Homework 1

Sarah Adilijiang

Problem 1

(a)

The generalized linear model (GLM) for binary response here is:
likelihood: $P(spikes_i = y_i|p_i) = p_i^{y_i}(1-p_i)^{1-y_i},\ y_i = 0\ or\ 1$
linear predictor: $\eta_i = \beta_0 + \beta_1 xN_i + \beta_2 yN_i + \epsilon_i$
link function (logit): $\eta_i = log\frac{p_i}{1-p_i}$

```
load("HIPP.Rdata")
model.glm = glm(spikes~xN+yN, family=binomial, data=HIPP)
summary(model.glm)
```

```
##
## Call:
## glm(formula = spikes ~ xN + yN, family = binomial, data = HIPP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5637  -0.3806  -0.2653  -0.1925   2.9658
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.20040    0.02814 -113.73   <2e-16 ***
## xN          -0.53624    0.04872  -11.01   <2e-16 ***
## yN          -1.24991    0.04635  -26.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16306  on 41327  degrees of freedom
## Residual deviance: 15332  on 41325  degrees of freedom
## AIC: 15338
##
## Number of Fisher Scoring iterations: 6
```

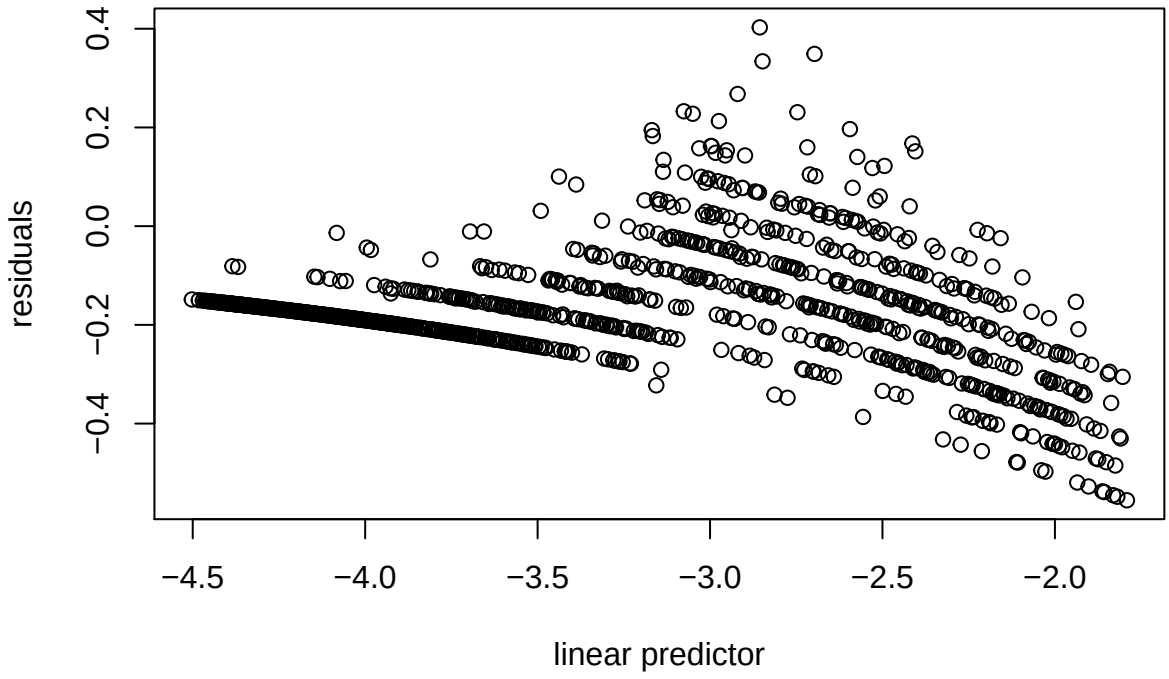
```
# binned deviance residuals plot against the linear predictor eta
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
HIPP2 = data.frame(HIPP)
HIPP2 = mutate(HIPP2, residuals=residuals(model.glm), linpred=predict(model.glm))
gdf = group_by(HIPP2, cut(linpred, breaks= unique(quantile(linpred, (1:1000)/1001)))) # each bin ~ 40 p
diagdf = summarise(gdf, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf, xlab="linear predictor")
```



Comment:

In the binned residuals plot against the liner predictor η, we see that there are nonlinearity trends in the residuals and there is a clear pattern, thus the model assumptions are not appropriate with respect to this data set.

(b)

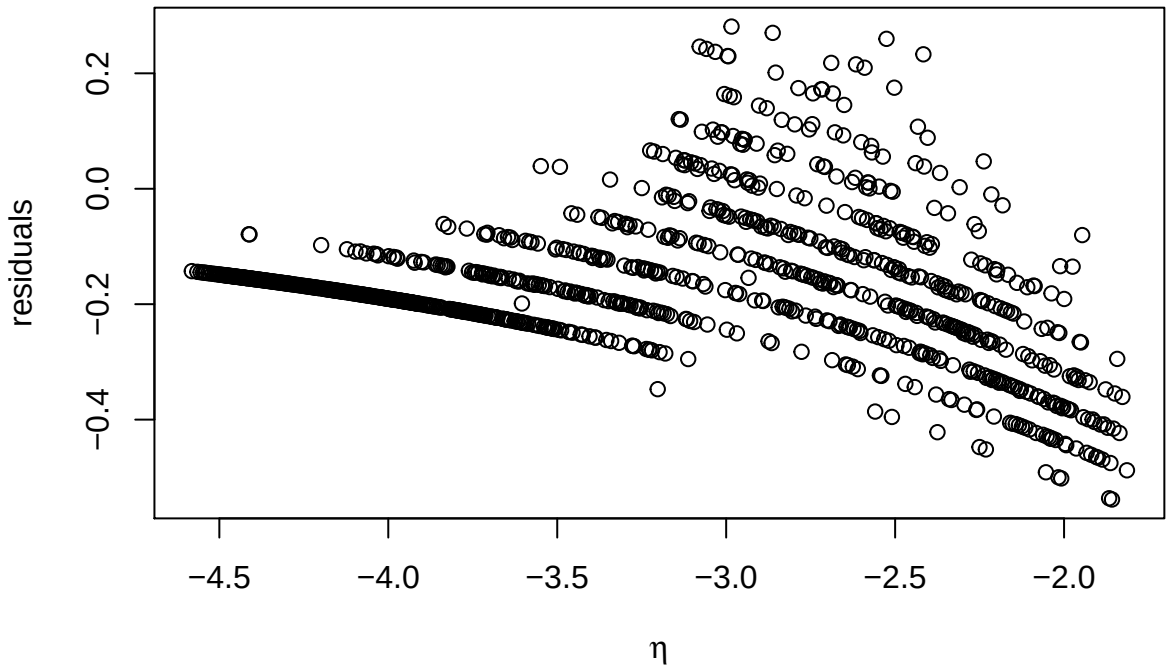
Fit the GLM model with the linear predictor η including the interactions of xN and yN:
linear predictor: $\eta_i = \beta_0 + \beta_1 xN_i + \beta_2 yN_i + \beta_3 xN_i : yN_i + \epsilon_i$

```
model.glm2 = glm(spikes~xN*yN, family=binomial, data=HIPP)
summary(model.glm2)
```

```
##
## Call:
## glm(formula = spikes ~ xN * yN, family = binomial, data = HIPP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5573  -0.3821  -0.2676  -0.1907   2.9747
```

```
##
## Coefficients:
##             Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -3.20264    0.02831 -113.113   <2e-16 ***
## xN          -0.57038    0.05637  -10.118   <2e-16 ***
## yN          -1.26036    0.04722  -26.690   <2e-16 ***
## xN:yN       -0.14722    0.12084   -1.218    0.223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16306  on 41327  degrees of freedom
## Residual deviance: 15330  on 41324  degrees of freedom
## AIC: 15338
##
## Number of Fisher Scoring iterations: 6
```

```r
# binned deviance residuals plot against the linear predictor eta
library(dplyr)
HIPP2 = data.frame(HIPP)
HIPP2 = mutate(HIPP2, residuals=residuals(model.glm2), linpred=predict(model.glm2))
gdf = group_by(HIPP2, cut(linpred, breaks= unique(quantile(linpred, (1:1000)/1001)))) # each bin ~ 40 p
diagdf = summarise(gdf, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf, xlab=expression(eta))
```



```r
# compare the two models
anova(model.glm2, model.glm, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: spikes ~ xN * yN
## Model 2: spikes ~ xN + yN
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     41324      15330
## 2     41325      15332 -1  -1.4876   0.2226
```

Answer:

There is stil nonlinearity trends and a clear pattern in the residuals plot, and the AIC value is the same with the original model in question (a). And the anova chi-square test statistics has a p-value = 0.2226, so we do not reject the null hypothesis, thus the smaller model without interaction terms is preferred in this case.

Therefore, fit the model with interaction terms of xN and yN will not improve the fit.

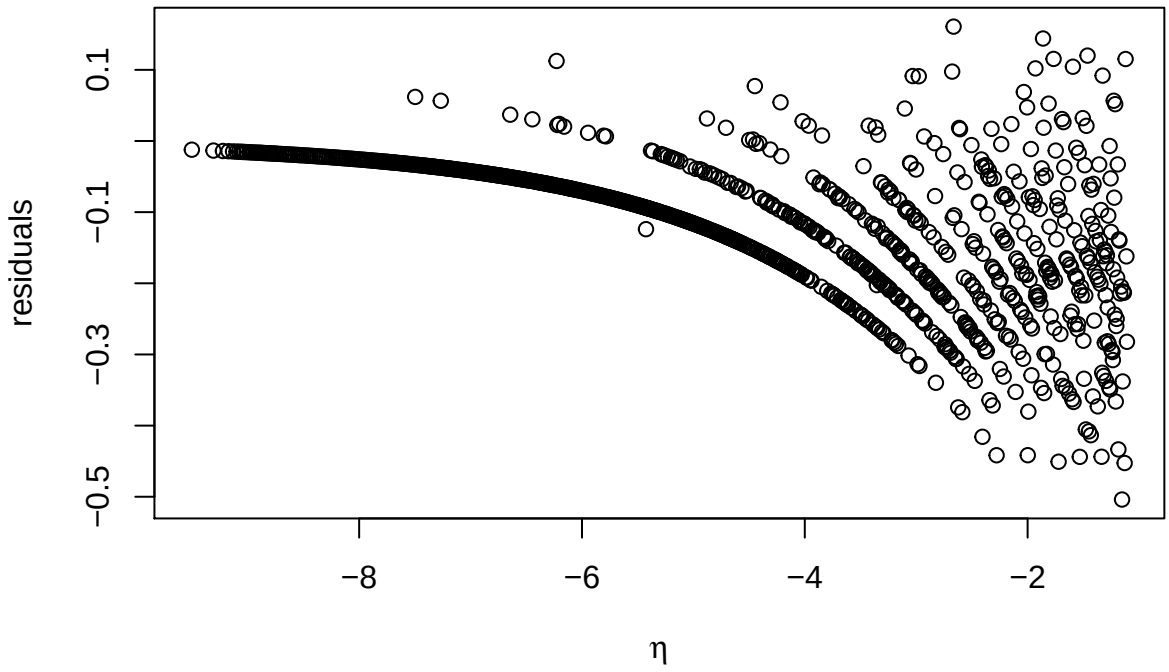
Then, fit the GLM model with the linear predictor η including second-order terms of xN and yN:
linear predictor: $\eta_i = \beta_0 + \beta_1 xN_i + \beta_2 yN_i + \beta_3 xN_i^2 + \beta_4 yN_i^2 + \epsilon_i$

```r
model.glm3 = glm(spikes~xN+yN+I(xN^2)+I(yN^2), family=binomial, data=HIPP)
summary(model.glm3)
```

```
##
## Call:
## glm(formula = spikes ~ xN + yN + I(xN^2) + I(yN^2), family = binomial,
##     data = HIPP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7571  -0.3462  -0.1537  -0.0528   3.8714
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.62104    0.04011  -40.42   <2e-16 ***
## xN          -1.57066    0.09582  -16.39   <2e-16 ***
## yN          -2.59063    0.09990  -25.93   <2e-16 ***
## I(xN^2)     -6.31350    0.20830  -30.31   <2e-16 ***
## I(yN^2)     -3.98895    0.14218  -28.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16306  on 41327  degrees of freedom
## Residual deviance: 12938  on 41323  degrees of freedom
## AIC: 12948
##
## Number of Fisher Scoring iterations: 8
```

```r
# binned deviance residuals plot against the linear predictor eta
library(dplyr)
HIPP2 = data.frame(HIPP)
HIPP2 = mutate(HIPP2, residuals=residuals(model.glm3), linpred=predict(model.glm3))
```

```r
gdf = group_by(HIPP2, cut(linpred, breaks= unique(quantile(linpred, (1:1000)/1001)))) # each bin ~ 40 p
diagdf = summarise(gdf, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf, xlab=expression(eta))
```




```r
# compare the two models
anova(model.glm3, model.glm, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: spikes ~ xN + yN + I(xN^2) + I(yN^2)
## Model 2: spikes ~ xN + yN
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1     41323      12938
## 2     41325      15332 -2  -2393.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:

Though there is stil nonlinearity trends and a clear pattern in the residuals plot, the AIC value of this model is much lower than the original model in question (a). And the anova chi-square test statistics has a p-value $<$ 2.2e-16, so we reject the null hypothesis, thus the larger model with second-order terms is preferred in this case.

Therefore, fit the model with second-order terms of xN and yN will improve the fit.

(c)

```r
# sample the data every 20 points
HIPP2 = data.frame(HIPP)
HIPP3 = data.frame(matrix(0,1,ncol(HIPP2)))
colnames(HIPP3) = colnames(HIPP2)
for (i in 1:2067) {
    HIPP3[i, ] = HIPP2[1+20*(i-1), ]
}

# fit the GLM model same as in question (a)
model.glm4 = glm(spikes~xN+yN, family=binomial, data=HIPP3)
summary(model.glm4)
```

```
##
## Call:
## glm(formula = spikes ~ xN + yN, family = binomial, data = HIPP3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5418  -0.3763  -0.2649  -0.1955   2.9424
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.2052     0.1252 -25.610  < 2e-16 ***
## xN           -0.5415     0.2197  -2.464   0.0137 *
## yN           -1.1798     0.2072  -5.695 1.24e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 800.82  on 2066  degrees of freedom
## Residual deviance: 757.01  on 2064  degrees of freedom
## AIC: 763.01
##
## Number of Fisher Scoring iterations: 6
```

Answer:

When sampling the data every 20 points, the value of fitted parameters are similar, but the coefficient of xN becomes less significant comparing with the model in question (a).

(d)

```r
# construct the new dataset
HIPP3_1 = HIPP3[HIPP3$spikes==1, ]
HIPP3_0 = HIPP3[HIPP3$spikes==0, ]

n = nrow(HIPP3) - nrow(HIPP3[HIPP3$spikes==1, ])
indices = sample(1:n, 100, replace=FALSE)
HIPP3_0 = HIPP3_0[indices, ]

HIPP4 = rbind(HIPP3_0, HIPP3_1)

# fit the linear predictor with second-order terms same as in question (b)
model.glm5 = glm(spikes~xN+yN+I(xN^2)+I(yN^2), family=binomial, data=HIPP4)
```

```r
summary(model.glm5)
```

```
##
## Call:
## glm(formula = spikes ~ xN + yN + I(xN^2) + I(yN^2), family = binomial,
##     data = HIPP4)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9057  -0.6499   0.2501   0.7319   3.0292
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.2305     0.3028   4.064 4.83e-05 ***
## xN           -1.4678     0.5614  -2.615  0.00893 **
## yN           -2.7395     0.6234  -4.394 1.11e-05 ***
## I(xN^2)      -6.0562     1.1862  -5.106 3.30e-07 ***
## I(yN^2)      -4.0851     0.9651  -4.233 2.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 277.26  on 199  degrees of freedom
## Residual deviance: 185.30  on 195  degrees of freedom
## AIC: 195.3
##
## Number of Fisher Scoring iterations: 5
```

Answer:

The coefficients are not the same, and the coefficient of xN becomes much less significant comparing with the model with second-order terms in question (b).

When using the original dataset, the predictors are fixed and the outcome is observed, thus we are doing prospective sampling. In this case, the probabilities that a neuron is included in the study are the same whether or not the neuron is fired.

However, when using the new dataset in this question with same number of fired and not fired neurons, the outcome is fixed and the predictors are observed, thus we are doing retrospective sampling. In this case, the probability that a fired neuron is included in the study is larger than a not fired neuron.

If π_0 is the probabiltiy that a neuron is included in the study if it is fired, while π_1 is the probability of inclusion if it is not fired. In prospective study, $\pi_0 = \pi_1$, while in retrospective study, usually $\pi_1 > \pi_0$.

Let's set the unconditional probability that a neuron is fired in the prosepctive study as $p(x)$, and the conditional probability that a neuron is fired in the retrosepctive study as $p^*(x)$.

So we have $p^*(x) = \frac{\pi_1 p(x)}{\pi_1 p(x) + \pi_0(1-p(x))} \quad \Rightarrow \quad logit(p^*(x)) = log\frac{\pi_1}{\pi_0} + logit(p(x))$, so the only difference between two kinds of studies would be the difference in the intercept: $log\frac{\pi_1}{\pi_0}$

(e)

Fit the GLM model with the linear predictor η including the variabel spikes.hist:
linear predictor: $\eta_i = \beta_0 + \beta_1 xN_i + \beta_2 yN_i + \beta_3 spikes.hist + \epsilon_i$

```r
model.glm6 = glm(spikes~xN+yN+spikes.hist, family=binomial, data=HIPP)
summary(model.glm6)
```

```
##
## Call:
## glm(formula = spikes ~ xN + yN + spikes.hist, family = binomial,
##     data = HIPP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8601  -0.2832  -0.1819  -0.1433   3.3975
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.98842    0.04205 -94.850  < 2e-16 ***
## xN            -0.41805    0.06210  -6.732 1.67e-11 ***
## yN            -0.82702    0.05608 -14.746  < 2e-16 ***
## spikes.hist1  -2.83822    0.18801 -15.096  < 2e-16 ***
## spikes.hist2  -1.57590    0.12220 -12.897  < 2e-16 ***
## spikes.hist3  -0.48330    0.09562  -5.055 4.31e-07 ***
## spikes.hist4   0.83538    0.07732  10.804  < 2e-16 ***
## spikes.hist5   1.62140    0.07262  22.328  < 2e-16 ***
## spikes.hist6   1.57906    0.07763  20.341  < 2e-16 ***
## spikes.hist7   1.28072    0.08341  15.354  < 2e-16 ***
## spikes.hist8   0.84931    0.08698   9.764  < 2e-16 ***
## spikes.hist9   0.57482    0.08611   6.675 2.47e-11 ***
## spikes.hist10  0.51384    0.08341   6.161 7.24e-10 ***
## spikes.hist11  0.44226    0.08407   5.261 1.43e-07 ***
## spikes.hist12  0.49832    0.08402   5.931 3.01e-09 ***
## spikes.hist13  0.60565    0.08344   7.258 3.92e-13 ***
## spikes.hist14  0.48414    0.08638   5.605 2.08e-08 ***
## spikes.hist15  0.60580    0.08339   7.265 3.73e-13 ***
## spikes.hist16  0.46846    0.08452   5.543 2.98e-08 ***
## spikes.hist17  0.49804    0.08294   6.005 1.91e-09 ***
## spikes.hist18  0.37698    0.08438   4.468 7.90e-06 ***
## spikes.hist19  0.32819    0.08323   3.943 8.04e-05 ***
## spikes.hist20  0.24809    0.08223   3.017  0.00255 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16306  on 41327  degrees of freedom
## Residual deviance: 12962  on 41305  degrees of freedom
## AIC: 13008
##
## Number of Fisher Scoring iterations: 7
```

```r
# compare the two models
anova(model.glm, model.glm6, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: spikes ~ xN + yN
```

```
## Model 2: spikes ~ xN + yN + spikes.hist
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1     41325      15332
## 2     41305      12962 20     2370 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:

The anova chi-square test statistics has a p-value $<$ 2.2e-16, so we reject the null hypothesis, thus the larger model adding the spikes.hist variable is preferred in this case. So we may say that the history of spike activity of the neuron improve the prediction of the spikes relative to the location covarites.

Looking at the coefficients fo the history data, we see that, overall, all the 20 time points are significant predictors. However, the most recent 8 time points have larger absolute values of coefficents and are much more siginicant than the other previous time points. On the other hand, the most recent 3 time points have negative effect on the response, and the following other time points all have positive effect on the response.

Problem 2

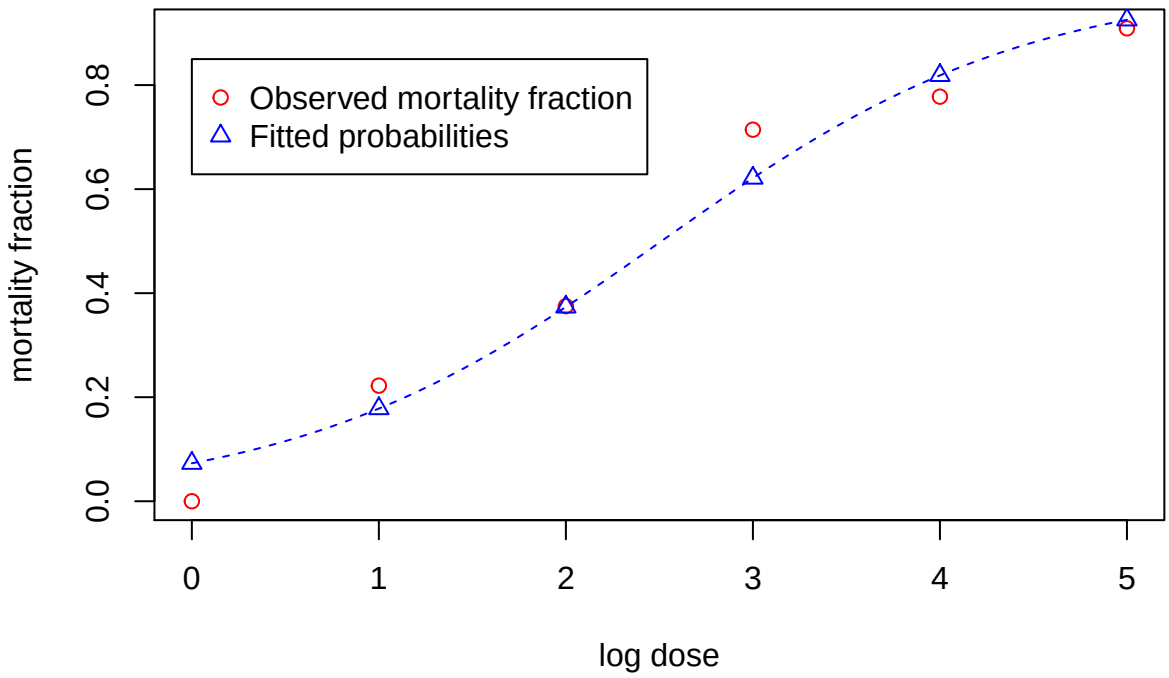
(a)

The generalized linear model (GLM) for binomial (proportion) response here is:
likelihood: $P(numdeath_i = y_i|p_i) = \binom{m_i}{y_i} p_i^{y_i}(1-p_i)^{m_i-y_i},\ y_i = 0, 1, ..., m_i$
linear predictor: $\eta_i = \beta_0 + \beta_1 log_2(dose)_i + \epsilon_i$
link function (logit): $\eta_i = log\frac{p_i}{1-p_i}, where\ mortality\ fraction\ p_i = y_i/m_i$

```r
# construct the binomial dataset
log_dose = c(0:5)
numdead = c(0,2,3,5,7,10)
m = c(7,9,8,7,9,11)
numalive = m-numdead
SF = cbind(numdead, numalive)
LD = data.frame(SF, log_dose)

# fit the binomial model
model.glm = glm(SF~log_dose, family=binomial, data=LD)

# plot
mortality_frac = numdead/m
plot(mortality_frac~log_dose, col=2, xlab="log dose", ylab="mortality fraction")
predprob = predict(model.glm, type="response")
points(predprob~log_dose, pch=2, col=4)
legend(0,0.85, c("Observed mortality fraction","Fitted probabilities"), pch=c(1,2), col=c(2,4))
ld = seq(0,5,by=0.001)
lines(ld, predict(model.glm, data.frame(log_dose=ld), type="response"), lty=2, col=4)
```



(b)

LD_{50} is the dose that causes a 50% mortality rate, so we set $p = 0.5$ and get $\eta = log\frac{p}{1-p} = 0$

Since $\hat{\eta} = \hat{\beta}_0 + \hat{\beta}_1 log_2(dose) = 0$, so we get $\widehat{log_2(LD_{50})} = -\hat{\beta}_0/\hat{\beta}_1$

```
beta = coef(model.glm)
log_LD50 = - beta[1]/beta[2]; log_LD50
```

```
## (Intercept)
##    2.510815
```

Then, to get the standard error:

Since $\hat{\theta} = (\hat{\beta}_0, \hat{\beta}_1)^T$, and $\widehat{log_2(LD_{50})} = -\hat{\beta}_0/\hat{\beta}_1 = g(\hat{\theta})$

$\Rightarrow g'(\hat{\theta}) = (\frac{\partial g(\hat{\theta})}{\partial \hat{\beta}_0}, \frac{\partial g(\hat{\theta})}{\partial \hat{\beta}_1})^T = (-\frac{1}{\hat{\beta}_1}, \frac{\hat{\beta}_0}{\hat{\beta}_1{}^2})^T$

Then use $Var[g(\hat{\theta})] \approx g'(\hat{\theta})^T Var(\hat{\theta}) g'(\hat{\theta})$ to compute the standard error.

```
dr = c(-1/beta[2], beta[1]/beta[2])
cov_matrix = summary(model.glm)$cov.unscaled
se = sqrt(t(dr) %*% cov_matrix %*% dr)
CI_0.95 = c(log_LD50-1.96*se, log_LD50+1.96*se); CI_0.95
```

```
## [1] 1.798816 3.222814
```

Answer:

Using the delta method, the point estimation of $log_2(LD_{50})$ is 2.510815, and its 95% CI is (1.798816, 3.222814).

(c)

Here we compute a 95% CI for $\rho = \beta_0/\beta_1$
First, the point estimation is: $\hat{\rho} = \hat{\beta}_0/\hat{\beta}_1$
Then, we compute the 95% CI for ρ:
Since $T_p = \frac{\beta_0 - \rho\beta_1}{\sqrt{C_{00} - 2\rho C_{01} + \rho^2 C_{11}}}$, and $\{\rho : |T_p| \leq z_{1-\alpha/2} = z_{0.975} = 1.96\} \quad \Rightarrow \quad |T_p|^2 \leq z^2$
$\Rightarrow (\beta_1^2 - z^2 C_{11})\rho^2 - 2(\beta_0\beta_1 - z^2 C_{01})\rho \ + (\beta_0^2 - z^2 C_{00}) \leq 0$
$\Rightarrow \frac{\beta_0\beta_1 - z^2 C_{01} \ - \sqrt{(\beta_0\beta_1 - z^2 C_{01})^2 - (\beta_0^2 - z^2 C_{00})(\beta_1^2 - z^2 C_{11})}}{\beta_1^2 - z^2 C_{11}} \leq \rho \leq \frac{\beta_0\beta_1 - z^2 C_{01} \ + \sqrt{(\beta_0\beta_1 - z^2 C_{01})^2 - (\beta_0^2 - z^2 C_{00})(\beta_1^2 - z^2 C_{11})}}{\beta_1^2 - z^2 C_{11}}$

```
beta0 = beta[1]; beta1 = beta[2]
rho = beta0/beta1; rho
```

```
## (Intercept)
##   -2.510815
```

```
C00 = cov_matrix[1,1]; C01 = cov_matrix[1,2]; C11 = cov_matrix[2,2]
z = 1.96
a = beta1^2 - z^2*C11
b = - 2*(beta0*beta1 - z^2*C01)
c = beta0^2 - z^2*C00
se = c( (-b - sqrt(b^2-4*a*c))/(2*a) , (-b + sqrt(b^2-4*a*c))/(2*a) ); se
```

```
## (Intercept) (Intercept)
##   -3.313109   -1.656959
```

Answer:

Using the Fieller's method, the point estimation of ρ is -2.510815, and its 95% CI is (-3.313109, -1.656959).

Since $log_2(LD_{50}) = -\beta_0/\beta_1 = -\rho$, so the point estimation of $log_2(LD_{50})$ is 2.510815, and its 95% CI is (1.656959, 3.313109).

(d)

```
# construct a new data frame including mi's and fitted pi's at each dose level
predprob = predict(model.glm, type="response")
LD = cbind(LD, m, predprob)
```

```
# nonparametric bootstrap
set.seed(101)
log_LD50_boot = NULL

for (i in 1:1000) {
    boot_data = data.frame(matrix(0,nrow(LD),4))
    colnames(boot_data) = c("numdead","numalive","log_dose","m")

    for (j in 1:nrow(LD)) {
        boot_data$m[j] = LD$m[j]
        boot_data$log_dose[j] = LD$log_dose[j]
        boot_data$numdead[j] = rbinom(n=1, size=LD$m[j], prob=LD$numdead[j]/LD$m[j])
        boot_data$numalive[j] = boot_data$m[j] - boot_data$numdead[j]
    }

    model.glm_boot = glm(cbind(numdead,numalive)~log_dose, family=binomial, data=boot_data)
    beta = coef(model.glm_boot)
    log_LD50_boot[i] = - beta[1]/beta[2]
```

```
}

mu = mean(log_LD50_boot); mu
```

```
## [1] 2.519667
```

```
se = sd(log_LD50_boot)
CI_0.95 = c(mu-se*1.96, mu+se*1.96); CI_0.95
```

```
## [1] 1.805743 3.233591
```

Answer:

Using nonparametric bootstrp, the point estimation of $log_2(LD_{50})$ is 2.519667, and its 95% CI is (1.805743, 3.233591).

```
# parametric bootstrap
set.seed(101)
log_LD50_boot = NULL

for (i in 1:1000) {
    boot_data = data.frame(matrix(0,nrow(LD),4))
    colnames(boot_data) = c("numdead","numalive","log_dose","m")

    for (j in 1:nrow(LD)) {
        boot_data$m[j] = LD$m[j]
        boot_data$log_dose[j] = LD$log_dose[j]
        boot_data$numdead[j] = rbinom(n=1, size=LD$m[j], prob=LD$predprob[j])
        boot_data$numalive[j] = boot_data$m[j] - boot_data$numdead[j]
    }

    model.glm_boot = glm(cbind(numdead,numalive)~log_dose, family=binomial, data=boot_data)
    beta = coef(model.glm_boot)
    log_LD50_boot[i] = - beta[1]/beta[2]
}

mu = mean(log_LD50_boot); mu
```

```
## [1] 2.485934
```

```
se = sd(log_LD50_boot)
CI_0.95 = c(mu-se*1.96, mu+se*1.96); CI_0.95
```

```
## [1] 1.761000 3.210868
```

Answer:

Using parametric bootstrp, the point estimation of $log_2(LD_{50})$ is 2.485934, and its 95% CI is (1.761000, 3.210868).

(e)

The restriction is : $\beta_0 + 4 * \beta_1 = 0 \quad \Rightarrow \quad \beta_0 = -4\beta_1$

So now the restricted GLM sub-model for binomial (proportion) response here is:
likelihood: $P(numdeath_i = y_i|p_i) = \binom{m_i}{y_i} p_i^{y_i}(1-p_i)^{m_i-y_i},\ y_i = 0, 1, ..., m_i$
linear predictor: $\eta_i = \beta_0 + \beta_1 log_2(dose)_i + \epsilon_i = -4\beta_1 + \beta_1 log_2(dose)_i + \epsilon_i = \beta_1(log_2(dose)_i - 4) + \epsilon_i$, which has no intercept term here.

link function (logit): $\eta_i = log\frac{p_i}{1-p_i}, where\ mortality\ fraction\ p_i = y_i/m_i$

```
# fit the sub-model
log_dose_new = log_dose - 4
model.glm_sub = glm(SF~log_dose_new-1, family=binomial)
summary(model.glm_sub)
```

```
##
## Call:
## glm(formula = SF ~ log_dose_new - 1, family = binomial)
##
## Deviance Residuals:
##       1        2        3        4        5        6
## -1.2332   0.4465   0.7601   1.8597   1.7152   2.1147
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## log_dose_new   0.5412     0.1776   3.048   0.0023 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25.969  on 6  degrees of freedom
## Residual deviance: 13.170  on 5  degrees of freedom
## AIC: 26.633
##
## Number of Fisher Scoring iterations: 4
```

Then we compute the log likelihood ratio statistic: $LR(4) = -2log\frac{L_{Small}}{L_{Large}} = (-2log(L_{Small})) - (-2log(L_{Large})) = D_{Small} - D_{Large}$
If the null hypothesis is true: $LR(4) = D_{Small} - D_{Large} \sim \chi^2_{l-s} = \chi^2_1$

```
# compute the log likelihood ratio statistic
LR4 = deviance(model.glm_sub) - deviance(model.glm); LR4
```

```
## [1] 11.59227
```

```
# compute the p-value
p_val = 1 - pchisq(LR4, 1); p_val
```

```
## [1] 0.0006622646
```

Answer:

The log likelihood ratio statistic LR(4) is 11.59227, and the p-value for the hypothesis test is 0.0006622646. Since p-value < 0.001, so we reject the null hypothesis (the restricted sub-model).