MATHEMATICAL STATISTICS

Semester 1	Generalized Linear Models - Assignment 2	2016

Due Tuesday, 1 Nov, 2016.

1. The maximum likelihood estimates for the parameters in the generalized linear model are obtained via an iterative weighted least squares algorithm with modified response variable

$$z_i = \eta_i + (y_i - \mu_i)/\frac{\partial \mu_i}{\partial \eta_i}$$

and weights given by

$$W_i = \frac{1}{V_i}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2$$

where $\mu_i = E(Y_i)$, $V_i = \mathrm{Var}(Y_i)$ and $\eta_i = \sum_{j=1}^{p} x_{ij}\beta_j$.

Let Y_i be a Poisson random variable with mean μ_i.

(a) Calculate the weight $\boldsymbol{W}$ for fitting the model

$$\ln \mu_i = \beta_0 + \beta_1 x_i, \quad i = 1, \ldots, n.$$

(b) Obtain an expression for the asymptotic variance of the maximum likelihood estimator for β_1.

2. The following data provide the times until subjects showed signs of motion sickness when placed in a cubical cabin mounted on a hydraulic piston and subjected to vertical motion for up to 120 minutes. The censored times are marked with an asterisk.

$$30,\ 50,\ 50^*,\ 51,\ 66^*, 82,\ 92,\ 120,\ 120^*,\ 120^*,\ 120^*,\ 120^*,\ 120^*.$$

(a) Find the Kaplan Meier (KM) estimator of $\widehat{S}(t)$ and plot $S(t)$ using R. Show your calculation of the probability $S(90)$ of surviving at least 90 minutes before showing signs of motion sickness.

(b) The variance of the KM estimator for the survival probability is given by

$$\mathrm{Var}[\widehat{S}(t)] \simeq [\widehat{S}(t)]^2 \sum_{j:t_j \le t} \frac{d_j}{r_j(r_j - d_j)}. \tag{1}$$

Prove the result using the following steps and apply the result to calculate the variance of the KM estimator in (a).

(i) Let X be a random variable with mean μ and variance σ^2. If $Y = f(X)$, where f is a differentiable function, show that $\text{Var}(Y) \simeq [f'(\mu)]^2\sigma^2$.

(ii) Let $\widehat{L}(t) = \ln[\widehat{S}(t)]$. Find an estimator for $\text{Var}[\widehat{L}(t)]$.

(iii) Hence show the result of $\text{Var}[\widehat{S}(t)]$ in (1).

3. Shouki and Pause (1999) report data from a study of hormonal therapy for the treatment of cattle with cystic ovarian disease. One group received the hormone treatment of cattle with cystic ovarian disease and the other a placebo. The time recorded was the time to the complete disappearance of all cysts. An asterisk denotes a right-censored time. The data were

Treatment group: 7, 12, 15, 16, 18, 22*
Control group: 19, 24, 18*, 20*, 22*, 27*, 30*.

(a) Estimate the hazard ratio for the treatment group using a proportional hazard model assuming that the survival time T follows a Weibull distribution with scale parameter λ and shape parameter α. Provide an approximate 95% confidence interval for the hazard ratio of the treatment group to control group. Plot the hazard and survival functions for the two groups.

(b) Fit a Cox's proportional hazards model to the data. Estimate the hazard ratio and an approximate 95% confidence interval for the hazard ratio. Note that when the number at risk for any group equals to zero, set it to be 0.001. Compare the result using the Cox's proportional hazard model model with the proportional hazard model in (a).

4. The following table reports the number of errors y_i out of t_i (in thousand) signals received from two transmitters. Let $g_i = 0, 1$ indicate the first and second transmitters respectively.

y_i	0	0	0	2	5	1	5	14	3	19	3	14	22
t_i	25	37	41	42	94	16	63	126	5	31	7	24	36
g_i	0	0	0	0	0	0	0	0	1	1	1	1	1

Conduct the following analyses.

(a) Fit the model

$$Y_i \sim \mathcal{P}(\lambda_i t_i) \quad \text{where} \quad \lambda_i = \exp(\beta_0 + \beta_1 g_i)$$

to the data using a `glm` module in R. Refit the model using negative binomial distribution. Compare the model fits.

(b) Suppose the information on the number of signal received t_i is unavailable, refit the models in (a). Compare the model fits and state the cause(s) for the discrepancy of model fits in (a) and (b).

(c) Suppose the information on the number of signal received t_i and the transmitter b_i are both unavailable and the *zero-inflated Poisson* model with parameters λ for the mean of Poisson distribution and π for the proportion of zeros is fitted to the data.

(i) Prove the result for moment estimators

$$\tilde{\lambda} = \frac{s^2 - \bar{y} + \bar{y}^2}{\bar{y}} \text{ and } \tilde{\pi} = \frac{s^2 - \bar{y}}{s^2 - \bar{y} + \bar{y}^2}$$

where $\bar{y}$ is the sample mean and s^2 is the sample variance. Use the result to find the moment estimates, $\tilde{\lambda}$ and $\tilde{\pi}$.

(ii) Prove the result for maximum likelihood (ML) estimators

$$\bar{y}\left(1 - e^{-\hat{\lambda}}\right) = \hat{\lambda}\left(1 - \frac{n_0}{n}\right) \text{ and } \hat{\pi} = 1 - \frac{\bar{y}}{\hat{\lambda}}$$

where n_0 is the number of zeros in the data, $\hat{\lambda}$ is the solution to the first equation and $\hat{\pi}$ depends on $\hat{\lambda}$. Use the R module `uniroot` to solve for the first equation to obtain $\hat{\lambda}$ and then calculate $\hat{\pi}$ using $\hat{\lambda}$.

5. The epilepsy data were collected from a clinical trial of 59 epileptics by Leppik et al. (1985). In the randomized controlled trial, $m = 59$ patients suffering from simple or complex partial seizures were assigned to either an antiepileptic drug progabide ($x_i = 1$) or a placebo ($x_i = 0$) with no intrinsic therapeutic value. The seizure counts were recorded at a two-week interval for an eight-week period ($n_i = 4$) with no dropout or missing cases.

Let $y_{ij}, i = 1, \ldots, 59;\ j = 1, \ldots, 4$ be the observed data. The covariates are `time` (1 to 4), `type` (of treatment $x_i = 0, 1$), `base` and `age`. The data are stored in `epilepsy.txt`. The two covariates `base` and `age` are tested to be insignificant. Three mixture Poisson regression models, namely 1-group mixture (M1), 2-group mixture with group-specific intercept (M2) and 2-group mixture with group-specific intercept and time effect (M3), are considered. The mean functions for these 3 models are given below:

$$\text{M1:} \quad \mu_{ij} = \exp(\beta_0 + \beta_1 \times j + \beta_2 x_i)$$

$$\text{M2:} \quad \begin{cases} \mu_{ij1} = \exp(\beta_{01} + \beta_1 \times j + \beta_2 x_i) \\ \mu_{ij2} = \exp(\beta_{02} + \beta_1 \times j + \beta_2 x_i) \end{cases}$$

$$\text{M3:} \quad \begin{cases} \mu_{ij1} = \exp(\beta_{01} + \beta_{11} \times j + \beta_2 x_i) \\ \mu_{ij2} = \exp(\beta_{02} + \beta_{12} \times j + \beta_2 x_i) \end{cases}$$

```
> dat=read.table("Epilepsy.txt",header = T)
> pat=dat[,1]; time=dat[,2]; y=dat[,3]; treatment=dat[,4]; base=dat[,5]; age=dat[,6]
```

(a) Provide summary statistics for the seizure counts across time for all patients and for patients in treatment and placebo groups.

(b) Fit the three models using `optim`. Report parameter estimates, standard error and AIC.

(c) Choose the best model according to AIC. Report the group memberships, group sizes and observed and fitted means across time for the four groups of high-low by treatment-placebo. Plot the observed and fitted means across time for these four groups. By comparing the observed mean and variance across time for the four groups, comment if other distribution will provide better model-fit.

Please attach your R commands.

Solution:

1. (a) For Poisson variable, $\theta_i = \eta_i = \ln \mu_i$ $\dfrac{\partial \eta_i}{\partial \mu_i} = \dfrac{1}{\mu_i}$ and $V_i = \mu_i$. Hence

$$W_i = \frac{1}{V_i}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 = \left[V_i\left(\frac{\partial \eta_i}{\partial \mu_i}\right)^2\right]^{-1} = \frac{\mu_i^2}{\mu_i} = \mu_i$$

and

$$\boldsymbol{W} = \begin{pmatrix} \mu_1 & 0 & \dots & 0 \\ 0 & \mu_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mu_n \end{pmatrix}$$

(b) For GLM, $\mathrm{Cov}(\widehat{\boldsymbol{\beta}}) \simeq (\boldsymbol{X'WX})^{-1}$.

$$\begin{aligned} \boldsymbol{X'WX} &= \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} \mu_1 & 0 & \dots & 0 \\ 0 & \mu_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mu_n \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^n \mu_i & \sum_{i=1}^n \mu_i x_i \\ \sum_{i=1}^n \mu_i x_i & \sum_{i=1}^n \mu_i x_i^2 \end{pmatrix} \\ (\boldsymbol{X'WX})^{-1} &= \frac{1}{\Big(\sum_{i=1}^n \mu_i\Big)\Big(\sum_{i=1}^n \mu_i x_i^2\Big) - \Big(\sum_{i=1}^n \mu_i x_i\Big)^2} \begin{pmatrix} \sum_{i=1}^n \mu_i x_i^2 & -\sum_{i=1}^n \mu_i x_i \\ -\sum_{i=1}^n \mu_i x_i & \sum_{i=1}^n \mu_i \end{pmatrix} \end{aligned}$$

Thus

$$\mathrm{Var}(\widehat{\beta}_1) \simeq \frac{\sum_{i=1}^n \mu_i}{\Big(\sum_{i=1}^n \mu_i\Big)\Big(\sum_{i=1}^n \mu_i x_i^2\Big) - \Big(\sum_{i=1}^n \mu_i x_i\Big)^2}$$

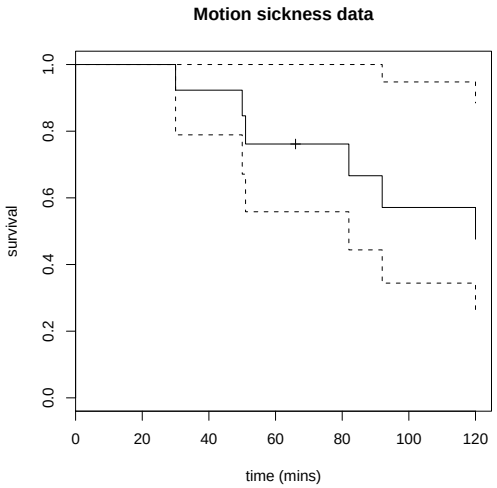
2. (a) The KM estimator for $S(90) = \Pr(T \geq 90)$ is

$$\begin{aligned} \widehat{S}(90) &= \prod_{j:t_j \leq 90} \left(1 - \frac{d_j}{r_j}\right) \\ &= \left(1 - \frac{1}{13}\right)\left(1 - \frac{1}{12}\right)\left(1 - \frac{1}{10}\right)\left(1 - \frac{1}{8}\right) \\ &= \frac{12}{13}\frac{11}{12}\frac{9}{10}\frac{7}{8} = \frac{693}{1040} = 0.6663 \end{aligned}$$

```
> t=c(30,50,50,51,66,82,92,120,120,120,120,120,120)
> w=c(1,1,0,1,0,1,1,1,0,0,0,0,0)
> surv=survfit(Surv(t,w)~1)
> plot(surv,xlab="time (mins)", ylab="survival",main="Motion sickness data")
> summary(surv)
Call: survfit(formula = Surv(t, w) ~ 1)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   30     13       1    0.923  0.0739        0.789        1.000
   50     12       1    0.846  0.1001        0.671        1.000
   51     10       1    0.762  0.1206        0.558        1.000
   82      8       1    0.666  0.1381        0.444        1.000
   92      7       1    0.571  0.1476        0.344        0.948
  120      6       1    0.476  0.1506        0.256        0.885
> t1=c(30,50,51,66,82,92,120)
> n=length(t1)
> d1=c(1,1,1,0,1,1,1)
> w1=c(0,1,0,1,0,0,5)
> sum(d1+w1)
[1] 13
> r1=c(13,12,10,9,8,7,6)
> h1=d1/r1
> S=cumprod(1-h1)
> v=d1/(r1*(r1-d1))
> se=S*sqrt(cumsum(v))
> result=cbind(t1,r1,d1,1-h1,S,se)
> result
      t1 r1 d1                   S        se
[1,]  30 13  1 0.9230769 0.9230769 0.0739053
[2,]  50 12  1 0.9166667 0.8461538 0.1000683
[3,]  51 10  1 0.9000000 0.7615385 0.1206435
[4,]  66  9  0 1.0000000 0.7615385 0.1206435
[5,]  82  8  1 0.8750000 0.6663462 0.1381030
[6,]  92  7  1 0.8571429 0.5711538 0.1475787
[7,] 120  6  1 0.8333333 0.4759615 0.1505853
```

Thus $\widehat{S}(90) = 0.666$ by the KM estimate of survival function.



(b) Use the Taylor's expansion around μ, we get

$$y = f(x) = f(\mu) + (x-\mu)f'(\mu) + \frac{1}{2}(x-\mu)^2 f''(\theta)$$

where θ lies between x and μ. Ignoring terms beyond the first 2, we have

$$\mathrm{Var}(Y) \simeq [f'(\mu)]^2 \mathrm{Var}(X) = \sigma^2 [f'(\mu)]^2.$$

Hence, writing $f_1(\hat{h}_j) = \ln(1-\hat{h}_j)$, we have $f_1'(\hat{h}_j) = \dfrac{-1}{1-\hat{h}_j}$ and

$$\begin{aligned}
\mathrm{Var}[\widehat{L}(t)] &= \mathrm{Var}[\ln \widehat{S}(t)] = \mathrm{Var}[\sum_{j:t_j \le t} \ln(1-\hat{h}_j)] \\
&\simeq \sum_{j:t_j \le t} \left(\frac{1}{1-\hat{h}_j}\right)^2 \mathrm{Var}(\hat{h}_j) \\
&= \sum_{j:t_j \le t} \left(\frac{r_j}{r_j - d_j}\right)^2 \frac{\hat{p}_j(1-\hat{p}_j)}{r_j} \quad \text{since } \hat{h}_j = \frac{d_j}{r_j} \text{ is a proportion} \\
&= \sum_{j:t_j \le t} \frac{r_j^2}{(r_j-d_j)^2} \frac{d_j(r_j-d_j)}{r_j \cdot r_j^2} \\
&= \sum_{j:t_j \le t} \frac{d_j}{(r_j-d_j)\, r_j}.
\end{aligned}$$

Since $\widehat{S}(t) = \exp[\widehat{L}(t)] = f_2[\widehat{L}(t)]$, $\widehat{S}'(t) = -\exp[\widehat{L}(t)] = f_2'[\widehat{L}(t)]$. Then

$$\mathrm{Var}[\widehat{S}(t)] \simeq \{\exp[\widehat{L}(t)]\}^2 \mathrm{Var}[\widehat{L}(t)] = [\widehat{S}(t)]^2 \sum_{j:t_j \le t} \frac{d_j}{(r_j-d_j)\, r_j}.$$

Hence the variance for $\widehat{S}(90)$ is

$$\begin{aligned}\mathrm{Var}[\widehat{S}(90)] &\simeq [\widehat{S}(90)]^2 \sum_{j:t_j \le 90} \frac{d_j}{(r_j - d_j)\, r_j} \\ &= \frac{693^2}{1040^2}\left[\frac{1}{13(13-1)} + \frac{1}{12(12-1)} + \frac{1}{10(10-1)} + \frac{1}{8(8-1)}\right] \\ &= 0.01907\end{aligned}$$

Note that $se(\widehat{\beta}_1) = \sqrt{0.01907} = 0.138103$ as given from `summary(surv)`.

3. (a) For the proportional hazard model $h(t, \boldsymbol{x}) = \lambda^\alpha \alpha t^{\alpha-1} \exp(\boldsymbol{x}'\boldsymbol{\beta})$ when the survival time $T \sim \mathcal{W}ei(\alpha, \lambda)$, the output using R is

```
> t=c(7,12,15,16,18,22,19,24,18,20,22,27,30)
> w=c(1,1,1,1,1,0,1,1,0,0,0,0,0)
> c=factor(c(rep(1,6),rep(0,7)))
> l=log(t)
> sur1=survreg(Surv(t,w)~c,dist="weibull")
> summary(sur1)

Call:
survreg(formula = Surv(t, w) ~ c, dist = "weibull")
             Value Std. Error     z        p
(Intercept)  3.535      0.230 15.35 3.80e-53
c1          -0.677      0.272 -2.49 1.27e-02
Log(scale)  -1.225      0.316 -3.88 1.04e-04

Scale= 0.294

Weibull distribution
Loglik(model)= -25.6   Loglik(intercept only)= -29.4
        Chisq= 7.49 on 1 degrees of freedom, p= 0.0062
Number of Newton-Raphson Iterations: 6
n= 13

> alpha=1/sur1$scale
> par=sur1$coeff
> beta0=-par[1]*alpha
> beta1=-par[2]*alpha
> c(alpha,beta0,beta1)
            (Intercept)          c1
```

```
   3.404920  -12.035809    2.304355
> time=seq(0,30,0.1)
> haz1=alpha*time^(alpha-1)*exp(beta0)
> haz2=alpha*time^(alpha-1)*exp(beta0+beta1)
> surv1=exp(-time^alpha*exp(beta0))
> surv2=exp(-time^alpha*exp(beta0+beta1))
>
> par(mfrow=c(2,2))
> plot(time,haz1,xlab="time", ylab="hazard",xlim=c(0,30),ylim=c(0,0.8),
  col="blue",cex=0.2,main="Hazard functions")
> points(time,haz2,pch=20,col='red',cex=0.2)
> plot(time,surv1,xlab="time", ylab="survival",xlim=c(0,30),ylim=c(0,1),
  col="blue",cex=0.2,main="Survival functions")
> points(time,surv2,pch=20,col='red',cex=0.2)
> covm=vcov(sur1)
> covm
             (Intercept)          c1  Log(scale)
(Intercept)  0.05306119 -0.05464474  0.03146429
c1          -0.05464474  0.07373180 -0.03648016
Log(scale)   0.03146429 -0.03648016  0.09966286
> se=alpha*sqrt(covm[2,2]-2*par[2]*covm[2,3]+par[2]^2*covm[3,3])
> se
       c1
0.9008695
> CI=c(beta1-1.96*se,beta1,beta1+1.96*se)
> CI
       c1        c1        c1
0.5386506 2.3043549 4.0700592
> exp(CI)
       c1        c1        c1
 1.713693 10.017714 58.560430
```

We have $\alpha = 1/0.294 = 3.404920$, $\beta_0 = -3.535/0.294 = -12.035809$ and $\beta_1 = 0.677/0.294 = 2.304355$. Since $\beta_1 = -\rho \times \alpha = -\rho \times \exp(-\gamma) = f(\rho, \gamma)$ where ρ is

the parameter for the treatment effect and $\gamma = \ln(1/\alpha)$ is the log(scale) from `survreg`,

$$\begin{aligned}
\mathrm{Var}(\beta_1) &= \begin{pmatrix} \frac{\partial\beta_1}{\partial\rho} & \frac{\partial\beta_1}{\partial\gamma} \end{pmatrix} \begin{pmatrix} \mathrm{Var}(\rho) & \mathrm{Cov}(\rho,\gamma) \\ \mathrm{Cov}(\rho,\gamma) & \mathrm{Var}(\gamma) \end{pmatrix} \begin{pmatrix} \frac{\partial\beta_1}{\partial\rho} \\ \frac{\partial\beta_1}{\partial\gamma} \end{pmatrix} \\
&= (-\alpha\ \rho\alpha) \begin{pmatrix} \mathrm{Var}(\rho) & \mathrm{Cov}(\rho,\gamma) \\ \mathrm{Cov}(\rho,\gamma) & \mathrm{Var}(\gamma) \end{pmatrix} \begin{pmatrix} -\alpha \\ \rho\alpha \end{pmatrix} \\
&= \alpha^2\mathrm{Var}(\rho) - 2\rho\alpha^2\mathrm{Cov}(\rho,\gamma) + \rho^2\alpha^2\mathrm{Var}(\gamma) \\
&= \alpha^2[\mathrm{Var}(\rho) - 2\rho\mathrm{Cov}(\rho,\gamma) + \rho^2\mathrm{Var}(\gamma)] \\
&= 3.404920^2[0.07373180 - 2(-0.6767721)(-0.03648016) + (-0.6767721)^2(0.09966286)] \\
&= 0.8115659
\end{aligned}$$

where

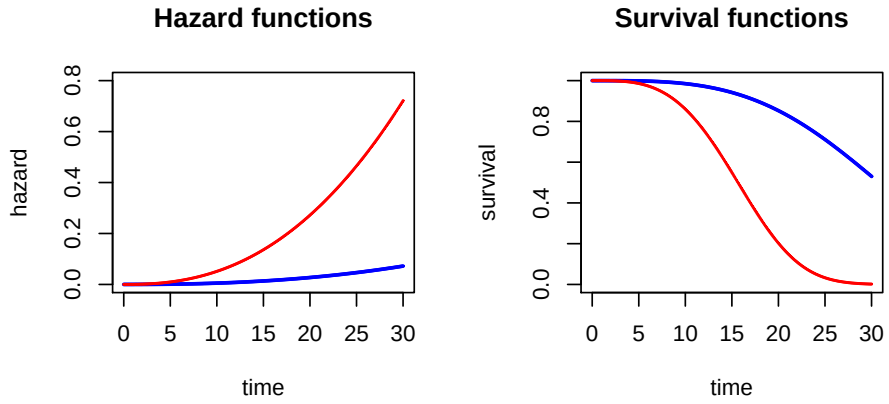
$$\frac{\partial\beta_1}{\partial\rho} = -\exp(-\gamma) = -\alpha, \qquad \frac{\partial\beta_1}{\partial\gamma} = \rho\exp(-\gamma) = \rho\alpha$$

and $\alpha = \exp(-\gamma)$. with s.e. $\sqrt{0.8115659} = 0.9008695$. A 95% CI for β_1 is

$$2.304355 \pm 1.96 \times 0.9008695 = (0.5386506,\ 4.0700592)$$

Hence the estimated hazard ratio is $\exp(\beta) = \exp(2.304355) = 10.017714$ and a 95% CI for $\exp(\beta)$, the hazard ratio is

$$(\exp(0.5386506),\ \exp(4.0700592)) = (1.713693,\ 58.560430)$$



If you swap the labels for treatment and control groups, the results are

```
> c=factor(c(rep(0,6),rep(1,7)))  #swap label
...
> summary(sur1)

Call:
```

```
survreg(formula = Surv(t, w) ~ c, dist = "weibull")
             Value Std. Error     z         p
(Intercept)  2.858      0.132 21.60 1.69e-103
c1           0.677      0.272  2.49  1.27e-02
Log(scale)  -1.225      0.316 -3.88  1.04e-04

Scale= 0.294

Weibull distribution
Loglik(model)= -25.6   Loglik(intercept only)= -29.4
        Chisq= 7.49 on 1 degrees of freedom, p= 0.0062
Number of Newton-Raphson Iterations: 6
n= 13

> covm
             (Intercept)          c1   Log(scale)
(Intercept)  0.017503516 -0.01908706 -0.005015873
c1          -0.019087063  0.07373180  0.036480160
Log(scale)  -0.005015873  0.03648016  0.099662862
> se=alpha*sqrt(covm[2,2]-2*par[2]*covm[2,3]+par[2]^2*covm[3,3])
> se
       c1
0.9008695
> CI=c(beta1-1.96*se,beta1,beta1+1.96*se)
> CI
        c1         c1         c1
-4.0700592 -2.3043549 -0.5386506
> exp(CI)
        c1         c1         c1
0.01707638 0.09982318 0.58353515
```

(b) As compare with the proportional hazard model, Cox proportional hazard model does not assume any distribution for T.

To code the data for Cox regression focus on the treatment group, the failure times t_j are

$$7,\ 12,\ 15,\ 16,\ 18,\ 19,\ 24$$

Let d_j be the number of failure at time t_j,
d_{1j} be the number of failure from the treatment group,
r_{1j} be the number at risk from treatment group at time t_j and

r_{2j} be the number at risk in the control group. Then

Time t_j	7	12	15	16	18	19	24
d_{1j}	1	1	1	1	1	0	0
d_{2j}	0	0	0	0	0	1	1
r_{1j}	6	5	4	3	2	1	0
r_{2j}	7	7	7	7	7	6	3

```
> d1=c(1,1,1,1,1,0,0)
> d=c(1,1,1,1,1,1,1)
> d2=d-d1
> r1=c(6,5,4,3,2,1,0.001)
> r2=c(7,7,7,7,7,6,3)
> l=log(r1/r2)
> dd=cbind(d1,d2)
> coxp=glm(dd~offset(l),family=binomial)
> summary(coxp)

Call:
glm(formula = dd ~ offset(l), family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5036   0.1648   0.4594   0.5469   0.7007

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.532      1.118   2.265   0.0235 *
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3.7492  on 6  degrees of freedom
Residual deviance: 3.7492  on 6  degrees of freedom
AIC: 5.7492

Number of Fisher Scoring iterations: 5

> beta=coxp$coeff
```

```
> var=vcov(coxp)[1]
> se=sqrt(var)
> CI=c(beta-1.96*se,beta,beta+1.96*se)
> CI
(Intercept) (Intercept) (Intercept)
  0.3408433   2.5321168   4.7233903
> exp(CI)
(Intercept) (Intercept) (Intercept)
   1.406133   12.580107  112.549185
```

The estimated hazard ratio is $\exp(\widehat{\beta}) = \exp(2.532) = 12.57864$ so the treatment group is 12.64 times more likely than the control group to be clear of cysts t days after commencing treatment.

A 95% CI for β is

$$2.579 \pm 1.96 \times 1.118 = (0.34072,\ 4.72328)$$

Hence a 95% CI for $\exp(\beta)$, the hazard ratio is

$$(\exp(0.34072),\ \exp(4.72328)) = (1.40596,\ 112.5368)$$

An alternative way is

```
> t=c(7,12,15,16,18,22,19,24,18,20,22,27,30)
> w=c(1,1,1,1,1,0,1,1,0,0,0,0,0)
> c=factor(c(rep(1,6),rep(0,7)))
> summary(coxph( Surv(t,w)~c))
Call:
coxph(formula = Surv(t, w) ~ c)

  n= 13

     coef exp(coef) se(coef)    z Pr(>|z|)
c1  2.537    12.646    1.123 2.26   0.0238 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

   exp(coef) exp(-coef) lower .95 upper .95
c1     12.65    0.07907     1.401     114.2

Rsquare= 0.43   (max possible= 0.9 )
```

```
Likelihood ratio test= 7.3  on 1 df,   p=0.00688
Wald test            = 5.11  on 1 df,   p=0.02382
Score (logrank) test = 7.79  on 1 df,   p=0.005256
```

So the hazard ratio and 95% CI are 12.65 and (1.401, 114.2) respectively. These results are preferred as there is no approximation of $0 \simeq 0.001$. Comparing the two methods, the estimates of hazard ratio $\exp(\beta)$ are 2.532 and 2.537 and their s.e. are 1.118 and 1.123 using `glm` and `coxph` respectively. They are similar and are also close to 2.3043549 using Weibull model but the CI of (1.713693, 58.560430) shows that the Weibull model gives more a precise estimate.

If one swaps the labels, the hazard ratio and 95% CI are given by their reciprocals.

4. (a) The R codes are:

```
> library(MASS)
> y=c(0,0,0,2,5,1,5,14,3,19,3,14,22)
> t=c(25,37,41,42,94,16,63,126,5,31,7,24,36)
> g=c(rep(0,8),rep(1,5))
> mean(y);var(y)
[1] 6.769231
[1] 59.52564
> lt=log(t)
> m1=glm(y~offset(lt)+g,family=poisson)
> summary(m1)

Call:
glm(formula = y ~ offset(lt) + g, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2330  -0.5921  -0.0568   0.1464   2.0498

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.8000     0.1925 -14.549   <2e-16 ***
g             2.2761     0.2312   9.847   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

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

```
    Null deviance: 124.138  on 12  degrees of freedom
Residual deviance:  17.676  on 11  degrees of freedom
AIC: 57.94

Number of Fisher Scoring iterations: 5

> m2=glm.nb(y~offset(lt)+g)
Warning messages:
1: In theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace = control$trace >  :
  iteration limit reached
2: In theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace = control$trace >  :
  iteration limit reached
> summary(m2)

Call:
glm.nb(formula = y ~ offset(lt) + g, init.theta = 86681.7163,
    link = log)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.23301  -0.59210  -0.05678   0.14641   2.04973

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.8000     0.1925 -14.549   <2e-16 ***
g             2.2761     0.2312   9.847   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for Negative Binomial(86681.72) family taken to be 1)

    Null deviance: 124.128  on 12  degrees of freedom
Residual deviance:  17.675  on 11  degrees of freedom
AIC: 59.94

Number of Fisher Scoring iterations: 1
```

```
              Theta:  86682
          Std. Err.:  2544075
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -53.94
```

The fit of Negative binomial (NB) distribution is 59.94 according to AIC showing an increase by 2 (57.94) of Poisson distribution and hence a worse model fit. The large estimated shape parameter $r = 86682$ of NB distribution shows that Poisson is a suitable data distribution and the increase of 2 in the AIC is due to the extra shape parameter r in NB distribution. This is because the data is under-dispersed after allowing for t_i.

```
> r=y/t
> c(mean(r),var(r))
[1] 0.24536199 0.07317279
```

(b) Dropping the information of t_i,

```
> summary(glm(y~g,family=poisson))

Call:
glm(formula = y ~ g, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1596  -2.5981  -0.8106   0.8249   4.3110

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.2164     0.1925   6.321 2.61e-10 ***
g             1.2850     0.2312   5.559 2.71e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 107.246  on 12  degrees of freedom
Residual deviance:  72.966  on 11  degrees of freedom
AIC: 113.23

Number of Fisher Scoring iterations: 6
```

```
> c(mean(y),var(y))
[1]  6.769231 59.525641
> summary(glm.nb(formula = y ~ g, link = log))

Call:
glm.nb(formula = y ~ g, link = log, init.theta = 0.9387015914)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6921  -1.0359  -0.4036   0.3651   1.6499

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.2164     0.4126   2.948  0.00319 **
g             1.2850     0.6322   2.033  0.04208 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for Negative Binomial(0.9387) family taken to be 1)

    Null deviance: 19.423  on 12  degrees of freedom
Residual deviance: 15.151  on 11  degrees of freedom
AIC: 79.027

Number of Fisher Scoring iterations: 1

              Theta:  0.939
          Std. Err.:  0.489

 2 x log-likelihood:  -73.027
```

Now, the fit NB distribution is 79.027 according to AIC showing a drop as compared to 113.23 of Poisson distribution. Hence the model fit improves substantially. The much lower estimated shape parameter $r = 0.9387$ of NB distribution shows that NB is a suitable data distribution. This is because the data is over-dispersed without allowing for t_i.

```
> c(mean(y),var(y))
[1]  6.769231 59.525641
```

(c) Zero-inflated Poisson model:

(i) We have shown that $E(Y) = \lambda(1-\pi) = \bar{y}$ and $Var(Y) = \lambda(1-\pi)(1+\lambda\pi) = s^2$. Then we equate them to sample estimates to obtain the moment estimators. Hence

$$\begin{aligned}
\bar{y} &= \lambda(1-\pi) \\
s^2 &= \lambda(1-\pi)(1+\lambda\pi) = \bar{y}\left(1 + \frac{\bar{y}}{1-\pi}\pi\right) \\
\Rightarrow s^2(1-\pi) &= \bar{y}(1-\pi+\bar{y}\pi) \\
\Rightarrow s^2 - \bar{y} &= (s^2 - \bar{y} + \bar{y}^2)\pi \\
\tilde{\pi} &= \frac{s^2-\bar{y}}{s^2-\bar{y}+\bar{y}^2} \\
\tilde{\lambda} &= \frac{\bar{y}}{1-\pi} = \frac{\bar{y}}{1-\frac{s^2-\bar{y}}{s^2-\bar{y}+\bar{y}^2}} = \frac{\bar{y}(s^2-\bar{y}+\bar{y}^2)}{s^2-\bar{y}+\bar{y}^2-(s^2-\bar{y})} = \frac{s^2-\bar{y}+\bar{y}^2}{\bar{y}}
\end{aligned}$$

To calculate the moment estimates,

```
> n=length(y)
> n0=length(y[y==0])
> m=mean(y)
> v=var(y)
> lamh=(v+m^2-m)/m
> ph=(v-m)/(v+m^2-m)
> c(lamh,ph)  #moment est
[1] 14.5627914  0.5351694
```

Hence $\tilde{\lambda} = 14.5627914$ and $\tilde{\pi} = 0.5351694$.

(ii) For the ML estimator, let n_0 denote the number of zero and $n_1 = n - n_0$. We have

$$L = \prod_{i=1}^{n_0}[\pi + (1-\pi)e^{-\lambda}] \prod_{i=n_0+1}^{n}\left[(1-\pi)\frac{e^{-\lambda}\lambda_i^y}{y_i!}\right]$$

Note that in the first product for zero observations, it should NOT be

$$\prod_{i=1}^{n_0}\pi^{I_{i0}}[(1-\pi)e^{-\lambda}]^{1-I_{i0}}$$

where I_{i0} is the indicator for the i-th zero in the degenerated zero group because I_{i0} is unobserved. The two likelihoods $\prod_{i=1}^{n_0}[\pi + (1-\pi)e^{-\lambda}]$ and $\prod_{i=1}^{n_0}\pi^{I_{i0}}[(1-\pi)e^{-\lambda}]^{1-I_{i0}}$ are called observed and complete data likelihood for the zero observations. Then the log-likelihood is

$$\ell = n_0 \ln[\pi + (1-\pi)e^{-\lambda}] + n_1 \ln(1-\pi) + \ln\lambda \sum_{i=n_0+1}^{n} y_i - n_1\lambda - \sum_{i=n_0+1}^{n} \ln y_i!$$

$$
\begin{aligned}
\frac{\partial \ell}{\partial \pi} &= \frac{n_0(1-e^{-\lambda})}{\pi+(1-\pi)e^{-\lambda}} - \frac{n_1}{1-\pi} = 0 \\
\Rightarrow n_0(1-e^{-\lambda})(1-\pi) &= n_1(\pi+(1-\pi)e^{-\lambda}) \\
n_0(1-e^{-\lambda}) &= n_1\left(\frac{\pi}{1-\pi}+e^{-\lambda}\right) \\
\frac{\pi}{1-\pi} &= \frac{n_0}{n_1}(1-e^{-\lambda}) - e^{-\lambda} = \frac{n_0}{n_1} - \frac{n}{n_1}e^{-\lambda} \qquad (2) \\
n_1\pi &= (n_0 - ne^{-\lambda})(1-\pi) = n_0 - n_0\pi - ne^{-\lambda} + ne^{-\lambda}\pi \\
\pi(n-ne^{-\lambda}) &= n_0 - ne^{-\lambda} \\
\pi &= \frac{n_0 - ne^{-\lambda}}{n-ne^{-\lambda}} = \frac{n-ne^{-\lambda}-n_1}{n-ne^{-\lambda}} \\
&= 1 - \frac{n_1}{n(1-e^{-\lambda})} = 1 - \frac{\bar{y}}{\lambda} \text{ using (3) below}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \ell}{\partial \lambda} = \frac{-n_0(1-\pi)e^{-\lambda}}{\pi+(1-\pi)e^{-\lambda}} + \frac{1}{\lambda}\sum_{i=n_0+1}^{n} y_i - n_1 &= 0 \\
\Rightarrow -n_0(1-\pi)e^{-\lambda}\lambda + n\bar{y}(\pi+(1-\pi)e^{-\lambda}) &= n_1\lambda(\pi+(1-\pi)e^{-\lambda}) \\
n\bar{y}\pi + n\bar{y}(1-\pi)e^{-\lambda} &= n_1\lambda\pi + n\lambda(1-\pi)e^{-\lambda} \\
(\bar{y}-\lambda)(1-\pi)e^{-\lambda} &= \pi\left(\frac{n_1}{n}\lambda - \bar{y}\right) \\
(\bar{y}-\lambda)e^{-\lambda} &= \left(\frac{n_0}{n_1} - \frac{n}{n_1}e^{-\lambda}\right)\left(\frac{n_1}{n}\lambda - \bar{y}\right) \text{ using (2)} \\
\bar{y}e^{-\lambda} - \lambda e^{-\lambda} &= \frac{n_0}{n}\lambda - \frac{n_0}{n_1}\bar{y} - \lambda e^{-\lambda} + \frac{n}{n_1}e^{-\lambda}\bar{y} \\
\bar{y}(e^{-\lambda} - \frac{n}{n_1}e^{-\lambda} + \frac{n_0}{n_1}) &= \frac{n_0}{n}\lambda \\
\bar{y}\frac{n_0}{n_1}\left(1-e^{-\lambda}\right) &= \lambda\frac{n_0}{n} \\
\bar{y}\left(1-e^{-\lambda}\right) &= \lambda\left(1-\frac{n_0}{n}\right) \Rightarrow \frac{\bar{y}}{\lambda} = \frac{n_1}{n(1-e^{-\lambda})} \qquad (3)
\end{aligned}
$$

Hence

$$
\bar{y}\left(1-e^{-\hat{\lambda}}\right) = \hat{\lambda}\left(1-\frac{n_0}{n}\right) \text{ and } \hat{\pi} = 1 - \frac{\bar{y}}{\hat{\lambda}}
$$

To calculate the ML estimates,

```
> a=m
> b=1-n0/n
> f=function(x) a*(exp(-x)-1)+b*x
> lamh=uniroot(f,lower=1,upper=50)$root
> ph=1-m/lamh
> c(lamh,ph) #ML est
[1] 8.7986718 0.2306531
```

Hence $\hat{\lambda} = 8.7986718$ and $\hat{\pi} = 0.2306531$.

For the hurdle model, the loglikelihood function is

$$\ell(\pi, \lambda) = n_0 \ln \pi + n_1 \ln(1-\pi) + \ln \lambda \sum_{i=n_0+1}^{n} y_i - n_1\lambda - \sum_{i=n_0+1}^{n} y_i! - n_1 \ln(1-e^{-\lambda}).$$

Hence we have

$$\begin{aligned}
\frac{\partial \ell}{\partial \pi} &= \frac{n_0}{\pi} - \frac{n_1}{1-\pi} = 0 \Rightarrow \pi = \frac{n_0}{n} \\
\frac{\partial \ell}{\partial \lambda} &= \frac{1}{\lambda} \sum_{i=n_0+1}^{n} y_i - n_1 - \frac{n_1 e^{-\lambda}}{(1-e^{-\lambda})} = 0 \\
\Rightarrow & \quad \sum_{i=n_0+1}^{n} y_i = n_1\lambda + \frac{n_1 \lambda e^{-\lambda}}{(1-e^{-\lambda})} \\
\Rightarrow & \quad n\bar{y}(1-e^{-\lambda}) = \lambda n_1 (1-e^{-\lambda}) + \lambda n_1 e^{-\lambda} \\
\Rightarrow & \quad n\bar{y}(1-e^{-\lambda}) = \lambda n_1 \Rightarrow \bar{y}(1-e^{-\lambda}) = \lambda \left(1 - \frac{n_0}{n}\right)
\end{aligned}$$

where $\bar{y}$ refers to overall mean.

5. (a) Basic descriptive statistics:

```
> dat=read.table("Epilepsy.txt",header = T)
> pat=dat[,1]; time=dat[,2]; w=dat[,3]; treatment=dat[,4]; base=dat[,5]; age=dat[,6]
> t=max(time)
> n=nrow(dat)
> m=nrow(dat)/t
> y=w
> ym=matrix(0,m,t)
> for (i in 1:m) ym[i,]=y[pat==i]
> t(ym)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,]    5    3    2    4    7    5    6   40    5    14    26    12     4     7
[2,]    3    5    4    4   18    2    4   20    6    13    12     6     4     9
[3,]    3    3    0    1    9    8    0   23    6     6     6     8     6    12
[4,]    3    3    5    4   21    7    2   12    5     0    22     5     2    14
     [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26]
[1,]    16    11     0    37     3     3     3     3     2     8    18     2
[2,]    24     0     0    29     5     0     4     4     3    12    24     1
[3,]    10     0     3    28     2     6     3     3     3     2    76     2
[4,]     9     5     3    29     5     7     4     4     5     8    25     1
     [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
[1,]     3    13    11     8     0     3     2     4    22     5     2     3
```

```
[2,]     1    15    14     7     4     6     6     3    17     4     4     7
[3,]     4    13     9     9     3     1     7     1    19     7     0     7
[4,]     2    12     8     4     0     3     4     3    16     4     4     7
     [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]
[1,]     4     2     0     5    11    10    19     1     6     2   102     4
[2,]    18     1     2     4    14     5     7     1    10     1    65     3
[3,]     2     1     4     0    25     3     6     2     8     0    72     2
[4,]     5     0     0     3    15     8     7     4     8     0    63     4
     [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59]
[1,]     8     1    18     6     3     1     2     0     1
[2,]     6     3    11     3     5    23     3     0     4
[3,]     5     1    28     4     4    19     0     0     3
[4,]     7     5    13     0     3     8     1     0     2
> ind=rep(0,m)
> mu1=matrix(0,2,t)
> mu2=matrix(0,2,t)
> typem=type[4*(1:m-1)+1]+1
> tc=unname(table(typem))
> yt1=ym[1:tc[1],]
> yt2=ym[(tc[1]+1):m,]
> ytm=apply(ym,2,mean)
> yt1m=apply(yt1,2,mean)
> yt2m=apply(yt2,2,mean)
> ytv=apply(ym,2,var)
> yt1v=apply(yt1,2,var)
> yt2v=apply(yt2,2,var)
> rbind(ytm,yt1m,yt2m,ytv,yt1v,yt2v)
           [,1]       [,2]       [,3]       [,4]
ytm    8.949153   8.355932   8.440678   7.338983
yt1m   9.357143   8.285714   8.785714   8.000000
yt2m   8.580645   8.419355   8.129032   6.741935
ytv  220.083577 103.784921 200.181765  92.883109
yt1v 102.756614  66.656085 215.285714  57.925926
yt2v 332.718280 140.651613 193.049462 126.664516
```

From the descriptive statistics, the epilepsy count decrease for treatment and across time. Hence we expect $\beta_1, \beta_2 < 0$.

(b) Fit three models:

```
> l=c()
```

```
> #M1 1gp
> logl1 <- function(pars) {
+ b01=pars[1]; b1=pars[2]; b2=pars[3]  #1gp
+ mu1=exp(b01+b1*time+b2*treatment)
+ for (i in 1:n) {
+ l[i]=prod(exp(-mu1[i])*mu1[i]^y[i]/factorial(y[i]))  #1gp
+ }
+ ll=sum(log(l))
+ return(-ll)
+ }
>
> mixreg1=optim(c(1.6,-0.01,-0.3),logl1,method = "BFGS",hessian=T) #1gp
> OIm1<-solve(mixreg1$hessian)
> sem1<-sqrt(diag(OIm1))
> parm1=mixreg1$par
> rbind(parm1,sem1)
            [,1]        [,2]        [,3]
parm1 2.29399393 -0.05739791 -0.07715474
sem1  0.05882923  0.02027285  0.04527167
> AICm1=mixreg1$value*2+length(parm1)*2
> AICm1
[1] 3280.348
>
> l1=c()
> l2=c()
> #M2 2gp common time & type
> logl2 <- function(pars) {
+ b01=pars[1]; b1=pars[2]; b2=pars[3]; b02=pars[4]; pi=pars[5]
+ mu1=exp(b01+b1*time+b2*treatment) #2gp common time & type
+ mu2=exp(b02+b1*time+b2*treatment)
+ l1i=exp(-mu1)*mu1^y/factorial(y)
+ l2i=exp(-mu2)*mu2^y/factorial(y)
+
+ for (i in 1:m) {
+ l1[i]=prod(l1i[((i-1)*4+1):(i*4)])
+ l2[i]=prod(l2i[((i-1)*4+1):(i*4)])
+ l[i]=pi*l1[i]+(1-pi)*l2[i]
+ }
+ ll=sum(log(l))
```

```
+ return(-ll)
+ }
>
> mixreg2=optim(c(1.5,-0.057,-0.08,2.5,0.7),logl2,method = "BFGS",hessian=T)
There were 17 warnings (use warnings() to see them)
> OIm2<-solve(mixreg2$hessian)
> sem2<-sqrt(diag(OIm2))
> parm2=mixreg2$par
> rbind(parm2,sem2)
            [,1]        [,2]       [,3]       [,4]       [,5]
parm2 1.40971520 -0.05741594 0.30803576 3.20400595 0.79767884
sem2  0.07211718  0.02027329 0.05215058 0.06389241 0.05254764
> AICm2=mixreg2$value*2+length(parm2)*2
> AICm2
[1] 1927.981
>
> l1=c()
> l2=c()
> #M3 2gp common type only
> logl3 <- function(pars) {
+ b01=pars[1]; b11=pars[2]; b2=pars[3]; b02=pars[4]; b12=pars[5]; pi=pars[6]
+ mu1=exp(b01+b11*time+b2*treatment) #2gp common type only
+ mu2=exp(b02+b12*time+b2*treatment)
+ l1i=exp(-mu1)*mu1^y/factorial(y)
+ l2i=exp(-mu2)*mu2^y/factorial(y)
+
+ for (i in 1:m) {
+ l1[i]=prod(l1i[((i-1)*4+1):(i*4)])
+ l2[i]=prod(l2i[((i-1)*4+1):(i*4)])
+ l[i]=pi*l1[i]+(1-pi)*l2[i]
+ }
+ ll=sum(log(l))
+ return(-ll)
+ }
>
> mixreg3=optim(c(1.5,-0.057,-0.08,2.5,-0.04,0.7),logl3,method = "BFGS",hessian=T) #
Warning messages: ...
> OIm3<-solve(mixreg3$hessian)
> sem3<-sqrt(diag(OIm3))
```

```
> parm3=mixreg3$par
> rbind(parm3,sem3)
            [,1]        [,2]       [,3]       [,4]        [,5]      [,6]
parm3 1.44139842 -0.07141155 0.30996937 3.17766644 -0.04735096 0.7973734
sem3  0.08973745  0.03175831 0.04982183 0.07639458  0.02655026 0.0524565
> AICm3=mixreg3$value*2+length(parm3)*2
> AICm3
[1] 1929.648
```

The AIC for M1-3 are respectively 3280.348, 1927.981 and 1929.648. Hence M2 is chosen and the treatment effect β_2 changes from -0.0771 (1 gp: not sign) to 0.3080 (2 gp: sign).

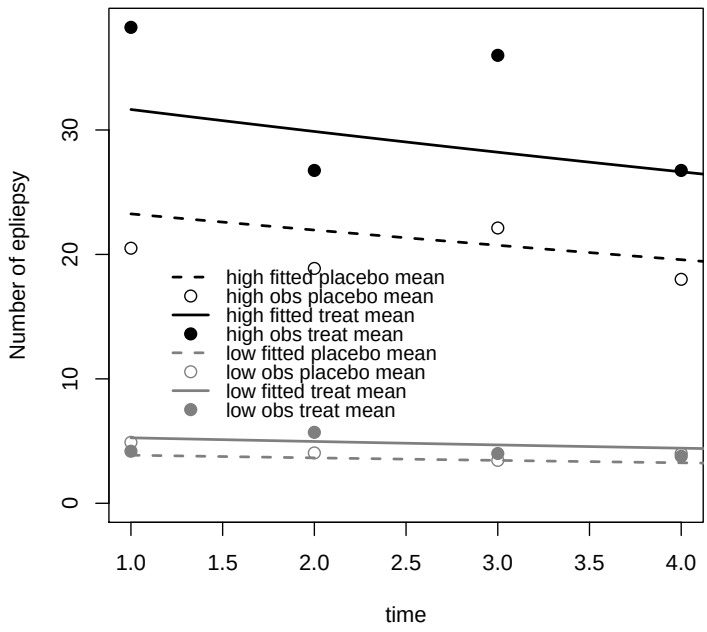
(c) Based on M2, 47 and 11 patients are assigned to low and high level groups respectively and 27 and 4 patients in the 2 groups received treatment. For both low and high level groups, the mean counts in treatment groups are higher than those in placebo groups and hence the β_2 for treatment effect become positive.

```
> b01=parm2[1]; b1=parm2[2]; b2=parm2[3]; b02=parm2[4]; pi=parm2[5]  #M2 chosen
> mu1=exp(b01+b1*time+b2*treatment)
> mu2=exp(b02+b1*time+b2*treatment)
> l1i=exp(-mu1)*mu1^y/factorial(y)
> l2i=exp(-mu2)*mu2^y/factorial(y)
>
> for (i in 1:m) {
+ l1[i]=prod(l1i[((i-1)*4+1):(i*4)])
+ l2[i]=prod(l2i[((i-1)*4+1):(i*4)])
+ ind[i]=pi*l1[i]/(pi*l1[i]+(1-pi)*l2[i])
+ }
> ind=round(ind,0)
> ind
 [1] 1 1 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1
[39] 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1
> wt=sum(ind)/m
> wt #agree with pi estimate
[1] 0.7966102
> mi=1:m
> gp1=mi[ind==1]; gp2=mi[ind==0]
> gp1
 [1]  1  2  3  4  6  7  9 10 12 13 16 17 19 20 21 22 23 24 26 27 29 30 31 32 33
[26] 34 36 37 38 39 40 41 42 44 45 46 47 48 50 51 52 54 55 56 57 58 59
```

```
> gp2
 [1]  5  8 11 14 15 18 25 28 35 43 49 53
> treatmenti=treatment[time==1]
> gp11=mi[ind==1 & treatmenti==0]; gp12=mi[ind==1 & treatmenti==1];
> gp21=mi[ind==0 & treatmenti==0]; gp22=mi[ind==0 & treatmenti==1]
> n1=length(gp1); n2=length(gp2); n11=length(gp11); n12=length(gp12);
> n21=length(gp21); n22=length(gp22)
> N=c(n1,n2,n11,n12,n21,n22)
> N
[1] 47 12 20 27  8  4
> p=c(n12/n1,n22/n2)
> p
[1] 0.5744681 0.3333333
> y1=ym[gp1,]; y2=ym[gp2,]
> y11=ym[gp11,]; y12=ym[gp12,]; y21=ym[gp21,]; y22=ym[gp22,]
> ym1=apply(y1,2,mean); ym2=apply(y2,2,mean);
> ym11=apply(y11,2,mean)
> ym12=apply(y12,2,mean)
> ym21=apply(y21,2,mean)
> ym22=apply(y22,2,mean)
> rbind(ym11,ym12,ym21,ym22) #gp mean across time
          [,1]      [,2]   [,3]      [,4]
ym11  4.900000  4.050000  3.450  4.000000
ym12  4.185185  5.703704  4.000  3.777778
ym21 20.500000 18.875000 22.125 18.000000
ym22 38.250000 26.750000 36.000 26.750000
>
> c(mean(ym11),mean(ym12),mean(ym21),mean(ym22)) #mean over time
[1]  4.100000  4.416667 19.875000 31.937500
> yv1=apply(y1,2,var); yv2=apply(y2,2,var)
> rbind(yv1,yv2)
         [,1]     [,2]      [,3]      [,4]
yv1  15.60315  21.0000  12.48751   6.20074
yv2 677.35606 222.8182 544.20455 211.35606
>
> c11=function(ti) exp(b01+b1*ti+b2*0)   #M2 chosen
> c12=function(ti) exp(b01+b1*ti+b2*1)
> c21=function(ti) exp(b02+b1*ti+b2*0)
> c22=function(ti) exp(b02+b1*ti+b2*1)
```

```
>
> maxy=max(ym21,ym22)
> plot(1:t,ym11,ylim=c(0,maxy),xlab="time",ylab="Number of epliepsy",
   cex=1.2,pch=1,col='gray50' )
> points(1:t,ym12,type="p",cex=1.2,pch=19,col='gray50')
> points(1:t,ym21,type="p",cex=1.2,pch=1,col='black')
> points(1:t,ym22,type="p",cex=1.2,pch=19,col='black')
> curve(c11,1,8,add=TRUE,col='gray50',lwd=2,lty=2)
> curve(c12,1,8,add=TRUE,col='gray50',lwd=2)
> curve(c21,1,8,add=TRUE,col='black',lwd=2,lty=2)
> curve(c22,1,8,add=TRUE,col='black',lwd=2)
```

```
> legend(0.3*t,0.52*maxy,text.width=0.8,y.intersp=0.8,bty="n",c("high fitted placebo mean",
  "high obs placebo mean","high fitted treat mean","high obs treat mean","low fitted placebo mean"
  "low obs placebo mean","low fitted treat mean","low obs treat mean"),col=c("black","black",
  "black","black","gray50","gray50","gray50","gray50"),lty=c(2,NA,1,NA,2,NA,1,NA),lwd=c(2,NA,2,NA,
  2,NA,2,NA),pch=c(NA,1,NA,19,NA,1,NA,19),pt.cex=c(NA,1.2,NA,1.2,NA,1.2,NA,1.2),cex=0.9)
```



Obviously, the variance is much greater than the mean and so the generalized Poisson distribution should provide better model fit.