506CourseWork

Table of contents

For the love of god don't forget

Do not display too much raw R output (e.g. don't display the full output of 'summary(model)'), but edit this down to the essentials. Ensure to include justification for each step of your analyses, providing comments alongside your R code to explain what you are doing and add appropriate titles and labelled axes to your plots.

Question 1

We have the model:

$Yi \sim N(\frac{\theta 1 xi}{\theta 2 + xi}, \sigma^2)$

Question 1 a)

Due to the visible non-linearity of the model, we would be required to significantly transform our data to get a linear model that would have an acceptable fit of the data. We can also see that the data seems to be only positive while a normal distribution goes from]-∞,∞[. Such arbitrary transformation increases the complexity of the model, making it less interpretable and not respect the nature of the data.

Linear regression models are based on the assumption that the relationship between the independent and dependent variables is linear. If the relationship between the variables is non-linear, a linear regression model is not be appropriate. In such cases, transforming the data to make the relationship linear may not result in an accurate representation of the true relationship, and can lead to overfitting or underfitting. Additionally, transforming the data can result in a loss of interpretability of the results, as it can be difficult to understand the meaning of the transformed variables.

Another issue with using a linear regression model for non-linear data is that the residuals, which represent the difference between the observed and predicted values, may not be normally distributed, which is another assumption of linear regression models. This can lead to biased or incorrect results.

In conclusion, when the data is non-linear, a linear regression model may not be the best choice for modelling the relationship between the variables, and alternative methods need to be considered.

Question 1 b)

The Yi are independent so the likelihood is a product of the individual pdfs.

Likelihood of a normal distribution where $L(yi|\mu,\sigma^2)$ =

$= \Pi_{i=1}^{n} fX(yi|\mu,\sigma^2)$

$= \Pi_{i=1}^{n}((2\pi\sigma^2)^{-\frac{1}{2}} * exp\left(-\frac{1}{2} * \frac{(yi-\mu)^2}{\sigma^2}\right)) =$

$= (2\pi\sigma^2)^{-\frac{n}{2}} * exp\left(-\frac{1}{2\sigma^2} * \sum_{i=1}^{n}(yi-\mu)^2\right)$

Replacing the μ with the respective θs formula and n, we have the likelihood as:

$L(\beta 0, \beta 1, \sigma^2; x, y) =$

$= \prod_{i=1}^{n} p(\beta 0, \beta 1, \sigma^2; x, y) =$

$= (2\pi\sigma^2)^{-\frac{100}{2}} * exp\left(-\frac{1}{2\sigma^2} * \sum_{i=1}^{100}\left(yi - \frac{\theta 1 xi}{\theta 2 + xi}\right)^2\right)$

The log-likelihood of a normal distribution is:

$l(yi|\mu,\sigma^2) =$

$= ln\left(L(yi|\mu,\sigma^2)\right) =$

$= ln\left((2\pi\sigma^2)^{-\frac{n}{2}} * exp\left(-\frac{1}{2\sigma^2} * \sum_{i=1}^{n}(yi-\mu)^2\right)\right) =$

$= ln\left((2\pi\sigma^2)^{-\frac{n}{2}}\right) + ln\left(exp\left(-\frac{1}{2\sigma^2} * \sum_{i=1}^{n}(yi-\mu)^2\right)\right) =$

$= -\frac{n}{2}ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} * \sum_{i=1}^{n}(yi-\mu)^2) =$

$= -\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\sigma^2) - \frac{1}{2\sigma^2} * \sum_{i=1}^{n}(yi-\mu)^2)$

Once again replacing the μ with the respective θs formula and n, we have the log-likelihood as:

$l(\beta 0, \beta 1, \sigma^2; x, y) =$

$= -\frac{100}{2}ln(2\pi) - \frac{100}{2}ln(\sigma^2) - \frac{1}{2\sigma^2} * \sum_{i=1}^{100}\left(yi - \frac{\theta 1 xi}{\theta 2 + xi}\right)^2) =$

$= -50ln(2\pi) - 50ln(\sigma^2) - \frac{1}{2\sigma^2} * \sum_{i=1}^{100}\left(yi - \frac{\theta 1 xi}{\theta 2 + xi}\right)^2)$

Question 1 c)

```
n = nrow(nlmodel)


### Create a function to evaluate minus the log-likelihood
myLike = function(variables) {

    theta1 = variables[1]  #theta1
    theta2 = variables[2]  #theta2
    sigma = variables[3]  #sigma

    mu = ((theta1 * nlmodel$x)/(theta2 + nlmodel$x))

    # Log-likelihood
    result = (-(n/2) * log(2 * pi)) - ((n/2) * log(sigma^2)) - (1/(2 *
(sigma^2))) *
        (sum((nlmodel$y - mu)^2))

    # Returning negative log-likelihood
    return(-result)
}
```

Question 1 d)

From the graph data we can clearly see that the deviantion is approximatly 15 from the value scatter which is easier to destinguish and measure between the [0.5,1] interval.

We can observethat when x -> 1 y is approximately 215 and as x -> 0 y is approximately to 50

To determine the thetas we will first see see that when x approximates to zero we can observe that:

$$\lim_{x->0} \frac{\theta 1xi}{\theta 2 + xi} \Longrightarrow \frac{1}{\theta 2}$$

As such $ Y$ $\sim N\left(\frac{\theta 1xi}{\theta 2+xi}\right)$ becomes $50 \sim \frac{1}{\theta 2}$

Solving it for $\theta 2$ we get:

$\theta 2 = 1/50 \Leftrightarrow \theta 2 = 0.02$

Now that we have $\theta 2$ we can use the approximation of x to 1 to determine the value of $\theta 1$

$$\lim_{x->1} \frac{\theta 1xi}{\theta 2 + xi} \Longrightarrow \frac{\theta 1}{\theta 2 + x}$$

As such $Y \sim N\left(\frac{\theta 1xi}{\theta 2+xi}\right)$ becomes $215 \sim \frac{\theta 1*1}{\theta 2+1}$

Solving it for $\theta 1$ we get:

$\theta 1 = 215/1.02 \Leftrightarrow \theta 1 \approx 210.7$

```
# Estimating the MLE
out <- nlm(myLike,
  p = c(210.7,0.02,15), #plugging in the starting values
  hessian = T,
  iterlim = 10000,
  steptol = 1e-10)

# Reporting estimates
variableEstimates = out$estimate
out$estimate
```

```
[1] 214.65008415   0.06353447  13.61564428
```

Question 1 e)

```
# Invert the negated Hessian to obtain the Observed Information Matrix
OIM <- solve(out$hessian)

# The diagonal entries are the variances of beta0 and beta1
# respectively so # obtain them
VarianceBeta <- diag(OIM)

# and then square root them to obtain standard errors
stand_error <- sqrt(VarianceBeta)
```

```
# reporting standard errors
stand_error

[1] 2.674798031 0.005140379 0.963024267
```

The formula to calculate a 99% confidence interval is: $\beta \pm 2.576 * SE(\beta)$

```
# Estimating CIs
CIs <- cbind(variableEstimates - 2.576 * stand_error, variableEstimates +
    2.576 * stand_error)

# Reporting the CIs
CIs

             [,1]         [,2]
[1,] 207.75980442 221.54036388
[2,]   0.05029285   0.07677609
[3,]  11.13489377  16.09639479
```

Question 1 f)

H0 : θ2 = 0,08 vs. H1 : θ2 ≠ 0,08

```
## Hypothesis thest without using confidence interval

z_stat <- (variableEstimates[2] - 0.08)/stand_error[2]

# Print the test values
z_stat  ## significance tests

[1] -3.203174
```

So now we need to decide if this value of the z-statistic is extreme at the 10% significance level

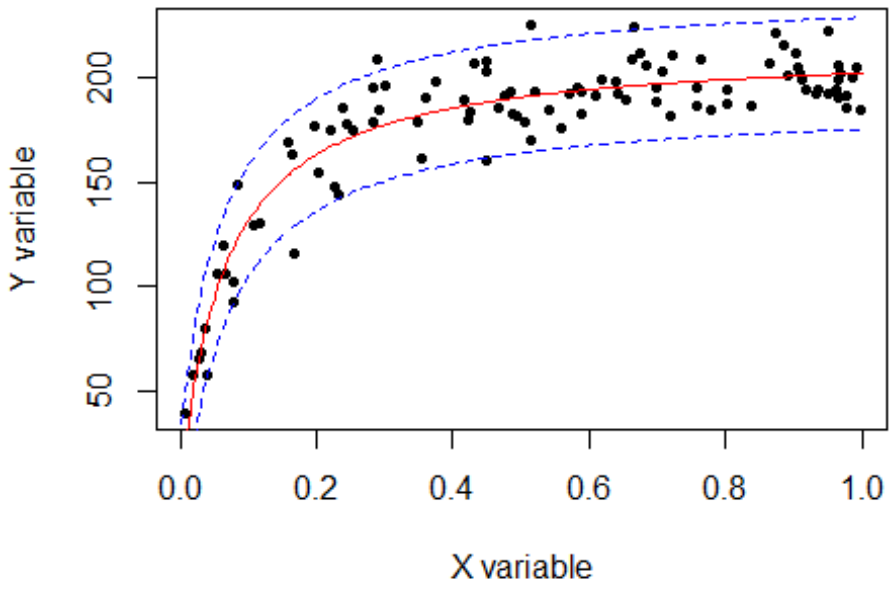
```
### Note that equivalently we can look at the 95% quantile of N(0,1)
qnorm(0.95, 0, 1)

[1] 1.644854

qnorm(0.05, 0, 1)

[1] -1.644854
```

As we can see the value from the z-statistician test is considerably lower than -1,645, as the value is not between the [-1.644854, 1.644854] is an extreme value and therefore we reject the null hypothesis that $\theta 2$ is 0,08.

Question 1 g)

```
# Plotting initial starting guess
xx <- seq(0, 1, len = 200)
```

```
# Estimating mean relationship (mean mu = )
mu <- (out$estimate[1] * xx)/(out$estimate[2] + xx)

# Getting standard Deviation
standardDeviation <- out$estimate[3]


# Getting 95% interval from the quantiles of a Normal distribution
plot(nlmodel$x, nlmodel$y, pch = 20, xlab = "X variable", ylab = "Y
variable")

lines(xx, qnorm(0.025, mean = mu, sd = standardDeviation), col = "blue",
    lty = "dashed")
lines(xx, qnorm(0.975, mean = mu, sd = standardDeviation), col = "blue",
    lty = "dashed")

lines(xx, qnorm(0.5, mean = mu, sd = standardDeviation), col = "red")
```



From the estimations produced through our model we can see that the current model with a 95% prediction fits the data quite nicely, having only 4 of the 100 observations shortly out of the 95% prediction interval.

However it should be noted that even though the model has a good performance, a normal distribution has the assumption that data can take any value in the real line however this data is bounded between the [0,1] interval.

Considering that the variance increases trough the model it further indicates that a normal distribution should be switched for another distribution that better respects the nature of our data.

Question 2

Model 2:

$Yi \sim Pois(\lambda i)$

$$log(\lambda i) = \beta 0 + \beta 1 xi$$

Model 1:

$Yi \sim N(\mu i, \sigma^2)$

$$log(\mu i) = \gamma 0 + \gamma 1 xi$$

Question 2 a)

As we can from the graph and what we can determine from the nature of the data represented in such graph the recorded number of AIDS cases is a count variable and the counts are non-negative integers.

The first model, a Poisson distribution, would be a more appropriate choice. The Poisson distribution is a discrete distribution that models count data which respects the nature of the data

The second model, a Normal distribution, would not be the best fit since its range is from]-∞,∞[and expects continuous values, not respecting the nature of the data.

The log-link function in both models ensures that the predicted values are always positive. This behaviour is standard for a poisson distribution but is inadequate for a normal distribution.

Question 2 b)

The Yi are independent so the likelihood is a product of the individual pdfs.

$$L(\theta 1, theta2, \sigma^2; y, x)$$

```
# Fitting in R

model2 = glm(cases ~ date, data = aids, family = poisson(link = "log"))

# Summarise the model
summary(model2)


Deviance Residuals:
   Min      1Q  Median      3Q     Max
```

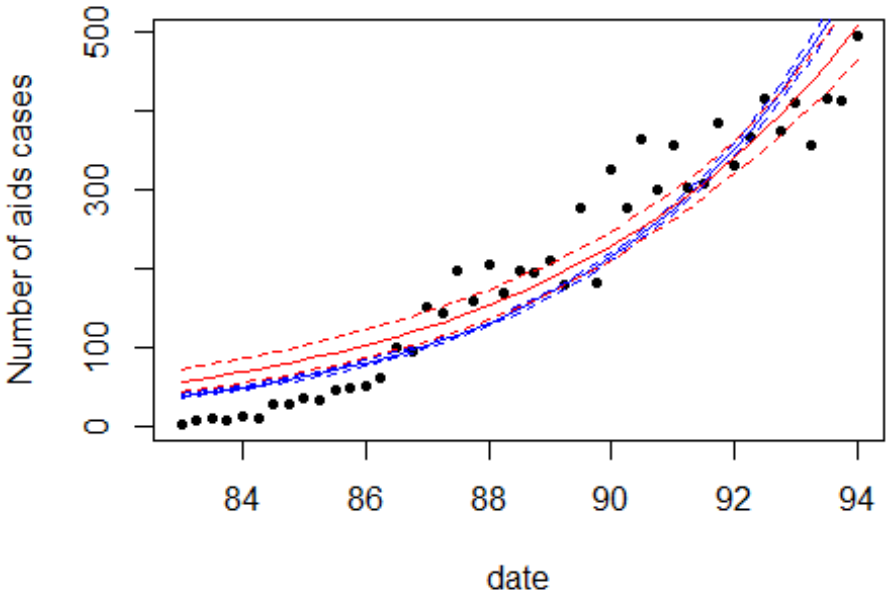
```
-7.768  -4.042  -0.335   3.048   7.281

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -16.875879   0.350353  -48.17   <2e-16 ***
date          0.247169   0.003856   64.10   <2e-16 ***

    Null deviance: 5738.16  on 44  degrees of freedom
Residual deviance:  854.02  on 43  degrees of freedom
AIC: 1153.9
```

```
# Fitting model 1
model1 = glm(cases ~ date, data = aids, family = gaussian(link = "log"))

# Summarise the model
summary(model1)
```

```

Call:
glm(formula = cases ~ date, family = gaussian(link = "log"),
    data = aids)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-79.54  -50.35  -12.50   24.94  112.83

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -12.6047     1.2663  -9.954 9.94e-13 ***
date          0.2004     0.0138  14.523  < 2e-16 ***

(Dispersion parameter for gaussian family taken to be 2448.904)

    Null deviance: 974004  on 44  degrees of freedom
Residual deviance: 105306  on 43  degrees of freedom
AIC: 482.81
```

```
## Sorry Matthew, but it seems I have switched the model numbers and I
## only noticed after finished the exercise



## We can use this to obtain 95% confidence intervals on our estimated
## relationship
xx <- seq(83, 94, len = 45)

## predict Poisson
predPoisson = predict(model2, newdata = aids, type = "link", se.fit = T)
```

```r
## Mean and Confidence Intervals estimation
muPoisson = exp(predPoisson$fit)
muPoisson_upper = exp(predPoisson$fit + qnorm(1 - 1.96/2) *
predPoisson$se.fit)  ##exp(p$fit+qnorm(1-a/2)*p$se.fit)
muPoisson_lower = exp(predPoisson$fit - qnorm(1 - 1.96/2) *
predPoisson$se.fit)


## predict normal
predNormal = predict(model1, newdata = aids, type = "link", se.fit = T)

## Mean and Confidence Intervals estimation
muNormal = exp(predNormal$fit)
muNormal_upper = exp(predNormal$fit + qnorm(1 - 1.96/2) *
predNormal$se.fit)  ## p$fit+qnorm(1-a/2)*p$se.fit
muNormal_lower = exp(predNormal$fit - qnorm(1 - 1.96/2) *
predNormal$se.fit)  ## p$fit-qnorm(1-a/2)*p$se.fit

## plot the data

plot(aids$date, aids$cases, pch = 20, xlab = "date", ylab = "Number of
aids cases")

lines(xx, muPoisson, col = "blue")
lines(xx, muPoisson_upper, col = "blue", lty = "dashed")
lines(xx, muPoisson_lower, col = "blue", lty = "dashed")

lines(xx, muNormal, col = "red")
lines(xx, muNormal_upper, col = "red", lty = "dashed")
lines(xx, muNormal_lower, col = "red", lty = "dashed")
```




```
## even if the Confidence Intervals are not exactly clear, it is still
## correct to make the exp of the mean and CI When we are using a
## log-link function
```

As we can see from the plot alone, neither the gaussian model nor the poisson model fits the data very well.

The poisson model is clearly not a good model since the big majority of the model is either overestimating or underestimating the data, which is a clear indicator that this model is inadequate.

The Normal distribution, seems to fit the data almost as equally poorly, with too much overpredicitons or underpredictions, making it also a not very adequate fit to the data, even if the distribution was well choosen for this type of data.

Question 2 c)

In this question we will be using the AIC

```
# Model comparison aka AIC time
# the formula to for the AIC: -2l + 2p

AIC(model1) # Gaussian
```

```
[1] 482.8128
```

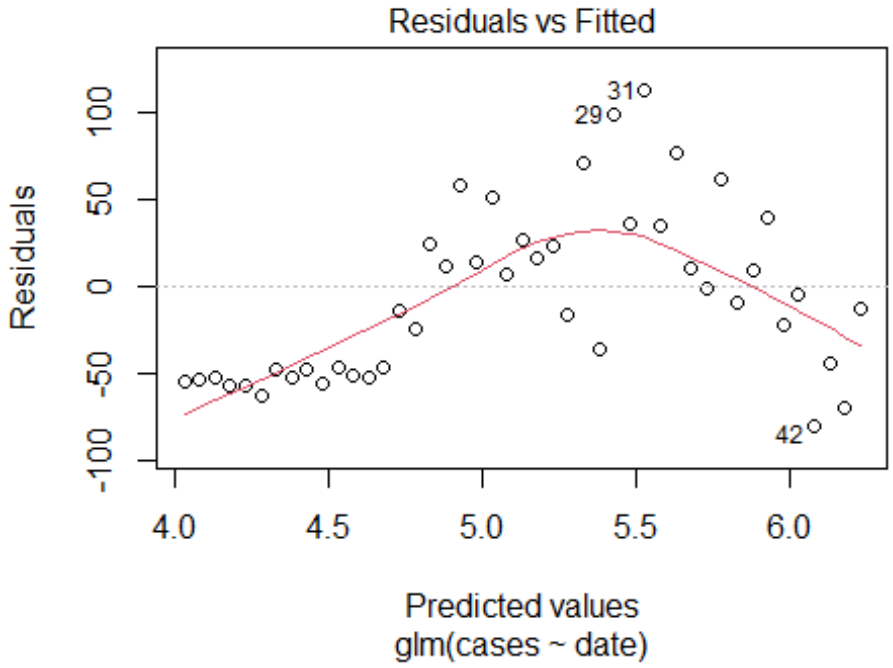
```
AIC(model2) #Poisson
```

```
[1] 1153.873
```

As we can see from the Akaike Information Criterion, model 1 (Gaussian) as a much lower AIC, meaning model 1 has a much better fit to the data than model 2 (Poisson) since when comparing AICs, the model with the lower values has the better fit to the data.

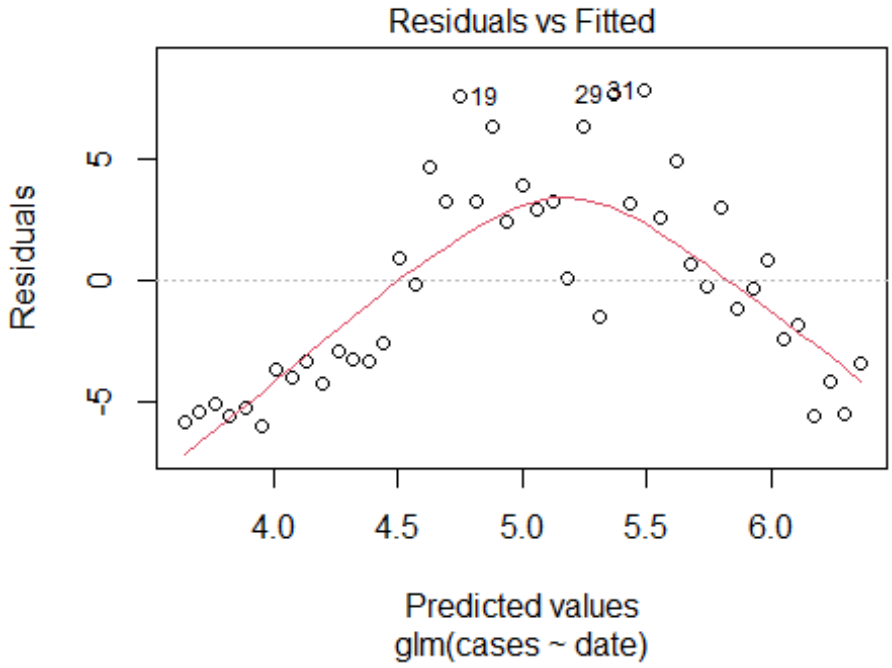
Question 2 d)

```
plot(model1, 1)
```



As we can see from the Residual vs fitted model, there seems to be quadratic pattern in how the model overfits and underfits the data in a way the would require more flexibility from a quadratic term. As such I propose we extend the module by adding a quadratic term of the data variable.

```
plot(model2, 1)
```



The Poisson model residuals seem to follow a quadratic function with some slight curves along the quadric function, this mean it will required at least quadratic or higher polynomial term to add the necessery flexibility to better fit the data.

As such I propose we extend the module by adding a quadratic and a cubic term of the data variable.

Question 2 e)

As we commented on the previous exercise we will be adding the quadratic and cubic term values directly to the aids so we can use them in our model

```
aids$dataSq = aids$date^2

aids$dataCubic = aids$date^3

# Fitting model 2 improvement
model2Improved = glm(cases ~ date + dataSq + dataCubic, data = aids,
family = poisson(link = "log"))

summary(model2Improved)
Call:
glm(formula = cases ~ date + dataSq + dataCubic, family = poisson(link =
"log"),
    data = aids)

Deviance Residuals:
```

```
    Min       1Q   Median       3Q      Max
-5.5861  -1.2939  -0.3798   1.1213   4.1190

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.151e+03  4.358e+02  -9.525   <2e-16 ***
date         1.356e+02  1.460e+01   9.290   <2e-16 ***
dataSq      -1.477e+00  1.630e-01  -9.059   <2e-16 ***
dataCubic    5.362e-03  6.062e-04   8.845   <2e-16 ***

Null deviance: 5738.16  on 44  degrees of freedom
Residual deviance:  166.68  on 41  degrees of freedom
AIC: 470.53


# Fitting model 1 improvement
model1Improved = glm(cases ~ date + dataSq, data = aids, family =
gaussian(link = "log"))

summary(model1Improved)
```

Call:

glm(formula = cases ~ date + dataSq, family = gaussian(link = "log"),

  data = aids)

Deviance Residuals:

  Min 1Q Median 3Q Max

-73.95 -21.04 -10.03 13.01 75.17

Coefficients:

        Estimate Std. Error t value Pr(>|t|)

(Intercept) -2.222e+02 3.560e+01 -6.241 1.79e-07 ***

date 4.847e+00 7.848e-01 6.176 2.21e-07 ***

dataSq -2.574e-02 4.324e-03 -5.952 4.65e-07 ***

  Null deviance: 974004 on 44 degrees of freedom

Residual deviance: 46327 on 42 degrees of freedom

AIC: 447.86

```
## compare the full and reduced models using ANOVA (this order)
anova(model2Improved, model2, test = "Chisq")

Analysis of Deviance Table

Model 1: cases ~ date + dataSq + dataCubic
Model 2: cases ~ date
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        41     166.68
2        43     854.02 -2  -687.34 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the p-value is less than 0.05 from the Chi-Squared and therefore we can reject the null hypothesis and conclude that the model with the quadratic and cubic terms is better than the linear Poisson model statistically so we should choose the improved model.

```
## compare the full and reduced models using ANOVA (this order)
anova(model1Improved, model1, test = "Chisq")

Analysis of Deviance Table

Model 1: cases ~ date + dataSq
Model 2: cases ~ date
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        42      46327
2        43     105306 -1   -58979 2.627e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can once again see that the p-value is less than 0.05 from the Chi-Squared and therefore we can reject the null hypothesis and conclude that the model with the quadratic term is better than the linear Gaussian model statistically so we should choose the improved model.

Question 2 f)

Firstly I will talk about each model flaws.

Starting from the improved Gaussian model, its main flaw is how a normal distribution simply does not respect the nature of this data, that is count data with only values on the real line for the reasons already mentioned on question 2 a).

The improved Poisson model although respects the nature of the nature, the initial model was underestimating and overestimating too much of the data and required a clear flexibility improvement by adding quadratic and cubic terms, however such performance improvement come at the cost of interpretability as it is not very clear what quadratic and cubic time real represent in this new model.

Now that we have established that both models have their flaws I will begin by comparing both models.

We can extract the deviance from the model and calculate a p-value to check whether the model fits the data using an LRT:

```
# Deviance goodness of fit of model 1 looks OK:
1 - pchisq(model1Improved$deviance, model1Improved$df.residual)

[1] 0

# Deviance goodness of fit of model 1 looks OK:
1 - pchisq(model2Improved$deviance, model2Improved$df.residual)

[1] 0
```

The p-value for both models are smaller than 0.05, so it means that both models although already improved still aren't a good fit for the data.

Comparing both models using AIC we can see that:

```
AIC(model1Improved)

[1] 447.8615

AIC(model2Improved)

[1] 470.5294
```

Both models have a relatively close AIC result.

Leaving us to choose neither of the models as good fit for the data, although the improved Poisson would still be preferred to the improved Gaussian since it is the only one of the improved models respecting the nature of the data and the AIC difference between these two improved models is no longer as big as the initial models difference.

Question 2 g)

```
# Fit the model
modelNegativeBinom <- glm.nb(cases ~ (date + dataSq + dataCubic), data =
aids)

# Model summary
summary(modelNegativeBinom)


Call:
glm.nb(formula = cases ~ (date + dataSq + dataCubic), data = aids,
    init.theta = 64.08286877, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.5965  -0.6448  -0.1348   0.6123   2.1889

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.203e+03  7.696e+02  -5.462 4.71e-08 ***
date         1.374e+02  2.594e+01   5.296 1.18e-07 ***
dataSq      -1.496e+00  2.913e-01  -5.135 2.81e-07 ***
dataCubic    5.432e-03  1.090e-03   4.986 6.17e-07 ***


Null deviance: 1812.085  on 44  degrees of freedom
Residual deviance:   45.007  on 41  degrees of freedom
AIC: 405.92

              Theta:  64.1
          Std. Err.:  20.4

 2 x log-likelihood:  -395.915
```

Now that we have extended the improved Poisson into a negative Binomial, we will compare this new model against the previously improved Poisson model and the improved Gaussian model.

```
# Deviance goodness of fit of the negative model looks OK:
1 - pchisq(modelNegativeBinom$deviance, modelNegativeBinom$df.residual)

[1] 0.3078234

# Deviance goodness of fit of improved model 2 looks OK:
1 - pchisq(model2Improved$deviance, model2Improved$df.residual)

[1] 0

# Deviance goodness of fit of improved model 1 looks OK:
1 - pchisq(model1Improved$deviance, model1Improved$df.residual)

[1] 0
```

As we can see, the negative binomial model is the only model with a p-value is larger than 0.05, meaning that only negative binomial is considered a good fit to the data.

Lastly from the Akaike Information Criterion test we can see that the Negative Binomial has the best goodness of fit per penalised complexity meaning it simply performs better than the other 2 models.

```
AIC(modelNegativeBinom)

[1] 405.9155

AIC(model1Improved)

[1] 447.8615
```

```
AIC(model2Improved)
```

```
[1] 470.5294
```

To conclude the negative binomial model not only is preferable to both the improved poisson and improved gaussian model, it is also a good fit for the data.