STAT2011 Statistical Models
Computer Exercise Week 11

Comments, where required, are indicated in **boldface**.

1. Repeat the Poisson model fit and check (obtaining standardised residuals `sr1`) for the milk/bacterial clumps data from week 9, question 2.

2. This week we shall fit the following form of the *negative binomial distribution*:

$$P(Y=y) = \binom{y+r-1}{y}(1-p)^y p^r\,, \quad \text{for } y = 0, 1, 2, \ldots \tag{1}$$

where the unknown parameters are $0 < p < 1$ and $r > 0$; note this permits r to not necessarily be an integer, in which case we interpret the binomial coefficient appearing in the pdf as

$$\binom{y+r-1}{y} = \frac{y+r-1}{y}\,\frac{y+r-2}{y-1}\cdots\frac{r}{1}\,. \tag{2}$$

From lectures we have seen that if Y has this pdf then $E(Y) = r(1-p)/p$ and $Var(Y) = r(1-p)/p^2$. Use these formulae to derive method-of-moments estimators of r and p as functions of the sample mean `m` and variance `v`. Call the resulting estimates `r.mom` and `p.mom` respectively.

3. Using the `dnbinom(...,size=r.mom,prob=p.mom)` function obtain fitted proportions and hence expected frequencies; note you will need to pool classes as you did in week 9 so that we have 12 classes: 0,1,2,...,10 and 11+.

4. Compute standardised residuals `sr2` and then form a 5-column matrix using `cbind()` so as to display and compare the observed frequencies; expected frequencies and standardised residuals for the Poisson fit; expected frequencies and standardised residuals for the negative binomial fit.

5. We shall now obtain *maximum likelihood estimates* (mle's) `r.mle` and `p.mle`. The mle for r is obtained by solving the equation

$$n\log\left(1+\frac{r}{\mathtt{m}}\right) = \sum_{j=1}^{y_{\max}} M_j\left(\frac{1}{j+r-1}\right) \tag{3}$$

where M_j is the number of data values $\geq j$. The solution $\hat{r}$ is then plugged into the formula $\hat{p} = \hat{r}/(\mathtt{m}+\hat{r})$ to obtain an estimate of p.

Some code is given below to solve the above equation in r; you need to have a vector `y` of the "original data" (i.e. ungrouped), which can be obtained using

```
x=c(0:10,19)
freq=c(56, 104, 80, 62, 42, 27, 9, 9, 5, 3, 2, 1)
y=rep(x,freq)
```

6. Here is the code that actually "solves" the equation (3) above. It uses the (iterative) Newton-Raphson method, starting at the method-of-moments estimator obtained earlier (**this is not examinable**, just copy-and-paste it into your `process()` file):

```
M=0
for(i in 1:max(y)) M[i]=sum(y>=i)
m=mean(y)
v=var(y)
r.mom=(m^2)/(v-m)
r=r.mom
eps=1
n=length(y)
tol=.0000000001
while(eps>tol){
rold=r
j=1:max(y)
g=sum((1/(j+r-1))*M)-n*log(1+m/r)
gd=(n*m)/(r*(m+r)) - sum( M/((j+r-1)^2))
r=r-g/gd
eps=abs(r-rold)/rold
print(r)
}
```

7. Having obtained your estimate `r.mle`, obtain the mle of the parameter p, calling it `p.mle`.

8. Obtain expected frequencies using the mle's and hence standardised residuals `sr3`, and add these to the 5 columns from the matrix in question 4 to obtain a new 7-column matrix. **Does the fit seem to be better?**

9. We shall perform a simulation to compare the performance of the method-of-moments and maximum-likelihood estimates. We shall simulate 1000 samples from the mle fit:

```
r0=r.mle
p0=p.mle
ests=matrix(0,1000,4)
for (k in 1:1000){
y=rnbinom(400,r0,p0)
...       # the code from question 6 above goes here with one small change:
...       # REMOVE the line ‘‘print(r)’’.
p.mom=...
r.mle=...
p.mle=...
ests[k,]=c(r.mom,r.mle,p.mom,p.mle)
}
```

10. Obtain average squared errors using

```
true=matrix(c(r0,r0,p0,p0),1000,4,byrow=T)
sq.errors=(ests-true)^2
avg.sq.err=apply(sq.errors,2,mean)
```

Comment on the relative performance of the estimators.