Basic Statistical Analysis

Rodrigo Labouriau

Department of Mathematics, Aarhus University

Module 3, Day 6 - Poisson Models - 2024
(One-way and two-ways classification structures)

1

General Remark

This material is only for internal use in the course.

Please, do not circulate and **do not record**.

Outline

Review, GLM in R

The Poisson one-way model

The Poisson two-ways model

Summing-up and the idea of generalised linear models

2

Important concepts:

- Statistical models
- Parameter in statistical model
- Point estimation
- Likelihood function and Maximum likelihood estimate
- Confidence interval and hypothesis test
- Likelihood ratio test
- One-way and two-ways binomial model.
- Binomial regression and binomial covariance analysis models
- Poisson regression models

Poisson as a law of rare events

- Suppose that we observe a binomial random variable, $Y \sim Bi(n,p)$.

  (*e.g.* throw n times a coin with probability p of head)

- Recall that the probability function of the distribution of Y is

$$P(Y=y) = \left(\begin{array}{c} n \\ y \end{array} \right) p^y (1-p)^{n-y} \, .$$

- We show that if p is very small the binomial distribution can be approximated by a Poisson distribution

  (in the sense given below).

Poisson as a law of rare events

Suppose that $n \to \infty$ and $p \to 0$ in such a way that np remains finite and tends to a number λ (*i.e.* , $np \to \lambda$), then

$$\begin{pmatrix} n \\ y \end{pmatrix} p^y (1-p)^{n-y} = \frac{n!}{(n-y)!y!}\frac{\lambda^y}{n^y}\left(1-\frac{\lambda}{n}\right)^{n-y}$$

$$\approx \frac{\sqrt{2\pi}\exp(-n)n^{n+1/2}}{\sqrt{2\pi}(n-y)^{n-y+1/2}\exp(-n+y)n^y}\frac{\lambda^r}{r!}\exp(-\lambda)$$

$$\approx \frac{1}{\left(1-\frac{y}{n}\right)^n \exp(y)}\frac{\lambda^r}{r!}\exp(-\lambda) \approx \frac{\lambda^r}{r!}\exp(-\lambda)\,.$$

Example: Horse-kicks

- The data are registers of Prussian military persons killed by kicks of horses.
- Ten corps observed (separately) during 20 years: 1875-1894

  (4 less representative corps were eliminated)
- The table below (next slide) displays the data
- The frequencies of number of deaths per year are:

Deaths					
0	1	2	3	4	≥ 5
109	65	22	3	1	0

- We are facing a rare event!

  (122 occurrences in 20 years 6.1 / year 0.61 per corp year)

  We will try to use the Poisson distribution!

The complete data:

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	Total
1875	0	0	0	0	1	1	0	0	1	0	3
1876	0	0	1	0	0	0	0	0	1	1	3
1877	0	0	0	0	1	0	0	1	2	0	4
1878	2	1	1	0	0	0	0	1	1	0	6
1879	0	1	1	2	0	1	0	0	1	0	6
1880	2	1	1	1	0	0	2	1	3	0	11
1881	0	2	1	0	1	0	1	0	0	0	5
1882	0	0	0	0	0	1	1	2	4	1	9
1883	1	2	0	1	1	0	1	0	0	0	6
1884	1	0	0	0	1	0	0	2	1	1	6
1885	0	0	0	0	0	0	2	0	0	1	3
1886	0	0	1	1	0	0	1	0	3	0	6
1887	2	1	0	0	2	1	1	0	2	0	9
1888	1	0	0	1	0	0	0	0	1	0	3
1889	1	1	0	1	0	0	1	2	0	2	8
1890	0	2	0	1	2	0	2	1	2	2	12
1891	0	1	1	1	1	1	0	3	1	0	9
1892	2	0	1	1	0	1	1	0	1	0	7
1893	0	0	0	1	2	0	0	1	0	0	4
1894	0	0	0	0	0	1	0	1	0	0	2

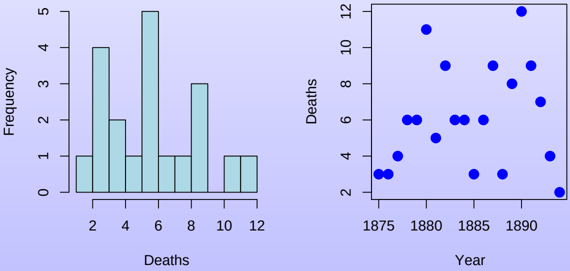
One-way Poisson model

- We start by analysing the total number of deaths per year

  We sum, for each year, the number of deaths occurred in each corp.

- The question is whether the number of deaths per year varies.
- Y_{year} number of deaths occurred in this year
- $Y_{year} \sim$ Poisson
- Two possible models:
  - Common intensity model: $Y_{year} \sim Po(\lambda)$
  - Saturated model: $Y_{year} \sim Po(\lambda_{year})$

```
> attach(kicks.data)
> print(kicks.data)
   year deaths
1  1875      3
2  1876      3
3  1877      4
4  1878      6
5  1879      6
6  1880     11
7  1881      5
8  1882      9
9  1883      6
10 1884      6
11 1885      3
12 1886      6
13 1887      9
14 1888      3
15 1889      8
16 1890     12
17 1891      9
18 1892      7
19 1893      4
20 1894      2
```


Frequency
0
1
2
3
4
5
2
4
6
8
10
12
Deaths
Deaths
2
4
6
8
10
12
1875
1880
1885
1890
Year

Fitting a Poisson model in R

- glm(formula= ... , family=poisson(link='log'))
- glm(formula= ... , family=poisson(link='identity'))
- formula:

  response variable $\sim$ explanatory variable1 $\sharp$ explanatory variable2 $\sharp$ ...

Common intensity model: $Y_{year} \sim Po(\lambda)$

```
> common <- glm(deaths ˜ 1 , family=poisson(link=’log’) )
> deviance(common)
[1] 25.25287
> summary(common)
....
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.80829    0.09054   19.97   <2e-16 ***
...
```

Saturated model: $Y_{year} \sim Po(\lambda_{year})$

```
> saturated <- glm(deaths ~ factor(year) , family=poisson(link='log') )
> deviance(saturated)
[1] -3.108624e-15
> summary(saturated)
...
Coefficients:
                     Estimate Std. Error   z value Pr(>|z|)
(Intercept)         1.099e+00  5.774e-01     1.903   0.0571 .
factor(year)1876    5.428e-17  8.165e-01  6.65e-17   1.0000
factor(year)1877    2.877e-01  7.638e-01     0.377   0.7064
...
factor(year)1894   -4.055e-01  9.129e-01    -0.444   0.6569
 ...
```

Testing differences of mortality among the years

Common intensity model: $Y_{year} \sim Po(\lambda)$
Saturated model: $Y_{year} \sim Po(\lambda_{year})$

```
> # common <- glm(deaths ˜ 1 , family=poisson(link=’log’) )
> # saturated <- glm(deaths ˜ 0 + factor(year), family=poisson(link=’log’) )
> anova(common, saturated, test="Chisq")
Analysis of Deviance Table
Model 1: deaths ˜ 1
Model 2: deaths ˜ factor(year)
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        19    25.2529
2         0 -3.109e-15 19  25.2529    0.1524
```

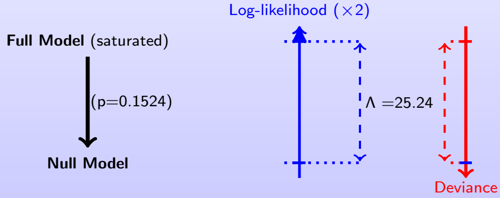
Conclusion: No evidence of differences in mortality (by horse kicks) among the years

Question: Which test did we use?

Summarising the analysis performed

Model	DF	Δ DF	Deviance	Δ Deviance	p-value
Full	0		0		
Null	19	19	25.24	25.24	0.1524

Summarising the analysis performed



The Poisson two-ways model

Now we analyse the data with different observations for each year and corp

```
      G1   G2   G3   G4  G5   G6   G7   G8  G9  G10  Total
1875   0    0    0    0   1    1    0    0   1    0      3
1876   0    0    1    0   0    0    0    0   1    1      3
1877   0    0    0    0   1    0    0    1   2    0      4
1878   2    1    1    0   0    0    0    1   1    0      6
1879   0    1    1    2   0    1    0    0   1    0      6
1880   2    1    1    1   0    0    2    1   3    0     11
1881   0    2    1    0   1    0    1    0   0    0      5
1882   0    0    0    0   0    1    1    2   4    1      9
1883   1    2    0    1   1    0    1    0   0    0      6
1884   1    0    0    0   1    0    0    2   1    1      6
1885   0    0    0    0   0    0    2    0   0    1      3
1886   0    0    1    1   0    0    1    0   3    0      6
1887   2    1    0    0   2    1    1    0   2    0      9
1888   1    0    0    1   0    0    0    0   1    0      3
1889   1    1    0    1   0    0    1    2   0    2      8
1890   0    2    0    1   2    0    2    1   2    2     12
1891   0    1    1    1   1    1    0    3   1    0      9
1892   2    0    1    1   0    1    1    0   1    0      7
1893   0    0    0    1   2    0    0    1   0    0      4
1894   0    0    0    0   0    1    0    1   0    0      2
```

- Now we analyse the data with different observations for each year and corp
(i.e. not the sum of the 10 corps)
- $Y_{year,corp}$ number of deaths occurred in this year at this corp
- $Y_{year,corp} \sim$ Poisson
- Several possible models!
- Saturated model: $Y_{year,corp} \sim Po(\lambda_{year,corp})$

- Several possible models!
- Saturated model:
  $Y_{year,corp} \sim Po(\lambda_{year,corp})$
- Multiplicative model:
  $Y_{year,corp} \sim Po(\lambda_{year,corp})$
  $\log(\lambda_{year,corp}) = \mu_{year} + \nu_{corp}$
- Only year:
  $Y_{year,corp} \sim Po(\lambda_{year,corp})$
  $\log(\lambda_{year,corp}) = \mu_{year}$
- Only corp:
  $Y_{year,corp} \sim Po(\lambda_{year,corp})$
  $\log(\lambda_{year,corp}) = \nu_{corp}$
- Null model:
  $Y_{year,corp} \sim Po(\lambda_{year,corp})$
  $\log(\lambda_{year,corp}) = \lambda$

```
> ################################################################################
> # Two-ways Poisson model
> ################################################################################

> attach(kicks.dataG)

> str(kicks.dataG)
'data.frame':   200 obs. of  3 variables:
 $ year  : int  1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 ...
 $ group : int  1 1 1 1 1 1 1 1 1 1 ...
 $ deaths: int  0 0 0 2 0 2 0 0 1 1 ...

> mean.year  <- tapply(deaths, factor(year), mean)   ; mean.year
1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894
 0.3  0.3  0.4  0.6  0.6  1.1  0.5  0.9  0.6  0.6  0.3  0.6  0.9  0.3  0.8  1.2  0.9  0.7  0.4  0.2

> mean.group <- tapply(deaths, factor(group), mean)  ; mean.group
   1    2    3    4    5    6    7    8    9   10
0.60 0.60 0.40 0.55 0.60 0.35 0.65 0.75 1.20 0.40
>
```

Fitting various models

```
> year <- factor(year); group <- factor(group)
> saturated <- glm(deaths ˜ year +  group + year:group,
+                                 family=poisson(link=’log’))
> multiplicative <- glm(deaths ˜ year +  group , family=poisson(link=’log’))
> only.year <- glm(deaths ˜ year , family=poisson(link=’log’))
> only.group <- glm(deaths ˜ group , family=poisson(link=’log’))
> null.model <- glm(deaths ˜ 1 , family=poisson(link=’log’))
> anova(multiplicative, saturated, test="Chisq")
  Analysis of Deviance Table
  Model 1: deaths ~ year + group
  Model 2: deaths ~ year + group + year:group
    Resid. Df  Resid. Dev  Df  Deviance  P(>|Chi|)
  1       171  171.640
  2         0  3.028e-10  171  171.640   0.472
```

```
> anova(only.year, multiplicative, test="Chisq")
Analysis of Deviance Table
Model 1: deaths ~ year
Model 2: deaths ~ year + group
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1       180    187.218
2       171    171.640   9   15.578     0.076
```

```
> anova(null.model, only.year, test="Chisq")
Analysis of Deviance Table
Model 1: deaths ~ 1
Model 2: deaths ~ year
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1       199    212.471
2       180    187.218  19   25.253     0.152
```

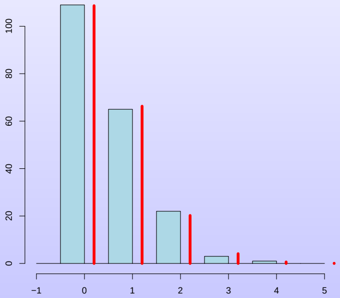
```
> anova(null.model, only.group, test="Chisq")
Analysis of Deviance Table
Model 1: deaths ~ 1
Model 2: deaths ~ group
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1       199    212.471
2       190    196.892   9   15.578     0.076
```

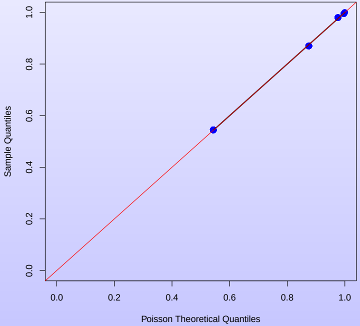
```
> anova(null.model, saturated, test="Chisq")
Analysis of Deviance Table
Model 1: deaths ~ 1
Model 2: deaths ~ year + group + year:group
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1       199    212.471
2         0  3.028e-10 199  212.471     0.244
```

Alternative (shorter) way to get the same tests

```
> # Alternative (shorter) way to get the same tests
> saturated <- glm(deaths ~ year +  group + year:group,
+                  family=poisson(link='log'))
> anova(saturated, test='Chisq')
Analysis of Deviance Table
Terms added sequentially (first to last)
            Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                          199    212.471
year        19   25.253       180    187.218     0.152
group        9   15.578       171    171.640     0.076
year:group 171  171.640         0  3.028e-10     0.472
```

```
> saturated <- glm(deaths ~ group + year+ year:group,
+                  family=poisson(link=’log’))
> anova(saturated, test=’Chisq’)
Analysis of Deviance Table
Model: poisson, link: log
Response: deaths
Terms added sequentially (first to last)
            Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                          199    212.471
group        9   15.578       190    196.892     0.076
year        19   25.253       171    171.640     0.152
group:year 171  171.640         0  3.028e-10     0.472
```

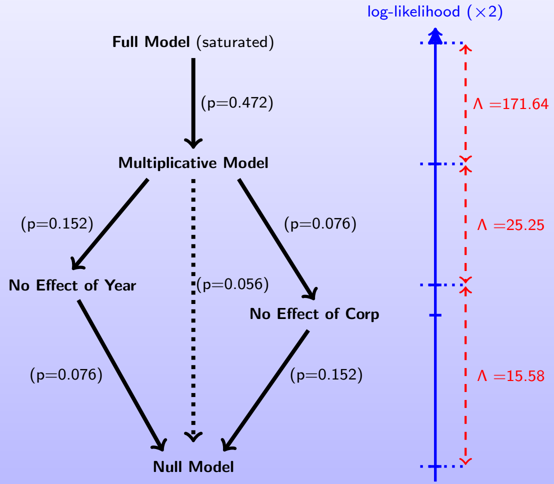

0
20
40
60
80
100
−1
0
1
2
3
4
5


Sample Quantiles
Poisson Theoretical Quantiles
0.0
0.2
0.4
0.6
0.8
1.0

Summarising the analysis performed

Model	DF	Δ DF	Deviance	Δ Deviance	p-value
Full	0		0		
Multiplicative	171	171	171.64	171.64	0.472
No effect of year	190	19	196.89	25.25	0.155
Null	199	9	212.47	15.58	0.076

Summarising the analysis performed



```
> # Calculating the expected number 4 deaths, assuming a common intensity
> intensity <- exp(coef(null.model)); intensity
(Intercept)
       0.61

> mean(deaths)
[1] 0.61

> prob.of.4 <- dpois(4, lambda=intensity); prob.of.4
[1] 0.003134646

> n.obs <- length(deaths); n.obs
[1] 200

> expected.4s <- n.obs *  prob.of.4; expected.4s
[1] 0.6269291

> # Calculating the expected number 3 deaths, assuming a common intensity
> prob.of.3 <- dpois(3, lambda=exp(coef(null.model))); prob.of.3
[1] 0.02055505

> expected.3s <- n.obs *  prob.of.3; expected.3s
[1] 4.111011

> # Calculating the expected number 2 deaths, assuming a common intensity
> prob.of.2 <- dpois(2, lambda=exp(coef(null.model))); prob.of.2
[1] 0.1010904

> expected.2s <- n.obs *  prob.of.2; expected.2s
[1] 20.21809
```

```
>  # Calculating the expected number 1 deaths, assuming a common intensity
> prob.of.1 <- dpois(1, lambda=exp(coef(null.model))); prob.of.1
[1] 0.331444

> expected.1s <- n.obs *  prob.of.1; expected.1s
[1] 66.2888

> # Calculating the expected number 0 deaths, assuming a common intensity
> prob.of.0 <- dpois(0, lambda=exp(coef(null.model))); prob.of.0
[1] 0.5433509

> expected.0s <- n.obs *  prob.of.0; expected.0s
[1] 108.6702
>
```

```
> Expected <- c(expected.0s, expected.1s, expected.2s, expected.3s, expected.4s)

> Expected
[1] 108.6701738  66.2888061  20.2180859   4.1110108   0.6269291

> Observed <- table (deaths) ; Observed
deaths
  0   1   2   3   4
109  65  22   3   1

> Chi.square.comp <- (Observed-Expected)^2 /Expected  ; Chi.square.comp
deaths
         0          1          2          3          4
0.00100106 0.02505734 0.15704840 0.30025341 0.22200573

> cbind(Observed, Expected, Observed-Expected, Chi.square.comp)
  Observed    Expected            Chi.square.comp
0      109 108.6701738  0.3298262      0.00100106
1       65  66.2888061 -1.2888061      0.02505734
2       22  20.2180859  1.7819141      0.15704840
3        3   4.1110108 -1.1110108      0.30025341
4        1   0.6269291  0.3730709      0.22200573

> Chisq.statistic <- sum(Chi.square.comp) ;  Chisq.statistic
[1] 0.7053659

> pchisq( Chisq.statistic , df= 5-1, lower.tail=F)
[1] 0.9506656
```

The glm function in R

- The glm function fits **G**eneralized **L**inear **M**odels, a large class of models.
- Examples of Generalized Linear Models:

  Binomial one/two/three ... way models, Binomial regressions, Poisson models, log-linear models (contingency tables), normal linear regression, normal anova, analysis of covariance models, gamma models, inverse gaussian models, some survival models, etc.
- Call: glm (formula= , family= , ...)
  Two important parts: formula and family

Generalized Linear Models

- Generalized Linear Models is a class of statistical models.
- A response variable, Y and
  a collection of explanatory variables, $X_1, \ldots, X_k$.
- The model specifies that Y follows a given probability laws and that the expectation of Y is related to the explanatory variables by

  $$g\left(E(Y)\right) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k,$$

  here g is a given function called the link function.

Specification of the family in glm

- The "family" parameter in glm specifies two characteristics of the generalized linear model:
  The class of probability laws and the link function.
- Common used distributions: Normal, Poisson, Binomial, Gamma, etc.
- Link function.

$$g\left(E(Y)\right) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k,$$

Common link functions: identity, log, inverse, logit, probit, etc.

identity: $E(Y) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k$

log: $\log\left(E(Y)\right) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k$

inverse: $\frac{1}{E(Y)} = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k$

Some common family specifications:

```
binomial(link = "logit")

binomial(link = "probit")

binomial(link = "cloglog")

gaussian(link = "identity")

Gamma(link = "inverse")

inverse.gaussian(link = "1/mu^2")

poisson(link = "log")
```

Can specify your own family

(but we will not do that at this stage)

The formula in glm

- glm (formula= , family= , ...)
- The formula specifies the response variable, the explanatory variables

  and the way the explanatory variables act on the expectation of the response variable.
- The general form is:

$$Y \sim X_1 \sharp \ldots \sharp X_k$$

  Here Y is the response variable (or matrix)
  $X_1, \ldots, X_k$ are the explanatory variables
  $\sharp$ are "operators" connecting the variables
- Possibilities for the operator $\sharp$: "+", "*" and ":"

The formula in glm: defining models with discrete explanatory variables

- Convention: y response variable, A and B factors.

- Single classification analysis of variance model of y, with classes determined by A.

```
  y ~ A
```

- Single classification analysis of variance model of the logarithm transformed y, i.e. log(y), with classes determined by A.

```
  log(y) ~ A
```

- Single classification analysis of variance model of the square-root transformed variable y, i.e. sqrt(y), with classes determined by A.

```
sqrt(y) ~ A
```

- Two factor additive model of y on A and B.

```
y ~ A + B
```

- Two factor non-additive model of y on A and B.

```
y ~ A*B
y ~ A + B + A:B
```

The formula in glm: defining models with continuous explanatory variables

- Convention: y response variable, A and B factors, x, z, and y numeric variable.
- Simple linear regression model of y on x.

  ```
  y ~ x   or   y ~ 1 + x
  ```
- Simple linear regression of y on x through the origin (that is, without an intercept term).

  ```
  y ~ 0 + x  or  y ~ -1 + x   or   y ~ x - 1
  ```
- Multiple regression of the transformed variable, log(y), on x1 and x2 (with an implicit intercept term).

  ```
  log(y) ~ x1 + x2
  ```
- Single classification analysis of covariance model of y, with classes determined by A, and with covariate x.

  ```
  y ~ A + x
  ```
- Separate simple linear regression models of y on x within the levels of A.

  ```
  y ~ A * x
  ```