Profesor: Héctor Bahamonde, PhD.

e:hector.bahamonde@uoh.cl

w:www.hectorbahamonde.com

Curso: MLE.

TA: Gonzalo Barría.

I. Outcomes de Cuentas

En la primera mitad de la clase abordaremos el modelo Poisson. La idea es que cuando el supuesto clave de este modelo no se cumple, tenemos la alternativa del modelo Negative-Binomial que relaja ese supuesto

I. Modelo Poisson

Motivación Muchas veces estamos interesados en procesos de cuentas: en número de veces que visitas a un doctor, el número de asaltos que se produce en tu ciudad, arrestos, etc. Aunque muchos analistas insisten en que estos *data generating processes* pueden ser estimados vía modelos OLS, este tipo de prácticas seriamente sesga las estimaciones (Long 1997, p. 217). Es para esto que necesitamos métodos de MLE para estimarlos.

Este tipo de procesos los llamamos "de cuentas" porque se refieren a cuántas veces cierto proceso se repite. Si te fijas, cada realización del evento ocurre de manera discreta (es decir, *sin* números decimales) y de manera positiva. **Recuerda**: los métodos OLS suponen variables dependientes continuas—"con decimales" y donde $y_i = \{-\infty, \infty\}$. Esto *no* ocurre con los modelos Poisson.

Supuestos Distribucionales Los modelos Poisson son uno de los modelos MLE más básicos. Cada evento i está determinado por una distribución Poisson. Existe una característica muy importante en este tipo de distribuciones: la media (condicional) y la varianza (condicional) son iguales (Long 1997, p. 218).

En otras palabras,

$$E(y_i|\boldsymbol{x}_i) = \text{Var}(y_i|\boldsymbol{x}_i) = \mu \tag{1}$$

lo que llamamos **equidispersion**. Nota que μ es un escalar. Que la **media sea constante** es una consecuencia directa de que asumamos independencia entre las probabilidades de los eventos y la cantidad ("cuenta") de eventos: da lo mismo la cantidad de cuentas, todas siempre tendrán la misma probabilidad, y por tanto, la misma media μ. Esto es lo que mas abajo llamaremos **independencia estocástica**.

Este modelo se construye en base a supuestos que no siempre se cumplirán. En estos casos, existen otras distribuciones (y estimaciones) alternativas:

1. **Negative Binomial**: cuando la varianza no es igual que la media (**overdispersion**). (Esta clase).

2. **Zero-Inflate Poisson**: cuando la cantidad de ceros es excesiva. (Otra clase).

Continuando con la distribución Poisson, ésta se puede definir como,

$$Pr(y_i|\mu) = \frac{\exp(-\mu)\mu^{y_i}}{y_i!} \tag{2}$$

para cada $y_i = 0, 1, 2....$ El parámetro μ se conoce como "*rate*" y representa el número esperado de veces de que un evento ocurre en un espacio de tiempo. Más específicamente,

$$\mu = E(y_i|\boldsymbol{x}_i) = \exp(\boldsymbol{x}_i\boldsymbol{\beta}) \tag{3}$$

Como lo indica la Equation 3, el símbolo "|" denota condicionalidad. Y como en clases anteriores, $\boldsymbol{x}_i\boldsymbol{\beta}$ representa el modelo estructural.

Otro de los supuestos importantes, es que las realizaciones de los eventos son asumidas como independientes. Esta **independencia estocástica** implica que, por ejemplo, los procesos con realizaciones $y_i = 1$ son independientes de los procesos $y_i = 2$, y así sucesivamente. **Creemos en ese supuesto?** Piensa en la cantidad de *papers* que publica un profesor: la cantidad de *papers* que publicaste en el pasado no influye en la cantidad de *papers* que publicas en el presente. Te suena bien?

Estimación: Probabilidades y Likelihood Continuando con la Equation 3, la probabilidad condicional de y_i está dada por,

$$\Pr(y_i|\boldsymbol{x}_i) = \frac{\exp(-\mu)\mu^{y_i}}{y_i!} \tag{4}$$

Asumiendo independencia, tenemos que la función likelihood es la siguiente,

$$L(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{x}) = \prod_{i=1}^{N} \Pr(y_i|\mu) = \prod_{i=1}^{N} \frac{\exp(-\mu)\mu^{y_i}}{y_i!} \tag{5}$$

Tomando en cuenta la Equation 4 y la Equation 5, es fácil notar que $L(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{x})$ es simplemente la sumatoria de las probabilidades de cada evento y_i.

Sumando todas estas piezas de información en Equation 1 y Equation 3, tenemos que,

$$\mu = E(y_i|\boldsymbol{x}_i) = \text{Var}(y_i|\boldsymbol{x}_i) = \exp(\boldsymbol{x}_i\boldsymbol{\beta}) \tag{6}$$

Es por esto que decimos que este es uno de los modelos sencillos que tiene MLE para ofrecer.

II. Programación

Carguemos los datos:

```r
p_load(foreign)
dat = read.dta("https://github.com/hbahamonde/MLE/raw/master/Datasets/banks.dta")
# cambiemos el nombre de la variable dependiente exec_chg
p_load(tidyverse)
dat = dat %>% rename(exec.chg = exec_chg, regime.type = regime_type)
```

Hagamos un resumen,

```r
summary(dat)

##      ccode             year         guerilla      demonstrations
##  Min.   :  10.0   Min.   :1815   Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 327.0   1st Qu.:1887   1st Qu.: 0.000   1st Qu.: 0.00
##  Median : 600.0   Median :1956   Median : 0.000   Median : 0.00
##  Mean   : 649.9   Mean   :1934   Mean   : 0.207   Mean   : 0.45
##  3rd Qu.: 990.0   3rd Qu.:1981   3rd Qu.: 0.000   3rd Qu.: 0.00
##  Max.   :1300.0   Max.   :1999   Max.   :34.000   Max.   :60.00
##                                  NA's   :5021     NA's   :5021
```

```
##                legislat_eff                  coalitions
##  None.  No legislature: 950   no coal. no opp      :3132
##  Ineffective          :2674   >1 party, no opposit.: 172
##  Part. Effect.        :1263   >1 party, opposit.   :1631
##  Effective            :1837   >1 party, no coalit. :1783
##  NA's                 :7002   NA's                 :7008
##
##
##               party_legit     party_frac            regime.type
##  No parties          :3036   Min.   :   0   Civilian        :11897
##  Exclusion           : 896   1st Qu.:   0   Military-Ciilian: 1009
##  1or+ extremist part.: 686   Median :4153   Military        :  277
##  No parties excluded :2100   Mean   :3486   Other           :  187
##  NA's                :7008   3rd Qu.:6429   NA's            :  356
##                              Max.   :9983
##                              NA's   :5122
##      coups         cabinet_size      exec.chg      num_elections
##  Min.   :0.0000   Min.   :  0.0   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:  8.0   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median : 13.0   Median :0.0000   Median :0.0000
##  Mean   :0.0408   Mean   : 14.6   Mean   :0.2286   Mean   :0.2301
##  3rd Qu.:0.0000   3rd Qu.: 19.0   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :3.0000   Max.   :109.0   Max.   :8.0000   Max.   :3.0000
##  NA's   :357      NA's   :1427    NA's   :701      NA's   :2503
##   _est_poisson      _est_zinb
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.4497   Mean   :0.4497
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
##
```

En esta aplicación pensaremos en la variable `exec.chg`: número de veces en que los poderes ejecutivos cambian.

```
table(dat$exec.chg)

##
##     0     1     2     3     4     5     6     7     8
## 10577  2063   294    61    18     7     1     2     2
```

El paquete de `R` que usaremos se llama `glm`.

Estimemos el modelo,

```
modelo.p = glm(exec.chg ~ demonstrations + guerilla + regime.type, family="poisson", data=dat)
```

Hagamos una tabla.

```
p_load(texreg)
texreg(modelo.p) # usa "screenreg" no "texreg".
```

Nota que tenemos varios paises:

```
length(unique(dat$ccode)) # 230 paises
```

[1] 230

Esto significa que nuestros errores estándard están malos (porque nuestras observaciones no son independientes). Los coeficientes están bien, pero los errores estándard están sesgados (y en consecuencia, nuestros intervalos de confianza, y en consecuencia, nuestra significancia estadística). Muchas veces los analistas dejan esto así ("pooled standard errors"), sin embargo esto ha sido altamente criticado (Green, Kim, and Yoon 2001). Deberemos calcular los errores estandares por "cluster" (o controlando por el hecho de que hay "grupos" de países en este análisis).

Este análisis toma bastante tiempo porque usa técnicas de re-sampleo estadístico (*boostrapping*).

	Model 1
(Intercept)	-1.59^{***}
	(0.03)
demonstrations	0.04^{***}
	(0.01)
guerilla	0.07^{***}
	(0.01)
regime.typeMilitary-Ciilian	0.09
	(0.08)
regime.typeMilitary	0.71^{***}
	(0.10)
regime.typeOther	0.16
	(0.23)
AIC	10073.24
BIC	10115.64
Log Likelihood	-5030.62
Deviance	6695.27
Num. obs.	8659

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 1: *Statistical models*

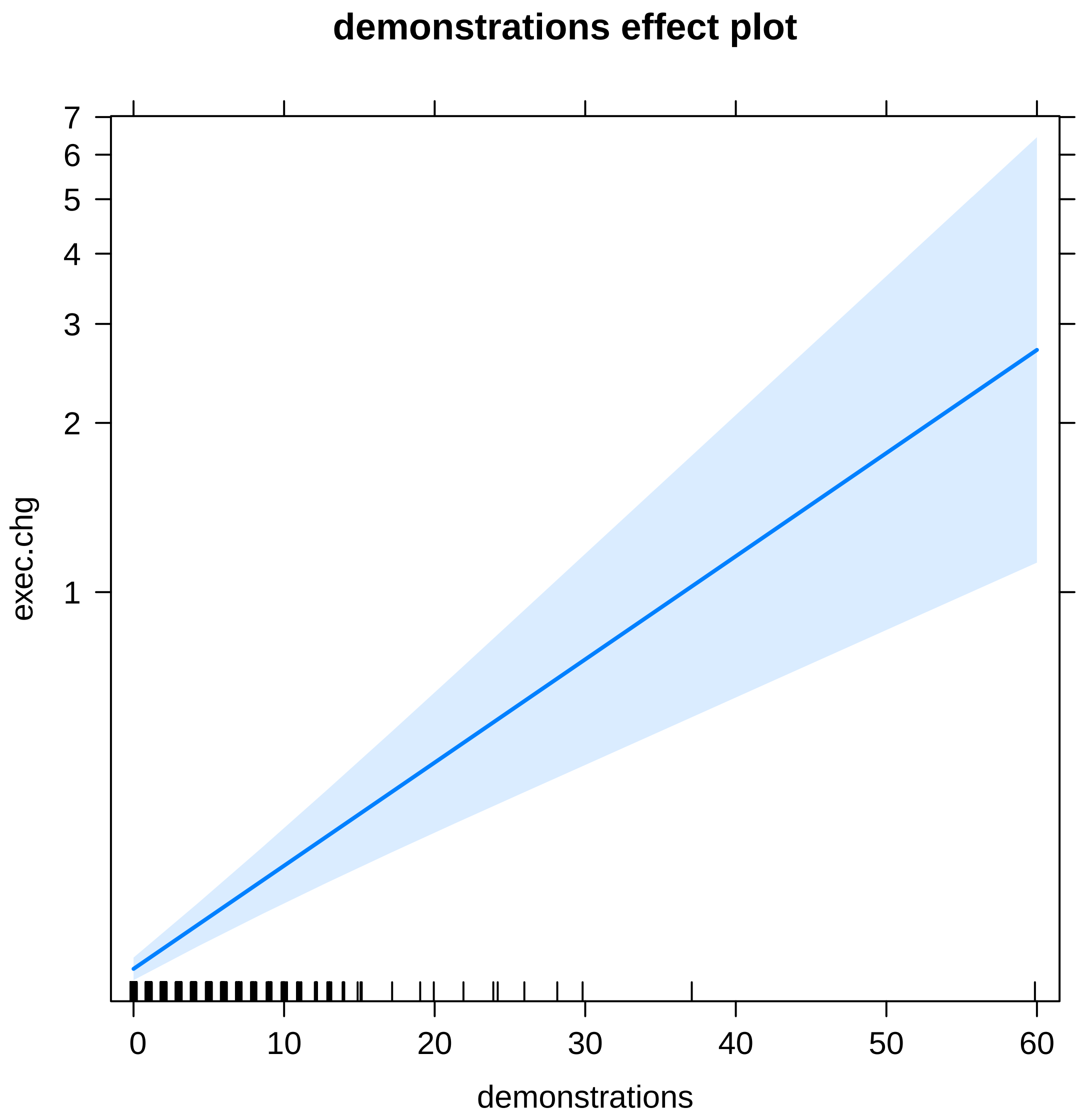
```
p_load(clusterSEs)
cluster.bs.glm(modelo.p, dat = dat, ~ ccode, seed=2020, prog.bar = FALSE)

##
##  Cluster Bootstrap p-values:
##
##              variable name     cluster bootstrap p-value
##                (Intercept)                             0
##             demonstrations            0.0580000000000001
##                   guerilla                             0
## regime.typeMilitary-Ciilian                       0.414
##        regime.typeMilitary                             0
##           regime.typeOther                         0.723
##
##  Confidence Intervals (derived from bootstrapped t-statistics):
##
```

```
##                variable name                 CI lower                CI higher
##                  (Intercept)        -1.72621474132988        -1.45153089874607
##               demonstrations     -0.00189187479121188       0.0864002868580099
##                     guerilla       0.0290397638432541        0.108481486804423
## regime.typeMilitary-Ciilian        -0.137654889684297        0.324078888405253
##          regime.typeMilitary         0.45250859153439        0.976476573734113
##             regime.typeOther        -1.03658924142159         1.35016862595064
```

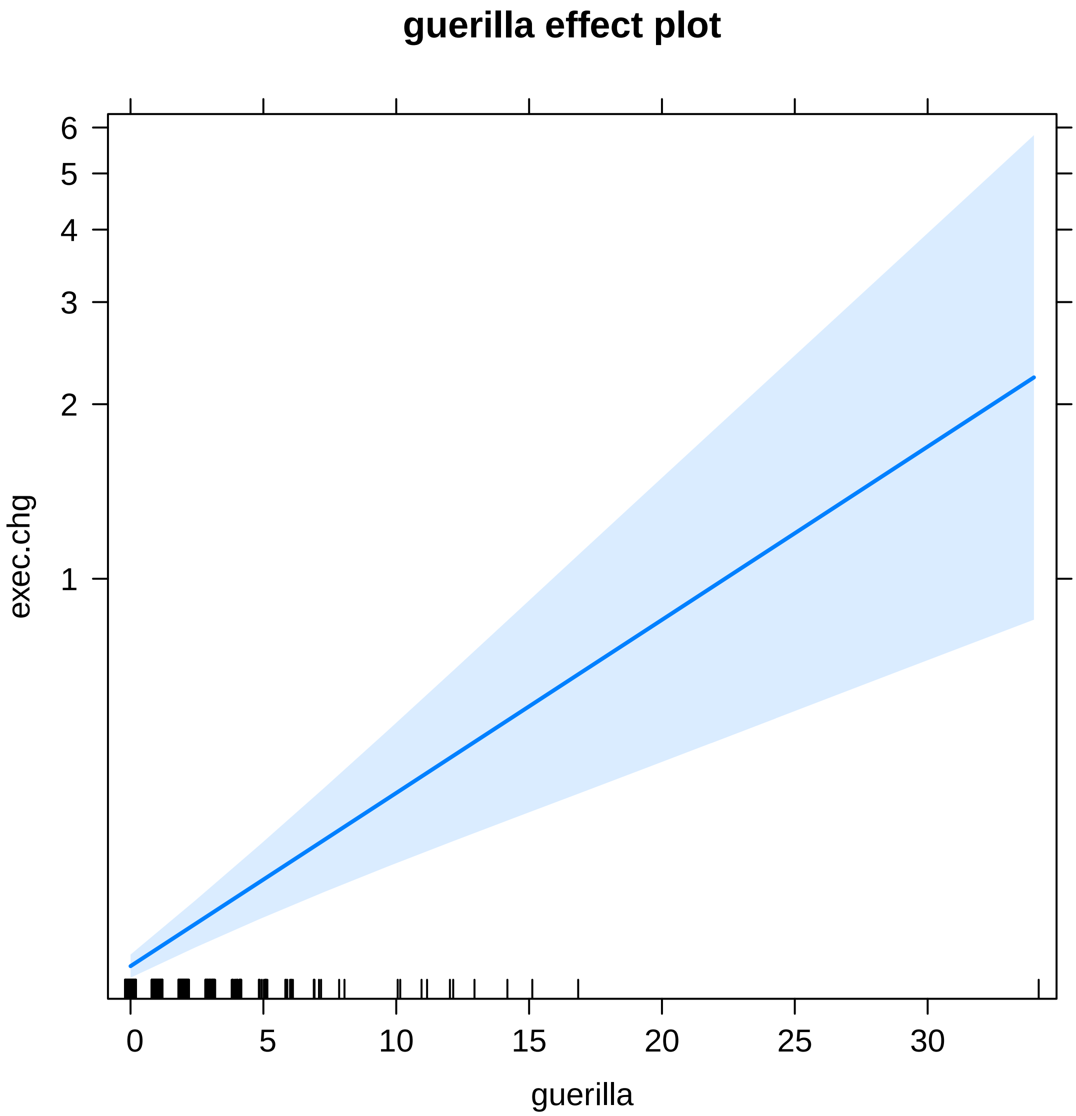
III. Interpretación

En cualquier caso, ya sabemos que la tabla poco valor tiene. Ahora procederemos a estimar los *predicted probabilities*.

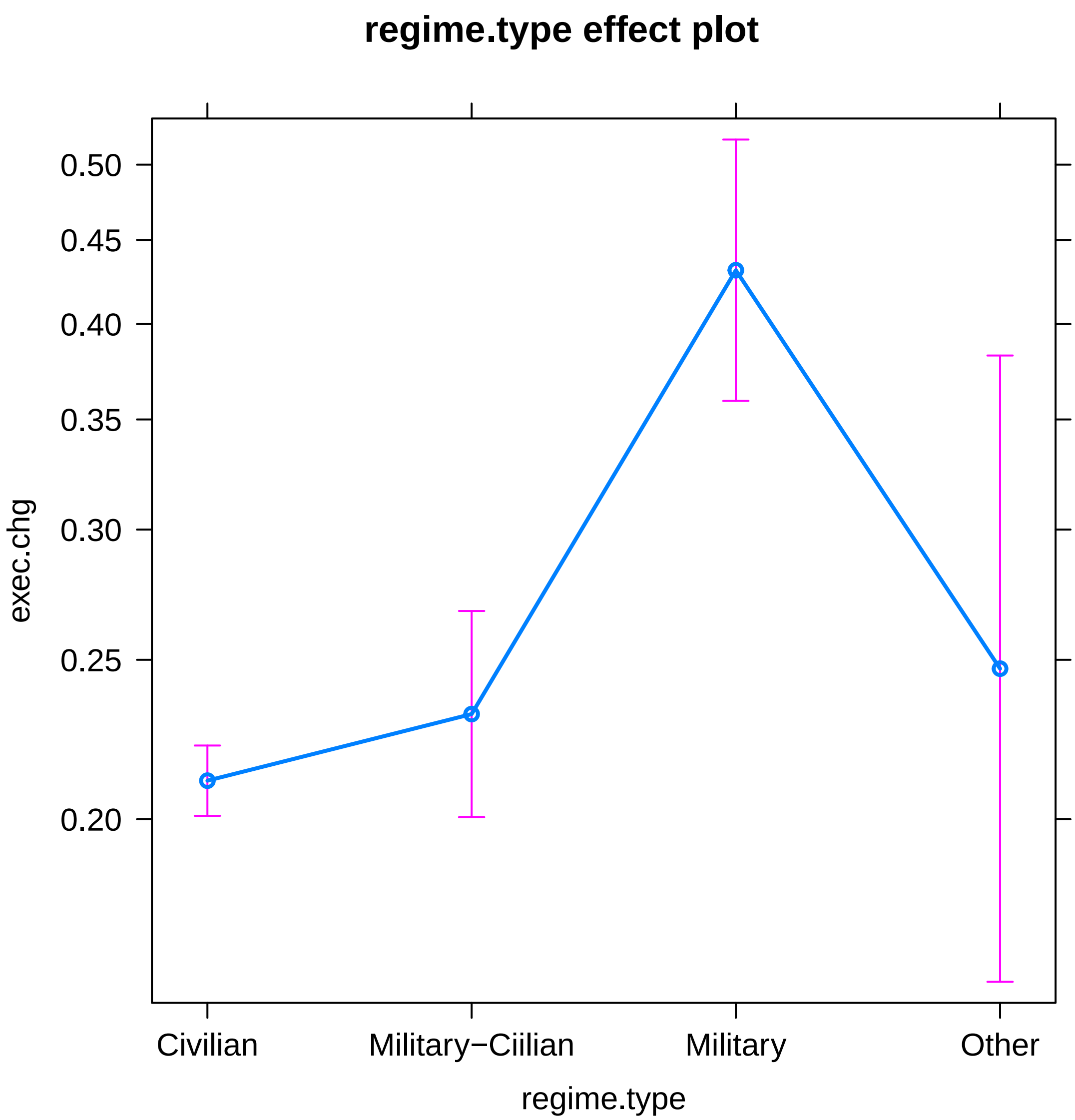
```
p_load(effects)
plot(effect("demonstrations", modelo.p))
```




```
plot(effect("guerilla", modelo.p))
```




```
plot(effect("regime.type", modelo.p))
```



Ahora, veamos si tenemos *overdispersion*. Para esto, usaremos el test desarrollado por Cameron and Trivedi (1990). Ellos explican que la hipótesis nula $E(y_i|\boldsymbol{x}_i) = \text{Var}(y_i|\boldsymbol{x}_i)$. Si el p-value es significativo, tendremos evidencia suficiente para sugerir que la media y la varianza no son iguales.

```
p_load(AER)
dispersiontest(modelo.p)

##
```

```
##  Overdispersion test
##
## data:  modelo.p
## z = 5.6944, p-value = 0.000000006192
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##   1.229601
```

El test es altamente significativo, por lo que sí tenemos evidencia a favor de overdispersion. Entonces deberemos pasar el modelo Negative-Binomial.

I. Modelo Negative-Binomial

Muy rara vez el modelo Poisson cumple su supuesto (Long 1997, p. 230). Lo que haremos será generalizar Equation 4, y permitir que exista heterogenidad no-observada vía la introducción de un residuo e_i (que hasta el momento el modelo Poisson no lo permitía).

En consecuencia, ahora tenemos que el "*rate*" del modelo Negative-Binomial está dado por,

$$\tilde{\mu}_i = \exp(\boldsymbol{x}_i \boldsymbol{\beta}) + \exp(\epsilon_i) \tag{7}$$

Nota que ahora μ_i es un vector (i.e. una distribución), no un escalar. Esto se traduce en el hecho de que tendremos distintos valores de μ_i según las distintas combinamos de valores en las variables independientes $\boldsymbol{x}_i$.

Supuestos Distribucionales El supuesto distribucional es que $\exp(\epsilon_i)$ (o muchas veces llamado el parámetro δ) toma el valor de $E(\exp(\epsilon_i)) = 1$.

- De manera muy importante, esta sigue siendo una distribución Poisson, pero "modificada" (de hecho el valor esperado sigue siendo el mismo, sólo cambia la varianza).
- Nota además que δ (como cualquier varianza) es *desconocida* (es un "*population parameter*").
- Finalmente, parte del supuesto es que δ se distribuye siguiendo la distribución Gamma.

	Model 1	Model 2
(Intercept)	-1.59^{***}	-1.60^{***}
	(0.03)	(0.03)
demonstrations	0.04^{***}	0.05^{***}
	(0.01)	(0.01)
guerilla	0.07^{***}	0.09^{***}
	(0.01)	(0.02)
regime.typeMilitary-Ciilian	0.09	0.09
	(0.08)	(0.08)
regime.typeMilitary	0.71^{***}	0.71^{***}
	(0.10)	(0.11)
regime.typeOther	0.16	0.15
	(0.23)	(0.25)
AIC	10073.24	9954.78
BIC	10115.64	10004.25
Log Likelihood	-5030.62	-4970.39
Deviance	6695.27	5398.93
Num. obs.	8659	8659

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 2: *Statistical models*

Contagio La flexibilidad del modelo Negative-Binomial permite modelar situaciones de **contagio**. Esto se refiere a situaciones donde la probabilidad de obtener ciertas cuentas está correlacionada con el número de cuentas. Volviendo al ejemplo de *papers* publicados, en la especificación Negative-Binomial podríamos modelar la situación donde los académicos tienen más probabilidades de publicar *papers* mientras más *papers* tengan publicados! Esto se refiere a que el modelo permite tomar en cuenta *data generating processes* que **no asuman independencia estocástica**.

Estimación Ahora procedamos a estimar el modelo.

```
p_load(MASS)
modelo.nb = glm.nb(exec.chg ~ demonstrations + guerilla + regime.type, data=dat)
```

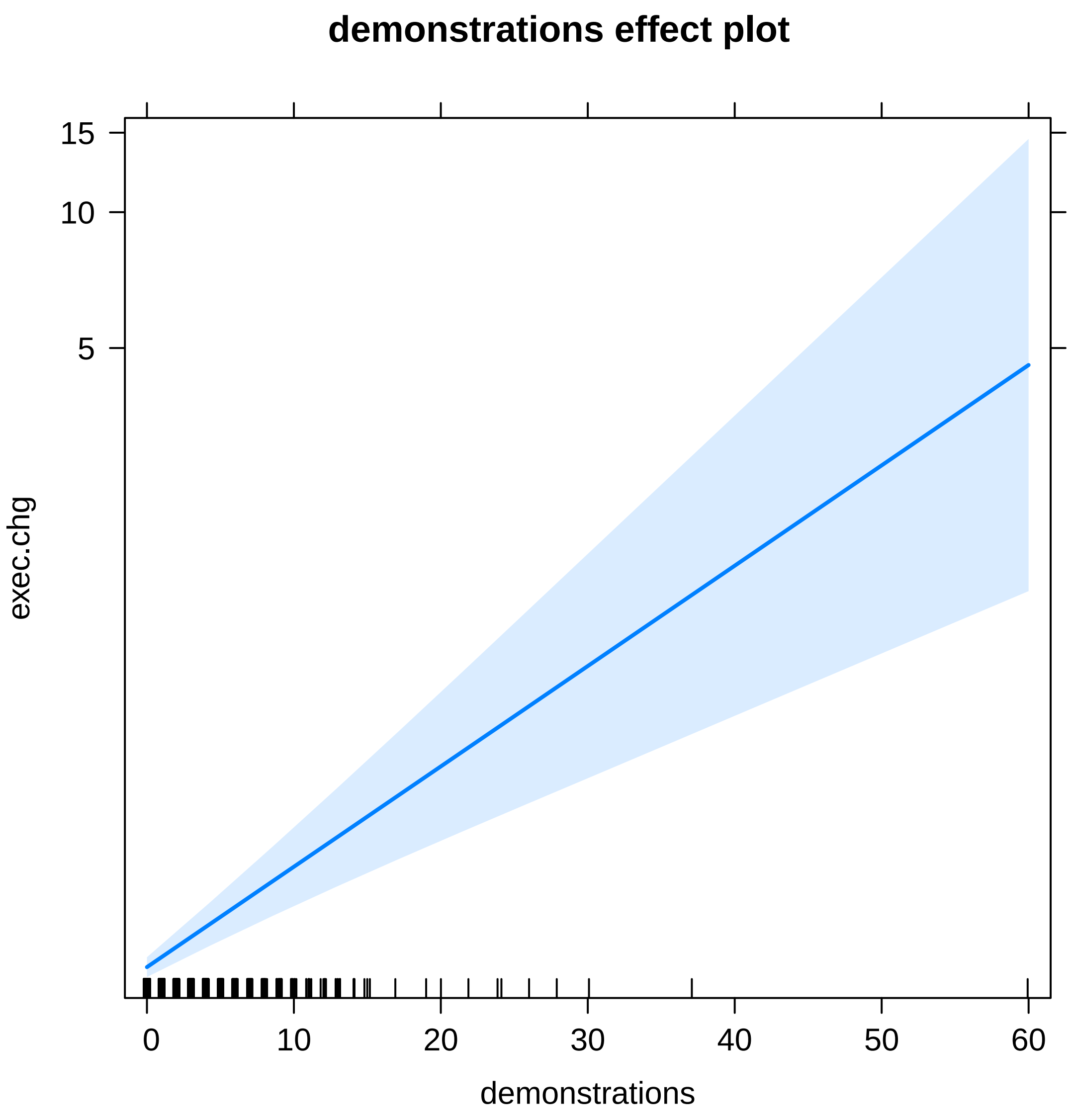
Y de hecho comparemos ambos modelos,

```
p_load(texreg)
texreg(list(modelo.p, modelo.nb)) # usa "screenreg" no "texreg".
```

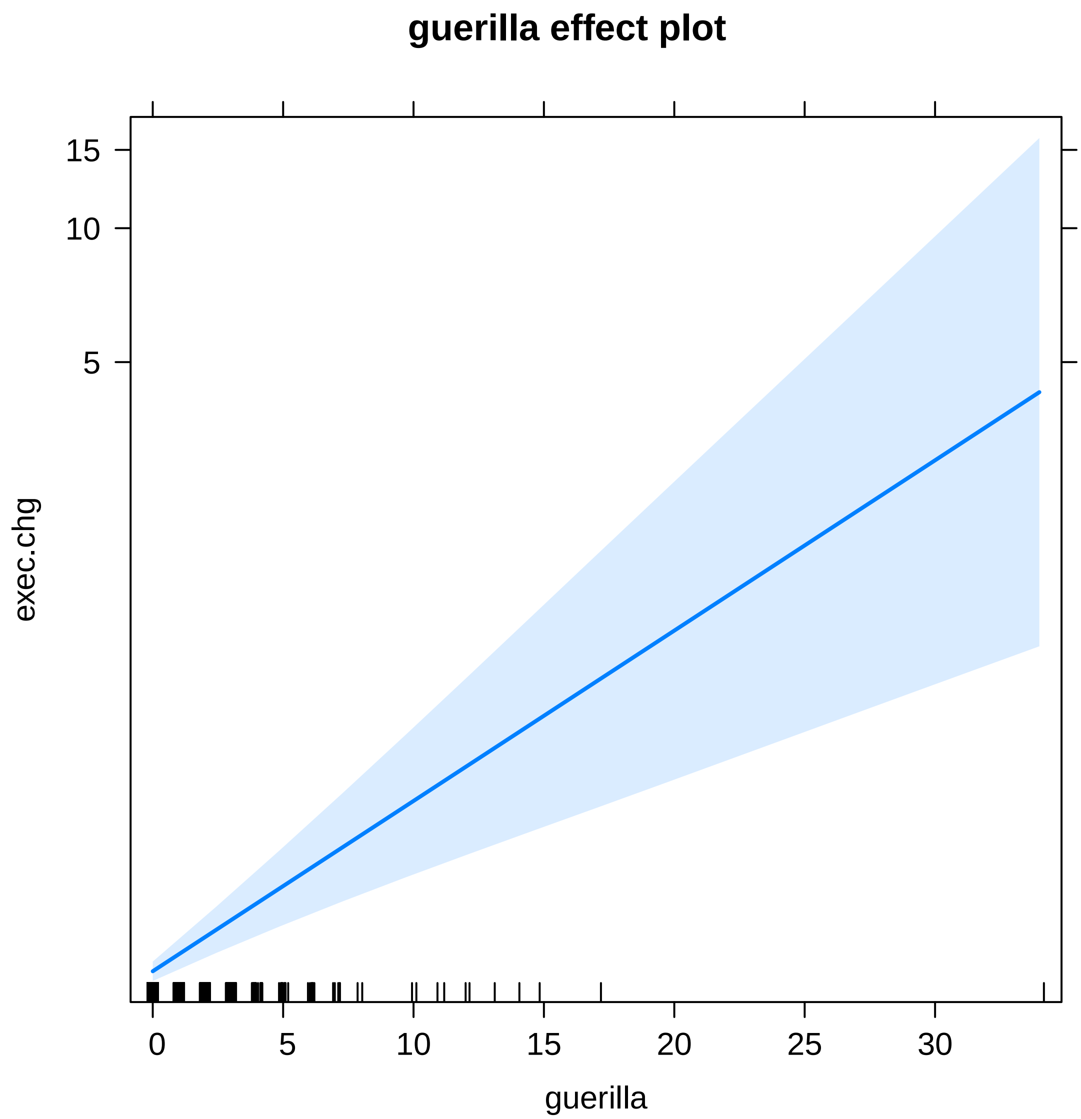
Encontramos—sin sorpresa—que los modelos son altamente parecidos.

Interpretación En cualquier caso, ya sabemos que la tabla poco valor tiene. Ahora procederemos a estimar los *predicted probabilities*.

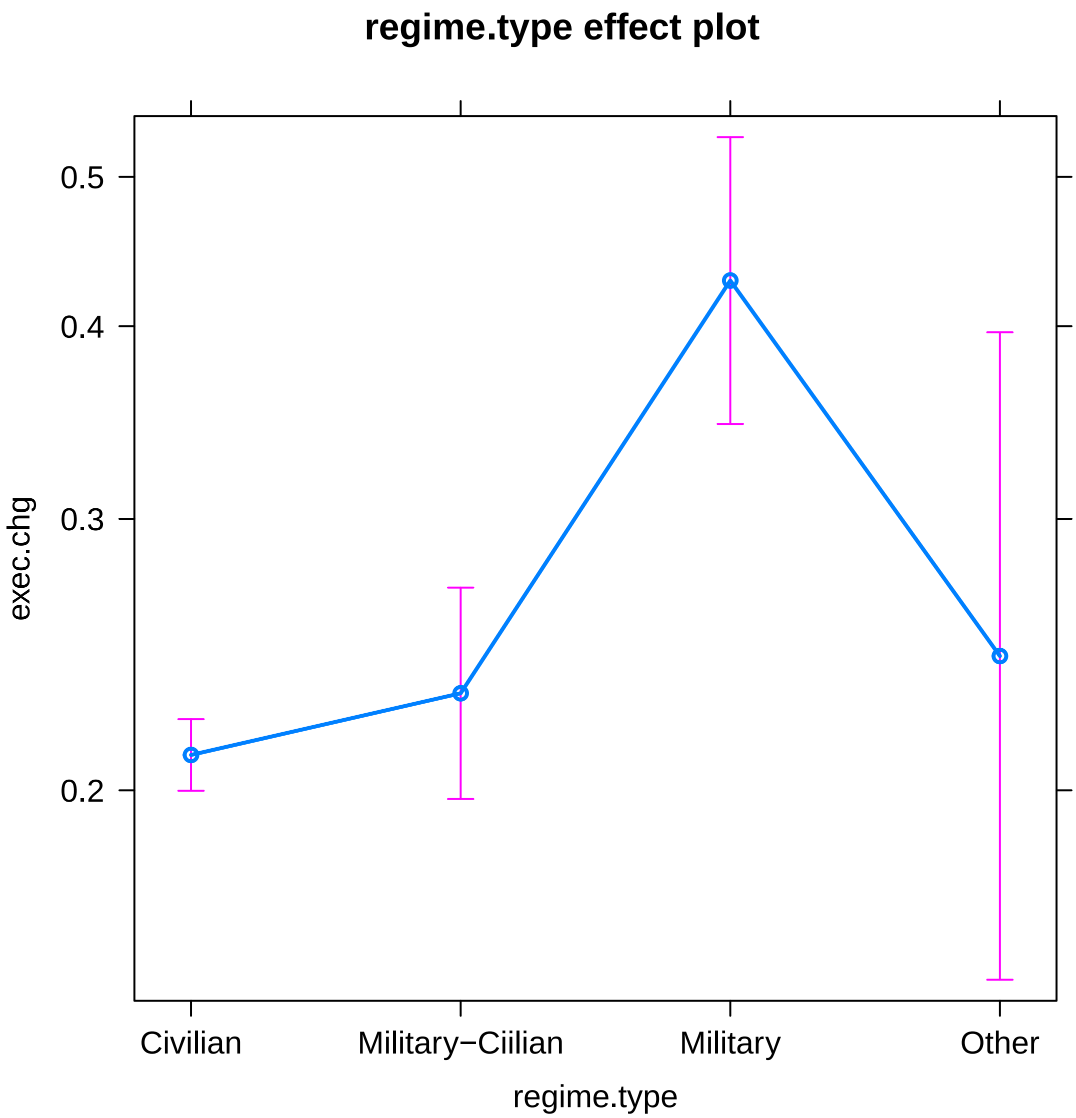
```
p_load(effects)
plot(effect("demonstrations", modelo.nb))
```



```r
plot(effect("guerilla", modelo.nb))
```




```r
plot(effect("regime.type", modelo.nb))
```




```
knitr::purl('Multinomial.Rnw')

## Error in file(con, "r"):  no se puede abrir la conexi'on

Stangle('Multinomial.Rnw')

## Error in SweaveReadFile(file, syntax, encoding = encoding):  no Sweave file with
name 'Multinomial.Rnw' found
```