CHL 5209 Assignment 1

Belina Jang

```
# load the library
library(tidyverse)
library(rjags)
```

Question 1

(a) Specify the regression model corresponding to the above output. How do you interpret the parameters in the model? What does this model assume about HIV incidence over time?

$\log \lambda_z = \alpha + \beta z$

λ_z : the incidence rate of HIV infection in group z

z : indicator for intervention ($z=0$: control, $z=1$: intervention)

e^{α} : the incidence rate of HIV infection in the control group $= e^{-4.3225}$

$e^{\alpha+\beta}$: the incidence rate of HIV infection in the intervention group $= e^{-4.3225-0.7039}$

$\frac{e^{\alpha+\beta}}{e^{\alpha}} = e^{\beta}$: incidence rate ratio of HIV infection in intervention / control. $= e^{-0.7039} \approx 0.495$

$\Rightarrow$ the intervention (male circumcision) reduces incidence rate of HIV infection by approximately 50.5 %.

$\therefore$ The regression model used to create the output is a log-linear model. The model assumed that the HIV incidence follows a poisson distribution and that the log incidence rate of the HIV infection is constant over time.

(b) Based on the fitted model, calculate

i. The two-year risks of HIV infection in the intervention and control arms.

```
alpha <- -4.3225 # intercept
beta <- -0.7039 # z
se_alpha <- 0.1491
se_beta <- 0.2601

# incidence rates
ir_control <- exp(alpha)
ir_intervention <- exp(alpha + beta)

# approximate two-year risks = 1 - 2yrs survival probability
r_control <- 1 - exp(-ir_control * 2)
r_intervention <- 1 - exp(-ir_intervention * 2)

cat("Control arm: incidence rate =", ir_control, ", two-year risk =", r_control, ".\n")
```

```
Control arm: incidence rate = 0.01326668 , two-year risk = 0.02618443 .
```

```
cat("Intervention arm: incidence rate =", ir_intervention,
    ", two-year risk =", r_intervention, ".\n")
```

```
Intervention arm: incidence rate = 0.006562393 , two-year risk = 0.01303903 .
```

ii. A 95% confidence interval for the HIV incidence rate ratio between the intervention and control arms.

```
cil <- exp(beta - 1.96 * se_beta)
ciu <- exp(beta + 1.96 * se_beta)
cat("95% CI for HIV incidence rate ratio: (", cil, ",", ciu, ").")
```

```
95% CI for HIV incidence rate ratio: ( 0.2970972 , 0.8235722 ).
```

(c) Derive the maximum likelihood estimators (MLEs) for the model parameters and verify the model-based point estimates. For checking the maximum likelihood point, recall that a $P \times P$ symmetric matrix A is negative definite if and only if the quadratic form $\mathbf{x}^T A\mathbf{x} < 0$ for all $\mathbf{x} = (x1, ..., xp) \in \mathbf{R}^p$.

c)

$$\frac{e^{-\mu_0}\mu_0^{D_0}}{D_0!} \cdot \frac{e^{-\mu_1}\mu_1^{D_1}}{D_1!} \quad \mu = \lambda \cdot Y \Rightarrow \quad \frac{e^{-\lambda_0 Y_0}(\lambda_0 Y_0)^{D_0}}{D_0!} \cdot \frac{e^{-\lambda_1 Y_1}(\lambda_1 Y_1)^{D_1}}{D_1!}$$

$$\lambda_0 = e^{\alpha} \Rightarrow \lambda_1 = e^{\alpha+\beta} \quad \frac{e^{-(e^{\alpha} Y_0)}(e^{\alpha} Y_0)^{D_0}}{D_0!} \cdot \frac{e^{-(e^{\alpha+\beta} Y_1)} \cdot (e^{\alpha+\beta} Y_1)^{D_1}}{D_1!}$$

taking log $\Rightarrow -e^{\alpha} Y_0 + D_0(\alpha + \log Y_0) - (e^{\alpha+\beta} Y_1) + D_1(\alpha+\beta) + \log Y_1$ (C, C)

it gives you, $l(\alpha, \beta) = D_0\alpha - e^{\alpha} Y_0 + D_1(\alpha + \beta) - e^{\alpha+\beta} Y_1.$

$$S_1(\alpha, \beta) = \frac{\partial l(\alpha,\beta)}{\partial \beta} = D_1 - e^{\alpha+\beta} Y_1 = 0$$

$$D_1 = e^{\alpha+\beta} Y_1$$

$$\frac{D_1}{Y_1} = e^{\alpha+\beta} \Rightarrow \log\left(\frac{D_1}{Y_1}\right) = \alpha + \beta$$

$$\hat{\beta}(\alpha) = \log\left(\frac{D_1}{Y_1}\right) - \alpha$$

$$S_2(\alpha, \beta) = \frac{\partial l(\alpha,\beta)}{\partial \alpha} = D_0 - e^{\alpha} Y_0 + D_1 - e^{\alpha+\beta} Y_1$$

$$S_2(\alpha, \hat{\beta}(\alpha)) = D_0 - e^{\alpha} Y_0 + D_1 - e^{\alpha + (\log \frac{D_1}{Y_1}) - \alpha} Y_1$$

$$= D_0 + D_1 - e^{\alpha} Y_0 - e^{\log \frac{D_1}{Y_1}} Y_1$$

$$= D_0 + D_1 - e^{\alpha} Y_0 - \frac{D_1}{Y_1} Y_1$$

$$= D_0 - e^{\alpha} Y_0 = 0$$

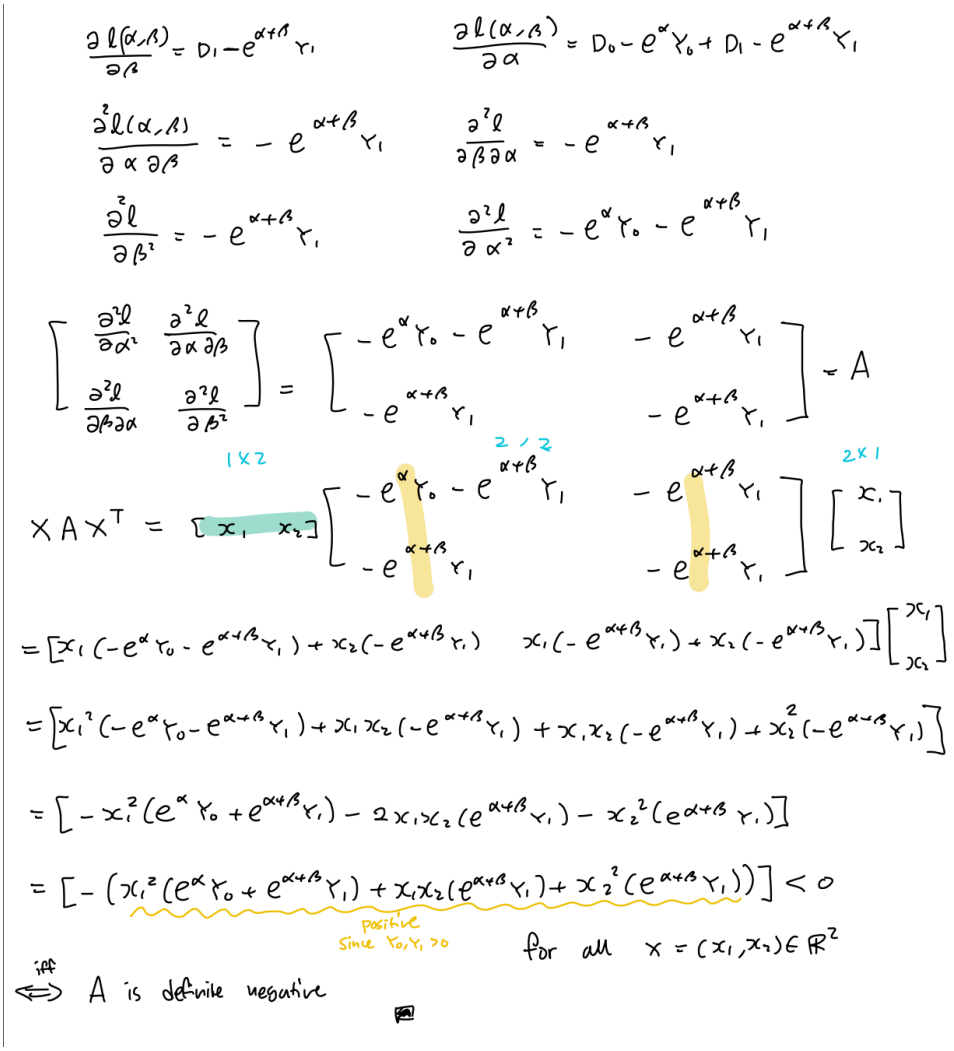
$$D_0 = e^{\alpha} Y_0$$

$$\frac{D_0}{Y_0} = e^{\alpha}$$

$$\hat{\alpha} = \log\left(\frac{D_0}{Y_0}\right)$$

$$\hat{\beta} = \hat{\beta}(\hat{\alpha}) = \log\left(\frac{D_1}{Y_1}\right) - \hat{\alpha} = \log\left(\frac{D_1}{Y_1}\right) - \log\left(\frac{D_0}{Y_0}\right) = \log\left(\frac{D_1}{Y_1} \times \frac{Y_0}{D_0}\right)$$

$$= \log\left(\frac{D_1 Y_0}{Y_1 D_0}\right)$$



```r
q1data <- data.frame(
  n = c(2430, 2387), #N0, N1
  d = c(45, 22), #D0, D1
  z = c(0, 1), #Z0, Z1
  y = c(3391.8, 3352.4) #Y0, Y1
)

#model <- glm(d ~ z + offset(log(y)), family = poisson(link = "log"), data = q1data)
model <- glm(d ~ z, offset = log(y), family = poisson(link = "log"), data = q1data)

summary(model)
```

```
Call:
glm(formula = d ~ z, family = poisson(link = "log"), data = q1data,
    offset = log(y))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.3225     0.1491 -28.996  < 2e-16 ***
z            -0.7039     0.2601  -2.706  0.00681 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance:  7.7920e+00  on 1  degrees of freedom
Residual deviance: -4.4409e-15  on 0  degrees of freedom
AIC: 14.585

Number of Fisher Scoring iterations: 3
```

```r
logLik(model)
```

```
'log Lik.' -5.292369 (df=2)
```

```r
n = c(2430, 2387) #N0, N1
d = c(45, 22) #D0, D1
z = c(0, 1) #Z0, Z1
y = c(3391.8, 3352.4) #Y0, Y1

# alpha_hat = log(D0/Y0)
alpha_hat <- log(q1data$d[1] / q1data$y[1])
# beta_hat = log((D1*Y0) / (Y1*D0))
beta_hat <- log((q1data$d[2]*q1data$y[1] ) / (q1data$y[2]*q1data$d[1]))

cat("MLE for alpha:", alpha_hat, "\n")
```

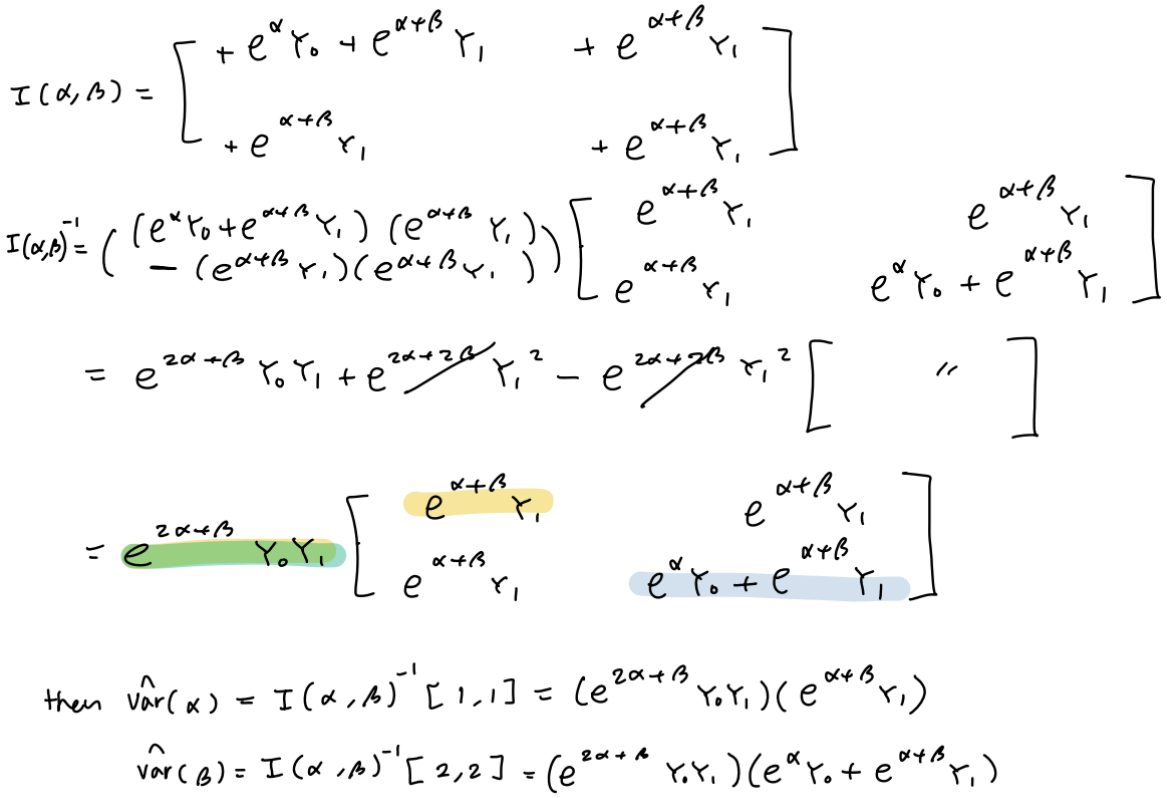
```
MLE for alpha: -4.322454
```

```r
cat("MLE for beta:", beta_hat, "\n")
```

```
MLE for beta: -0.7039358
```

$\therefore$ Manual calculation gives the same MLEs as the glm output.

(d) Derive variance estimators for the MLEs by inverting the observed information matrix and verify the model-based standard error estimates. Verify also the Wald test p-values shown in the model output.



```
var_alpha_hat <- (exp(alpha_hat+beta_hat)*y[2]) / (exp(2*alpha_hat+beta_hat)*y[1]*y[2])
var_beta_hat <- (exp(alpha_hat)*y[1] + exp(alpha_hat+beta_hat)*y[2]) /
  (exp(2*alpha_hat+beta_hat)*y[1]*y[2])

se_alpha_hat <- sqrt(var_alpha_hat)
se_beta_hat <- sqrt(var_beta_hat)

cat("Standard errors of maximum likelihood estimators: \n")
```

```
Standard errors of maximum likelihood estimators:
```

```r
cat("se(alpha_hat) =", se_alpha_hat, "\n")
```

```
se(alpha_hat) = 0.1490712
```

```r
cat("se(beta_hat) =", se_beta_hat, "\n")
```

```
se(beta_hat) = 0.2601476
```

∴ It produces the same standard errors as the glm output.

```r
# Verify Wald test p-values
wald_alpha <- alpha_hat / se_alpha_hat
p_alpha <- 2 * (1 - pnorm(abs(wald_alpha)))
cat("Wald test statistic for alpha:", wald_alpha, "\n")
```

```
Wald test statistic for alpha: -28.9959
```

```r
cat("p-value for alpha:", p_alpha, "\n")
```

```
p-value for alpha: 0
```

```r
wald_beta <- beta_hat / se_beta_hat
p_beta <- 2 * (1 - pnorm(abs(wald_beta)))

cat("Wald test statistic for beta:", wald_beta, "\n")
```

```
Wald test statistic for beta: -2.705909
```

```r
cat("p-value for beta:", p_beta, "\n")
```

```
p-value for beta: 0.006811764
```

∴ It produces the same p-values as the glm output.

Question 2

(a) Find the log-likelihood function for the parameters in the model fitted in 1st lecture slides, slide 31, and calculate its value at the maximum likelihood point. Here and in (b) we take the glm estimates to be the ML estimates.

a) $\log(\lambda_{zx}) = \alpha + \beta z + \nu_1 \cdot 1_{\{x=1\}} + \nu_2 \cdot 1_{\{x=2\}}$

$$L(\alpha, \beta, \nu_1, \nu_2) = \prod_{z,x} \frac{\mu_{zx}^{d_{zx}} e^{-\mu_{zx}}}{d_{zx}!}$$

$$= \prod_{z,x} \frac{(\lambda_{zx} y_{zx})^{d_{zx}} e^{-\lambda_{zx} y_{zx}}}{d_{zx}!} = \prod_{z,x} \frac{(\exp(\log \lambda_{zx}) \cdot y_{zx})^{d_{zx}} e^{-\exp(\log \lambda_{zx}) \cdot y_{zx}}}{d_{zx}!}$$

$$\ell(\alpha, \beta, \nu_1, \nu_2) = \sum \left[d_{zx} (\log \lambda_{zx} + \log y_{zx}) - y_{zx} \cdot \exp(\log \lambda_{zx}) - \log(d_{zx}!) \right]$$

$$= \sum \left[d_{zx} (\log \lambda_{zx} + \log y_{zx}) - y_{zx} \lambda_{zx} - \log(d_{zx}!) \right]$$

$$= \sum \left[d_{zx} (\alpha + \beta z + \nu_1 \cdot 1_{\{x=1\}} + \nu_2 \cdot 1_{\{x=2\}} + \log y_{zx}) - (\alpha + \beta z + \nu_1 \cdot 1_{\{x=1\}} + \nu_2 \cdot 1_{\{x=2\}}) y_{zx} - \log(d_{zx}!) \right]$$

where $\alpha = -5.4177$, $\beta = 0.8697$, $\nu_1 = 0.1290$, $\nu_2 = 0.6920$

value @ maximum likelihood point: $\ell(\hat{\alpha}, \hat{\beta}, \hat{\nu}_1, \hat{\nu}_2) = -11.89823$ (df = 4)

```
d <- c(4, 5, 8, 2, 12, 14)
y <- c(607.9, 1272.1, 888.9, 311.9, 878.1, 667.5)
z <- c(0, 0, 0, 1, 1, 1)
x <- c(0, 1, 2, 0, 1, 2)

# manual calculation using MLEs from glm output
alpha_hat <- -5.4177
beta_hat <- 0.8697
gamma1_hat <- 0.1290
gamma2_hat <- 0.6920

data <- data.frame(
  d = d,
  y = y,
  z = z,
  x = x
```

```
)

llh <- 0
# for loop for each row(unique combination of x and z)
for (i in 1:nrow(data)) {
  di <- data$d[i]
  yi <- data$y[i]
  zi <- data$z[i]
  xi <- data$x[i]

  lambda_zx <- exp(alpha_hat + beta_hat * zi + gamma1_hat * (xi == 1) +
                     gamma2_hat * (xi == 2))
  llh <- llh + di * (log(lambda_zx) + log(yi)) - yi * lambda_zx - log(factorial(di))
}

cat("The Value of the log-likelihood function at the maximum likelihood point:",llh)
```

```
The Value of the log-likelihood function at the maximum likelihood point: -11.89823
```

```
# double check
model <- glm(d ~ z + as.factor(x) + offset(log(y)), family = poisson(link = "log"))
summary(model)
```

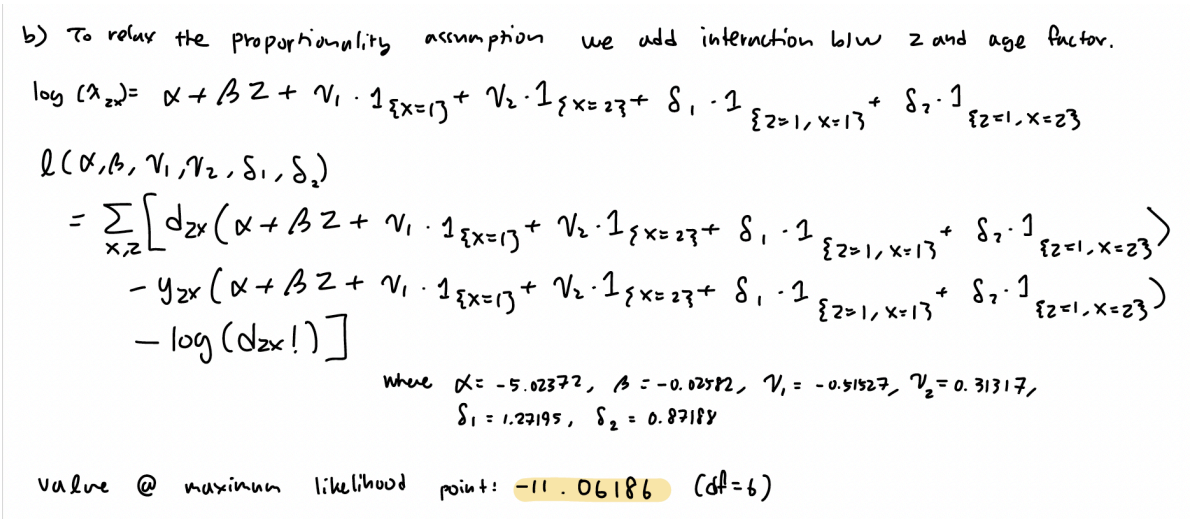
```
Call:
glm(formula = d ~ z + as.factor(x) + offset(log(y)), family = poisson(link = "log"))

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.4177     0.4421 -12.256  < 2e-16 ***
z              0.8697     0.3080   2.823  0.00476 **
as.factor(x)1  0.1290     0.4754   0.271  0.78609
as.factor(x)2  0.6920     0.4614   1.500  0.13366
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 14.5780  on 5  degrees of freedom
Residual deviance:  1.6727  on 2  degrees of freedom
AIC: 31.796
```

```
Number of Fisher Scoring iterations: 4
```

```
logLik(model)
```

```
'log Lik.' -11.89823 (df=4)
```

(b) The model in (a) assumed that the exposure effect is constant across the age groups (the proportionality assumption). Write the log-likelihood function for the parameters in a model that relaxes this assumption. Fit this model using glm and calculate the value of the log-likelihood function at the maximum likelihood point.




```
# fit the glm nodel
model_interaction <- glm(d ~ z * as.factor(x) + offset(log(y)),
                         family = poisson(link = "log"))

summary(model_interaction)
```

```
Call:
glm(formula = d ~ z * as.factor(x) + offset(log(y)), family = poisson(link = "log"))

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)     -5.02372    0.50000 -10.047   <2e-16 ***
z               -0.02582    0.86603  -0.030    0.976
as.factor(x)1   -0.51527    0.67082  -0.768    0.442
as.factor(x)2    0.31317    0.61237   0.511    0.609
z:as.factor(x)1  1.27195    1.01653   1.251    0.211
z:as.factor(x)2  0.87188    0.97285   0.896    0.370
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1.4578e+01  on 5  degrees of freedom
Residual deviance: 4.4409e-16  on 0  degrees of freedom
AIC: 34.124

Number of Fisher Scoring iterations: 3
```

```r
logLik(model_interaction)
```

```
'log Lik.' -11.06186 (df=6)
```

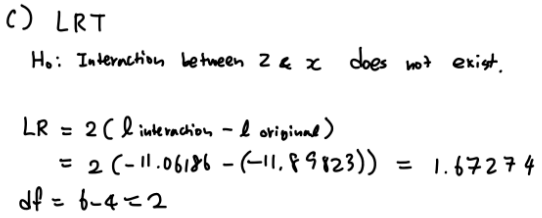
```r
# manual calculation using MLEs from glm output
alpha_hat <- -5.02372
beta_hat <- -0.02582
gamma1_hat <- -0.51527
gamma2_hat <- 0.31317
delta1_hat <- 1.27195
delta2_hat <- 0.87188

data <- data.frame(
  d <- c(4, 5, 8, 2, 12, 14),
  y <- c(607.9, 1272.1, 888.9, 311.9, 878.1, 667.5),
  z <- c(0, 0, 0, 1, 1, 1),
  x <- c(0, 1, 2, 0, 1, 2)
)

llh <- 0
# for loop for each row(unique combination of x and z)
for (i in 1:nrow(data)) {
  di <- data$d[i]
  yi <- data$y[i]
  zi <- data$z[i]
```

```
  xi <- data$x[i]

  lambda_zx <- exp(alpha_hat + beta_hat * zi + gamma1_hat * (xi == 1) +
                     gamma2_hat * (xi == 2) + delta1_hat * zi * (xi == 1) +
                     delta2_hat * zi * (xi == 2))
  llh <- llh + di * (log(lambda_zx) + log(yi)) - yi * lambda_zx - log(factorial(di))
}

llh
```

```
[1] -11.06186
```

∴ Using manual calculation, we get the value of the log-likelihood function at the maximum likelihood point: -11.06186.

(c) Use the likelihood ratio test to test the assumption mentioned in (b).




```
D <- 2 * (logLik(model_interaction) - logLik(model))
p_value <- pchisq(D, df = 2, lower.tail = FALSE)

cat("LRT Statistic (D):", D, "\n")
```

```
LRT Statistic (D): 1.672747
```

```
cat("p-value:", p_value, "\n")
```

```
p-value: 0.4332791
```

∴ We fail to reject the null hypothesis that there's no interaction between **z** and **x** (p-value>0.05).

(d) How is the residual deviance reported in the glm output related to the quantities calculated in (a) and (b)?

```
r = residuals(model, type = "deviance")
r
```

```
          1           2           3           4           5           6
 0.73940382 -0.58410194  0.04254837 -0.77384901  0.42799819 -0.03191168
```

```
cat("Residual Deviance:", sum(r^2), "\n")
```

```
Residual Deviance: 1.672747
```

∴ The residual deviances reported in the glm output are for individual observations, which are 0.73940, −0.58410, 0.04255, −0.77385, 0.42800 and −0.03191. The sum of the squared residuals is the same as the likelihood ratio statistic calculated in (c), which is twice the difference of the log-likelihoods of the two models from parts (a) and (b).

Question 3

(a) Specify a model for the mortality rate, including an intercept term, age group effects, and marital status effect. Assume that the marital status effect is proportional over age.

a) Since the data is a count data so it's reasonable to assume it closely follows a poisson distribution.
then $d_{MA} \sim \text{Poisson}(\lambda_{MA} y_{MA})$ where d_{MA} is the # of death, λ_{MA} is the mortality rate and y_{MA} is the person year in age group A with marriage status M.

Since marital status is proportional over age, there's no interaction term between marital status and age in the model. Also assume the rate of change in mortality is constant over all age gmp.

model: $\log(\lambda_{MA}) = \beta_0 + \beta_1 M + \sum_k \nu_k \cdot 1_{\{A=k\}}$

β_0: intercept
M: marriage status (0 = single, 1 = married)
β_1: marriage status effect
A: age group (A = 22, 23, ... 29)
ν_k: effect of age group $A = k$

(b) Show how the regression coefficient parameter for marital status can be interpreted in terms of a rate ratio.

$\therefore$ The regression coefficient parameter for marital status can be interpreted as the log rate ratio of the mortality rate between married and single individuals. The rate ratio is the $e^{log\ rate\ ratio}$.

(c) Enter the data, fit the model and interpret the results.

```r
# enter data
q3data <- data.frame(
  age = rep(c(22, 23, 24, 25, 26, 27, 28, 29), each=2),
  marital_status = rep(c(0, 1), times = 8),  # 0 = single, 1 = married
  deaths = c(433, 24, 412, 36, 373, 66, 331, 102,
             287, 138, 242, 171, 215, 185, 192, 200),
  person_years = c(91444, 8556, 86835, 12708, 75892, 23203, 63241, 35415, 52023,
                   46207, 42123, 55675, 36915, 60470, 32215, 64770)
)

# fit the model
model <- glm(deaths ~ factor(marital_status) + as.factor(age),
             offset = log(person_years),
             family = poisson(link = "log"),
             data = q3data)

summary(model)
```

```
Call:
glm(formula = deaths ~ factor(marital_status) + as.factor(age),
    family = poisson(link = "log"), data = q3data, offset = log(person_years))

Coefficients:
                       Estimate Std. Error  z value Pr(>|z|)
(Intercept)            -5.34833    0.04682 -114.228  < 2e-16 ***
factor(marital_status)1 -0.61115    0.04175  -14.639  < 2e-16 ***
as.factor(age)23         0.00493    0.06649    0.074  0.94089
as.factor(age)24         0.04224    0.06694    0.631  0.52802
as.factor(age)25         0.09898    0.06751    1.466  0.14259
as.factor(age)26         0.14755    0.06837    2.158  0.03092 *
```

```
as.factor(age)27          0.18266    0.06962    2.624  0.00870 **
as.factor(age)28          0.18736    0.07070    2.650  0.00805 **
as.factor(age)29          0.20167    0.07162    2.816  0.00487 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 232.550  on 15  degrees of freedom
Residual deviance:   1.047  on  7  degrees of freedom
AIC: 130.23

Number of Fisher Scoring iterations: 3
```

∴ Interpretations:

alpha = -5.34833: baseline death rate for the reference group (marital_status="single" and age=22) is e^alpha which is 0.004756087.

beta = -0.61115: the log rate ratio of the death rate of married with respect to single is beta. The rate ratio is e^beta which is 0.5427264, which means marriage status = married lowers the death rate by approximately 45.7%. The difference between the death rates of married and single individuals was statistically significant.

gamma(k) = coefficient of as.factor(age)(k): the log rate ratio of the death rate between age group k and the reference group (age=22) is gamma(k). The rate ratio is e^gamma(k). The difference between the death rates of age group k and the reference group (age=22) was statistically significant when k is at least 26.

(d) Calculate the expected number of events in each age/marital status category. Compare the expected numbers to observed event counts to assess the overall model fit using the chi-squared goodness of fit test.

```r
q3data$expected_deaths <- predict(model, type = "response")
q3data %>% select(age, marital_status, deaths, expected_deaths)
```

```
   age marital_status deaths expected_deaths
1   22              0    433       434.91472
2   22              1     24        22.08528
3   23              0    412       415.03520
4   23              1     36        32.96480
```

```
5   24              0    373       376.52267
6   24              1     66        62.47733
7   25              0    331       332.07338
8   25              1    102       100.92662
9   26              0    287       286.76421
10  26              1    138       138.23579
11  27              0    242       240.48842
12  27              1    171       172.51158
13  28              0    215       211.74800
14  28              1    185       188.25200
15  29              0    192       187.45340
16  29              1    200       204.54660
```

```r
# chi-squared goodness of fit test
x2 <- sum((q3data$deaths - q3data$expected_deaths)^2 / q3data$expected_deaths)
cat("Chi-square value:", x2, ".\n")
```

```
Chi-square value: 1.06335 .
```

```r
# p-value
p <- 1 - pchisq(x2, df = nrow(q3data)-1)
cat("p-value:",p)
```

```
p-value: 0.9999996
```

∴ The expected number of events (deaths) in each age/marital status category seems to be very close to the observed event counts. The chi-squared value is 1.06335 and the p-value is very big (p=0.9999996), so the difference between the observed and expected distributions is not statistically significant. Therefore, it seems like the model fits the data well in overall.

(e) Fit also a model that allows for interaction between marital status and age. What can you say about the model fit now?

```r
# fit the model
interaction_model <- glm(deaths ~ factor(marital_status) * factor(age),
                         offset = log(person_years),
                         family = poisson(link = "log"),
                         data = q3data)

summary(interaction_model)
```

```
Call:
glm(formula = deaths ~ factor(marital_status) * factor(age),
    family = poisson(link = "log"), data = q3data, offset = log(person_years))

Coefficients:
                                     Estimate Std. Error  z value Pr(>|z|)
(Intercept)                         -5.352744   0.048057 -111.383  < 2e-16
factor(marital_status)1             -0.523590   0.209705   -2.497  0.01253
factor(age)23                        0.002003   0.068823    0.029  0.97679
factor(age)24                        0.037256   0.070643    0.527  0.59793
factor(age)25                        0.100155   0.073011    1.372  0.17013
factor(age)26                        0.152785   0.076117    2.007  0.04472
factor(age)27                        0.193333   0.080260    2.409  0.01600
factor(age)28                        0.207009   0.083430    2.481  0.01309
factor(age)29                        0.230052   0.086705    2.653  0.00797
factor(marital_status)1:factor(age)23  0.007864   0.272362    0.029  0.97697
factor(marital_status)1:factor(age)24 -0.023304   0.248613   -0.094  0.92532
factor(marital_status)1:factor(age)25 -0.073738   0.238330   -0.309  0.75702
factor(marital_status)1:factor(age)26 -0.090084   0.233895   -0.385  0.70013
factor(marital_status)1:factor(age)27 -0.102622   0.232285   -0.442  0.65864
factor(marital_status)1:factor(age)28 -0.120222   0.232449   -0.517  0.60502
factor(marital_status)1:factor(age)29 -0.133998   0.232776   -0.576  0.56485

(Intercept)                         ***
factor(marital_status)1             *
factor(age)23
factor(age)24
factor(age)25
factor(age)26                       *
factor(age)27                       *
factor(age)28                       *
factor(age)29                       **
factor(marital_status)1:factor(age)23
factor(marital_status)1:factor(age)24
factor(marital_status)1:factor(age)25
factor(marital_status)1:factor(age)26
factor(marital_status)1:factor(age)27
factor(marital_status)1:factor(age)28
factor(marital_status)1:factor(age)29
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2.3255e+02  on 15  degrees of freedom
Residual deviance: 1.0703e-13  on  0  degrees of freedom
AIC: 143.19

Number of Fisher Scoring iterations: 3
```

```r
q3data$interaction_expected_deaths <- predict(interaction_model, type = "response")
q3data %>% select(age, marital_status, deaths, interaction_expected_deaths)
```

```
   age marital_status deaths interaction_expected_deaths
1   22              0    433                         433
2   22              1     24                          24
3   23              0    412                         412
4   23              1     36                          36
5   24              0    373                         373
6   24              1     66                          66
7   25              0    331                         331
8   25              1    102                         102
9   26              0    287                         287
10  26              1    138                         138
11  27              0    242                         242
12  27              1    171                         171
13  28              0    215                         215
14  28              1    185                         185
15  29              0    192                         192
16  29              1    200                         200
```

```r
# anova
anova(model, interaction_model, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: deaths ~ factor(marital_status) + as.factor(age)
Model 2: deaths ~ factor(marital_status) * factor(age)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         7      1.047
2         0      0.000  7    1.047    0.994
```

∴ By observing the table with the expected number of events in each age/marital status category after including the interaction between marital status and age, you can see that the

expected number of deaths are exactly the same as observed counts. In fact, the new model with interaction is a fully saturated model, which means it fits the data perfectly. The p-value from the anova test (comparing the two models) is 0.994, which is very close to 1, which means the model with interaction is not significantly better than the model without interaction. So it's better to use the simpler model without the interaction.

Question 4

```
# reset the environment
rm(list=ls())

# a part of cases_hosp.r below
# by Dr. Olli Saarela

# Case and hospitalization rates by vax status
# (crude rates, no age adjustment, interpret with caution)
# Datasets downloaded from https://data.ontario.ca/dataset/covid-19-vaccine-data-in-ontario

#setwd('c:/Users/ollis/Dropbox/work/CHL5209H_2025/data')

# Case counts:

cases <- read.csv('cases_by_vac_status.csv')

cases$date <- as.Date(cases$Date, "%Y-%m-%d")
cases$unvac <- ifelse(!is.na(cases$covid19_cases_notfull_vac),
                      cases$covid19_cases_notfull_vac,
                      cases$covid19_cases_unvac + cases$covid19_cases_partial_vac)
cases$vac <- ifelse(is.na(cases$covid19_cases_boost_vac),
                    cases$covid19_cases_full_vac,
                    cases$covid19_cases_full_vac + cases$covid19_cases_boost_vac)

# Hosp. counts:

hosp <- read.csv('vac_status_hosp_icu.csv')

hosp$date <- as.Date(hosp$date, "%Y-%m-%d")
hosp$vac <- hosp$hospitalnonicu_full_vac + hosp$icu_full_vac
hosp$unvac <- hosp$icu_unvac + hosp$icu_partial_vac +
              hosp$hospitalnonicu_unvac + hosp$hospitalnonicu_partial_vac
```

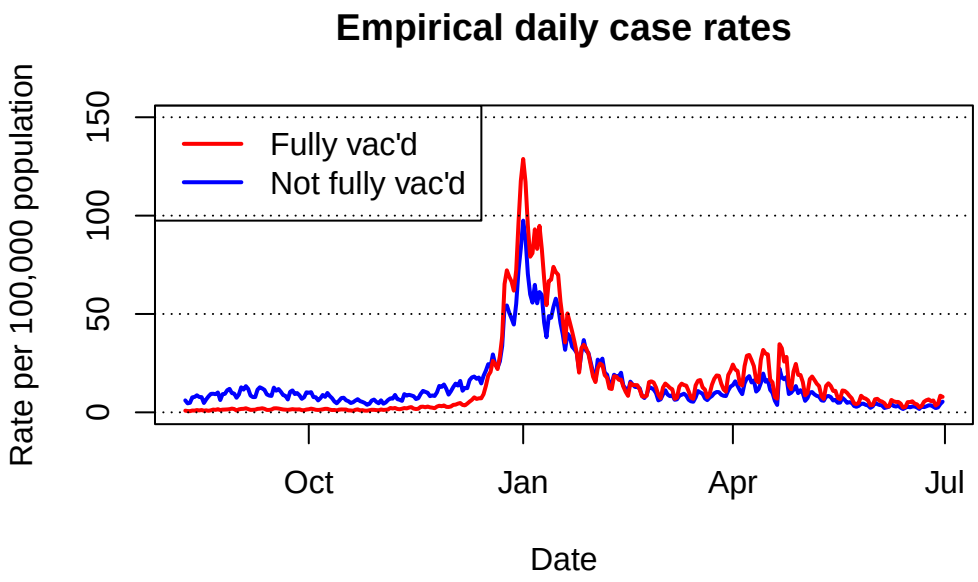
```
hosp$tot <- hosp$unvac + hosp$vac
hosp$prop <- hosp$vac/hosp$tot

# Get population denominators by vax status:

vac <- read.csv('vaccine_doses.csv')

vac$vacpop <- vac$total_individuals_fully_vaccinated
# Use Q4 Ontario population from https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=171000
vac$totalpop <- 14915270
vac$unvacpop <- vac$totalpop - vac$vacpop

vac$date <- as.Date(vac$report_date, "%Y-%m-%d")
vac <- vac[,c('date','unvacpop','vacpop','totalpop')]

intersect(names(cases), names(vac))
```

```
[1] "date"
```

```
cases <- merge(cases, vac)

intersect(names(hosp), names(vac))
```

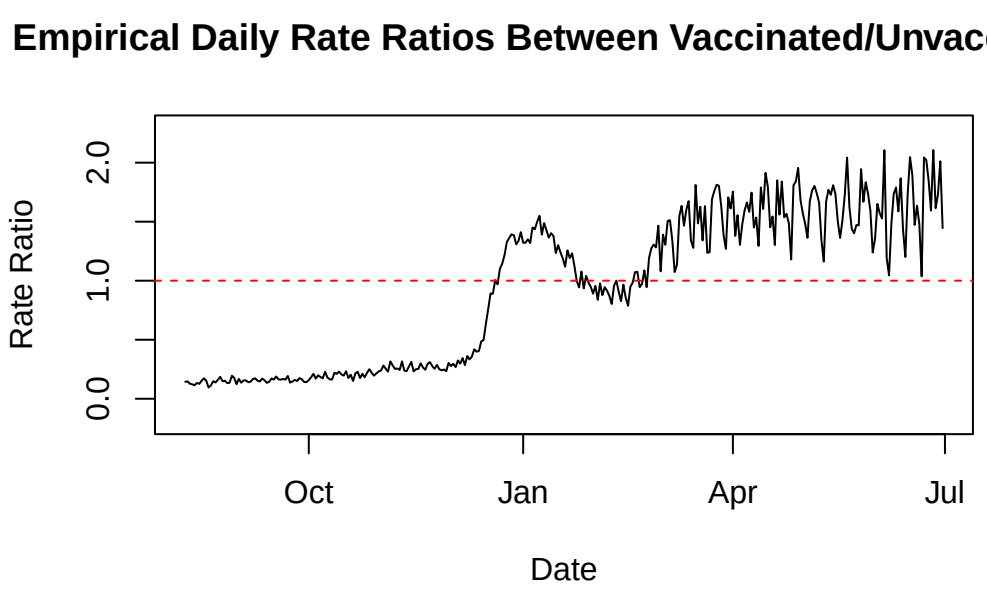
```
[1] "date"
```

```
hosp <- merge(hosp, vac)

# Case rate:

cases$vacrate <- cases$vac/cases$vacpop * 100000
cases$unvacrate <- cases$unvac/cases$unvacpop * 100000

plot(cases$date, cases$unvacrate, type='l', ylim=c(0,150), lwd=2, col='blue',
     xlab='Date', ylab='Rate per 100,000 population',
     main='Empirical daily case rates')
lines(cases$date, cases$vacrate, lwd=2, col='red')
abline(h=seq(0,150,50), lty='dotted')
legend('topleft', legend=c('Fully vac\'d','Not fully vac\'d'), lwd=2, col=c('red','blue'))
```



(a) The code plots the empirical daily case rates among vaccinated/unvaccinated (we take the latter to comprise both unvaccinated and partially vaccinated). Produce a plot of the empirical daily rate ratios between vaccinated/unvaccinated.

```
# Compute empirical daily rate ratios
cases$rate_ratio = cases$vacrate / cases$unvacrate

plot(x = cases$date, y = cases$rate_ratio, type = "l", xlab = "Date",
     ylab = "Rate Ratio", ylim=c(-0.2,2.3),
     main = "Empirical Daily Rate Ratios Between Vaccinated/Unvaccinated")
abline(h = 1, col = "red", lty = 2)
```



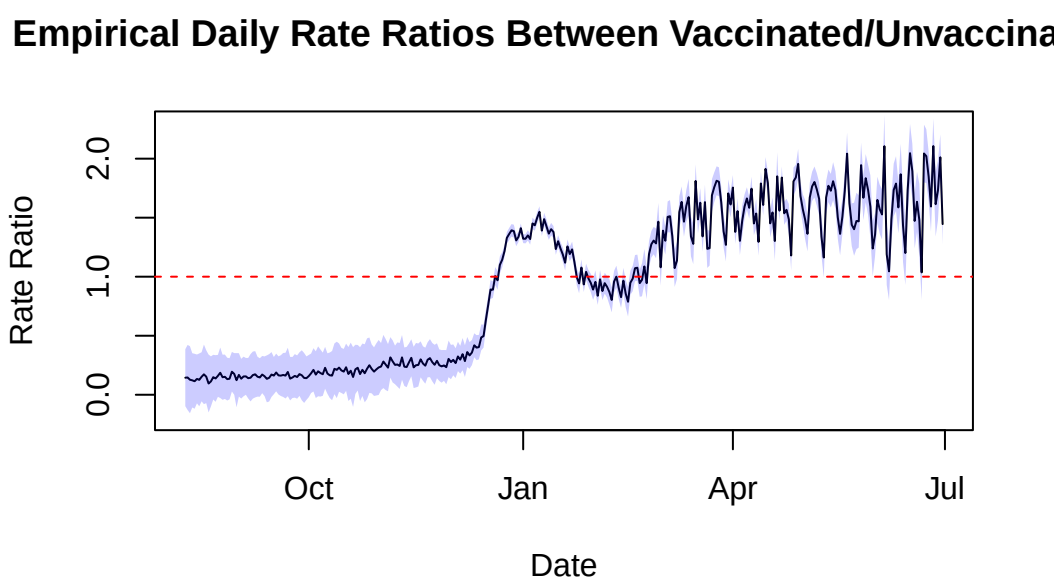
(b) Fit an appropriate saturated Poisson regression model to verify that you can use the model to reproduce the same daily rate ratios. Add also the 95% confidence bands into the plot.

```
# question 4(b) preparation
# Long format dataset:

case_long <- cases[,c('date','unvac','vac','unvacpop','vacpop')]
case_long <- reshape(case_long,
                     varying=list(c('unvac','vac'),c('unvacpop', 'vacpop')),
                     direction='long', times=c(0,1))
case_long <- case_long[order(case_long$date),]
names(case_long) <- c('date', 'fullyvac', 'd', 'y', 'id')
head(case_long)
```

```
          date fullyvac   d       y id
1.0 2021-08-09        0 339 5572010  1
1.1 2021-08-09        1  82 9343260  1
2.0 2021-08-10        0 254 5534148  2
2.1 2021-08-10        1  63 9381122  2
```

```
3.0 2021-08-11        0 266 5496638  3
3.1 2021-08-11        1  58 9418632  3
```

```r
# 4(b)

# fit the model
poisson_model <- glm(
  d ~ -1 + fullyvac:factor(date) + factor(date),
  offset = log(y),
  family = poisson(link = "log"),
  data = case_long
)

coefficients <- coef(poisson_model)

# vaccinated/unvaccinated log-rate ratios
log_rateratios <- coefficients[grep("fullyvac:factor\\(date\\)", names(coefficients))]
rateratios <- exp(log_rateratios)


plot(x = cases$date, y = rateratios, ylim=c(-0.2,2.3), type = "l", xlab = "Date",
     ylab = "Rate Ratio",
     main = "Empirical Daily Rate Ratios Between Vaccinated/Unvaccinated")

# 95% confidence bands
se_rateratios <- summary(poisson_model)$coefficients[grep("fullyvac:factor\\(date\\)",
                   rownames(summary(poisson_model)$coefficients)), "Std. Error"]
cil_rateratios <- (rateratios - 1.96*se_rateratios)
ciu_rateratios <- (rateratios + 1.96*se_rateratios)

polygon(c(unique(cases$date), rev(unique(cases$date))),
        c(cil_rateratios, rev(ciu_rateratios)), col = rgb(0, 0, 1, 0.2), border = NA)
abline(h = 1, col = "red", lty = 2)
```



What can you say about the "vaccination effect" on preventing infection/positive test over time? Would the assumption of constant vaccination effect be reasonable in this case? How can you test this?

∴ Before around December-January, the rate ratio was consistently around 0.25, which means vaccinated individuals had lower positive test rates than unvaccinated individuals. However, the rate ratio increased rapidly over 1.5 in January, which means unvaccinated individuals had lower positive test rates than vaccinated individuals. The rate ratio decreased around and below 1 around mid-February. Then from March, the rate ratio increased again and then started fluctuating rapidly between 1 and 2, which means unvaccinated individuals might have an even lower rate of positive test than vaccinated individuals during this time.

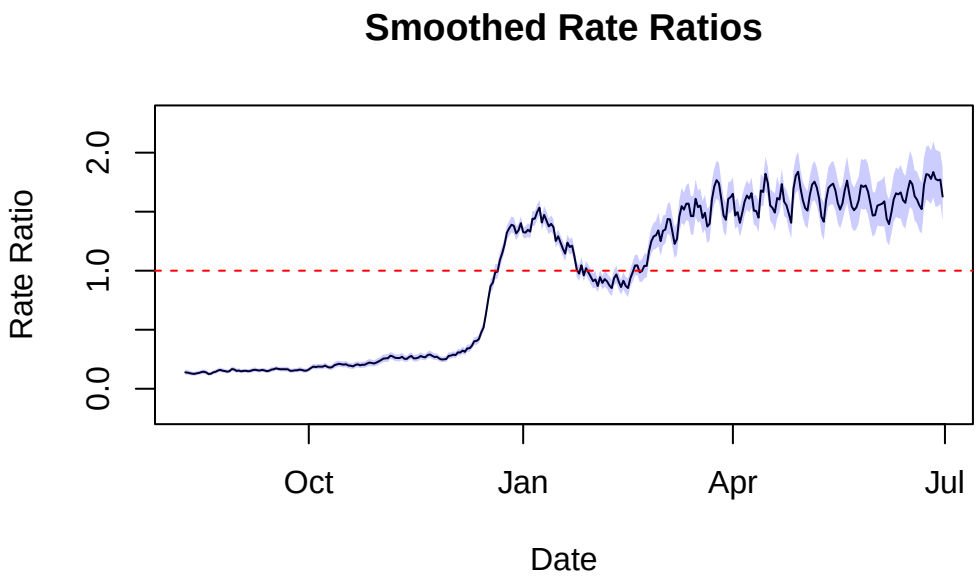
The assumption of constant vaccination effect would not be reasonable in this case, since the rate ratio showed an opposite effect of the vaccine in the later time period while it showed a decent effect in the earlier time period. To test this, we can fit a model with an interaction between vaccination status and time, and compare it to the model without interaction using the anova likelihood ratio test.

(c) The daily rate ratios are quite noisy due to small counts. Using JAGS, fit an appropriate Bayesian Poisson regression model to smooth the daily rates, and plot the resulting posterior mean rate ratios along with the 95% credible intervals. Compare the results to the unsmoothed estimates.

```
# jags model with smoothing prior
smooth_model <- "
model {
  for (i in 1:N) {
    d[i] ~ dpois(mu[i])
    log(mu[i]) <- log(y[i])+beta[date[i]]+fullyvac[i]*gamma[date[i]]
  }

  for (j in 1:ndates) {
    beta[j] ~ dnorm(0.0,0.0001)
  }

  gamma[1] ~ dnorm(0.0,0.0001)
  for (j in 2:ndates) {
    gamma[j] ~ dnorm(gamma[j-1],tau_gamma)
  }

  tau_gamma ~ dgamma(0.0001,0.0001)
}
"

# data list
datalist <- list(
  d = case_long$d,
  y = case_long$y,
  fullyvac = case_long$fullyvac,
  date = as.numeric(factor(case_long$date)),
  N = nrow(case_long),
  ndates = length(unique(case_long$date))
)

# initial list
initslist <- list("beta"=rep(0,datalist$ndates), "gamma"=rep(0,datalist$ndates))

# fit the model
model <- jags.model(textConnection(smooth_model), data=datalist, inits=initslist,
                    n.chains=2, quiet=FALSE)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 652
   Unobserved stochastic nodes: 653
   Total graph size: 5873

Initializing model
```

```r
samples <- coda.samples(model, c("beta","gamma"), n.iter=10000, n.burnin=5000,
                        thin = 10)

# log rate ratios
logrrs <- as.matrix(samples[, grep("gamma", colnames(as.matrix(samples)))])
# rate ratios
rrs <- exp(logrrs)

# posterior mean and credible interval
rr <- colMeans(rrs)
ciu <- apply(rrs, 2, quantile, probs = 0.975)
cil <- apply(rrs, 2, quantile, probs = 0.025)

plot(unique(case_long$date), rr, ylim = c(-0.2, 2.3), type = "l", xlab = "Date",
     ylab = "Rate Ratio", main = "Smoothed Rate Ratios")
polygon(c(unique(case_long$date), rev(unique(case_long$date))), c(cil, rev(ciu)),
        col = rgb(0, 0, 1, 0.2), border = NA)
abline(h = 1, col = "red", lty = 2)
```



∴ The smoothed rate ratios plot shows a similar trajectory as the unsmoothed rate ratios plot with smaller overall noise, especially in the later time period. Now it's more clear to see that the rate ratio was constantly around 1.5 from April to July with some fluctuations, which means unvaccinated individuals had lower positive test rates than vaccinated individuals during this time period. So being vaccinated increased the rate of positive test by around 50% between April and July.

(d) The results in (a)-(c) should be interpreted with caution because the analysis does not adjust for age. Explain why.

∴ This is because age could have a strong effect on both the vaccination status and the risk of infection, which affects the positive test rates. The older population was prioritized for vaccination and they were also at higher risk of infection due to weaker immune systems. Also, we need to consider that the lifestyles are significantly different between younger populations and older populations. For example, younger individuals tend to meet or come across more people during their school and/or work while older individuals tend to not meet as many people often.

Therefore, when the age is not adjusted in the model, the differences we observe in the rate ratio might be due to the differences in the age distribution within the data rather than the vaccination status itself.

Complete Code

```r
# load the library
library(tidyverse)
library(rjags)

# question 1
alpha <- -4.3225 # intercept
beta <- -0.7039 # z
se_alpha <- 0.1491
se_beta <- 0.2601

# incidence rates
ir_control <- exp(alpha)
ir_intervention <- exp(alpha + beta)

# approximate two-year risks = 1 - 2yrs survival probability
r_control <- 1 - exp(-ir_control * 2)
r_intervention <- 1 - exp(-ir_intervention * 2)

cat("Control arm: incidence rate =", ir_control, ", two-year risk =", r_control, ".\n")
cat("Intervention arm: incidence rate =", ir_intervention,
    ", two-year risk =", r_intervention, ".\n")

cil <- exp(beta - 1.96 * se_beta)
ciu <- exp(beta + 1.96 * se_beta)

cat("95% CI for HIV incidence rate ratio: (", cil, ",", ciu, ").")

q1data <- data.frame(
  n = c(2430, 2387), #N0, N1
  d = c(45, 22), #D0, D1
  z = c(0, 1), #Z0, Z1
  y = c(3391.8, 3352.4) #Y0, Y1
)

#model <- glm(d ~ z + offset(log(y)), family = poisson(link = "log"), data = q1data)
model <- glm(d ~ z, offset = log(y), family = poisson(link = "log"), data = q1data)

summary(model)
```

```r
logLik(model)

n = c(2430, 2387) #N0, N1
d = c(45, 22) #D0, D1
z = c(0, 1) #Z0, Z1
y = c(3391.8, 3352.4) #Y0, Y1

# alpha_hat = log(D0/Y0)
alpha_hat <- log(q1data$d[1] / q1data$y[1])
# beta_hat = log((D1*Y0) / (Y1*D0))
beta_hat <- log((q1data$d[2]*q1data$y[1] ) / (q1data$y[2]*q1data$d[1]))

cat("MLE for alpha:", alpha_hat, "\n")
cat("MLE for beta:", beta_hat, "\n")

var_alpha_hat <- (exp(alpha_hat+beta_hat)*y[2]) / (exp(2*alpha_hat+beta_hat)*y[1]*y[2])
var_beta_hat <- (exp(alpha_hat)*y[1] + exp(alpha_hat+beta_hat)*y[2]) /
  (exp(2*alpha_hat+beta_hat)*y[1]*y[2])

se_alpha_hat <- sqrt(var_alpha_hat)
se_beta_hat <- sqrt(var_beta_hat)

cat("Standard errors of maximum likelihood estimators: \n")
cat("se(alpha_hat) =", se_alpha_hat, "\n")
cat("se(beta_hat) =", se_beta_hat, "\n")

# Verify Wald test p-values
wald_alpha <- alpha_hat / se_alpha_hat
p_alpha <- 2 * (1 - pnorm(abs(wald_alpha)))
cat("Wald test statistic for alpha:", wald_alpha, "\n")
cat("p-value for alpha:", p_alpha, "\n")

wald_beta <- beta_hat / se_beta_hat
p_beta <- 2 * (1 - pnorm(abs(wald_beta)))

cat("Wald test statistic for beta:", wald_beta, "\n")
cat("p-value for beta:", p_beta, "\n")

# question 2
# reset the environment
rm(list=ls())
d <- c(4, 5, 8, 2, 12, 14)
```

```r
y <- c(607.9, 1272.1, 888.9, 311.9, 878.1, 667.5)
z <- c(0, 0, 0, 1, 1, 1)
x <- c(0, 1, 2, 0, 1, 2)

# manual calculation using MLEs from glm output
alpha_hat <- -5.4177
beta_hat <- 0.8697
gamma1_hat <- 0.1290
gamma2_hat <- 0.6920

data <- data.frame(
  d = d,
  y = y,
  z = z,
  x = x
)

llh <- 0
# for loop for each row(unique combination of x and z)
for (i in 1:nrow(data)) {
  di <- data$d[i]
  yi <- data$y[i]
  zi <- data$z[i]
  xi <- data$x[i]

  lambda_zx <- exp(alpha_hat + beta_hat * zi + gamma1_hat * (xi == 1) +
                     gamma2_hat * (xi == 2))
  llh <- llh + di * (log(lambda_zx) + log(yi)) - yi * lambda_zx - log(factorial(di))
}

cat("The Value of the log-likelihood function at the maximum likelihood point:",llh)

# double check
model <- glm(d ~ z + as.factor(x) + offset(log(y)), family = poisson(link = "log"))
summary(model)

logLik(model)

# fit the glm nodel
model_interaction <- glm(d ~ z * as.factor(x) + offset(log(y)),
                         family = poisson(link = "log"))
```

```r
summary(model_interaction)

logLik(model_interaction)

# manual calculation using MLEs from glm output
alpha_hat <- -5.02372
beta_hat <- -0.02582
gamma1_hat <- -0.51527
gamma2_hat <- 0.31317
delta1_hat <- 1.27195
delta2_hat <- 0.87188

data <- data.frame(
  d <- c(4, 5, 8, 2, 12, 14),
  y <- c(607.9, 1272.1, 888.9, 311.9, 878.1, 667.5),
  z <- c(0, 0, 0, 1, 1, 1),
  x <- c(0, 1, 2, 0, 1, 2)
)

llh <- 0
# for loop for each row(unique combination of x and z)
for (i in 1:nrow(data)) {
  di <- data$d[i]
  yi <- data$y[i]
  zi <- data$z[i]
  xi <- data$x[i]

  lambda_zx <- exp(alpha_hat + beta_hat * zi + gamma1_hat * (xi == 1) +
                     gamma2_hat * (xi == 2) + delta1_hat * zi * (xi == 1) +
                     delta2_hat * zi * (xi == 2))
  llh <- llh + di * (log(lambda_zx) + log(yi)) - yi * lambda_zx - log(factorial(di))
}

llh

D <- 2 * (logLik(model_interaction) - logLik(model))
p_value <- pchisq(D, df = 2, lower.tail = FALSE)

cat("LRT Statistic (D):", D, "\n")
cat("p-value:", p_value, "\n")

r = residuals(model, type = "deviance")
```

```r
r
cat("Residual Deviance:", sum(r^2), "\n")

# question 3
# reset the environment
rm(list=ls())

# enter data
q3data <- data.frame(
  age = rep(c(22, 23, 24, 25, 26, 27, 28, 29), each=2),
  marital_status = rep(c(0, 1), times = 8),  # 0 = single, 1 = married
  deaths = c(433, 24, 412, 36, 373, 66, 331, 102,
             287, 138, 242, 171, 215, 185, 192, 200),
  person_years = c(91444, 8556, 86835, 12708, 75892, 23203, 63241, 35415, 52023,
                   46207, 42123, 55675, 36915, 60470, 32215, 64770)
)

# fit the model
model <- glm(deaths ~ factor(marital_status) + as.factor(age),
             offset = log(person_years),
             family = poisson(link = "log"),
             data = q3data)

summary(model)

q3data$expected_deaths <- predict(model, type = "response")
q3data %>% select(age, marital_status, deaths, expected_deaths)

# chi-squared goodness of fit test
x2 <- sum((q3data$deaths - q3data$expected_deaths)^2 / q3data$expected_deaths)
cat("Chi-square value:", x2, ".\n")

# p-value
p <- 1 - pchisq(x2, df = nrow(q3data)-1)
cat("p-value:",p)

# fit the model
interaction_model <- glm(deaths ~ factor(marital_status) * factor(age),
                         offset = log(person_years),
                         family = poisson(link = "log"),
                         data = q3data)
```

```r
summary(interaction_model)

q3data$interaction_expected_deaths <- predict(interaction_model, type = "response")
q3data %>% select(age, marital_status, deaths, interaction_expected_deaths)

# anova
anova(model, interaction_model, test = "Chisq")

# question 4
# reset the environment
rm(list=ls())

# a part of cases_hosp.r below
# by Dr. Olli Saarela

# Case and hospitalization rates by vax status
# (crude rates, no age adjustment, interpret with caution)
# Datasets downloaded from https://data.ontario.ca/dataset/covid-19-vaccine-data-in-ontario

#setwd('c:/Users/ollis/Dropbox/work/CHL5209H_2025/data')

# Case counts:

cases <- read.csv('cases_by_vac_status.csv')

cases$date <- as.Date(cases$Date, "%Y-%m-%d")
cases$unvac <- ifelse(!is.na(cases$covid19_cases_notfull_vac),
                      cases$covid19_cases_notfull_vac,
                      cases$covid19_cases_unvac + cases$covid19_cases_partial_vac)
cases$vac <- ifelse(is.na(cases$covid19_cases_boost_vac),
                    cases$covid19_cases_full_vac,
                    cases$covid19_cases_full_vac + cases$covid19_cases_boost_vac)

# Hosp. counts:

hosp <- read.csv('vac_status_hosp_icu.csv')

hosp$date <- as.Date(hosp$date, "%Y-%m-%d")
hosp$vac <- hosp$hospitalnonicu_full_vac + hosp$icu_full_vac
hosp$unvac <- hosp$icu_unvac + hosp$icu_partial_vac +
  hosp$hospitalnonicu_unvac + hosp$hospitalnonicu_partial_vac
```

```r
hosp$tot <- hosp$unvac + hosp$vac
hosp$prop <- hosp$vac/hosp$tot

# Get population denominators by vax status:

vac <- read.csv('vaccine_doses.csv')

vac$vacpop <- vac$total_individuals_fully_vaccinated
# Use Q4 Ontario population from https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=171000
vac$totalpop <- 14915270
vac$unvacpop <- vac$totalpop - vac$vacpop

vac$date <- as.Date(vac$report_date, "%Y-%m-%d")
vac <- vac[,c('date','unvacpop','vacpop','totalpop')]

intersect(names(cases), names(vac))
cases <- merge(cases, vac)

intersect(names(hosp), names(vac))
hosp <- merge(hosp, vac)

# Case rate:

cases$vacrate <- cases$vac/cases$vacpop * 100000
cases$unvacrate <- cases$unvac/cases$unvacpop * 100000

plot(cases$date, cases$unvacrate, type='l', ylim=c(0,150), lwd=2, col='blue',
     xlab='Date', ylab='Rate per 100,000 population', main='Empirical daily case rates')
lines(cases$date, cases$vacrate, lwd=2, col='red')
abline(h=seq(0,150,50), lty='dotted')
legend('topleft', legend=c('Fully vac\'d','Not fully vac\'d'), lwd=2,
       col=c('red','blue'))

# 4a
# Compute empirical daily rate ratios
cases$rate_ratio = cases$vacrate / cases$unvacrate

plot(x = cases$date, y = cases$rate_ratio, type = "l", xlab = "Date",
     ylab = "Rate Ratio", ylim=c(-0.2,2.3),
     main = "Empirical Daily Rate Ratios Between Vaccinated/Unvaccinated")
abline(h = 1, col = "red", lty = 2)
```

```r
# question 4(b) preparation
# Long format dataset:

case_long <- cases[,c('date','unvac','vac','unvacpop','vacpop')]
case_long <- reshape(case_long, varying=list(c('unvac','vac'), c('unvacpop', 'vacpop')),
                     direction='long', times=c(0,1))
case_long <- case_long[order(case_long$date),]
names(case_long) <- c('date', 'fullyvac', 'd', 'y', 'id')
head(case_long)

# 4(b)

# fit the model
poisson_model <- glm(
  d ~ -1 + fullyvac:factor(date) + factor(date),
  offset = log(y),
  family = poisson(link = "log"),
  data = case_long
)

coefficients <- coef(poisson_model)

# vaccinated/unvaccinated log-rate ratios
log_rateratios <- coefficients[grep("fullyvac:factor\\(date\\)", names(coefficients))]
rateratios <- exp(log_rateratios)


plot(x = cases$date, y = rateratios, ylim=c(-0.2,2.3), type = "l", xlab = "Date",
     ylab = "Rate Ratio",
     main = "Empirical Daily Rate Ratios Between Vaccinated/Unvaccinated")

# 95% confidence bands
se_rateratios <- summary(poisson_model)$coefficients[grep("fullyvac:factor\\(date\\)",
                rownames(summary(poisson_model)$coefficients)), "Std. Error"]
cil_rateratios <- (rateratios - 1.96*se_rateratios)
ciu_rateratios <- (rateratios + 1.96*se_rateratios)

polygon(c(unique(cases$date), rev(unique(cases$date))),
        c(cil_rateratios, rev(ciu_rateratios)), col = rgb(0, 0, 1, 0.2), border = NA)
abline(h = 1, col = "red", lty = 2)

#4c
```

```
# jags model with smoothing prior
smooth_model <- "
model {
  for (i in 1:N) {
    d[i] ~ dpois(mu[i])
    log(mu[i]) <- log(y[i])+beta[date[i]]+fullyvac[i]*gamma[date[i]]
  }

  for (j in 1:ndates) {
    beta[j] ~ dnorm(0.0,0.0001)
  }

  gamma[1] ~ dnorm(0.0,0.0001)
  for (j in 2:ndates) {
    gamma[j] ~ dnorm(gamma[j-1],tau_gamma)
  }

  tau_gamma ~ dgamma(0.0001,0.0001)
}
"

# data list
datalist <- list(
  d = case_long$d,
  y = case_long$y,
  fullyvac = case_long$fullyvac,
  date = as.numeric(factor(case_long$date)),
  N = nrow(case_long),
  ndates = length(unique(case_long$date))
)

# initial list
initslist <- list("beta"=rep(0,datalist$ndates), "gamma"=rep(0,datalist$ndates))

# fit the model
model <- jags.model(textConnection(smooth_model), data=datalist, inits=initslist,
                    n.chains=2, quiet=FALSE)

samples <- coda.samples(model, c("beta","gamma"), n.iter=10000, n.burnin=5000,
                        thin = 10)

# log rate ratios
```

```r
logrrs <- as.matrix(samples[, grep("gamma", colnames(as.matrix(samples)))])
# rate ratios
rrs <- exp(logrrs)

# posterior mean and credible interval
rr <- colMeans(rrs)
ciu <- apply(rrs, 2, quantile, probs = 0.975)
cil <- apply(rrs, 2, quantile, probs = 0.025)

plot(unique(case_long$date), rr, ylim = c(-0.2, 2.3), type = "l", xlab = "Date",
     ylab = "Rate Ratio", main = "Smoothed Rate Ratios")
polygon(c(unique(case_long$date), rev(unique(case_long$date))), c(cil, rev(ciu)),
        col = rgb(0, 0, 1, 0.2), border = NA)
abline(h = 1, col = "red", lty = 2)
```