Poisson Regression

The Training Data

Office workers at a large insurance company are randomly assigned to one of 3 computer use training programmes, and their number of calls to IT support during the following month is recorded. Additional information on each worker includes years of experience and score on a computer literacy test (out of 100). It is reasonable to model calls to IT support as a Poisson process, and the question is whether training programme affects the rate of the process.

Could test H_0: $\lambda_1 = \lambda_2 = \lambda_3$ with a likelihood ratio test, but ...

```
> train = read.table("training.data.txt")
> train[1:4,]
  Program Experience Score Support
1       A       3.92    60       6
2       A       5.83    64       3
3       A       0.92    51       8
4       A       8.50    58       2
> attach(train)
> table(Support)
Support
 0  1  2  3  4  5  6  7  8  9 10 11 12
 6 27 42 61 70 39 23 17  9  2  2  1  1
> aggregate(Support,by=list(Program),FUN=mean)
  Group.1    x
1       A 4.07
2       B 3.47
3       C 4.05
> aggregate(Support,by=list(Program),FUN=length)
  Group.1   x
1       A 100
2       B 100
3       C 100
>
```

```
> model1 = glm(Support ~ Program, family=poisson)
> summary(model1)

Call:
glm(formula = Support ~ Program, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8531  -0.6319  -0.0348   0.4552   3.1765

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.403643   0.049567  28.318   <2e-16 ***
ProgramB    -0.159488   0.073066  -2.183   0.0291 *
ProgramC    -0.004926   0.070185  -0.070   0.9440
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 330.39  on 299  degrees of freedom
Residual deviance: 324.26  on 297  degrees of freedom
AIC: 1250.2

Number of Fisher Scoring iterations: 4

> anova(model1,test="Chisq") # Overall likelihood ratio test
Analysis of Deviance Table

Model: poisson, link: log

Response: Support

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                      299     330.39
Program  2    6.122       297     324.26  0.04684 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> # Include covariates
> model2 = glm(Support ~ Score+Experience+Program, family=poisson)
> summary(model2)

Call:
glm(formula = Support ~ Score + Experience + Program, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9625  -0.6957  -0.1018   0.5362   2.9386

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.992744   0.159223  12.515  < 2e-16 ***
Score       -0.009205   0.003019  -3.049  0.00230 **
Experience  -0.028014   0.010317  -2.715  0.00662 **
ProgramB    -0.170519   0.073163  -2.331  0.01977 *
ProgramC    -0.007833   0.070218  -0.112  0.91118
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 330.39  on 299  degrees of freedom
Residual deviance: 305.90  on 295  degrees of freedom
AIC: 1235.8

Number of Fisher Scoring iterations: 4

> anova(model2,test="Chisq") # Sequential
Analysis of Deviance Table

Model: poisson, link: log

Response: Support

Terms added sequentially (first to last)

           Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                         299     330.39
Score       1   9.9766       298     320.41 0.001585 **
Experience  1   7.6333       297     312.78 0.005730 **
Program     2   6.8767       295     305.90 0.032118 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> # Wald test for program
>
> Wtest
function(L,Tn,Vn,h=0) # H0: L theta = h
# Note Vn is the estimated asymptotic covariance matrix of Tn,
# so it's Sigma-hat divided by n. For Wald tests based on numerical
# MLEs, Tn = theta-hat, and Vn is the inverse of the Hessian.
     {
     Wtest = numeric(3)
     names(Wtest) = c("W","df","p-value")
     r = dim(L)[1]
     W = t(L%*%Tn-h) %*% solve(L%*%Vn%*%t(L)) %*%
          (L%*%Tn-h)
     W = as.numeric(W)
     pval = 1-pchisq(W,r)
     Wtest[1] = W; Wtest[2] = r; Wtest[3] = pval
     Wtest
     }

>
> Lprog = rbind(c(0,0,0,1,0),
+               c(0,0,0,0,1) )
> Wtest(L=Lprog,Tn=model2$coefficients,Vn=vcov(model2))
         W         df    p-value
6.73350088 2.00000000 0.03450157
> # Compare G^2 = 6.8767, df=2, p=0.032118
```