Ejemplo Regresión Poisson y sobredispersión.

Los datos corresponden a accidentes de trenes(colisiones, descarrilamientos, alcances) en Reino Unido, y los ultimos años se ha ido privatizando el servicio, **se quiere saber si esto ha representado una mejoría o no.**

```
> datos
   traincollisions trainroadcollisions trainkm anio
1                0                    3     518   28
2                1                    3     516   27
3                0                    4     508   26
4                1                    3     503   25
5                1                    2     505   24
6                0                    4     487   23
7                1                    1     463   22
8                2                    2     437   21
9                1                    2     423   20
10               2                    4     415   19
11               0                    4     425   18
12               1                    4     430   17
13               2                    6     439   16
14               1                    2     431   15
15               4                    4     436   14
16               2                    4     443   13
17               1                    6     397   12
18               2                   13     414   11
19               0                    5     418   10
20               5                    3     389    9
21               2                    7     401    8
22               2                    3     372    7
23               2                    2     417    6
24               2                    2     430    5
25               3                    3     426    4
26               2                    4     430    3
27               1                    8     425    2
28               2                   12     426    1
29               5                    2     436    0>
```

En este caso la longitud del tramo **trainkm** funciona como medida de exposición al riesgo. Y su logaritmo será el OFFSET.

```
> modelo0<-glm(trainroadcollisions~anio,offset=log(trainkm) ,family=poisson)
Call:
glm(formula = trainroadcollisions ~ anio, family = poisson, offset = log(trainkm))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0580  -0.7825  -0.0826   0.3775   3.3873

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.21142    0.15892  -26.50  < 2e-16 ***
anio        -0.03292    0.01076   -3.06  0.00222 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 47.376  on 28  degrees of freedom
Residual deviance: 37.853  on 27  degrees of freedom
AIC: 133.52

Number of Fisher Scoring iterations: 5
```

Para ver si el modelo ajusta bien vemos que tan grande es su devianza.

```
#pvalue
> 1-pchisq(modelo0$deviance,27)
[1] 0.08021703
```

Como0.10>0.08>0.05 consideramos el modelo como “regular”.

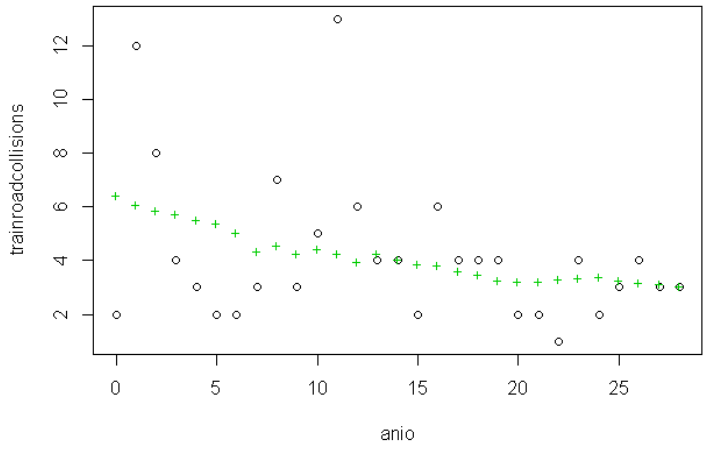
El modelo es:

$$e^{-4.21}(e^{-.0329})^x = 0.0148(0.968)^x = \mu_x$$

Claramente se ve que decrece la tasa de accidentes con el tiempo.

```
> plot(anio,trainroadcollisions)
> points(anio,fitted(modelo0),col=3,pch="+")
```

Esta gráfica muestra el modelo ajustado en verde y los valores observados en negro



ANALISIS DE RESIDUALES

A continuación se listan los residuales del modelo

```
> cbind(fitted(modelo0),modelo0$y,modelo0$residuals)
       [,1] [,2]        [,3]
1  3.055227    3 -0.01807632
2  3.145282    3 -0.04619039
3  3.200145    4  0.24994332
4  3.274689    3 -0.08388243
5  3.397735    2 -0.41137261
6  3.386283    4  0.18123629
7  3.327142    1 -0.69944175
8  3.245397    2 -0.38374265
9  3.246556    2 -0.38396259
10 3.291749    4  0.21515941
11 3.483884    4  0.14814390
12 3.642833    4  0.09804642
13 3.843540    6  0.56106072
14 3.899782    2 -0.48715079
15 4.077046    4 -0.01889752
16 4.281135    4 -0.06566840
17 3.964988    6  0.51324559
18 4.273146   13  2.04225482
19 4.458819    5  0.12137328
20 4.288340    3 -0.30042862
21 4.568568    7  0.53220880
22 4.380006    3 -0.31506951
23 5.074158    2 -0.60584592
24 5.407449    2 -0.63013986
25 5.536429    3 -0.45813444
26 5.775434    4 -0.30741141
27 5.899310    8  0.35609075
28 6.111080   12  0.96364623
29 6.463846    2 -0.69058668
```

Los residuals fluctuan alrededor del cero, para los conteos Poisson la desviación estándard va creciendo conforme crece μ, por eso conviene usar los

residuals estandarizados $= (y_i - \hat{\mu}_i) / \sqrt{\hat{\mu}_i (i - h_{ii})}$

estos tienen una distribución aproximadamente normal, así que, si estos residuales exceden al 2 o 3 en valor absoluto, debe ponerse atención y considerar hacer la regresión sin ellos.

Aquí la observación 18 excede a 2 por lo que se hace el análisis sin ella.

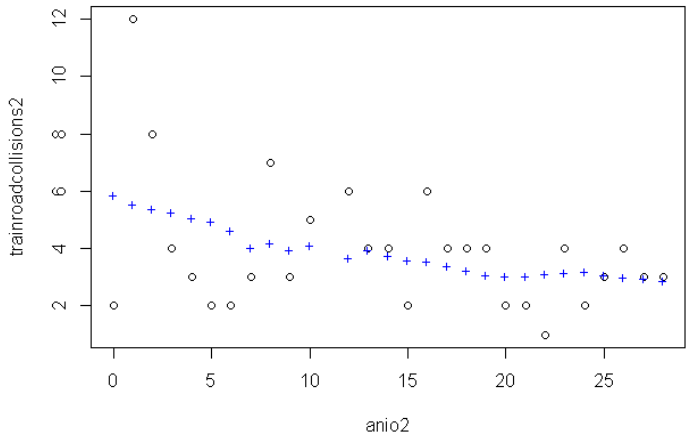
```
>
>
> trainroadcollisions2<- trainroadcollisions[-18]
> trainkm2<- trainkm[-18]
> anio2<-anio[-18]
>
> modelo2<-glm(trainroadcollisions2~anio2,offset=log(trainkm2) ,family=poisson)
> summary(modelo2)

Call:
glm(formula = trainroadcollisions2 ~ anio2, family = poisson,
    offset = log(trainkm2))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.86042  -0.65645   0.01735   0.45793   2.35466

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.30418    0.16726 -25.734  < 2e-16 ***
anio2       -0.03163    0.01117  -2.832  0.00463 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 33.831  on 27  degrees of freedom
```

```
Residual deviance: 25.702  on 26  degrees of freedom
AIC: 116.95

Number of Fisher Scoring iterations: 5
#pvalue
> 1-pchisq(modelo2$deviance,27)
[1] 0.5351741
```

0.53>>0.10
el ajuste del modelo mejoró bastante

Los residuales estandarizados todos son menores a 2

```
> cbind(fitted(modelo2),modelo2$y,modelo2$residuals)
       [,1] [,2]         [,3]
1  2.886771    3  0.039223311
2  2.968037    3  0.010769022
3  3.015924    4  0.326293481
4  3.082206    3 -0.026671117
5  3.193905    2 -0.373807381
6  3.179045    4  0.258239580
7  3.119505    1 -0.679436326
8  3.038947    2 -0.341877343
9  3.036121    2 -0.341264760
10 3.074425    4  0.301056477
11 3.249688    4  0.230887330
12 3.393581    4  0.178695869
13 3.575949    6  0.677876306
14 3.623607    2 -0.448063745
15 3.783444    4  0.057237805
16 3.967725    4  0.008134397
17 3.669994    6  0.634880060
18 4.116472    5  0.214632359
19 3.953989    3 -0.241272609
20 4.206949    7  0.663913566
21 4.028124    3 -0.255236480
22 4.660505    2 -0.570861966
23 4.960237    2 -0.596793438
24 5.072015    3 -0.408519144
25 5.284166    4 -0.243021476
26 5.390560    8  0.484075739
27 5.576884   12  1.151739245
28 5.891224    2 -0.660511955

>
> plot(anio2,trainroadcollisions2)
> points(anio2,fitted(modelo2),col=4,pch="+")
```



Sobredispersión

Para verificar si hay sobredispersión, se puede trabajar con distribución Binomial Negativa en vez de la Poisson, esta distribución permite que su varianza sea mayor que la media:

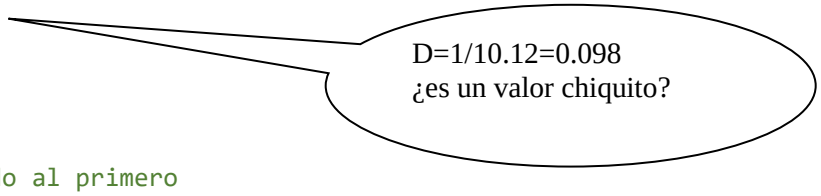
$$E(Y)=\mu \quad y \quad Var(Y)=\mu+D\mu^2$$

Podremos estimar D y ver si este valor es distinto o no de cero.
El comando de R cambia un poco para hacer este ajuste

```
> modelonb<-glm.nb(trainroadcollisions~anio+offset(log(trainkm) ) )
> summary(modelonb)

Call:
glm.nb(formula = trainroadcollisions ~ anio + offset(log(trainkm)),
    init.theta = 10.11828724, link = log)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.72370  -0.65461  -0.05868   0.32984   2.64065

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.19999    0.19584 -21.446  < 2e-16 ***
anio        -0.03367    0.01288  -2.615  0.00893 **
---
```

Las betas son muy parecidas a las del primer modelo,

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(10.1183) family taken to be 1)

    Null deviance: 32.045  on 28  degrees of freedom
Residual deviance: 25.264  on 27  degrees of freedom
AIC: 132.69

Number of Fisher Scoring iterations: 1


              Theta:  10.12
          Std. Err.:  8.00

 2 x log-likelihood:  -126.69
```



El modelo queda muy parecido al primero

$$e^{-4.19}(e^{-.033})^x = 0.0151(0.966)^x = \mu_x$$

Para probar si D es distinta de cero usamos esta otra librería:

```
> require(pscl)
> odTest(modelonb)
Likelihood ratio test of H0: Poisson, as restricted NB model:
n.b., the distribution of the test-statistic under H0 is non-standard
e.g., see help(odTest) for details/references

Critical value of test statistic at the alpha= 0.05 level: 2.7055
Chi-Square Test Statistic =  2.8293 p-value = 0.04628
```

(The p = .05 level is 2.7, instead of the usual 3.8)

Entonces la prueba con un alfa=0.05 se rechaza (H0. D=0) y entonces puede considerarse que sí hay sobredispersión.
El modelo que usa distribución binomial negative ajusta mejor.
Y la respuesta a la pregunta inicial es la misma:
La tasa de accidentes decrece con el tiempo, esto es con la privatización hay mejora.