

# STA841 HW2

Dipesh Gautam

October 1, 2015

## 1 Problem 1

See attached writeup.

## 2 Problem 2

### 2.1

We're given that the expected concentration after time  $t$  is  $\mu = e^{\alpha - \beta_T t}$ . So, the link function is a log link. Also assuming count data for concentration, I assumed poisson distribution for  $Y$ . The random component is the ascorbic acid determinants. Systematic component is the time with temperature factors. We also model with same intercept but different slope following the specification in the problem. The estimation is summarized below. And we are able to achieve low ratio for residual deviance and degrees of freedom.

Call:

```
glm(formula = acid ~ 1 + factor(temp):time, family = poisson(link = log),
     data = data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.258275	-0.056668	0.006663	0.128272	0.140947

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	3.835210	0.112360	34.133	< 2e-16 ***
factor(temp)0:time	-0.000733	0.023063	-0.032	0.975
factor(temp)10:time	-0.023747	0.023935	-0.992	0.321
factor(temp)20:time	-0.132440	0.029181	-4.539	5.66e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 37.09555 on 11 degrees of freedom  
Residual deviance: 0.23212 on 8 degrees of freedom  
AIC: 73.229

Number of Fisher Scoring iterations: 3

## 2.2

Time taken for 50% recution in the concentration can be calculated as below:

$$\begin{aligned}\frac{\mu_t}{\mu_0} &= \frac{e^{\alpha - \beta_T t}}{e^{\alpha}} \\ 0.5 &= e^{-\beta_T t} \\ \log(0.5) &= -\beta_T t \\ t &= \frac{\log 2}{\beta_T}\end{aligned}$$

	2.5 %	97.5 %
factor(temp)0:time	14.972	0.000
factor(temp)10:time	9.744	0.000
factor(temp)20:time	3.631	9.074

Times taken at the three temperatures for ascorbic acid concentration to be reduced to 50% of original values are:

temp	time for 50% reduction
0	945.603418278674
10	29.1889599071147
20	5.23366253870517

The confidence intervals are also shown above.

## 2.3

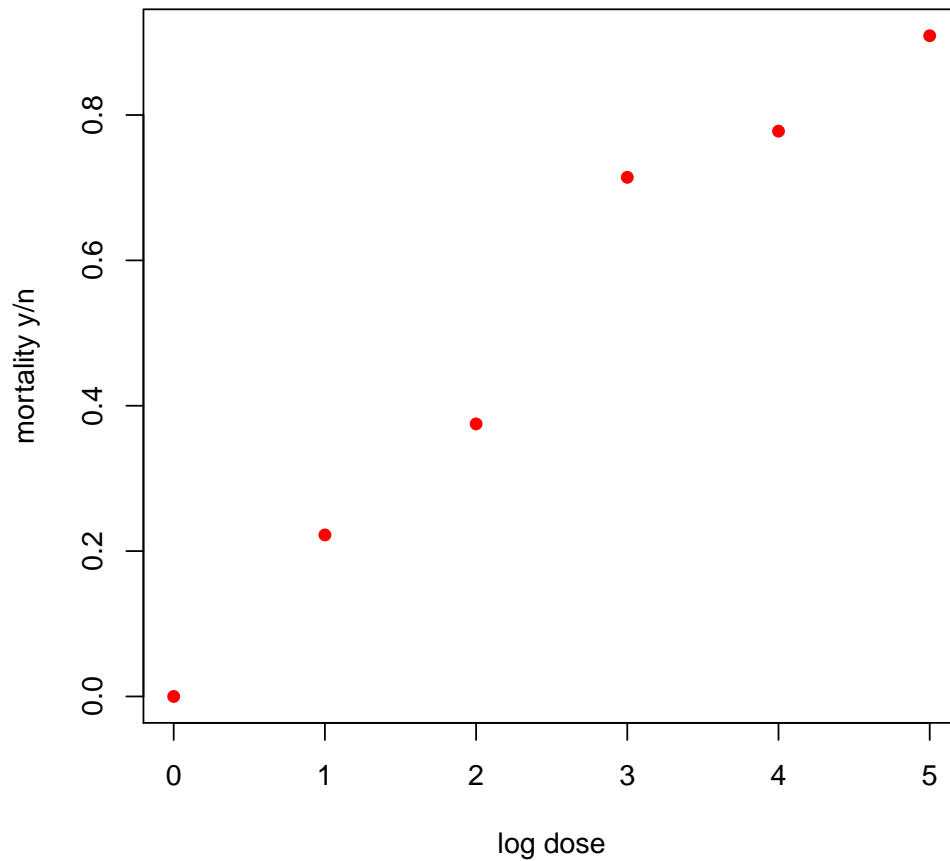
I use F-test to check the consistency of the given values of mean squared error and degrees of freedom for individual packages with the one we obtained from the grouped packages. We're given that for the individual packages,  $MSE = 0.706$  and  $df = 24$ . So the total sum of squares is  $.706 * 30$  on 24 degrees of freedom.

In the model we fit, we got deviance residual of 0.232 on 8 degrees of freedom. Using these values we used the F-test to compare the models and find that our model is better than the one with individual pacakges as the F-statistic is 54.25

### 3 Problem 3

#### 3.1

Mortality fraction is plotted against log dose below:



#### 3.2

Probit linear model is fitted where the probit of the mortality rate is linear in log dose. Summary of the glm fit is shown below. The fitted values are superimposed on the plot from the last part.

Call:

```
glm(formula = mortality ~ dose, family = binomial(link = probit),  
     data = data3, weights = data3$weights)
```

Deviance Residuals:

1	2	3	4	5	6
-0.96616	0.33030	0.00717	0.58395	-0.23383	-0.25443

Coefficients:

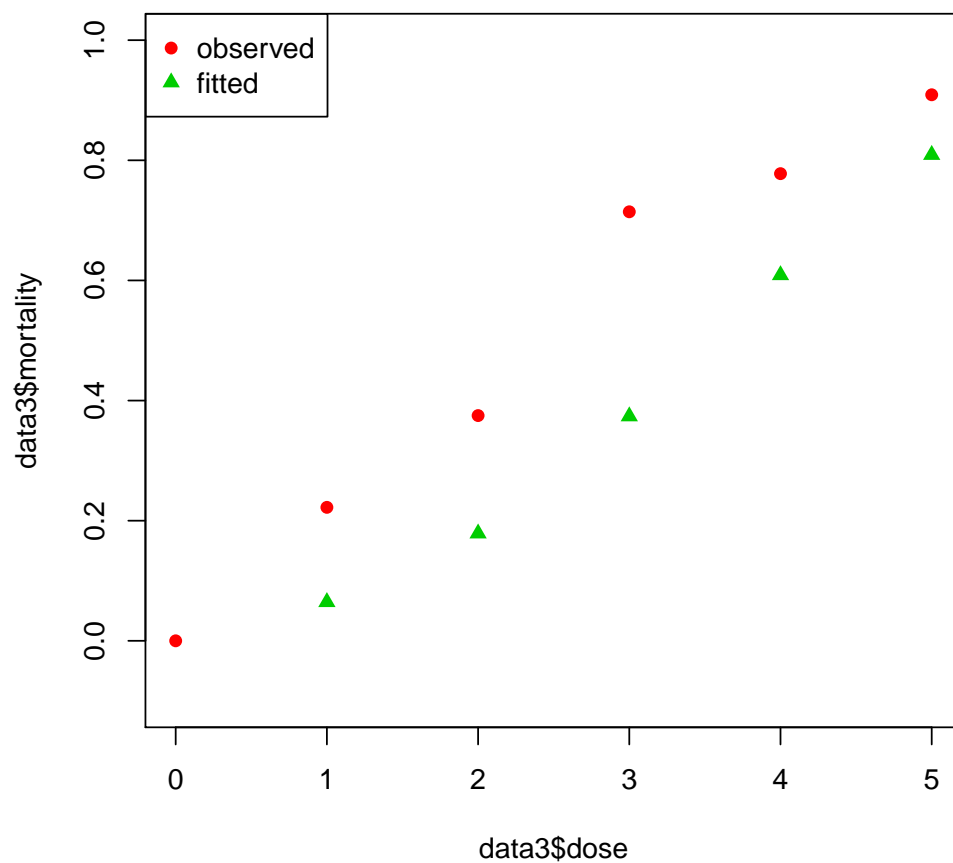
	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-1.5180	0.4219	-3.598	0.000321 ***
dose	0.5981	0.1391	4.300	1.71e-05 ***

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 25.792 on 5 degrees of freedom  
 Residual deviance: 1.503 on 4 degrees of freedom  
 AIC: 16.966

Number of Fisher Scoring iterations: 4



### 3.3

$\log_2 LD_{50}$  is given by the expression:

$$\log_2 LD_{50} = -\frac{\hat{\beta}_0}{\hat{\beta}_1}$$

which is evaluated below.

The confidence interval obtained by Fieller's method for  $\log_2 LD_{50}$  is: [1.738, 3.29]

### 3.4 Bootstrap CI

Bootstrap CI is calculated by obtaining 1000 bootstrap samples and calculating  $\log_2 LD_{50}$  estimate for each bootstrap sample and getting the 2.5% and 97.5% quantile for the estimates.

A potential concern with the current problem is the small sample size. Another potential concern might be that we have aggregate data and not individual level data. The bootstrap might be better if we had individual level data.

The confidence interval obtained by bootstrap method for  $\log_2 LD_{50}$  is: [1.78, 3.317]

Bootstrap CI is mostly similar but slightly less spread out than the one obtained in the previous part using Fieller's method.

### 3.5 Bayesian probit

I implemented the Bayesian probit linear model using the data augmentation technique by Albert and Chib as explained in the lecture notes.

Multivariate normal prior was used for  $\beta$  given by

$$\pi(\beta) = N(\beta_0, \Sigma_0)$$

and for the problem  $\beta_0 = (1,1)$  and  $\Sigma_0$  was defined with no covariance and 1 variance for both coefficients. First 100 samples were discarded as burnin.

Posterior mean estimate for  $\log_2 LD_{50}$  obtained by Bayesian Probit linear model using the data augmentation technique by Albert and Chib is 2.485.

The credible interval obtained using the same technique for  $\log_2 LD_{50}$  is: [1.743, 3.208]

## 4 Appendix

Codes used:

### Problem 2

```
> #2.1
> set.seed(1000)
> w2= c(45,45,34)
> w4=c(47,43,28)
> w6=c(46,41,21)
> w8=c(46,37,16)
> temp=rep(c(0,10,20),4)
> time=rep(c(2,4,6,8),each=3)
> acid= c(w2,w4,w6,w8)
> data = data.frame(acid=acid ,time=time,temp=temp)
> fit1 = glm(acid~ 1 + factor(temp):time, data = data, family=poisson(link=log))
> summary(fit1)
> #2.2
> beta_hat = coef(fit1)
> t_50_0 = -log(2)/beta_hat[2]
> t_50_10 = -log(2)/beta_hat[3]
> t_50_20 = -log(2)/beta_hat[4]
> int_beta=confint(fit1, c(2,3,4))
> CI_t_50=round(-log(2)/int_beta, 3)
> CI_t_50[CI_t_50<0]=0
> CI_t_50
```

```

> #2.3
>
> mse= 0.706
> df1 = 24
> df2= df.residual(fit1)
> resid.dev = deviance(fit1)
> F = ((mse*36-resid.dev)/24)/(resid.dev/8)

```

### Problem 3

```

> #3.1
> dose = seq(0,5)
> mortality = c(0/7, 2/9, 3/8, 5/7, 7/9, 10/11)
> weights= c(7,9,8,7,9,11)
> data3 = data.frame(dose=dose, mortality=mortality, weights=weights)
> plot(data3$dose, data3$mortality, pch=16, col=2, xlab="log dose", ylab="mortality y/n")
> #3.2
> probit.fit = glm(mortality~dose, family=binomial(link=probit),
+                 data=data3, weights=data3$weights)
> summary(probit.fit)
> fitted_values=fitted.values(probit.fit)
> plot(data3$dose, data3$mortality, pch=16, col=2, ylim=c(-.1,1))
> points(fitted_values, pch=17, col=3, xlab="log dose", ylab="mortality y/n")
> legend("topleft", c("observed","fitted"), col = c(2,3), pch=c(16,17) )
> #3.3
>
>
> set.seed(1000)
> beta_hat = coef(probit.fit)
> log_LD_50 = -beta_hat[1]/beta_hat[2]
> sigma_hat = vcov(probit.fit)
> sigma11= sigma_hat[1,1]
> sigma22= sigma_hat[2,2]
> sigma12= sigma_hat[1,2]
> t = 1.96
> b0 = beta_hat[1]
> b1 = beta_hat[2]
> a = b1^2 - t^2*sigma22
> b = -2*(b0*b1-t^2*sigma12) #check the 1.96 part
> c = b0^2-t^2*sigma11
> CI = -(-b+c(1,-1)*sqrt(b^2-4*a*c))/(2*a)
> #3.4
>
> set.seed(1000)
> N = 1000
> r.boot = double(N)
> mortality.boot = double(6)
> est.boot = matrix(NA, nrow=N, ncol=2)
> for (i in 1:N){
+   for (j in 1:length(data3$mortality)){
+     mortality.boot[j] = sum(rbinom(weights[j], 1, mortality[j]))/weights[j]
+   }
+   data3.boot = data.frame(mortality=mortality.boot, dose=dose, weights=weights)
+   data3.boot
+   probit.fit.boot = glm(mortality.boot~dose, family=binomial(link=probit),
+                         data = data3.boot, weights=weights)
+

```

```

+   beta_hat = coef(probit.fit.boot)
+   beta_hat
+   est.boot[i,] = beta_hat
+   r.boot[i] = -beta_hat[1]/beta_hat[2]
+
+ }
> CI.boot = quantile(r.boot, c(.025,.975))
> CI.boot
> #3.5
>
> set.seed(1000)
> dose = seq(0,5)
> mortality = c(0/7, 2/9, 3/8, 5/7, 7/9, 10/11)
> weights= c(7,9,8,7,9,11)
> data3 = data.frame(dose=dose, mortality=mortality, weights=weights)
> library(truncnorm)
> library(mvtnorm)
> Pi = mortality
> y1= rbinom(weights[1], 1, Pi[1])
> y2= rbinom(weights[2], 1, Pi[2])
> y3= rbinom(weights[3], 1, Pi[3])
> y4= rbinom(weights[4], 1, Pi[4])
> y5= rbinom(weights[5], 1, Pi[5])
> y6= rbinom(weights[6], 1, Pi[6])
> Y =c(y1,y2,y3,y4,y5,y6)
> X2=rep(dose,weights)
> M = length(Y)
> X1 = rep(1, M)
> X = cbind(X1,X2)
> sigma0 = matrix(c(1,0,0,1), nrow=2)
> beta0= c(1,1)
> N = 1000
> #Initial Z
> Z = rep(NA,M)
> for (i in 1:M){
+   Z[i] = rnorm(1, X[i,]*beta0, 1)
+ }
> BETA = matrix(rep(NA, N*2), nrow=N)
> sigma_hat = solve(solve(sigma0)+t(X)%*%X)
> beta_hat = sigma_hat*%(solve(sigma0)*%*beta0+t(X)%*%Z)
> for (i in 1:N){
+   beta = rmvnorm(1, beta_hat, sigma_hat)
+   for (j in 1:M){
+     if (Y[j]==1){
+       Z[j] = rtruncnorm(1,a=0, b=Inf, mean= X[j,]%*%t(beta), sd= 1 )
+     } else if(Y[j]==0){
+       Z[j] = rtruncnorm(1,a=-Inf, b=0, mean= X[j,]%*%t(beta), sd= 1 )
+     }
+   }
+   BETA[i,] = beta
+   beta_hat = sigma_hat*%(solve(sigma0)*%*beta0+t(X)%*%Z)
+ }
> LD.50.bayes = -BETA[100:N,1]/BETA[100:N,2]
> CI.bayes = quantile(LD.50.bayes, c(.025, .975))
> post.mean =round(mean(LD.50.bayes),3)

```