```
> rm(list = ls())
>
> #Question 2
> #(b)
>
> #Let S(p,y,n)=Score function=(y/p)-(n-y)/(1-p)
> S=function(p,y,n){
+   res=(y/p)-(n-y)/(1-p)
+   return(res)
+ }
>
> #Hessian matirxb
> H=function(p,y,n){
+   res=-(y/p^2)-(n-y)/(1-p)^2
+   return(res)
+ }
>
> #Newton-Raphson Function
> #t=iterition times
> #p0=starting value
> NR=function(p0,y,n,t){
+   p=p0
+   for(i in 1:t){
+     p=p-H(p,y,n)^(-1)*S(p,y,n)
+     cat("p=",p,"t=",t,"\n")
+   }
+   return(p)
+ }
>
> #(c)
> #p_hat=0.3 n=10 implies y=3
> #p0=(0.1,0.2,...,0.9)
> #t=6
>
> NR(0.1,3,10,6)
p= 0.172 t= 6
p= 0.2525236 t= 6
p= 0.2947439 t= 6
p= 0.299946 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
[1] 0.3
> NR(0.2,3,10,6)
p= 0.2727273 t= 6
p= 0.2983957 t= 6
p= 0.2999951 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
```

```
[1] 0.3
> NR(0.3,3,10,6)
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
[1] 0.3
> NR(0.4,3,10,6)
p= 0.2909091 t= 6
p= 0.2998355 t= 6
p= 0.2999999 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
[1] 0.3
> NR(0.5,3,10,6)
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
[1] 0.3
> NR(0.6,3,10,6)
p= 0.36 t= 6
p= 0.2952809 t= 6
p= 0.2999566 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
[1] 0.3
> NR(0.7,3,10,6)
p= 0.472973 t= 6
p= 0.2932591 t= 6
p= 0.2999105 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
[1] 0.3
> NR(0.8,3,10,6)
p= 0.626087 t= 6
p= 0.3847662 t= 6
p= 0.2923753 t= 6
p= 0.2998851 t= 6
p= 0.3 t= 6
p= 0.3 t= 6
[1] 0.3
```

```
> NR(0.9,3,10,6)
p= 0.8052632 t= 6
p= 0.6349771 t= 6
p= 0.3940117 t= 6
p= 0.2914389 t= 6
p= 0.2998545 t= 6
p= 0.3 t= 6
[1] 0.3
>
> #(d)
> #The more starting value get closed to true value, the faster speed of
> #convergence is
>
> #(e)
> #when pi_hat=0, implies that y/n=0, y=0
> #Score function is -10/(1-p)
> #Hessian matrix is -10/(1-p)^2
> #p(t)=2*p(t-1)-1
>
> #when pi_hat=1, implies that y/n=1, y=10
> #Score function is 10/p
> #Hessian matrix is -10/p^2
> #p(t)=2*p(t-1)
>
> #Thus, we can not have correct result because there is no convergence
>
```

```
> #Question 5
> #(a)
>
> A=cbind(c(8,7,6,6,3,4,7,2,3,4),rep(0,10))
> B=cbind(c(9,9,8,14,8,13,11,5,7,6),rep(1,10))
> data=data.frame(rbind(A,B))
> colnames(data)=c("Y","X")
> model1=glm(Y~X,family = poisson(),data = data)
> summary(model1)

Call:
glm(formula = Y ~ X, family = poisson(), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5280  -0.7622  -0.1699   0.6938   1.5399

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.6094     0.1414  11.380  < 2e-16 ***
X             0.5878     0.1764   3.332 0.000861 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 27.857  on 19  degrees of freedom
Residual deviance: 16.268  on 18  degrees of freedom
AIC: 94.349

Number of Fisher Scoring iterations: 4

>
> #When x=0
> #ua=e^a
>
> #When x=1
> #ub=e^(a+b)
>
> #So ub/ua=e^b
> exp(model1$coefficients[2])
  X
1.8
> ua=exp(model1$coefficients[1])
> ub=exp(model1$coefficients[1]+model1$coefficients[2])
> exp(model1$coefficients[2])==ub/ua
   X
TRUE
>
```

```
> #Interpretion
> #The coefficient for the intercept is 1.6094379. Thus the estimated
> #expectation for the number of seizures in Treatment A is e^1.6094379=5
> #The estimated expectation for the number of seizures in Treatment B
> #is e^(1.6094379+0.5877867)=9
>
> #(c)
> library(AER)
> dispersiontest(model1,trafo = 1)

	Overdispersion test

data:  model1
z = -1.1189, p-value = 0.8684
alternative hypothesis: true alpha is greater than 0
sample estimates:
     alpha
-0.1977778


>
> #Result shows that we can not reject c=0 for Var(y)=u+c*f(u)
> #Which means Var(y)=u=E(y). There is no dispersion
>
> #(e)
> library(MASS)
> model2=glm.nb(Y~X,data = data,link = log)
> summary(model2)

Call:
glm.nb(formula = Y ~ X, data = data, link = log, init.theta = 113420.3107)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5280  -0.7622  -0.1699   0.6937   1.5398

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.6094     0.1414  11.380  < 2e-16 ***
X             0.5878     0.1764   3.332 0.000861 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(113420.3) family taken to be
1)

    Null deviance: 27.855  on 19  degrees of freedom
Residual deviance: 16.267  on 18  degrees of freedom
AIC: 96.349
```

```
Number of Fisher Scoring iterations: 1


              Theta:  113420
          Std. Err.:  4076965
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -90.349
>
>
>
> #(f)
>
> #Poisson
> model1$coefficients[2]
        X
0.5877867
>
> #standard error is 0.1746
>
> #Negative Binomial
> model2$coefficients[2]
        X
0.5877867
>
> #standard error is 0.1746
>
> #(g)
> model3=glm(Y~1,family = poisson(),data = data)
> summary(model3)

Call:
glm(formula = Y ~ 1, family = poisson(), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2336  -0.9063   0.0000   0.4580   2.3255

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.94591    0.08451   23.02   <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: 27.857  on 19  degrees of freedom
Residual deviance: 27.857  on 19  degrees of freedom
AIC: 103.94
```

```
Number of Fisher Scoring iterations: 4

>
> model4=glm.nb(Y~1,data = data,link = log)
> summary(model4)

Call:
glm.nb(formula = Y ~ 1, data = data, link = log, init.theta = 18.2073559)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9810  -0.7836   0.0000   0.3859   1.9033

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.94591    0.09944   19.57   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(18.2074) family taken to be
1)

    Null deviance: 20.279  on 19  degrees of freedom
Residual deviance: 20.279  on 19  degrees of freedom
AIC: 104.77

Number of Fisher Scoring iterations: 1


              Theta:  18.2
          Std. Err.:  21.0

 2 x log-likelihood:  -100.767
>
> #(h)
> dispersiontest(model3,trafo = 1)

       Overdispersion test

data:  model3
z = 0.96575, p-value = 0.1671
alternative hypothesis: true alpha is greater than 0
sample estimates:
    alpha
0.3857143
>
> #Result shows that we can not reject c=0 for Var(y)=u+c*f(u)
> #Which means Var(y)=u=E(y). There is no dispersion
```