

## Questions

(1) Fit four separate negative binomial models to these data, one for each treatment. (p. 2-4)

	Treatment 1	Treatment 2	Treatment 3	Treatment 4
$\mu$ of model	4.033331	3.166660	1.483332	1.508332
$\theta$ of model	1.502888	1.760494	1.333131	1.153521

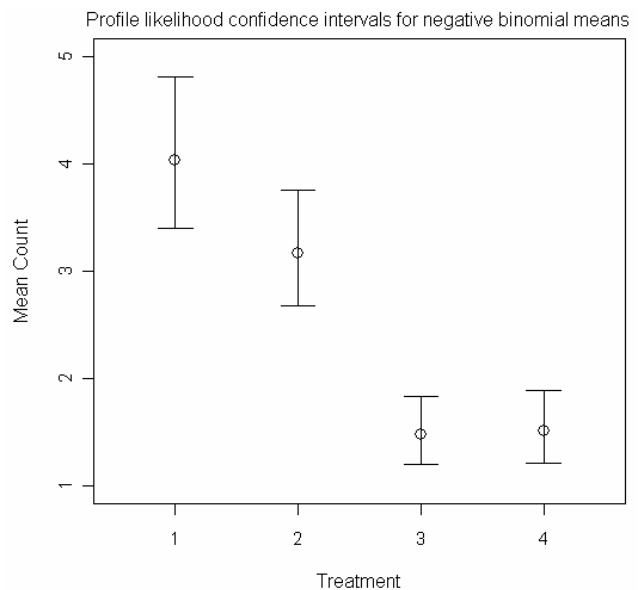
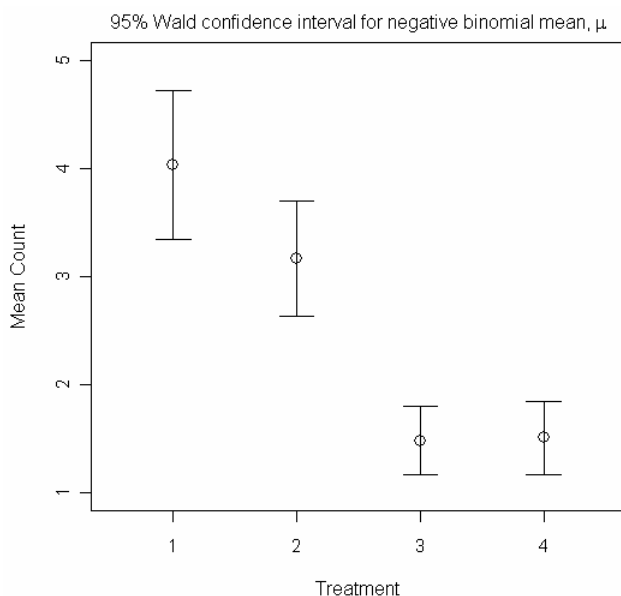
(2) Check the fit of the negative binomial for each treatment separately using a goodness of fit test. (p. 4-5)

Model:	Model 1	Model 2	Model 3	Model 4
p-value	0.5065	0.711	0.9554	0.337

Each of the models has a different goodness of fit to their respective data. The best fitting negative binomial model is the one for Treatment 3, judging from its highest p-value of the 4 models, 0.9554, which means it has a significant lack of lack-of-fit. The poorest fitting model is the one for Treatment 4, with a p-value of 0.337. Still, these are much better results than the p-value we got with the Poisson model using the aphid data, which when using `chisq.test` with 20000 replicates as was done with this problem, gave a p-value of 0.00015 (a significant lack of fit).

(3) Create an error bar plot for the results in which you display 95% Wald confidence intervals for the mean. (p. 5-6)

(4) Do a second error bar plot this time using profile likelihood confidence intervals for the mean (in which the dispersion parameter is fixed at its MLE). (p. 6-7)



The two confidence intervals are very similar to each other both visually and numerically (see p.7).

## R Code

### Problem 1

#### Inputting the data:

```
> num.borers1<-c(0:26)
> trt1<-c(19,12,18,18,11,12,7,8,4,4,1,0,1,1,0,1,0,1,0,1,0,0,0,0,0,0,1)
> dataset1<-rep(num.borers,trt1)

> num.borers2<-c(0:12)
> trt2<-c(24,16,16,18,15,9,6,5,3,4,3,0,1)
> dataset2<-rep(num.borers2,trt2)

> num.borers3<-c(0:8)
> trt3<-c(43,35,17,11,5,4,1,2,2)
> dataset3<-rep(num.borers3,trt3)

> num.borers4<-c(0:11)
> trt4<-c(47,23,27,9,7,3,1,1,0,0,1,1)
> dataset4<-rep(num.borers4,trt4)

> dataset1
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [17] 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 2
 [33] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [49] 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [65] 3 3 3 4 4 4 4 4 4 4 4 4 4 4 5 5
 [81] 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6
 [97] 6 7 7 7 7 7 7 7 7 7 8 8 8 8 9 9 9
 [113] 9 10 12 13 15 17 19 26

> dataset2
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [17] 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1
 [33] 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
 [49] 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3
 [65] 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4
 [81] 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5
 [97] 5 5 6 6 6 6 6 6 7 7 7 7 7 8 8 8
 [113] 9 9 9 9 10 10 10 12

> dataset3
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [25] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
 [49] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [73] 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3
 [97] 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 5 5 5 5 6 7 7 8 8

> dataset4
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [17] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [33] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 [49] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [65] 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
 [81] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [97] 2 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4
 [113] 4 5 5 5 6 7 10 11

>
```

Creating negative binomial positive and negative loglikelihood functions

```

> NBvec.pos<-function(dataset,p)
sum(log(dnbinom(dataset, mu=p[1],size=p[2])))
> NBvec.neg<-function(dataset,p) -NBvec.pos(dataset,p)

```

### Minimizing the negative loglikelihood for all treatments

```

> out1<-nlm(function(p) NBvec.neg(dataset1,p), c(2,1), hessian=TRUE)
> out2<-nlm(function(p) NBvec.neg(dataset2,p), c(2,1), hessian=TRUE)
> out3<-nlm(function(p) NBvec.neg(dataset3,p), c(2,1), hessian=TRUE)
> out4<-nlm(function(p) NBvec.neg(dataset4,p), c(2,1), hessian=TRUE)

```

```

> out1
$minimum
[1] 298.7527
$estimate
[1] 4.033331 1.502888
$gradient
[1] -1.409342e-07 -1.248152e-06
$hessian
      [,1]      [,2]
[1,] 8.0738560310 7.811502e-04
[2,] 0.0007811502 1.223846e+01
$code
[1] 1
$iterations
[1] 9

```

The value of the negative loglikelihood at the minimum is 298.7527 (which means the maximum of the positive loglikelihood occurs at -298.7527). The estimate of the MLE of  $\mu$  is 4.033331, and of  $\theta$ , 1.502888. Both gradients are close to zero, as they should be at both MLEs.

Thus, the negative binomial model that best fits Treatment 1 has  $\mu= 4.033331$ , and  $\theta= 1.502888$ .

```

> out2
$minimum
[1] 272.1343
$estimate
[1] 3.166660 1.760494
$gradient
[1] -6.779938e-05 4.132908e-05
$hessian
      [,1]      [,2]
[1,] 13.535582084 0.000749431
[2,] 0.000749431 6.174429709
$code
[1] 1
$iterations
[1] 9

```

```

> out3
$minimum
[1] 200.3049
$estimate
[1] 1.483332 1.333131
$gradient
[1] -3.391448e-05 3.070011e-06
$hessian

```

```

      [,1]      [,2]
[1,] 38.280697490 0.001096639
[2,] 0.001096639 7.171600728
$code
[1] 1
$iterations
[1] 7

> out4
$minimum
[1] 202.2679
$estimate
[1] 1.508332 1.153521
$gradient
[1] -5.068803e-06 9.855637e-08
$hessian
      [,1]      [,2]
[1,] 34.465919938 0.001262720
[2,] 0.001262720 10.324999683
$code
[1] 1
$iterations
[1] 7

```

## Problem 2

### Goodness of fit for Treatment 1

```

> expprobs1<-c(dnbinom(0:25,mu=out1$estimate[1], size=out1$estimate[2]), 1-
pnbinom(25, mu=out1$estimate[1], size=out1$estimate[2]))
> expprobs1
 [1] 0.1409076176 0.1542807623 0.1406610459 0.1196546187 0.0981321330
 [6] 0.0786832917 0.0621281113 0.0485142340 0.0375661123 0.0288974788
[11] 0.0221115550 0.0168455071 0.0127868562 0.0096760407 0.0073025534
[16] 0.0054985311 0.0041317805 0.0030991931 0.0023209524 0.0017356500
[21] 0.0012962769 0.0009669987 0.0007205963 0.0005364584 0.0003990182
[26] 0.0002965464 0.0008500800
> expprobs1*120->Ei1
> Ei1
 [1] 16.90891411 18.51369148 16.87932551 14.35855425 11.77585596
 [6] 9.44199500 7.45537336 5.82170809 4.50793348 3.46769745
[11] 2.65338660 2.02146085 1.53442274 1.16112489 0.87630640
[16] 0.65982373 0.49581366 0.37190317 0.27851429 0.20827800
[21] 0.15555323 0.11603984 0.08647156 0.06437501 0.04788218
[26] 0.03558556 0.10200960

```

Only the first 8 data points are above 5, so the rest looks like it will need to be pooled, but not if I use `chisq.test`.

```

> trt1->Oil
> Oil
 [1] 19 12 18 18 11 12 7 8 4 4 1 0 1 1 0 1 0 1 0 1 0 0
[23] 0 0 0 0 1
> chisq.test(Oil,p=expprobs1,simulate.p.value=TRUE,B=20000)

```

Chi-squared test for given probabilities with simulated p-value  
(based on 20000 replicates)

```
data: Oi1
X-squared = 22.8416, df = NA, p-value = 0.5065
```

### Goodness of fit for Treatment 2

```
> expprobs2<-c(dnbinom(0:11,mu=out2$estimate[1], size=out2$estimate[2]), 1-
pnbinom(11, mu=out2$estimate[1], size=out2$estimate[2]))
> expprobs2*120->Ei22
> trt2->Oi2
> chisq.test(Oi2,p=expprobs2,simulate.p.value=TRUE,B=20000)
```

Chi-squared test for given probabilities with simulated p-value  
(based on 20000 replicates)

```
data: Oi2
X-squared = 8.8655, df = NA, p-value = 0.711
```

### Goodness of fit for Treatment 3

```
> expprobs3<-c(dnbinom(0:7,mu=out3$estimate[1], size=out3$estimate[2]), 1-
pnbinom(7, mu=out3$estimate[1], size=out3$estimate[2]))
> expprobs3*120->Ei3
> trt3->Oi3
> chisq.test(Oi3,p=expprobs3,simulate.p.value=TRUE,B=20000)
```

Chi-squared test for given probabilities with simulated p-value  
(based on 20000 replicates)

```
data: Oi3
X-squared = 2.6892, df = NA, p-value = 0.9554
```

### Goodness of fit for Treatment 4

```
> expprobs4<-c(dnbinom(0:10,mu=out4$estimate[1], size=out4$estimate[2]), 1-
pnbinom(10, mu=out4$estimate[1], size=out4$estimate[2]))
> expprobs4*120->Ei4
> trt4->Oi4
> chisq.test(Oi4,p=expprobs4,simulate.p.value=TRUE,B=20000)
```

Chi-squared test for given probabilities with simulated p-value  
(based on 20000 replicates)

```
data: Oi4
X-squared = 11.988, df = NA, p-value = 0.337
```

## Problem 3

### Get x and y for plot

```
> treatment<-c(1:4)
> treatment
[1] 1 2 3 4
> mean.count<-c(
out1$estimate[1],out2$estimate[1],out3$estimate[1],out4$estimate[1] )
> mean.count
[1] 4.033331 3.166660 1.483332 1.508332
```

### Set up plot area

```

> plot(treatment, mean.count, axes=FALSE, xlab="Treatment", ylab="Mean
Count", xlim=c(0.5,4.5), ylim=c(0,15),cex=1.5)
> axis(1,cex.axis=.9)
> axis(2,cex.axis=.9)
> box()

```

#### Add title

```

> mtext(expression(paste('95% Wald confidence interval for negative binomial
mean, ',mu)), side=3, line=.5)

```

#### Getting lower and upper bounds for all 4 models

```

> out1$estimate[1]-qnorm(.975)*sqrt(diag(solve(out1$hessian))[1])
[1] 3.343556
> out1$estimate[1]+qnorm(.975)*sqrt(diag(solve(out1$hessian))[1])
[1] 4.723107
> out2$estimate[1]-qnorm(.975)*sqrt(diag(solve(out2$hessian))[1])
[1] 2.633927
> out2$estimate[1]+qnorm(.975)*sqrt(diag(solve(out2$hessian))[1])
[1] 3.699393
> out3$estimate[1]-qnorm(.975)*sqrt(diag(solve(out3$hessian))[1])
[1] 1.166551
> out3$estimate[1]+qnorm(.975)*sqrt(diag(solve(out3$hessian))[1])
[1] 1.800112
> out4$estimate[1]-qnorm(.975)*sqrt(diag(solve(out4$hessian))[1])
[1] 1.174481
> out4$estimate[1]+qnorm(.975)*sqrt(diag(solve(out4$hessian))[1])
[1] 1.842184

```

The Wald confidence interval for the mean is between **3.343 and 4.723** for Model 1, between **2.634 and 3.699** for Model 2, **1.167 and 1.800** for Model 3, and **1.174 and 1.842** for Model 4.

#### Drawing the arrows

```

#draws arrows from y0=meanval to y1=meanval+/-stderrval, x0=x1=1,2,3,or 4.
arrow.draw<-function(xpos,meanval,stderrval)
arrows(xpos,meanval+stderrval,xpos,meanval-
stderrval,length=0.15,angle=90,code=3)
arrow.draw(1,mean.count[1],qnorm(.975)*sqrt(diag(solve(out1$hessian))[1])
arrow.draw(2,mean.count[2],qnorm(.975)*sqrt(diag(solve(out1$hessian))[1])
arrow.draw(3,mean.count[3],qnorm(.975)*sqrt(diag(solve(out1$hessian))[1])
arrow.draw(4,mean.count[4],qnorm(.975)*sqrt(diag(solve(out1$hessian))[1])

```

### **Problem 4**

#### Getting profile likelihood confidence intervals

#This is done for each model and the CIs are outputted into vectors CI1-CI4

```

> library(Bhat)

> x.in<-
list(label=c('mu','theta'),est=out1$estimate,low=c(2,.01),high=c(8,10))
> plkhci(x.in, function(p) NBvec.neg(dataset1,p), 'mu')->CI1
neg. log. likelihood: 298.7527
> x.in<-
list(label=c('mu','theta'),est=out2$estimate,low=c(2,.01),high=c(8,10))
> plkhci(x.in, function(p) NBvec.neg(dataset2,p), 'mu')->CI2
neg. log. likelihood: 272.1343

```

```

> x.in<-list(label=c('mu','theta'),est=out3$estimate,low=c(1,1),high=c(2,2))
> plkhci(x.in, function(p) NBvec.neg(dataset3,p), 'mu')->CI3
neg. log. likelihood: 200.3049
> x.in<-list(label=c('mu','theta'),est=out4$estimate,low=c(1,1),high=c(2,2))
> plkhci(x.in, function(p) NBvec.neg(dataset4,p), 'mu')->CI4
neg. log. likelihood: 202.2679

> CI1
[1] 3.401741 4.804884
> CI2
[1] 2.675628 3.759181
> CI3
[1] 1.198229 1.838585
> CI4
[1] 1.210002 1.887630

```

The profile likelihood confidence interval for the mean is between **3.402 and 4.804** for Model 1, between **2.676 and 3.759** for Model 2, **1.198 and 1.839** for Model 3, and between **1.210 and 1.888** for Model 4.

Numerically, these aren't much different from the Wald confidence intervals:

	Wald CIs for mu, lower bound	Profile likelihood CIs for mu, lower bound	Wald CIs for mu, upper bound	Profile likelihood CIs for mu, upper bound	% similarity, lower bound	% similarity, upper bound
Model 1	3.343556	3.401741	4.723107	4.804884	98.289552	98.298044
Model 2	2.633927	2.675628	3.699393	3.759181	98.441450	98.409547
Model 3	1.166551	1.198229	1.800112	1.838585	97.356265	97.907467
Model 4	1.174481	1.210002	1.842184	1.887630	97.064385	97.592431

### Setting up error bar plot

```

treatment<-c(1:4)
plot(treatment, mean.count, axes=FALSE, xlab="Treatment", ylab="Mean Count",
xlim=c(0.5,4.5), ylim=c(1,5), cex=1.5)
axis(1,cex.axis=.9)
axis(2,cex.axis=.9)
box()
mtext("Profile likelihood confidence intervals for negative binomial means",
side=3, line=.5)

```

### Drawing error bars

#arrow.draw draws arrows from y0=lower bound given by CI[1] to y1=upper bound given by CI[2], x0=x1=1,2,3,or 4.

```

arrow.draw<-function(xpos,CI)
arrows(xpos,CI[1],xpos,CI[2],length=0.15,angle=90,code=3)

arrow.draw(1,CI1)
arrow.draw(2,CI2)
arrow.draw(3,CI3)
arrow.draw(4,CI4)

```