Ecology 145 Midterm
Dahl Winters
3/10/06

Problem 1

1. Estimate how the variance in the number of satellite males varies with the mean using carapace width to group your data.

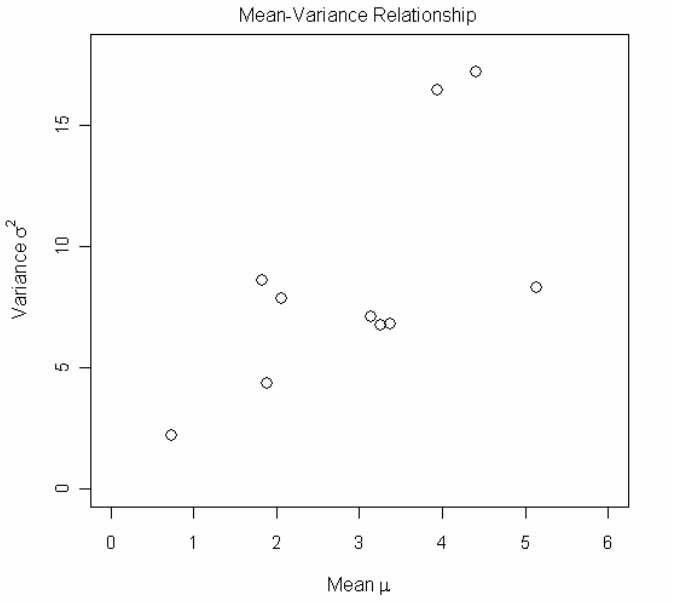
```
crabs<-
read.table('http://www.unc.edu/courses/2006spring/ecol/145/001/data/midterm/c
rabs.txt', header=TRUE, sep='')
```

```
table(crabs$num.satellites)
 0  1  2  3  4  5  6  7  8  9 10 11 12 14 15
62 16  9 19 19 15 13  4  6  3  3  1  1  1  1
```

```
table(crabs$width)
  21   22 22.5 22.9   23 23.1 23.2 23.4 23.5 23.7 23.8 23.9   24
   1    1    3    3    2    3    1    1    1    3    3    1    2
24.1 24.2 24.3 24.5 24.7 24.8 24.9   25 25.1 25.2 25.3 25.4 25.5
   1    2    2    7    5    1    3    6    2    2    1    3    3
25.6 25.7 25.8 25.9   26 26.1 26.2 26.3 26.5 26.7 26.8   27 27.1
   2    6    7    1    6    2    8    1    6    3    3    5    2
27.2 27.3 27.4 27.5 27.6 27.7 27.8 27.9   28 28.2 28.3 28.4 28.5
   2    1    3    6    1    2    2    2    3    4    3    2    4
28.7 28.9   29 29.3 29.5 29.7 29.8   30 30.2 30.3 30.5 31.7 31.9
   2    1    6    2    1    1    1    3    1    1    1    1    1
33.5
   1
```

```
table(cut(crabs$width, quantile(crabs$width, seq(0,1,.1)), include.lowest=
TRUE))
  [21,23.7] (23.7,24.5] (24.5,25.1] (25.1,25.7] (25.7,26.1]
         19          18          15          19          16
```

```
(26.1,26.7] (26.7,27.4] (27.4,28.2]   (28.2,29]   (29,33.5]
         18          16          16          22          14
width.decs<-table(cut(crabs$width, quantile(crabs$width, seq(0,1,.1)),
include.lowest=TRUE))

tapply(crabs$num.satellites, width.decs,mean)->means
tapply(crabs$num.satellites, width.decs,var)->vars

plot(means, vars, axes=FALSE, xlab=expression(paste("Mean ",mu)),
ylab=expression(paste("Variance ",sigma^2)), xlim=c(0,6), ylim=c(0,18),
cex=1.5)
axis(1,cex.axis=.9)
axis(2,cex.axis=.9)
box()
mtext("Mean-Variance Relationship", side=3, line=.5)
```



Since we have discrete count data, our choices for models are the Poisson and the negative binomial. Because the mean-variance relationship is heteroscedastic, a negative binomial model would be the best model of the two to use for describing the probability distribution.

$$\sigma^2 = \mu + \frac{\mu^2}{\theta}$$

NB2 – negative binomial regression

```
lm(vars~offset(means)+I(means^2)-1)
Call:
lm(formula = vars ~ offset(means) + I(means^2) - 1)

Coefficients:
I(means^2)
    0.4109
quad.coef<-coef(lm(vars~offset(means)+I(means^2)-1))
quad.coef
I(means^2)
 0.4109061
```

```
quad.func<-function(x) x+quad.coef*x^2
lines(seq(0,6,.01), quad.func(seq(0,6,.01)), col=1, lty=1, lwd=2) #solid
black line
```

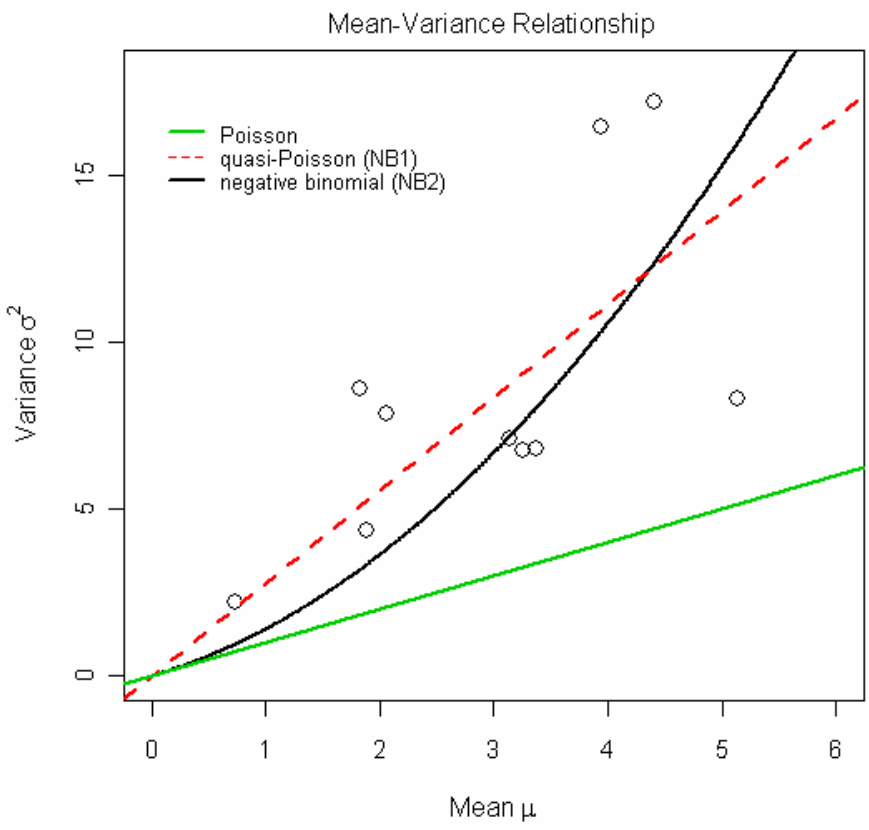
NB1 – negative binomial regression (quasi-Poisson) $\sigma^2 = k\mu$

```
lm(vars~means-1)
Call:
lm(formula = vars ~ means - 1)
Coefficients:
means
2.786
abline(0, coef(lm(vars~means-1)), col=2, lty=2, lwd=2) #dotted red line
```

Poisson model $\sigma^2 = \mu$

```
abline(0,1,col=3,lty=1,lwd=2)

#Adding the legend
legend(0,17, c('Poisson', 'quasi-Poisson (NB1)', 'negative binomial (NB2)'),
col=c(3,2,1), lwd=c(2,1,2), lty=c(1,2,1), cex=c(.8,.8,.8), bty='n')
```

2. Superimpose in a scatter plot of the variance versus the mean potential models for the mean-variance relationship drawn from appropriate probability models.



3. Based on your answer to question 2, what probability model might be most appropriate as a probability generating mechanism for these data?

Both NB models obviously have a better fit to the scatterplot than the Poisson, so we can reject the Poisson as a suitable model for our probability generating mechanism. To decide between the two NB models, I tried getting values for R^2:

```
# R2 for the NB1 model
(summary(lm(vars~means-1)))$r.squared
[1] 0.8683304
# R2 for the NB2 model
(summary(lm(vars~offset(means)+I(means^2)-1)))$r.squared
[1] 0.8094694
```

The R^2 for the NB1 is actually higher than for the NB2, so judging from this the NB1 fits the data better. However, since the mean-variance relationship is heteroscedastic, I'll opt for using the NB2. There are only 10 points in the scatterplot to use for determining a probability generating mechanism, so there isn't much data to go on.

4. Fit your probability model to the data. Obtain your answer by minimizing the negative loglikelihood of your data using your chosen probability model. I would like you to fit two models.

1. A model without predictors.

```
nbin1<-function(data,p)
{
mu<-p[1]
theta<-p[2]
negloglike<- -sum(log(dnbinom(data$num.satellites,mu=mu,size=theta)))
negloglike
}
> nlm(function(p) nbin1 (crabs,p),c(2.9,.75))->out1
> out1
$minimum
[1] 383.7046
$estimate
[1] 2.9190728 0.7577585
$gradient
[1] -1.135282e-05  5.684342e-07
$code
[1] 1
$iterations
[1] 4
```

2. A model in which you allow the mean number of satellites to vary as a function of the female's carapace width.

```
nbin2<-function(data,p)
{
mu<-exp(p[1]+p[3]*data$width)
theta<-p[2]
negloglike<- -sum(log(dnbinom(data$num.satellites,mu=mu,size=theta)))
```

```
negloglike
}
nlm(function(p) nbin2 (crabs,p),c(0,1,1))->out2
out2
$minimum
[1] 375.6455
$estimate
[1] -4.0526238  0.9045677  0.1920775
$gradient
[1] -2.356423e-08 -5.002221e-08 -5.866241e-07
$code
[1] 1
$iterations
[1] 30
```

5. Compare the two models you found in question 4 using AIC.

```
my.aic<-function(output) -2*(-output$minimum) + 2*length(output$estimate)
my.aic(out1)
[1] 771.4092
my.aic(out2)
[1] 757.291
```

According to the AICs, the second model where the mean is allowed to vary as a function of width has the lowest AIC, so I will consider it the better of the two.

6. Refit the two models of question 4 using an appropriate generalized linear model. Use a significance test to assess whether female carapace width is a significant predictor of the number of satellite males.

Model with no predictors

```
library(MASS)
model1<-glm.nb(num.satellites~1, data=crabs)
summary(model1)
Call:
glm.nb(formula = num.satellites ~ 1, data = crabs, init.theta =
0.75775903807072,
    link = log)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5471  -1.5471  -0.2720   0.4659   1.7999
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.07127    0.09802   10.93   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.7578) family taken to be 1)
    Null deviance: 192.93  on 172  degrees of freedom
Residual deviance: 192.93  on 172  degrees of freedom
AIC: 771.41
Number of Fisher Scoring iterations: 1
              Theta:  0.758
```

```
          Std. Err.:  0.126
2 x log-likelihood:  -767.409
```

Model with width as a predictor

```
model2<-glm.nb(num.satellites~width, data=crabs)
summary(model2)
Call:
glm.nb(formula = num.satellites ~ width, data = crabs, init.theta =
0.904568080033865,
    link = log)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7798  -1.4110  -0.2502   0.4770   2.0177
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.05251    1.17143  -3.459 0.000541 ***
width        0.19207    0.04406   4.360 1.30e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.9046) family taken to be 1)
    Null deviance: 213.05  on 172  degrees of freedom
Residual deviance: 195.81  on 171  degrees of freedom
AIC: 757.29
Number of Fisher Scoring iterations: 1
Correlation of Coefficients:
      (Intercept)
width -1.00
              Theta:  0.905
          Std. Err.:  0.161
 2 x log-likelihood:  -751.291
```

The significance test in the summary for the second model says that width has a value for Pr of 1.30e-05. This number has 3 stars next to it, which suggests that the value has a high level of significance. Thus, the significance test is telling me that width should be a significant predictor of the number of satellite males.

7. Hard: Check the fit of the best of the two models you've fit.

Finding expected values

```
exp.prob<-function(x) mean(dnbinom(x, mu=fitted(model2), size=model2$theta))
tail.prob<-function(x) mean(1-pnbinom(x, mu=fitted(model2),
size=model2$theta))
sapply(0:15,exp.prob)
 [1] 0.291810010 0.191020753 0.132979498 0.095034058 0.069107504 0.050944211
 [7] 0.037991838 0.028623861 0.021766360 0.016692956 0.012903271 0.010047449
[13] 0.007877736 0.006216708 0.004936003 0.003941892
sum(sapply(0:15,exp.prob))
[1] 0.9818941
tail.prob(15)
[1] 0.01810589
sum(sapply(0:15,exp.prob))+tail.prob(15)
```

```
[1] 1

expprobs<-c(sapply(0:15,exp.prob),tail.prob(15))
expprobs
 [1] 0.291810010 0.191020753 0.132979498 0.095034058 0.069107504 0.050944211
 [7] 0.037991838 0.028623861 0.021766360 0.016692956 0.012903271 0.010047449
[13] 0.007877736 0.006216708 0.004936003 0.003941892 0.018105892
expfreqs<-expprobs*173
expfreqs
 [1] 50.4831317 33.0465902 23.0054531 16.4408921 11.9555982  8.8133485
 [7]  6.5725880  4.9519279  3.7655803  2.8878813  2.2322659  1.7382086
[13]  1.3628484  1.0754905  0.8539286  0.6819474  3.1323193
sum(expfreqs)
[1] 173
```

The expected frequencies exceed 5 only from 0 through 6. Thus I will pool together 7 and 8, 9:11, and 12:15 including the tail probability.

Pooling expected values

```
expected<-c(
sapply(0:6,exp.prob),
sum(exp.prob(7),exp.prob(8)),
sum(exp.prob(9),exp.prob(10),exp.prob(11)),
sum(exp.prob(12),exp.prob(13),exp.prob(14),exp.prob(15),tail.prob(15))
)
exp.pooled<-expected*173
exp.pooled
[1] 50.483132 33.046590 23.005453 16.440892 11.955598  8.813348  6.572588
[8]  8.717508  6.858356  7.106534
```

Finding observed values

```
table(crabs$num.satellites)
 0  1  2  3  4  5  6  7  8  9 10 11 12 14 15
62 16  9 19 19 15 13  4  6  3  3  1  1  1  1

#adding in zero counts for 13
crabstable<-c(table(crabs$num.satellites)[0:13], 0,
table(crabs$num.satellites)[14:15])
crabstable
 0  1  2  3  4  5  6  7  8  9 10 11 12    14 15
62 16  9 19 19 15 13  4  6  3  3  1  1  0  1  1
names(crabstable)[14]=13
crabstable
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
62 16  9 19 19 15 13  4  6  3  3  1  1  0  1  1
```

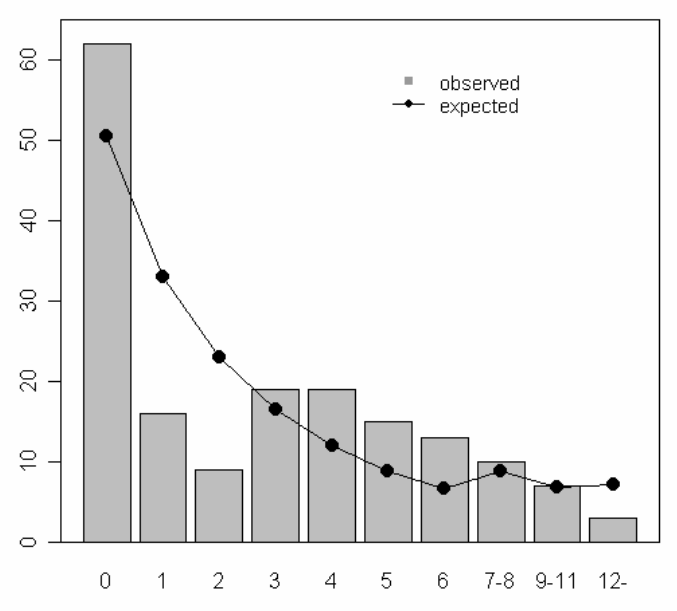
Pooling observed values

```
obs.pooled<-c(
crabstable[1:7],sum(crabstable[8:9]),sum(crabstable[10:12]),
sum(crabstable[13:16]) )
obs.pooled
 0  1  2  3  4  5  6
62 16  9 19 19 15 13 10  7  3
names(obs.pooled)<-c(0:6,'7-8','9-11','12-')
```

```
obs.pooled
   0    1    2    3    4    5    6  7-8 9-11  12-
  62   16    9   19   19   15   13   10    7    3
```

Plotting expected and observed values

```
coords<-barplot(obs.pooled,ylim=c(0,65))
points(coords,exp.pooled,pch=16,cex=1.5)
lines(coords,exp.pooled)
legend(coords[6],60, c('observed','expected'), pch=c(15,16),
col=c('gray60',1), lty=c(NA,1), cex=c(.9,.9), bty='n')
box()
```




```
pearson<-sum((obs.pooled-exp.pooled)^2/exp.pooled)
pearson
[1] 37.68878
df<-length(obs.pooled)-1-3 #(n-1-p)
df
[1] 6
p.val<-1-pchisq(pearson,df)
p.val
[1] 1.292206e-06
qchisq(.95,df)
[1] 12.59159
```

The extremely small p-value is far below 0.05, so it is significant in indicating a significant lack of fit of this model to the data. The observed value of the test statistic was 37.7, much greater than the critical 5% value given by qchisq (12.59), which means this model should be rejected.

I could already tell this model should be rejected because of the poor fit it has for lower counts of satellite males. The observed counts almost look like they require a model that would incorporate two different distributions in order to have a decent fit, given the high number of zeros and the rise around 3-5 counts of satellite males.

Problem 2

1. Fit a gamma distribution to the hurricane precipitation data.

```
hurrs<-
read.table('http://www.unc.edu/courses/2006spring/ecol/145/001/data/midterm/h
urricanes.csv', header=TRUE, sep=',')
names(hurrs)
[1] "Year"          "Name"          "Location"
[4] "Precipitation"
hurrs$Precipitation
 [1] 31.00  2.82  3.98  4.02  9.50  4.50 11.40 10.71  6.31  4.95
[11]  5.64  5.51 13.40  9.72  6.47 10.16  4.21 11.60  4.75  6.85
[21]  6.25  3.42 11.80  0.80  3.69  3.10 22.22  7.43  5.00  4.58
[31]  4.46  8.00  3.73  3.50  6.20  0.67
```

$$E(X) = \frac{\alpha}{\beta}$$

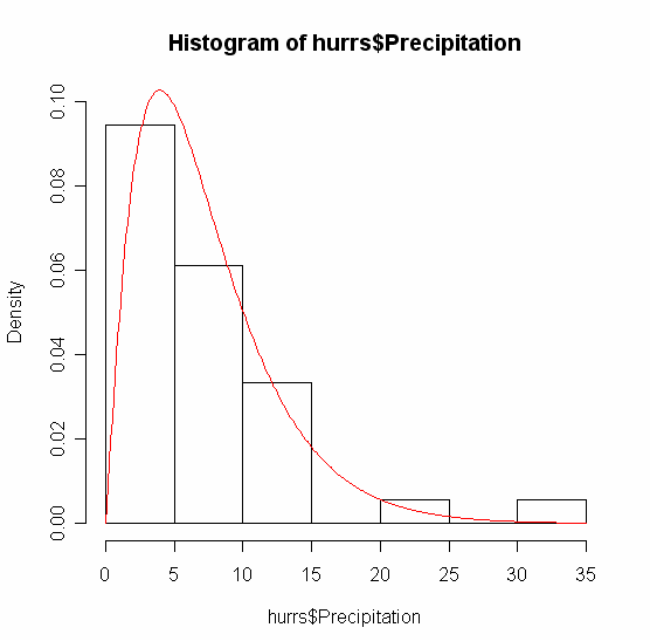
$$\mathrm{Var}(X) = \frac{\alpha}{\beta^2}$$

Solving for α and β, **$\alpha = \mu^2/\mathrm{Var}(X)$**, and **$\beta = \mu/\mathrm{Var}(X)$**. #shape = alpha, scale = beta. In R, scale = 1/rate.

```
shape = (mean(hurrs$Precipitation)^2) / var(hurrs$Precipitation)
shape
[1] 1.589584
scale = mean(hurrs$Precipitation)/ var(hurrs$Precipitation)
scale
[1] 0.2181248

gamma.func<-function(p) {
sh<-p[1]
sc<-p[2]
negloglike<- -sum(log(dgamma(hurrs$Precipitation,shape=sh,scale=1/sc)))
negloglike
}
nlm(function(p) gamma.func(p),c(shape,scale))->out.gamma
out.gamma
$minimum
[1] 102.3594
$estimate
[1] 2.1871929 0.3001288
$gradient
[1] -6.841660e-06 -3.667822e-05
$code
[1] 1
$iterations
[1] 8
my.aic(out.gamma)
[1] 208.7188
```

2. **Superimpose the distribution estimated in question 1 on top of a histogram of the data.**

```
histogram<-hist(hurrs$Precipitation,probability=TRUE,ylim=c(0,.1))
lines(
seq(0,35,.1),
dgamma(seq(0,35,.1),shape=out.gamma$estimate[1],scale=1/out.gamma$estimate[2]),
col=2
)
```



3. **Use a generalized linear model to fit the same distribution you did in question 1. Find one thing from the output of the glm fit (you may need to use a function to extract it) that matches one thing from your output of fitting the distribution directly in question 1.**

```
precip.glm<-glm(Precipitation~1, data=hurrs, family=Gamma)
summary(precip.glm)
Call:
glm(formula = Precipitation ~ 1, family = Gamma, data = hurrs)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7196  -0.5434  -0.2565   0.2837   1.9005
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.13722    0.01814   7.565 7.24e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for Gamma family taken to be 0.629095)
    Null deviance: 17.689  on 35  degrees of freedom
Residual deviance: 17.689  on 35  degrees of freedom
AIC: 208.82
Number of Fisher Scoring iterations: 6

logLik(precip.glm) #the loglikelihood from the glm
'log Lik.' -102.4119 (df=2)
```

The negative loglikelihood from question 1 was:

```
out.gamma$minimum
[1] 102.3594
```

so these match closely enough except for the negative sign.

Problem 3

1. Using only the egg data from 1935, find a distribution that is appropriate for these data, estimate its parameter(s), and check its fit.

```
gallfly<-
read.table('http://www.unc.edu/courses/2006spring/ecol/145/001/data/midterm/g
allfly.txt', header=TRUE, sep='')
> names(gallfly)
[1] "Freq"   "Number" "Year"   "Type"

num.eggs35<-c(29,38,36,23,8,5,5,2,1,0,0,1)
sum(num.eggs35)
[1] 148
eggs35.data<-rep(1:12,num.eggs35)
eggs35.data
  [1]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [21]  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2
 [41]  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
 [61]  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3  3  3  3  3
 [81]  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
[101]  3  3  3  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
[121]  4  4  4  4  4  4  5  5  5  5  5  5  5  5  6  6  6  6  6  7
[141]  7  7  7  7  8  8  9 12
```

Fitting truncated Poisson distribution

```
poi.trunc<-function(p) -sum(ifelse(eggs35.data==0,  0,
log(dpois(eggs35.data,lambda=p)/(1-dpois(0,lambda=p)))))
mean(eggs35.data)
[1] 3.02027
nlm(function(p) poi.trunc(p), 3)->out35
out35
$minimum
[1] 276.5876
$estimate
[1] 2.844621
$gradient
[1] -2.731645e-05
$code
[1] 1
$iterations
[1] 4
```

Adding in zero counts for 0, 10, and 11

```
table(eggs35.data)
eggs35.data
 1  2  3  4  5  6  7  8  9 12
```

```
29 38 36 23  8  5  5  2  1  1
```

```
eggs35table<-c(0, table(eggs35.data)[1:9], 0, 0, table(eggs35.data)[10])
names(eggs35table)=c('0',1:9, '10', '11', 12)
eggs35table
0  1  2  3  4  5  6  7  8  9 10 11 12
0 29 38 36 23  8  5  5  2  1  0  0  1
```

Finding expected values

```
#without truncation of the Poisson
exp<-c(dpois(0:12,lambda=out35$estimate), 1-ppois(12,lambda=out35$estimate))
sum(exp)
[1] 1

#with truncation of the Poisson
expprobs<-c(0,
dpois(1:12,lambda=out35$estimate)/(1-dpois(0,lambda=out35$estimate)),
1-ppois(12,lambda=out35$estimate) #tail probability
)
expprobs
 [1] 0.000000e+00 1.756477e-01 2.498255e-01 2.368863e-01 1.684629e-01
 [6] 9.584263e-02 4.543932e-02 1.846538e-02 6.565875e-03 2.075269e-03
[11] 5.903354e-04 1.526619e-04 3.618876e-05 9.322215e-06
sum(expprobs)
[1] 0.9999994 #doesn't quite add up to 1, but it is fairly close

expfreqs<-expprobs*173
expfreqs
 [1]  0.000000000 30.387049486 43.219816744 40.981330116 29.144086173
 [6] 16.580774835  7.861002872  3.194510343  1.135896327  0.359021593
[11]  0.102128030  0.026410502  0.006260655  0.001612743
sum(expfreqs)
[1] 172.9999 #very close to 173
```

The expected frequencies exceed 5 only from 1 through 7. Thus I will pool together everything that is 8 and above, including the tail probability.

Pooling expected values

```
expected<-c(expprobs[1:7],sum(expprobs[8:14]))
expected
[1] 0.00000000 0.17564768 0.24982553 0.23688630 0.16846293 0.09584263
[7] 0.04543932 0.02789503
exp.pooled<-expected*173
exp.pooled
[1]  0.000000 30.387049 43.219817 40.981330 29.144086 16.580775
[7]  7.861003  4.825840 #the last probability is close enough to 5
```

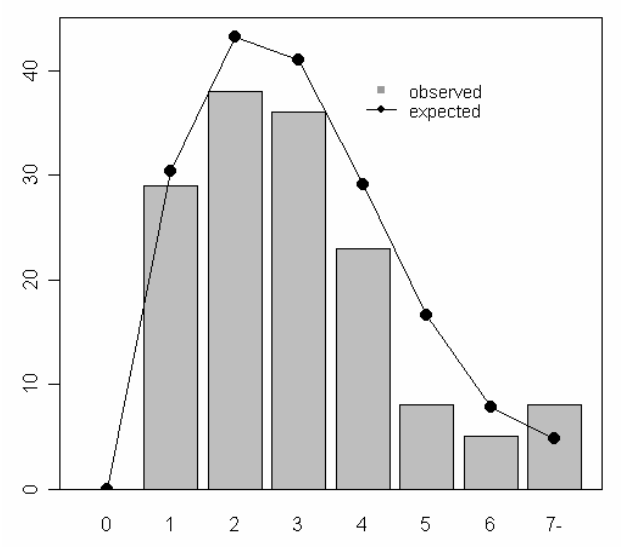
Pooling observed values

```
obs.pooled<-c(eggs35table[1:7],sum(eggs35table[8:12]))
obs.pooled
 0  1  2  3  4  5  6
 0 29 38 36 23  8  5  8
names(obs.pooled)<-c(0:6,'7-')
obs.pooled
```

```
 0  1  2  3  4  5  6 7-
 0 29 38 36 23  8  5  8
```

Plotting expected and observed values

```
coords<-barplot(obs.pooled,ylim=c(0,45))
points(coords,exp.pooled,pch=16,cex=1.5)
lines(coords,exp.pooled)
legend(coords[5],40, c('observed','expected'), pch=c(15,16),
col=c('gray60',1), lty=c(NA,1), cex=c(.9,.9), bty='n')
box()
```



Just from looking at the barplot, I see that the model consistently overpredicts values for the number of eggs that are counted (except in the last category, but this was the pooled category). However, the model does follow the general shape of the barplot.

Doing the goodness of fit test

```
exp.pooled
[1]  0.000000 30.387049 43.219817 40.981330 29.144086 16.580775
[7]  7.861003  4.825840
pearson.table<-(obs.pooled-exp.pooled)^2/exp.pooled
pearson.table
         0          1          2          3          4          5
       NaN 0.06331336 0.63041653 0.60548669 1.29528147 4.44066683
         6         7-
1.04125867 2.08777955
pearson<-sum(pearson.table[2:8])
pearson
[1] 10.16420

df<-length(obs.pooled)-1-1 #(n-1-p); p=1 because we only estimated lambda
df
[1] 6
p.val<-1-pchisq(pearson,df)
p.val
[1] 0.1179052
qchisq(.95,df)
[1] 12.59159
```

The p-value is above 0.05 (but barely), so it doesn't suggest a significant lack of fit of this model to the data. In short, this means the model has a good fit. Also, the observed value of the test statistic was 10.16, which is less than the critical 5% value given by qchisq (12.59). This also indicates that we shouldn't reject this model. However, when the model is superimposed on the barplot we see that it always overpredicts values, so even though the Pearson test suggests we shouldn't reject the model, this overprediction seems a bit problematic.

2. Does the same distribution work for the 1935 gall-cell counts?

To do this, I will use the model estimates from question 1 and pool the observed gall-cell counts in the same way as I did for the expected values from question 1. Then I will do a goodness-of-fit test to see how well the 1935 eggs distribution works for the 1935 gall-cell counts.

```
num.galls35<-c(287,272,196,79,29,20,2,0,1,0,0,0)
sum(num.galls35)
[1] 886
galls35.data<-rep(1:12,num.galls35)
galls35.data
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [26] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [51] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
. . .
[826] 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[851] 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6
[876] 6 6 6 6 6 6 6 6 7 7 9
```

Adding in zero counts for 0, 8, 10, 11, and 12

```
table(galls35.data)
galls35.data
  1   2   3   4   5   6   7   9
287 272 196  79  29  20   2   1

galls35table<-c(0, table(galls35.data)[1:7],0,table(galls35.data)[8],0,0,0)
names(galls35table)=c('0',1:7, '8', 9, '10', '11', '12')
galls35table
  0   1   2   3   4   5   6   7   8   9  10  11  12
  0 287 272 196  79  29  20   2   0   1   0   0   0
```

Finding and pooling expected values

The expected frequencies from the last model exceeded 5 only from 1 through 7. Thus I will pool together everything that is 8 and above, including the tail probability.

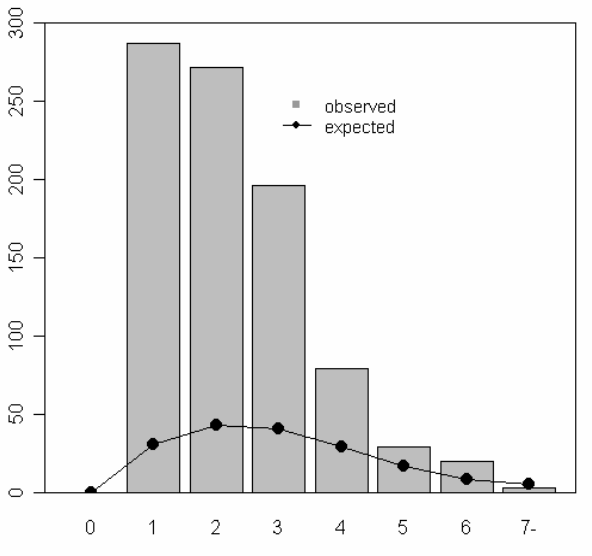
```
expected<-c(expprobs[1:7],sum(expprobs[8:14]))
expected
[1] 0.00000000 0.17564768 0.24982553 0.23688630 0.16846293 0.09584263
[7] 0.04543932 0.02789503
exp.pooled<-expected*173
exp.pooled
[1]  0.000000 30.387049 43.219817 40.981330 29.144086 16.580775
[7]  7.861003  4.825840

#I'll count the last probability in exp.pooled as being close enough to 5
```

Pooling observed values

```
obs.pooled<-c(galls35table[1:7],sum(galls35table[8:13]))
obs.pooled
  0   1   2   3   4   5   6
  0 287 272 196  79  29  20   3
names(obs.pooled)<-c(0:6,'7-')
obs.pooled
  0   1   2   3   4   5   6  7-
  0 287 272 196  79  29  20   3
```

Plotting expected and observed values

```
coords<-barplot(obs.pooled,ylim=c(0,300))
points(coords,exp.pooled,pch=16,cex=1.5)
lines(coords,exp.pooled)
legend(coords[4],260, c('observed','expected'), pch=c(15,16),
col=c('gray60',1), lty=c(NA,1), cex=c(.9,.9), bty='n')
box()
```



Just looking at the barplot, the fit looks very poor. The model consistently underpredicts values for the number of galls that should be counted, except for the last category where there were fewer observed counts than were expected from the model. The results from the goodness of fit test should also be poor.

Doing the goodness of fit test

```
exp.pooled  #these are the expected values from the model for question 1.
[1]  0.000000 30.387049 43.219817 40.981330 29.144086 16.580775  7.861003
[8]  4.825840
pearson.table<-(obs.pooled-exp.pooled)^2/exp.pooled
pearson.table
          0           1           2           3           4           5
        NaN 2167.0483803 1211.0271675  586.3837983   85.2870160    9.3021680
          6          7-
 18.7450957   0.6908004
pearson<-sum(pearson.table[2:7])
pearson
[1] 4077.794
```

```
df<-length(obs.pooled)-1-1 #(n-1-p); p=1 because we only estimated lambda
df
[1] 6
p.val<-1-pchisq(pearson,df)
p.val
[1] 0
qchisq(.95,df)
[1] 12.59159
```

The p-value was apparently so small that R had to return 0, so it is significant in indicating a significant lack of fit of this model to the data. In short, this model has a terrible fit. Also, the observed value of the test statistic was 4077.8, which is way beyond the critical 5% value given by qchisq (12.59). We should definitely reject this model.

The results of this goodness-of-fit test tells us that the truncated Poisson distribution that was fit for the 1935 eggs data does not work at all for this data. Perhaps there is a different probability generating mechanism at work for gall cells.

3. Is there any evidence of a change in distribution of eggs and gall-cells between 1935 and 1936? Based on your answer to question 2 you may want to answer this question separately for eggs and gall-cells.

The truncated Poisson distribution works reasonably for the 1935 eggs data, but not at all for the 1935 gall cells data. However, I will use it as suggested in the hints to compare AICs.

Inputting the eggs and galls data for 1936

```
num.eggs36<-c(22,18,18,11,9,6,3,0,1,0,0,0)
num.galls36<-c(90,96,57,26,10,4,5,0,1,0,0,0)
```

Combining eggs and galls for both years

```
eggs<-num.eggs35+num.eggs36
[1] 51 56 54 34 17 11  8  2  2  0  0  1
galls<-num.galls35+num.galls36
[1] 377 368 253 105  39  24   7   0   2   0   0   0

eggs.data<-rep(1:12,eggs)
galls.data<-rep(1:12,galls)

table(eggs.data)
eggs.data
 1  2  3  4  5  6  7  8  9 12
51 56 54 34 17 11  8  2  2  1
table(galls.data)
galls.data
  1   2   3   4   5   6   7   9
377 368 253 105  39  24   7   2
```

Fitting the 4 models

1. Common means truncated Poisson model for eggs

```
poi.trunc3<-function(p) -sum(ifelse(eggs.data==0,  0,
log(dpois(eggs.data,lambda=p)/(1-dpois(0,lambda=p)))))
mean(eggs.data)
```

```
[1] 3.025424
nlm(function(p) poi.trunc3(p), 3)->out3
out3
$minimum
[1] 443.1378
$estimate
[1] 2.850508
$gradient
[1] 1.774794e-06
$code
[1] 1
$iterations
[1] 3
```

2. Separate means truncated Poisson model for eggs

```
poi.trunc4<-function(p) {
year.dummy<-gallfly$Year-1935   #so that this reduces to 0 or 1
mylambda<-p[1]+p[2]*year.dummy
negloglike<- -sum(ifelse(eggs.data==0,  0,
log(dpois(eggs.data,lambda=mylambda)/(1-dpois(0,lambda=mylambda)))))
negloglike
}
mean(eggs.data)
[1] 3.025424
nlm(function(p) poi.trunc4(p), c(3,.01))->out4
out4
$minimum
[1] 443.0529
$estimate
[1] 2.80201469 0.09677711
$gradient
[1] -6.146847e-06 -1.176659e-05
$code
[1] 1
$iterations
[1] 5
```

3. Common means truncated Poisson model for galls

```
poi.trunc5<-function(p) -sum(ifelse(galls.data==0,  0,
log(dpois(galls.data,lambda=p)/(1-dpois(0,lambda=p)))))
mean(galls.data)
[1] 2.29617
nlm(function(p) poi.trunc5(p), 3)->out5
out5
$minimum
[1] 1780.905
$estimate
[1] 1.978739
$gradient
[1] 1.149084e-07
$code
[1] 1
$iterations
[1] 3
```

4. Separate means truncated Poisson model for galls

```
poi.trunc6<-function(p) {
```

```
year.dummy<-gallfly$Year-1935   #so that this reduces to 0 or 1
mylambda<-p[1]+p[2]*year.dummy
negloglike<- -sum(ifelse(galls.data==0,  0,
log(dpois(galls.data,lambda=mylambda)/(1-dpois(0,lambda=mylambda)))))
negloglike
}
mean(galls.data)
[1] 2.29617
nlm(function(p) poi.trunc6(p), c(2,.01))->out6
out6
$minimum
[1] 1780.900
$estimate
[1] 1.974121729 0.009239174
$gradient
[1] -6.864557e-05  6.559731e-04
$code
[1] 1
$iterations
[1] 7
```

Comparing the AICs of the four models

```
my.aic(out3)
[1] 888.2756  #Common means, eggs
my.aic(out4)
[1] 890.1058  #Separate means, eggs
my.aic(out5)
[1] 3563.811  #Common means, galls
my.aic(out6)
[1] 3565.801  #Separate means, galls
```

For both the eggs and galls data, the AIC is lower when the mean is not allowed to vary as a function of the year. The loglikelihoods are pretty much the same for both eggs models and both galls models (as expected, since the same data was used for each pair), so the difference in AIC must be due to a difference in the degrees of freedom. Thus, the better models for both datasets are the ones without any predictors. This means there is no evidence of a change in distribution for the eggs or galls between 1935 and 1936. The models I fit in questions 1 and 2 should work just as well with the 1936 eggs and galls as with the 1935 eggs and galls.