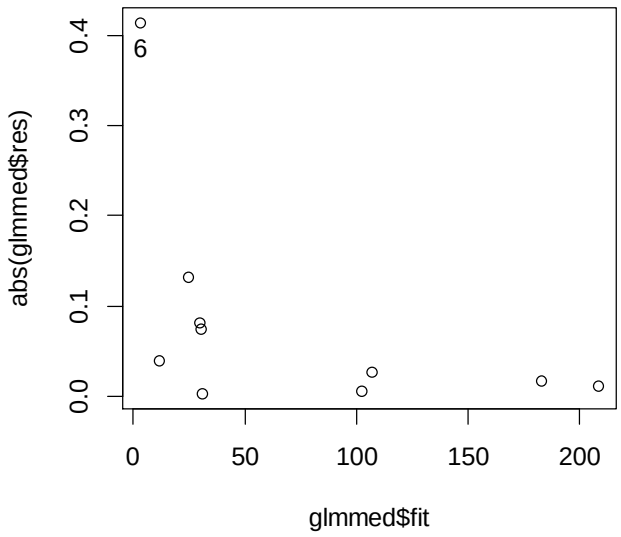
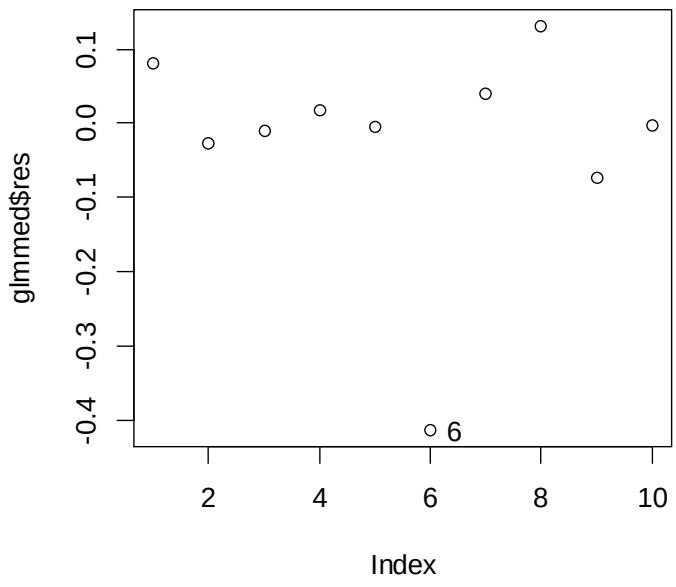
Ejemplo médicos

```
y=c(32,104,206,186,102,2,12,28,28,31)
smoke
 [1] 1 1 1 1 1 0 0 0 0 0
edad
 [1] 1 2 3 4 5 1 2 3 4 5
edad2
 [1]  1  4  9 16 25  1  4  9 16 25
smkage
 [1] 1 2 3 4 5 0 0 0 0 0
logpop
 [1] 10.866795 10.674706 10.261581  9.446440  8.578665
      9.841080  9.275472  8.649974  7.857481  7.287561

glmmed<-glm(y~edad+edad2+smoke+smkage, offset = logpop, family =
poisson,x=T)

summary(glmmed)

Call:
glm(formula = y ~ edad + edad2 + smoke + smkage, family = poisson,
    offset = logpop, x = T)

Deviance Residuals:
0.43820  -0.27329  -0.15265   0.23393  -0.05700  -0.83049
0.13404   0.64107  -0.41058  -0.01275

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.79176    0.45008 -23.978  < 2e-16 ***
edad          2.37648    0.20795  11.428  < 2e-16 ***
edad2        -0.19768    0.02737  -7.223 5.08e-13 ***
smoke         1.44097    0.37220   3.872 0.000108 ***
smkage       -0.30755    0.09704  -3.169 0.001528 **


Null deviance: 935.0673  on 9  degrees of freedom
Residual deviance:   1.6354  on 5  degrees of freedom
AIC: 66.703

Number of Fisher Scoring iterations: 4

1-pchisq(1.6354,5)
[1] 0.8969356

exp(coef(glmmed))
     edad        edad2           smoke           smkage
1.076692e+01  8.206353e-01    4.224800e+00      7.352475e-01
  10.767       0.821           4.225             0.735
exp(confint(glmmed))
                    2.5 %       97.5 %
edad         7.229960e+00 1.634429e+01      (7.23,16.34)
edad2        7.770875e-01 8.651400e-01      (0.78, 0.87)
smoke        2.088587e+00 9.011414e+00      (2.09, 9.01)
smkage       6.056596e-01 8.866280e-01      (0.61, 0.89)
```

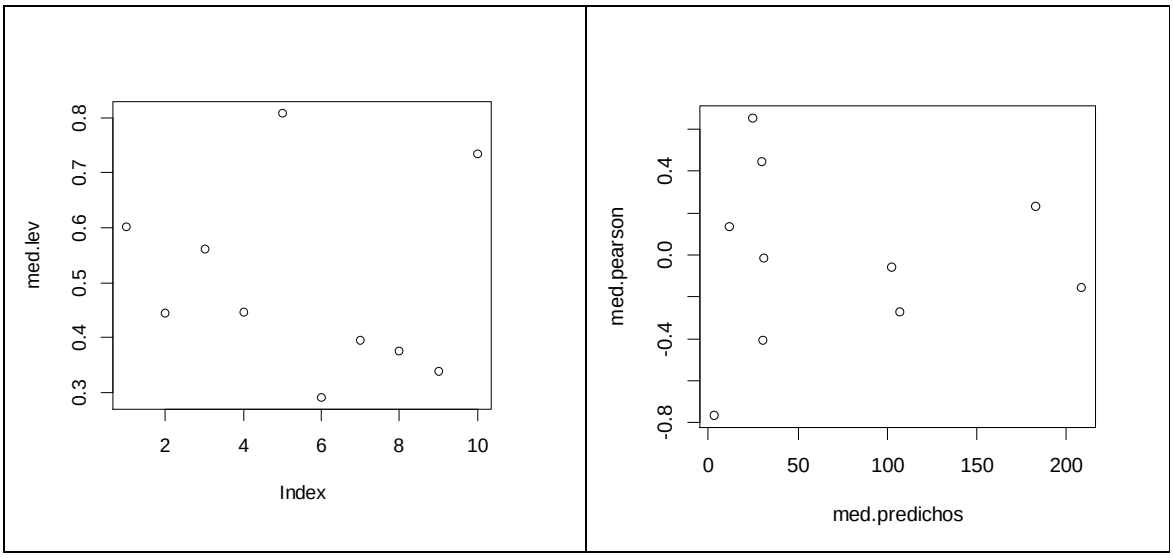
```
plot(glmmed$fit, abs(glmmed$res))
> identify(glmmed$fit, abs(glmmed$res),n=1)
[1] 6
```




```
plot(glmmed$res)
> identify(glmmed$res,n=1)
[1] 6
```



```
med.pearson<-residuals.glm(glmmed,type="pearson")
med.deviance<-residuals.glm(glmmed,type="deviance")
med.lev<-glm.diag(glmmed)$h
med.predichos<-glmmed$fit
```

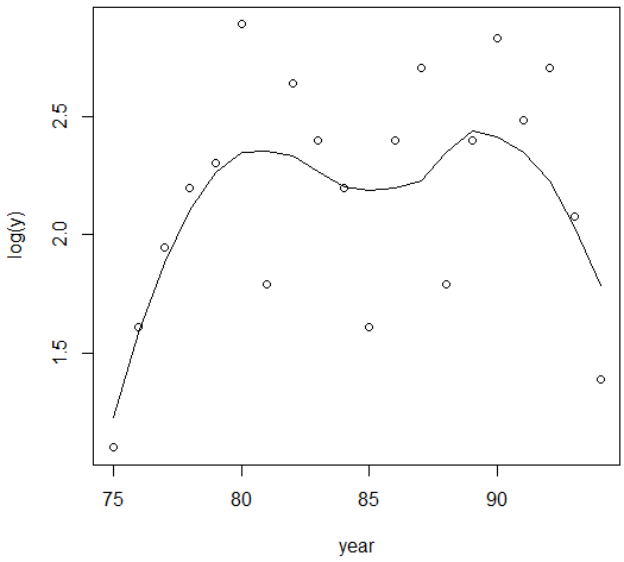


Los siguientes datos corresponden a un famoso estudio del siglo 19, de Vladislav Bortkiewicz, sobre el número de muertes en las tropas del ejército prusiano debido a patadas de caballos. La tabla muestra el total de muertes en 14 tropas del ejército prusiano entre 1875 y 1894:

Año:	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
Muertes:	3	5	7	9	10	18	6	14	11	9	5	11	15	6	11	17	12	15	8	4

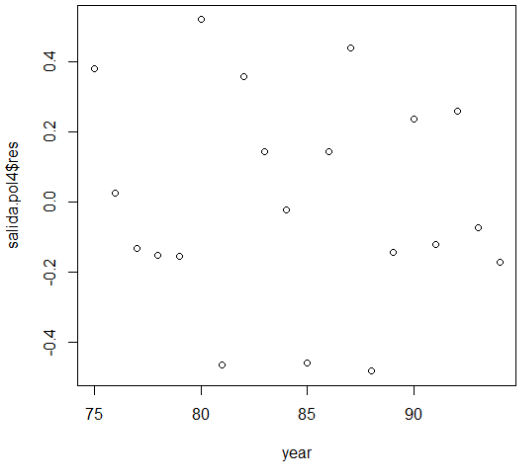
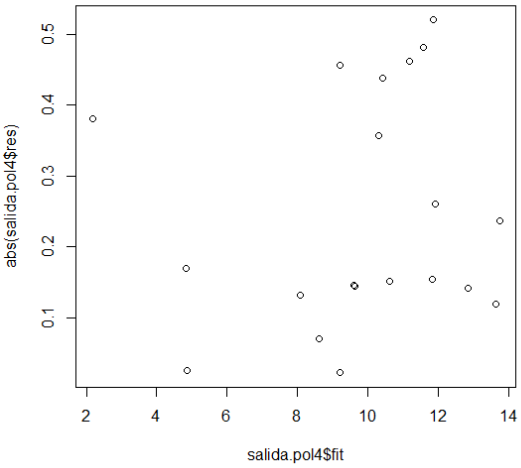
Este ejemplo aparece en muchos libros de texto para ilustrar la distribución Poisson. Estudiaremos si hay alguna tendencia lineal a través de los años.

```
year=75:94
y=c(3,5,7,9,10,18,6,14,11,9,5,11,15,6,11,17,12,15,8,4)
plot(year,log(y))

y.lo<-loess(log(y) ~ year)
predict(y.lo,year)

1.226664 1.585052 1.878879 2.101635 2.264813 2.347958
2.352566 2.332271 2.268641 2.202568 2.189957 2.196296
2.228585 2.349120 2.438450 2.412453 2.349269 2.226670
2.034226 1.782733

y.pred.lo<- predict(y.lo,year)
lines(year, y.pred.lo)
```



Si aparece una tendencia con el año, debe ser un polinomio de un grado alto, mínimo 4.

```
# Para evitar colinealidad centramos

tiempo= year-mean(year)
xx=cbind(t=tiempo, t2=tiempo^2, t3=tiempo^3, t4=tiempo^4)

# Ajusto Poisson
salida.pol4<-glm(y~ xx, family = poisson)

 summary(salida.pol4)

Call:
glm(formula = y ~ xx, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8063  -0.5099  -0.1399   0.6103   1.6613

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.2142611  0.1365518  16.216  < 2e-16 ***
xxt         -0.0006781  0.0315795  -0.021  0.98287
xxt2         0.0222576  0.0091181   2.441  0.01465 *
xxt3         0.0004726  0.0005855   0.807  0.41950
xxt4        -0.0003743  0.0001151  -3.251  0.00115 **

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 38.503  on 19  degrees of freedom
Residual deviance:  17.669  on 15  degrees of freedom
AIC: 108.32

Number of Fisher Scoring iterations: 4

> 1-pchisq(17.669,15)
[1] 0.2804668


Veamos unos gráficos


plot(year,salida.pol4$res)

plot(salida.pol4$fit, abs(salida.pol4$res))
```

No parece haber nada raro 	Acá da la impresión que la función de varianza no está bien elegida. En este modelo V(mu)=mu, pero quizás es algo que crece más rápido 

Ahora voy a ajustar con una binomial negativa

```
glm.nb(formula, data, weights, subset, na.action,start =
NULL, etastart, mustart,control = glm.control(...), method =
"glm.fit",model = TRUE, x = FALSE, y = TRUE, contrasts =
NULL, ...,init.theta, link = log)

#primero estimamos a lo bruto a kappa (theta=1/kappa)
ajustando un modelo con constante solamente


library(MASS)

kappa.net=seq(from=0.01,to=100,by=1)
largo=length(kappa.net)
thetas=numeric(largo)
```

```
for(i in 1:largo){
thetas[i]= glm.nb(y~1, link = log,
init.theta=1/kappa.net[i])$theta
}

thetas

  [1] 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269
10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269

 [15] 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269
10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269

 [29] 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269
10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269

 [43] 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269
10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269

 [57] 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269
10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269

 [71] 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269
10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269

 [85] 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269
10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269 10.47269

 [99] 10.47269 10.47269

salida<-glm.nb(y~xx, link = log,init.theta=10)

summary(salida)

Call:
glm.nb(formula = y ~ xx, init.theta = 96762.37134, link =
log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8062  -0.5099  -0.1399   0.6102   1.6612

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.2142619  0.1365587  16.215  < 2e-16 ***
xxt         -0.0006774  0.0315812  -0.021  0.98289
xxt2         0.0222575  0.0091186   2.441  0.01465 *
xxt3         0.0004726  0.0005855   0.807  0.41954
xxt4        -0.0003743  0.0001151  -3.250  0.00115 **

(Dispersion parameter for Negative Binomial(96762.64) family
taken to be 1)

    Null deviance: 38.499  on 19  degrees of freedom
Residual deviance: 17.667  on 15  degrees of freedom
AIC: 110.32
```

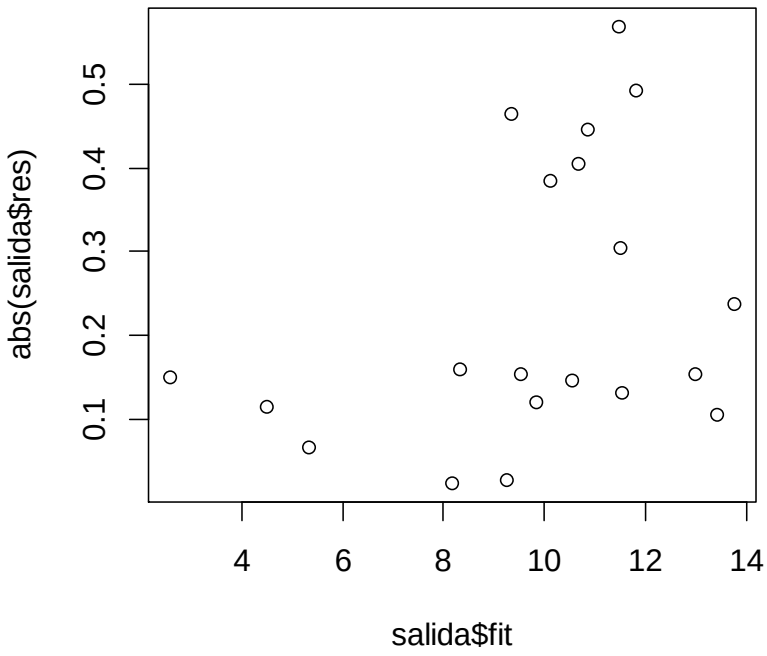
```
Number of Fisher Scoring iterations: 1

Theta:  96762
Std. Err.:  5408764
2 x log-likelihood:  -98.323
```

Esto es lo mismo que ajustaba la Poisson, esto es coherente porque el theta que usa es 96762 o sea que el kappa es aproximadamente 0, con lo que el modelo Binomial Negativo se reduce al Poisson.

Qué pasa si ahora uso Quasi-verosimilitud?

```
salida<-glm(formula = y~xx,family = quasipoisson)
> summary(salida)

Call:
glm(formula = y ~ xx, family = quasipoisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8063  -0.5099  -0.1399   0.6103   1.6613

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
xxint  2.2142611  0.1459644  15.170 1.66e-10 ***
xxt   -0.0006781  0.0337563  -0.020  0.98424
xxt2   0.0222576  0.0097466   2.284  0.03739 *
xxt3   0.0004726  0.0006258   0.755  0.46181
xxt4  -0.0003743  0.0001231  -3.041  0.00825 **

(Dispersion parameter for quasipoisson family taken to be 1.142613)
Null deviance: 38.503  on 19  degrees of freedom
Residual deviance: 17.669  on 15  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
17.669/1.142613 =15.4637
1-pchisq(15.4637,15)
[1] 0.4185601


sum(residuals.glm(salida.pol4,type="pearson")^2)/15

[1] 1.142612

salida<-glm(formula = y~xx,family = quasi(link = log,
variance = "mu^2"))

summary(salida)
```

```
Call:
glm(formula = y ~ xx, family = quasi(link = log, variance =
"mu^2"))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.60845  -0.15541  -0.04695   0.16568   0.48752

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
xxint  2.2241056  0.1386779  16.038 7.52e-11 ***
xxt    0.0094207  0.0321595   0.293  0.77358
xxt2   0.0208372  0.0088538   2.353  0.03266 *
xxt3   0.0002159  0.0004950   0.436  0.66897
xxt4  -0.0003525  0.0001002  -3.517  0.00312 **

(Dispersion parameter for quasi family taken to be 0.1081053)

Null deviance: 4.3613  on 19  degrees of freedom
Residual deviance: 1.7782  on 15  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
1.7782/0.1081053 = 16.448777
1-pchisq(16.4488,15)
[1] 0.3528671


plot(salida$fit, abs(salida$res))
```


abs(salida$res)
0.1
0.2
0.3
0.4
0.5
4
6
8
10
12
14
salida$fit