Poisson regression

Maths and Statistics Help Centre

Poisson regression is a technique similar to linear regression but the response variable (y) follows a Poisson distribution. Another assumption is that the logarithm of the expected value of the response can be modelled using a linear combination of unknown parameters. Poisson modelling is used for count data or contingency table data. Poisson models are also called log-linear models.

The maths:

For multiple Poisson regression a model of the following form can be used to predict the value of a response variable y using the values of a number of explanatory variables *x*:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_q x_q + \varepsilon$$

$$\beta_0 = \text{constant/intercept}, \beta_1, ..., \beta_q = \text{coefficients for q explanatory variables } x_1, ..., x_q$$

However if y is not normal a generalized linear model can be used. The assumptions are that y comes from a distribution such as binomial, Poisson, gamma, etc. The observations of y are independent and the mean of (y-μ) is related to the linear predictor η by a smooth invertible function g(). In the case of Poisson regression this is:

$$\eta = \log(\mu)$$

This means that the covariates affect the Poisson mean in a multiplicative way.

In *R* the command for Poisson regression is:

glm (*y***~** x_1 **+** x_2 **+..., data =** dataset, **family = poisson)**

Example

Use the data on eye colour in Glasgow, Sheffield and London given below.

	Blue	Brown	Green	Other
Glasgow	43	62	48	27
Sheffield	35	26	30	29
London	27	39	61	33

First create the dataset in R using the following commands:

```
# Define the variables and their corresponding data in R
count_eyes <- c(43,62,48,27,35,26,30,29,27,39,61,33)
city <- c("G","G","G","G","S","S","S","S","L","L","L","L")
colour_eye <- c(1,2,3,4,1,2,3,4,1,2,3,4)
# Combine all variables into a dataset
eye_data <- data.frame(count_eyes,city,colour_eye)
```

Declare the categorical explanatory variables as factors.

```
city_f <- as.factor(city)
colour_f <- as.factor(colour_eye)
```

We can now use this data to fit a Poisson model to the response variable (y) count_eyes using the explanatory variables (x_i) city_f and colour_f. This will allow produce a model which infers the number of people with a particular eye colour in a particular city given the information about the eye colour and city they live in.

```
# Define model1 to be the full model
model1 <- glm(count_eyes~city_f+colour_f, family=poisson)

# Obtain a summary of the model
summary(model1)

Call:
glm(formula = count_eyes ~ city_f + colour_f, family = poisson)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6528  -1.1260  -0.2491   1.2144   1.7478

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.7157     0.1136  32.708  < 2e-16 ***
city_fL      -0.1178     0.1087  -1.084 0.278353
city_fS      -0.4055     0.1179  -3.440 0.000581 ***
colour_f2     0.1902     0.1319   1.442 0.149247
colour_f3     0.2805     0.1293   2.170 0.030044 *
colour_f4    -0.1653     0.1441  -1.147 0.251206
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 43.924  on 11  degrees of freedom
Residual deviance: 18.232  on  6  degrees of freedom
AIC: 95.547

Number of Fisher Scoring iterations: 4
```

The output gives the coefficients for the parameters, the null deviance and the residual deviance.

The null model is the model with no explanatory variables, i.e. it only includes a constant.

```
# Define the null model model0
model0 <- glm(count_eyes~1, family=poisson)
# View the details of the null model
model0

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

Coefficients:
(Intercept)
      3.646

Degrees of Freedom: 11 Total (i.e. Null);  11 Residual
Null Deviance:      43.92
Residual Deviance: 43.92        AIC: 111.2
```

The deviance, also called G^2 (which corresponds to -2 * log-likelihood) is used to check the models. A small value of G^2 is preferable. A comparison between the deviance for the null model and the full model is sometimes called a G^2 test for goodness of fit. This involves comparing to difference of deviances (null model deviance – full model deviance) with a chi-squared value with degrees of freedom equal to the difference in the degrees of freedom of the two models.

The following command returns the p-value for the G^2 test for goodness of fit. A large p-value means that the full model is an improvement compared to the null model.

```
> pchisq(43.924-18.232  ,11-6)
[1] 0.9998976
```

Here we have a very large p-value and so we conclude that the full model is a great improvement on the null model.

To compare the two models we can also use an ANOVA. This will give the same result. Here we want a small p-value to show that the full model is an improvement compared to the null model. Notice that the p-value here and above sum to one, meaning that one is the greater than p-value and the other the less than p-value.

```
# Run an ANOVA on the two models using the Chi-squared test
anova(model1,model0,test="Chi")

Analysis of Deviance Table

Model 1: count_eyes ~ city_f + colour_f
Model 2: count_eyes ~ 1
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         6     18.232
2        11     43.924 -5  -25.692 0.0001024 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```