Profesor: Héctor Bahamonde, PhD.

e:hector.bahamonde@uoh.cl

w:www.hectorbahamonde.com

Curso: MLE.

TA: Gonzalo Barría.

I. Outcomes Poco Frecuentes

En esta clase seguiremos con los *outcomes* de "cuentas". En las ciencias sociales existen *data generating processes* que son raros. Es decir, ocurren con poca frecuencia. Ejemplos son número de atentados terroristas, secuestros, etc. Sería muy raro que por ejemplo al día y de manera constante, existiera mas de cero secuestro.

Aunque una opción sería dejar de lado un estudio así (porque "*quién se interesaría en estudiar algo que casi nunca pasa?*"), existen razones substantivas para estudiar fenómenos que ocurren rara vez. Que un evento ocurra rara vez no lo hace menos importante. Por ejemplo, atentados a autoridades ocurren rara vez, pero **sería incorrecto pensar que el fenómeno carece de interés**.

En la primera mitad de la clase abordaremos el modelo Zero-Inflated (con variantes Poisson/Negative-Binomial). En la segunda mitad del curso veremos una generalización del modelo logit pero calibrado para dar cuenta a eventos raros.

I. Zero-Inflated

Motivación El modelo zero-inflated justamente da cuenta de fenómenos donde los casos donde el evento ocurre es casi en su gran mayoría un "0". Es por esto que se llama "zero-inflated", i.e. la cantidad 0 está "inflada".

La característica principal de la regresión zero-inflated es que modela dos procesos de manera separada. Por un lado, modela aquellos casos donde $y_i > 0$, y aparte, modela aquellos casos donde $y_i = 0$.

Los eventos que no son ceros se modelan segun la distribución Poisson, donde

$$Pr(y_i|\mu) = \frac{\exp(-\mu)\mu^{y_i}}{y_i!} \tag{1}$$

donde $\mu = \exp(\boldsymbol{x}_i\boldsymbol{\beta})$. Si te fijas, hasta el momento, todo sigue igual al modelo Poisson.

Donde el modelo Zero-inflated (Poisson) se diferencia del modelo Poisson (a secas) es que los casos donde $y_i = 0$ son modelados de manera separada, donde $Pr(y_i) = 0 = \psi$. Lo interesante, es que estas probabilidades ψ son modeladas como función de las caracteristicas de los respondentes/firmas/ciudades, i.e. de la unidad de análisis. Más formalmente,

$$\psi_i = F(\boldsymbol{x}_i\boldsymbol{\beta}) \tag{2}$$

donde F es el *cumulative density function* de la distribución normal (Φ) o de la distribución logit (π), es decir $F = \{\Phi, \pi\}$. Si te fijas, el proceso que modela los 0's se guía por el mismo proceso que modelaba *outcomes* binarios. En este caso, el proceso que modela los 0's modela casos donde hay ceros (1) o no (0).

Veamos en más detalle cómo combinamos las probabilidades del modelo Poisson que modela los *outcomes* $y_i > 0$ y las probabilidades del modelo binario para *outcomes* $y_i = 0$,

$$\begin{aligned} \Pr(y_i = 0|\boldsymbol{x}_i) &= \psi_i + (1-\psi_i)\exp(-\mu) \\ \Pr(y_i|\boldsymbol{x}_i) &= (1-\psi_i)\frac{\exp(-\mu)\mu^{y_i}}{y_i!} \quad \text{for } y_i > 0 \end{aligned} \tag{3}$$

Equation 3 es lo que llamamos el modelo Zero-inflated Poisson ("ZIP"). Como ya sabemos, el modelo Poisson descansa sobre el supuesto de la **equidispersión**. Este supuesto no siempre se cumple. Afortunadamente, existe una extensión que (sorpresa!) se llama **Zero-Inflated Negative Binomial** ("ZINB"). Aunque no entraremos en el detalle de la notación de los modelos ZINB, ya sabemos que se obtiene modificando la varianza y el valor esperado de cuentas μ.

II. Rare-event Logistic

Los modelos generalizados son flexibles y existen muchas maneras posibles de abordar procesos que son similares. Tanto el ZIP como el ZINB parten de la base que el *data generating process* genera outcomes poco frecuentes. Ambos modelan cuentas. Qué pasa cuando no quieres estimar cuentas si no que un proceso binario (0,1) pero donde es muy raro encontrar 1's? Para estos casos existen modelos logísticos especiales llamados "*rare event logistic regressions*" (`relogit`).

King and Zeng (2001b) derivan el *rare-event logistic regression.* Particularmente, ellos lo piensan para casos de guerra (1) o paz (0), donde la guerra es (afortunadamente) mucho menos frecuente que la paz (King and Zeng 2001a).

Recordemos el modelo logit tradicional que está dado por,

$$\Pr(y_i = 1) = \pi_i \tag{4}$$

donde,

$$\pi_i = \frac{exp(\boldsymbol{x}_i'\boldsymbol{\beta})}{1 + exp(\boldsymbol{x}_i'\boldsymbol{\beta})} \tag{5}$$

Como explican King and Zeng (2001b, p. 149), el modelo logit para eventos raros, está caracterizado por un **corrector factor** C_i que **incrementa en 50% la contribución relativa** de instancias donde $y_i = 1$, i.e. haciendo "menos raras" las instancias $y_i = 0$. El entendido es que un "evento raro" ocurre menos de la mitad de las veces.

Formalmente, la probabilidad estimada del *rare event logistic regression* está dado por,

$$\Pr(y_i = 1) = \pi_i + C_i \tag{6}$$

donde

$$C_i = (0.5 - \pi_i)\pi_i(1 - \pi_i)\boldsymbol{x}V(\boldsymbol{\beta})\boldsymbol{x}^T \tag{7}$$

Como ellos explican, "[w]hen $\pi_i < 0.5$, as is usually the case for rare events, the correction factor adds to the estimated probability of an event" (King and Zeng 2001b, p. 149). Uno de los entendidos más importantes de esta solución es que en **finite samples** la incertidumbre relativa de los eventos del tipo $y_i = 1$ es mayor e inconsistente.

Esto se puede ver en la varianza de un modelo logit,

$$V(\boldsymbol{\beta}) = [\sum_{1}^{N} \pi_i(1 - \pi_i)\boldsymbol{x}_i^T\boldsymbol{x}_i]^{-1} \tag{8}$$

donde "[t]he part of this matrix affected by rare events is the factor $\pi_i(1 - \pi_i)$" (King and Zeng 2001b, p. 141). Como ellos explican, en los estudios con eventos raros, el resultado es que "$\pi_i(1 - \pi_i)$

will usually be larger for ones than zeros, and so the variance [...] will be smaller. In this situation, additional **ones** will cause the variance to drop more and hence **are more informative than additional zeros**."[1] La manera de compensar por este desbalance estructural del modelo logit aplicado en casos raros es añadiendo el **corrector factor** C_i.

II. Programación

Zero-inflated En esta sección estimaremos un `ZIP` y un `ZINB`.

Carguemos los datos:

```
p_load(foreign)
dat = read.dta("https://github.com/hbahamonde/MLE/raw/master/Datasets/banks.dta")
dat = na.omit(dat) # excluir NAs
```

Hagamos un resumen,

```
summary(dat)

##       ccode              year          guerilla        demonstrations
##  Min.   :  10.0   Min.   :1946   Min.   : 0.000   Min.   : 0.0000
##  1st Qu.: 310.0   1st Qu.:1967   1st Qu.: 0.000   1st Qu.: 0.0000
##  Median : 650.0   Median :1978   Median : 0.000   Median : 0.0000
##  Mean   : 658.7   Mean   :1976   Mean   : 0.221   Mean   : 0.5025
##  3rd Qu.:1000.0   3rd Qu.:1987   3rd Qu.: 0.000   3rd Qu.: 0.0000
##  Max.   :1300.0   Max.   :1999   Max.   :34.000   Max.   :60.0000
##                  legislat_eff                      coalitions
##  None.  No legislature: 879   no coal. no opp       :2926
##  Ineffective          :2431   >1 party, no opposit.: 155
##  Part. Effect.        :1116   >1 party, opposit.   :1438
##  Effective            :1747   >1 party, no coalit. :1654
##
##
```

[1] Mi énfasis.

```
##                party_legit     party_frac            regime_type
##  No parties          :2826   Min.   :   0   Civilian        :5262
##  Exclusion           : 775   1st Qu.:   0   Military-Ciilian: 612
##  1or+ extremist part.: 618   Median :1480   Military        : 238
##  No parties excluded :1954   Mean   :2990   Other           :  61
##                              3rd Qu.:6001
##                              Max.   :9956
##      coups          cabinet_size      exec_chg      num_elections
##  Min.   :0.00000   Min.   :  0.0   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.: 13.0   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median : 18.0   Median :0.0000   Median :0.0000
##  Mean   :0.03645   Mean   : 19.3   Mean   :0.1963   Mean   :0.2214
##  3rd Qu.:0.00000   3rd Qu.: 23.0   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :3.00000   Max.   :109.0   Max.   :7.0000   Max.   :2.0000
##   _est_poisson   _est_zinb
##  Min.   :1     Min.   :1
##  1st Qu.:1     1st Qu.:1
##  Median :1     Median :1
##  Mean   :1     Mean   :1
##  3rd Qu.:1     3rd Qu.:1
##  Max.   :1     Max.   :1
```

En esta aplicación pensaremos en la variable `coups`: número de golpes de estado.

```
table(dat$coups)
```

```
##
##    0    1    2    3
## 5964  195   12    2
```

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

Estimemos un `ZIP` y un `ZINB`.

	Model 1	Model 2
Count model: (Intercept)	-2.01^{***}	-2.04^{***}
	(0.12)	(0.13)
Count model: demonstrations	0.07^{*}	0.07^{*}
	(0.04)	(0.03)
Count model: guerilla	0.03	0.03
	(0.04)	(0.04)
Count model: party_frac	-0.00	-0.00
Zero model: (Intercept)	0.55^{*}	0.49^{*}
	(0.22)	(0.24)
Zero model: guerilla	-9.30	-10.73
		(54.72)
Count model: Log(theta)		1.53
		(1.05)
AIC	1767.97	1769.39
Log Likelihood	-877.98	-877.69
Num. obs.	6173	6173

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

Table 1: *Statistical models*

```
p_load(pscl)
modelo.zip = zeroinfl(coups ~ demonstrations + guerilla + party_frac | guerilla,
dist = 'poisson',
data = dat)
modelo.zinb = zeroinfl(coups ~ demonstrations + guerilla + party_frac | guerilla,
dist = 'negbin',
data = dat)
```

Fíjate cómo **antes** del símbolo "|" ponemos los procesos Poisson, y **después del símbolo** ponemos el proceso logit para estimar los ceros. Nota también que las variables se pueden repetir (o no). Esto depende de la teoría que tengas. **Nunca olvides justificar tu elección**!

Hagamos una tabla.

```
p_load(texreg)
texreg(list(modelo.zip,modelo.zinb)) # usa "screenreg" no "texreg".
```

III. Interpretación

En cualquier caso, ya sabemos que la tabla poco valor tiene. Ahora procederemos a estimar los *predicted probabilities*. Por simpleza sólo procederemos a ver las probabilidades del `ZINB`.

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

## Error in mod.matrix %*% scoef:  argumentos no compatibles

plot(effect("guerilla", modelo.zinb))

## Error in mod.matrix %*% scoef:  argumentos no compatibles

plot(effect("regime.type", modelo.zinb))

## Error in factors[, term2]:  sub'indice fuera de  los l'imites
```

I. Rare-event Logistic Regression

$$\tilde{\mu}_i = \exp(\boldsymbol{x}_i\boldsymbol{\beta}) + \exp(\epsilon_i) \tag{9}$$

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.

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)

## Error in eval(predvars, data, env):  objeto 'exec.chg' no encontrado
```

Y de hecho comparemos ambos modelos,

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

## Error in "list" %in% class(l)[1]:  objeto 'modelo.p' no encontrado
```

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))

## Error in effect("demonstrations", modelo.nb):  objeto 'modelo.nb' no encontrado

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

## Error in effect("guerilla", modelo.nb):  objeto 'modelo.nb' no encontrado

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

## Error in effect("regime.type", modelo.nb):  objeto 'modelo.nb' no encontrado
```

```
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
```