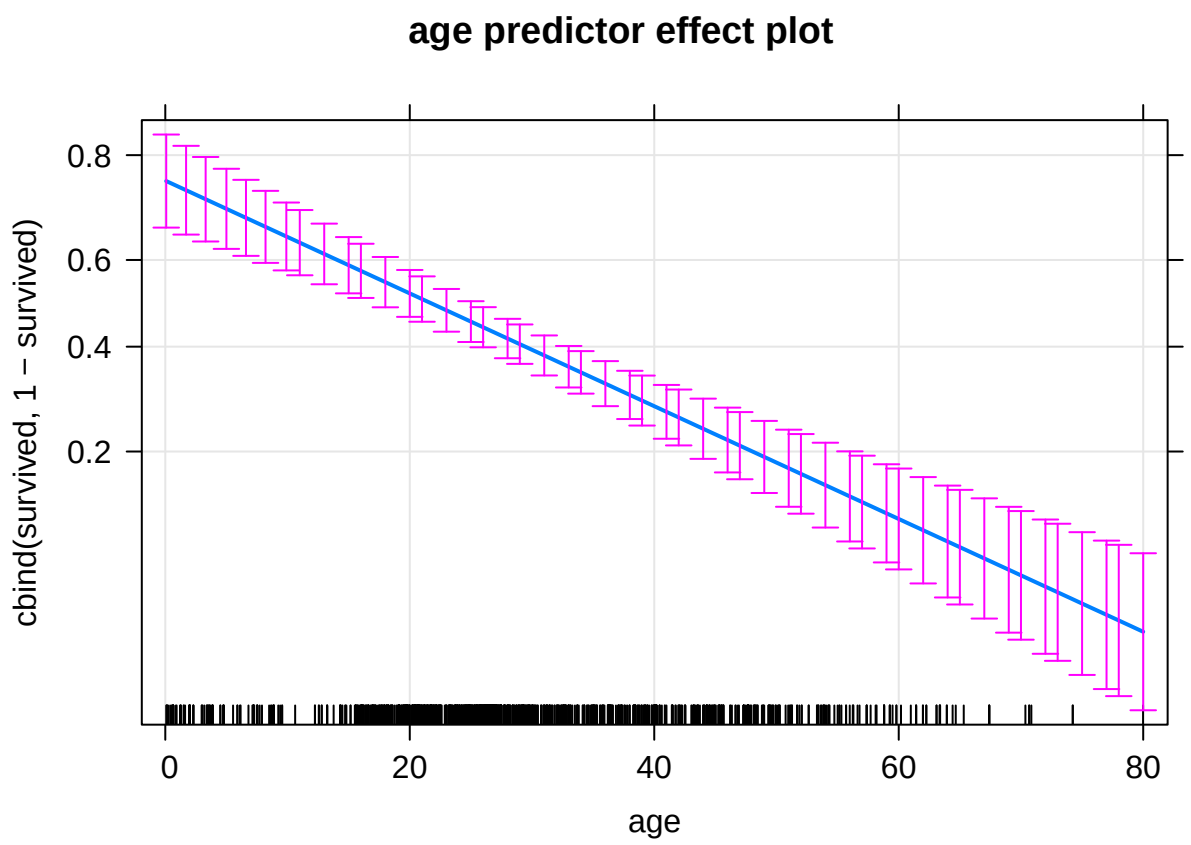
The plot below shows that age is a very strong predictor of survival. A younger a passenger has a larger chance of survival than an older passenger.

```
library(effects) # load effects library
```

```
## Loading required package: carData
```

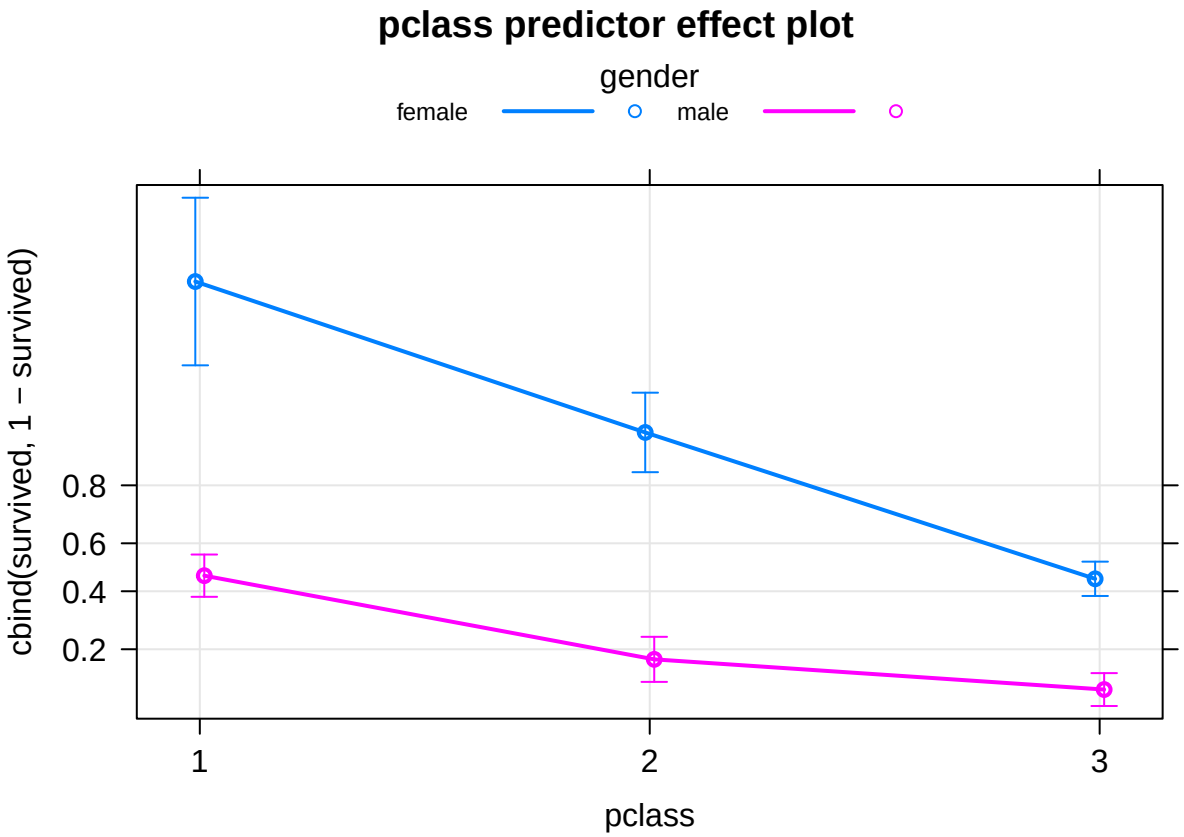
```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
plot(predictorEffects(model.Bern), lines=list(multiline=TRUE),axes=list(grid=TRUE),
     confint=list(style="bars"),1)
```



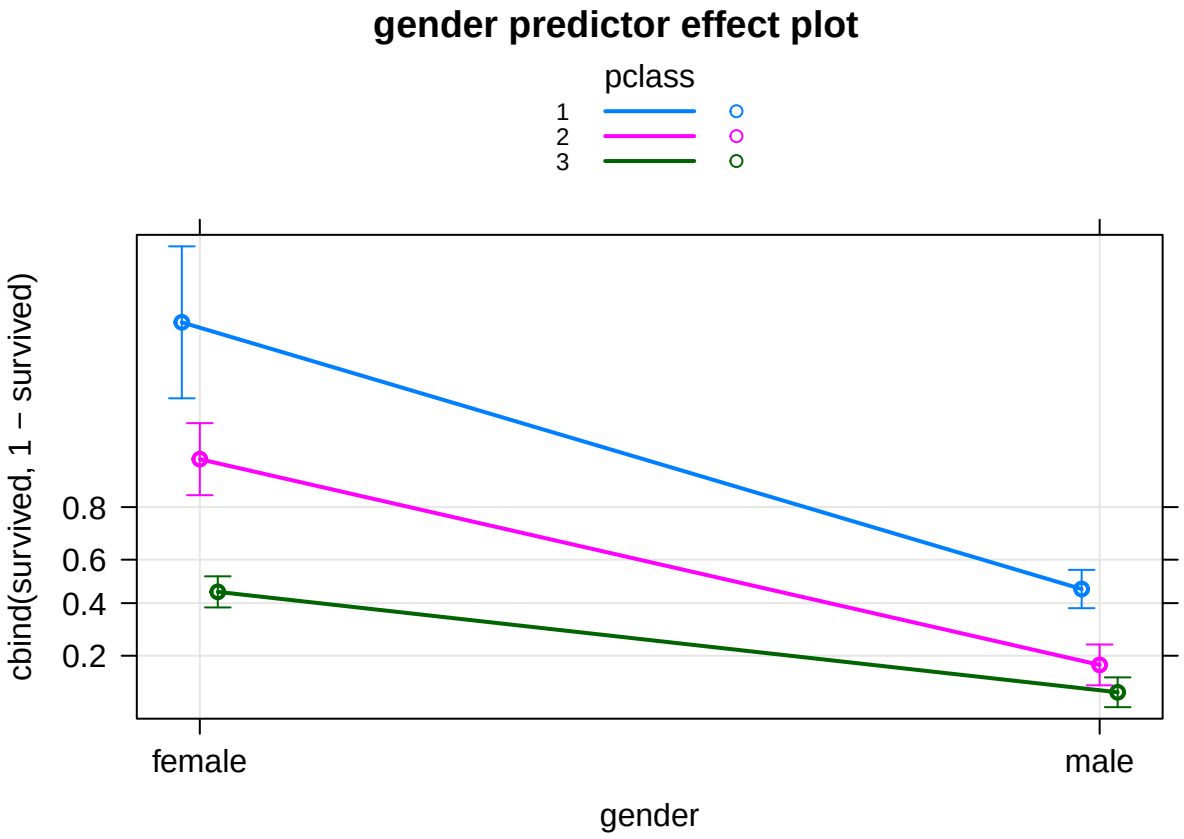
This plot shows that the survival probability of a passenger significantly decreases as the class of the passenger decreases. In all three classes, females have a higher probability of surviving than males in the same class.

```
plot(predictorEffects(model.Bern), lines=list(multiline=TRUE),axes=list(grid=TRUE),
     confint=list(style="bars"),2)
```



This plot reiterates the points in the plot above: women are more likely to survive than men and passengers in higher classes are more likely to survive than passengers in lower classes.

```
plot(predictorEffects(model.Bern), lines=list(multiline=TRUE),axes=list(grid=TRUE),
     confint=list(style="bars"),3)
```


gender predictor effect plot
pclass
1
2
3
cbind(survived, 1 − survived)
0.8
0.6
0.4
0.2
female
male
gender

Question 11

To understand the impact of the covariates on the proportions of fragments of animal bones, let us fit a model to the dataset containing all 3-way interactions, all 2-way interactions and all of the main effects.

```
model.Bones1 <- glm(Nbone~surface_geology*Era*Animal,data=bones,family=poisson)
```

Perform a Chai-squared test to see if any of the covariates should be removed from the model.

```
drop1(model.Bones1,test="Chi")
```

```
## Single term deletions
##
## Model:
## Nbone ~ surface_geology * Era * Animal
##                             Df Deviance    AIC    LRT Pr(>Chi)
## <none>                           0.0000 170.21
## surface_geology:Era:Animal  6   5.6434 163.85 5.6434   0.4643
```

The Chai-squared test tells us that the 3-way interaction term should be removed. I therefore fit the model again using all 2-way interactions and the main effects.

```
model.Bones2 <- glm(Nbone~(surface_geology+Era+Animal)^2,data=bones,family=poisson)
```

I perform another Chai-squared test to see if further covariates should be dropped.

```
model.Bones2$aic
```

```
## [1] 163.8539
```

```
drop1(model.Bones2,test="Chi")
```

```
## Single term deletions
##
## Model:
## Nbone ~ (surface_geology + Era + Animal)^2
##                         Df Deviance    AIC     LRT  Pr(>Chi)
## <none>                       5.643 163.85
## surface_geology:Era      3  225.562 377.77 219.919 < 2.2e-16 ***
## surface_geology:Animal   2   24.358 178.57  18.715 8.633e-05 ***
## Era:Animal               6   29.046 175.26  23.402 0.0006723 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that removing any of the covariates will increase the AIC. We have therefore created the finalized version of our model.

The model summary shows the relationship between the covariates of the mean of the model. The intercept is the mean (at logit scale) number of bones found of cattle in the early medieval period found in soil type other. The rest of the coefficients express differences from this baseline.

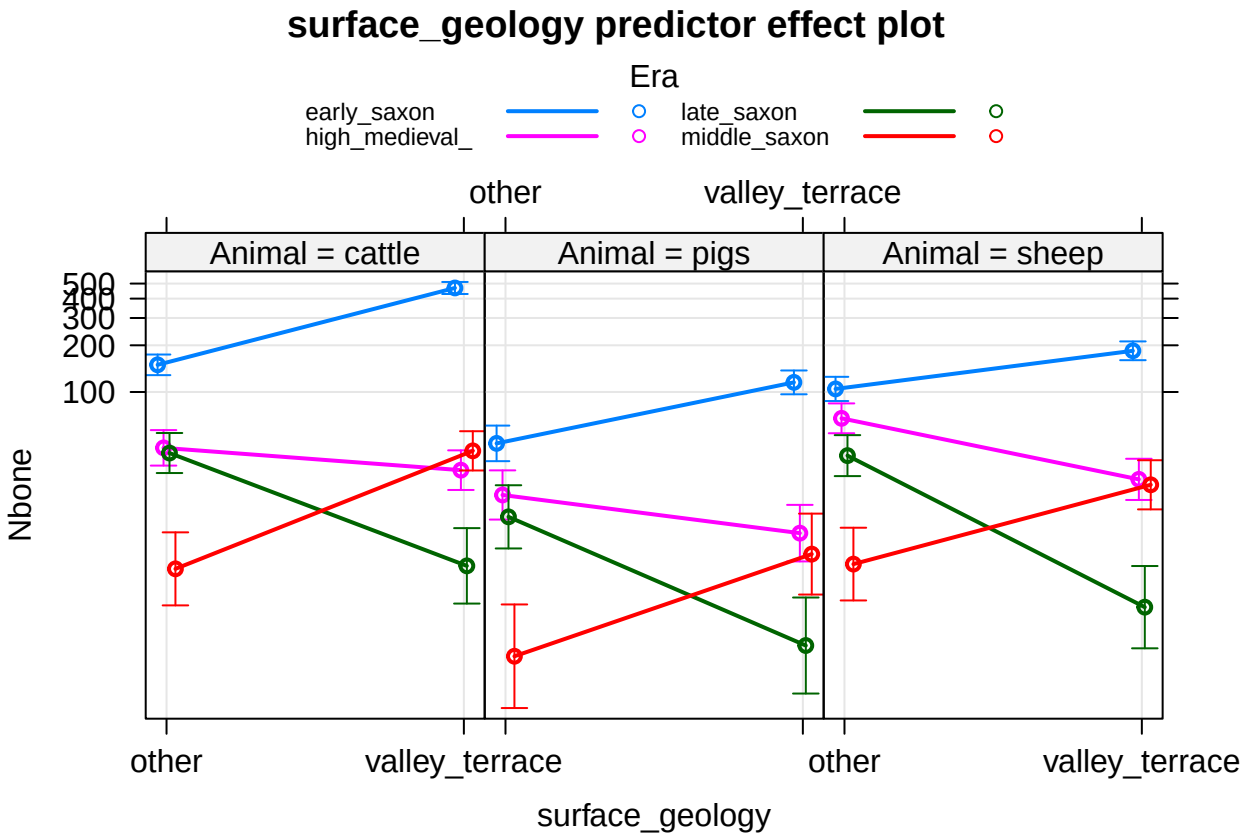
```
summary(model.Bones2)$coefficients
```

```
##                                                  Estimate Std. Error
## (Intercept)                                     5.0084864 0.07808657
## surface_geologyvalley_terrace                   1.1406696 0.08831883
## Erahigh_medieval_                              -1.2322108 0.14708988
## Eralate_saxon                                  -1.3090504 0.16826583
## Eramiddle_saxon                                -3.0282676 0.27901086
## Animalpigs                                     -1.1657980 0.15210205
## Animalsheep                                    -0.3576619 0.11551070
## surface_geologyvalley_terrace:Erahigh_medieval_ -1.4718317 0.16399564
## surface_geologyvalley_terrace:Eralate_saxon     -2.8151862 0.29536769
## surface_geologyvalley_terrace:Eramiddle_saxon    0.6109468 0.27873581
## surface_geologyvalley_terrace:Animalpigs        -0.2353934 0.17135527
## surface_geologyvalley_terrace:Animalsheep       -0.5747734 0.13251680
## Erahigh_medieval_:Animalpigs                     0.4664157 0.23262589
## Eralate_saxon:Animalpigs                         0.2186308 0.30718833
## Eramiddle_saxon:Animalpigs                      -0.1312220 0.34570826
## Erahigh_medieval_:Animalsheep                    0.7958213 0.17690268
## Eralate_saxon:Animalsheep                        0.3191517 0.23483346
## Eramiddle_saxon:Animalsheep                      0.4284169 0.23809272
##                                                    z value     Pr(>|z|)
## (Intercept)                                     64.1401805 0.000000e+00
## surface_geologyvalley_terrace                   12.9153616 3.687080e-38
## Erahigh_medieval_                               -8.3772645 5.417203e-17
## Eralate_saxon                                   -7.7796566 7.272165e-15
## Eramiddle_saxon                                -10.8535831 1.917576e-27
## Animalpigs                                      -7.6645777 1.794206e-14
## Animalsheep                                     -3.0963525 1.959173e-03
## surface_geologyvalley_terrace:Erahigh_medieval_ -8.9748218 2.838108e-19
## surface_geologyvalley_terrace:Eralate_saxon     -9.5311243 1.555890e-21
## surface_geologyvalley_terrace:Eramiddle_saxon    2.1918491 2.839040e-02
## surface_geologyvalley_terrace:Animalpigs        -1.3737155 1.695300e-01
## surface_geologyvalley_terrace:Animalsheep       -4.3373627 1.442026e-05
## Erahigh_medieval_:Animalpigs                     2.0050034 4.496268e-02
## Eralate_saxon:Animalpigs                         0.7117158 4.766408e-01
## Eramiddle_saxon:Animalpigs                      -0.3795744 7.042614e-01
## Erahigh_medieval_:Animalsheep                    4.4986389 6.838990e-06
## Eralate_saxon:Animalsheep                        1.3590553 1.741291e-01
## Eramiddle_saxon:Animalsheep                      1.7993700 7.196017e-02
```

Let us view plots to interpret the effects of the 2-way interactions of the model.

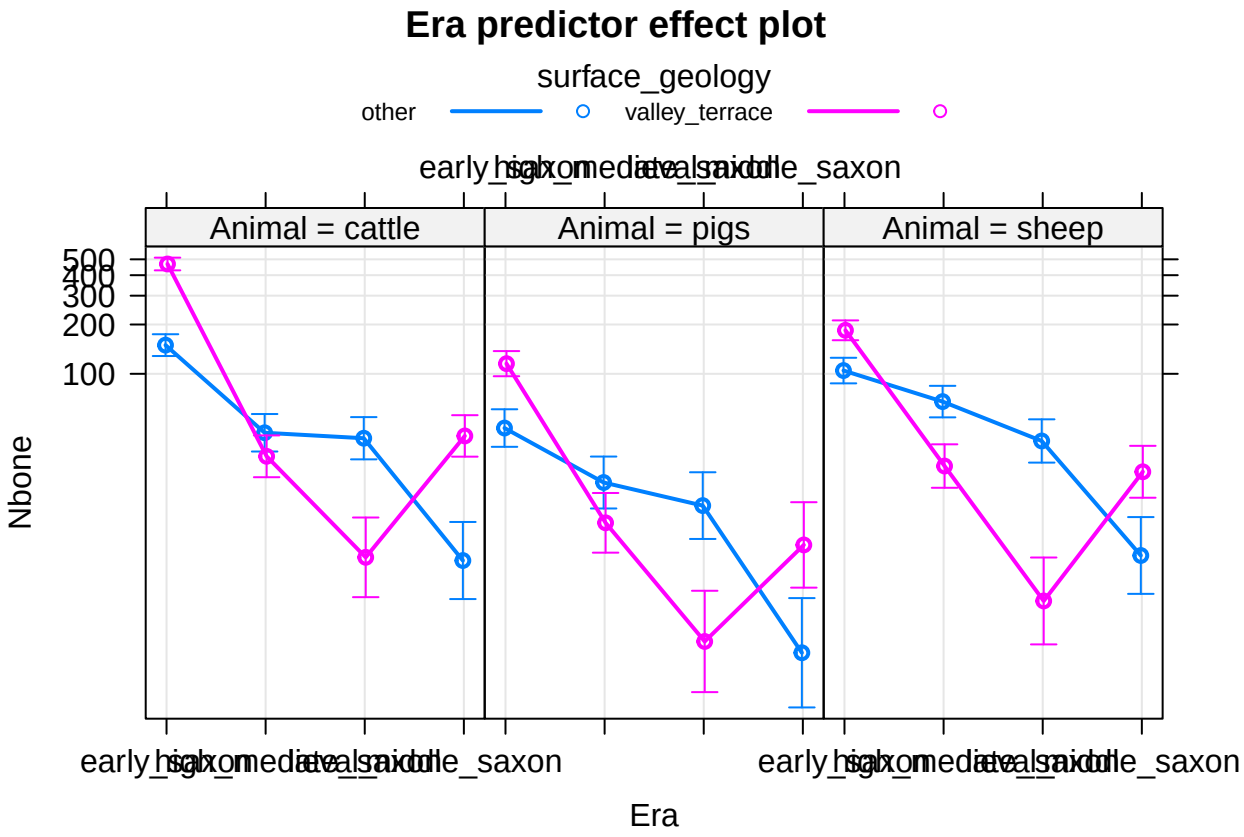
The plot below shows that cattle bones are significantly the most common bone type, then sheep and then pigs. The plot also shows that bones that come from the early to middle era of the Saxons are more likely to found in valley terraces than other geology types, regardless of animal. The inverse is true for bones that come form the late Saxon and high Medieval era.

```
plot(predictorEffects(model.Bones2),
     lines=list(multiline=TRUE),axes=list(grid=TRUE),confint=list(style="bars"), 1)
```



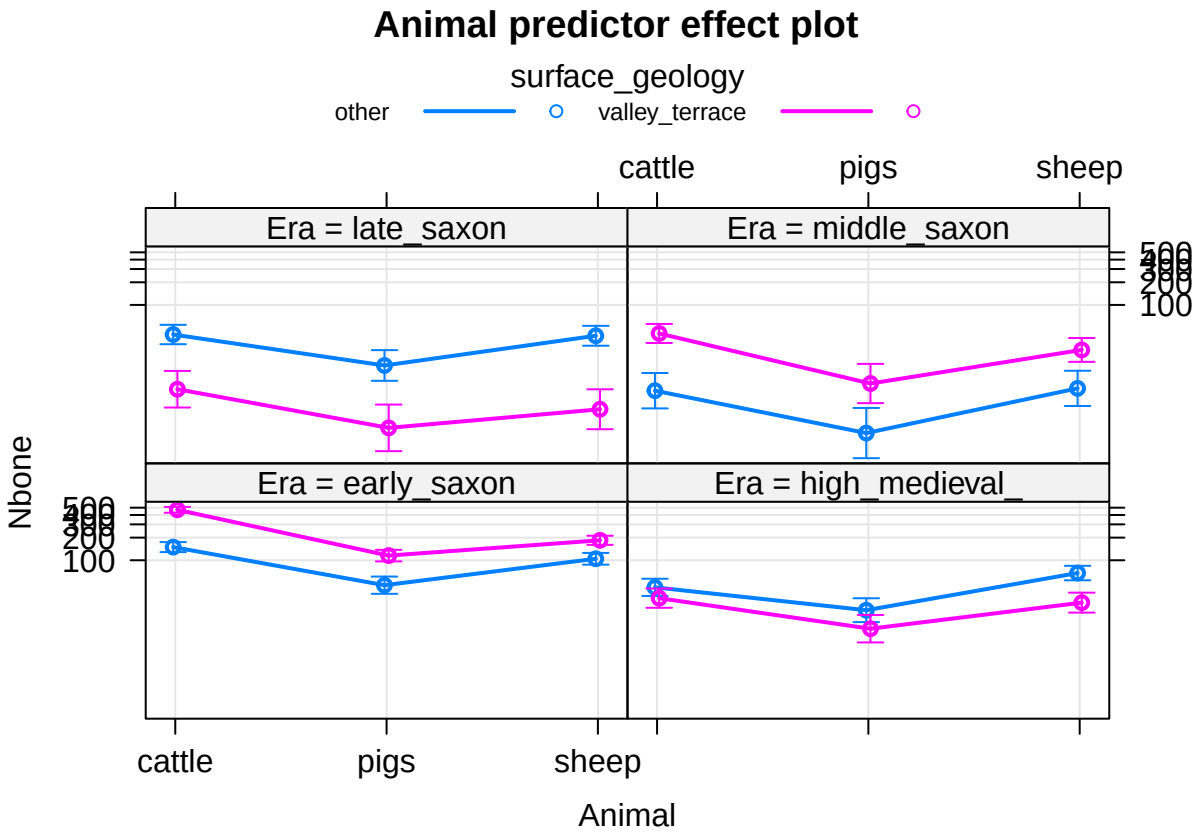
This graph shows the trends over time of the number of bones found in different geologies. For animal bones found in a valley terrace, the number of bones found significantly decreases from the early middle to late Saxon period, and then significantly increases in the high Medieval period, this is true for all three animal types. A different trend is found in other geology types, the number of bones for each animal decreases from the early to middle Saxon period, then the number of bones increases in the late Saxon period and finally remains constant in the late Medieval period.

```
plot(predictorEffects(model.Bones2),
     lines=list(multiline=TRUE),axes=list(grid=TRUE),confint=list(style="bars"), 2)
```



This plot shows that cattle bones were the most common type of bone found in the early Saxon period, however sheep bones are significantly more common in the three other periods. Across all periods, pig bones are significantly the least common bone to find. In the early and middle Saxon period, more bones were found in valley terraces than other geology types, in the late Saxon period however, the inverse in true. In the high Medieval period, the geology type does not have a significant effect on the number of cattle and pig bones found, however sheep bones are more common in other geologies than a valley terrace.

```
plot(predictorEffects(model.Bones2),
     lines=list(multiline=TRUE),axes=list(grid=TRUE),confint=list(style="bars"), 3)
```



I have used a Poisson GLM for the cell counts with a log-link function for the classifying factors to create a Poisson log-linear model. I have assumed that the bones were randomly found in a given time period, meaning that all of the covariates are free to vary. Therefore, I looked to include all of the covariates in the linear predictor.

Question 13

The gehan dataset contains data on a trial concerning 42 leukemia patients, some were treated with 6-mercaptopurine and the rest were controls. The trial was designed as matching pairs, both withdrawn from the trial when either patient came out of remission.

13a

The dataset contains data on patients that were removed from the experiment before they came out of remission. I create a dataset with these data points removed to simplify the modeling process.

```
reduced <- gehan[!(gehan$cens==0),] # censor the data
```

I ensure that the control patients are buried in the intercept. The Exponential distribution is equal to the Gamma distribution with the dispersion parameter set to 1. I therefore fit a Gamma model to the data and obtain summary statistics of the model with dispersion set to 1.

```
reduced$treat <- relevel(reduced$treat, "control")
model.exp <- glm(time~treat, data=reduced, family=Gamma(link='log'))
```

The model obtains a Chai-squared goodness of fit test that indicates that the model fits as well as the saturated model.

```
1-pchisq(deviance(model.exp),model.exp$df.residual)
```

```
## [1] 0.9567503
```

The model summary shows the relationship effect of the covariate on the model mean. The intercept is the mean (at logit scale) recovery time of the control group. The treat6-MP variable is the effect that being treated with 6-mercaptopurine has on recovery time with respect to the control group.

```
summary(model.exp, dispersion=1)$coefficients
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) 2.1594842  0.2182179 9.8960000 4.332583e-23
## treat6-MP   0.3346391  0.3984095 0.8399374 4.009435e-01
```

The p-value of the treat6-MP covariate estimation is 0.4, therefore the effect of 6-mercaptopurine is not significantly non-zero. Therefore, the effect of 6-mercaptopurine is unknown and more data should be collected.

13 b

The likelihood of the exponential distribution can be written as

$$
\begin{aligned}
L(\beta_0, \beta_1; y) &= \prod_{i=1}^{n} p(y_i; \lambda_i)^{c_i} [Pr(Y > y_i; \lambda_i)]^{1-c_i} \\
&= \prod_{i=1}^{n} (\lambda_i e^{-\lambda_i y_i})^{c_i} (e^{-\lambda_i y_i})^{1-c_i} \\
&= \prod_{i=1}^{n} \lambda_i^{c_i} e^{-\lambda_i y_i c_i} e^{-\lambda_i y_i} e^{\lambda_i y_i c_i} \\
&= \prod_{i=1}^{n} \lambda_i^{c_i} e^{-\lambda_i y_i} \\
&= \prod_{i=1}^{n} \lambda_i^{c_i} e^{-\lambda_i y_i} \times \frac{y_i^{c_i}}{y_i^{c_i}} \\
&= \prod_{i=1}^{n} \frac{\lambda_i^{c_i} y_i^{c_i} e^{-\lambda_i y_i}}{y_i^{c_i}} \\
&= \prod_{i=1}^{n} \frac{(\lambda_i y_i)^{c_i} e^{-\lambda_i y_i}}{y_i^{c_i}}
\end{aligned}
$$

13 c

Discarding data on patients that were removed before coming out of remission gets rid of potentially useful information. I therefore look for a model that can incorporate this data effectively. First I ensure that the control patients are buried in the intercept.

```
gehan$treat <- relevel(gehan$treat, "control")
```

The likelihood that I calculated in part (b) is equivalent to the Poisson likelihood up to a constant. I can therefore fit a Poisson model with a mean $\lambda_i y_i$ to the dataset, where λ is the predicted means from the Exponential model in part (a).

```
preds_i <- predict(model.exp, newdata=data.frame(treat=gehan$treat),type="response", se.fit=T)
lambda <- 1/preds_i$fit
```

Algebraic analysis gives us the link function that ensures that x_i has the same interpretation on λ_i as in part (a).

$$
\begin{aligned}
C_i \sim Pois(\mu_i) \qquad \mu_i = \lambda_i y_i \\
log(\mu_i) &= \beta_0 + \beta_1 x_i \\
log(\lambda_i y_i) &= \beta_0 + \beta_1 x_i \\
log(\lambda_i) + log(y_i) &= \beta_0 + \beta_1 x_i \\
log(\lambda_i) &= -log(y_i) + \beta_0 + \beta_1 x_i
\end{aligned}
$$

I now fit a Poisson model to the data that has a log link function that incorporates an offset of $-log(y_i)$.

```
gehan$log.time <- log(gehan$time)
model.pois <- glm(time~-offset(log.time)+treat, data=gehan, family=poisson(link='log'))
```

A Chai-squared goodness of fit test shows that the model fits the data well.

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

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

The summary of the extended model strongly suggests that the there is not a significant difference between the recovery time of the control group and the patients given 6-mercaptopurine. I therefore conclude that 6-mercaptopurine does not effect on the recovery of Leukemia patients.

```r
summary(model.pois)$coefficients
```

```
##                  Estimate Std. Error       z value Pr(>|z|)
## (Intercept)  3.643174e-17 0.07412493  4.914911e-16        1
## treat6-MP   -4.562215e-17 0.09099462 -5.013720e-16        1
```