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 $y_i = 0$.

Supuestos distribucionales Los eventos que no son ceros se modelan según 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.

Sin embargo, lo que sí discutiremos es como ayudar a dilucidar qué modelo hace un mejor trabajo maximizando el likelihood. Para esto usaremos el Vuong test que nos dice si el "ZIP" tiene un mejor *fit* que el "ZINB".

II. Rare-event Logistic

Motivación 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).

Parametrización 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" estas realizaciones. El entendido es que un "evento raro" ocurre menos de la mitad de las veces.

De manera muy similar a la regresión logit en Equation 4, la probabilidad estimada del *rare event logistic regression* está dado por,

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

donde el *corrector factor* está caracterizado por,

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

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

I. 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
```

[1] Mi énfasis.

```
##  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
##
##
##               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`.

```
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".
```

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`.

Estimemos las *predicted probabilities*,

	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*

```
zinb.y.p = data.frame(predict(modelo.zinb,type = "prob"))
```

Si te fijas, hemos estimado cuatro columnas: una para cada cuenta de golpes de estado:

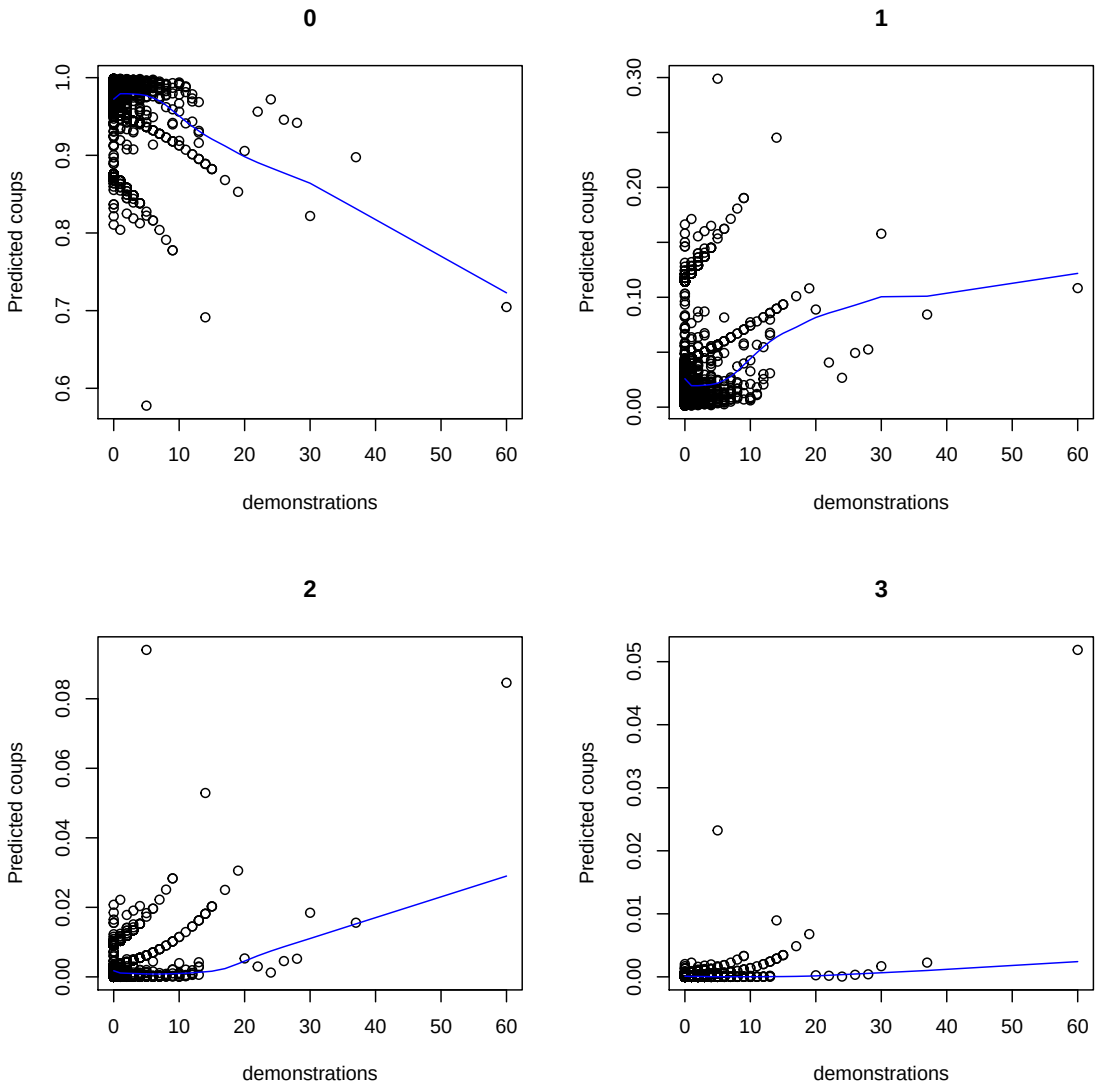
```
head(zinb.y.p)
```

```
##            X0         X1          X2           X3
## 35 0.9543118 0.04223456 0.003246963 0.0001959787
## 36 0.9543118 0.04223456 0.003246963 0.0001959787
## 39 0.9543118 0.04223456 0.003246963 0.0001959787
## 41 0.8759063 0.11440875 0.009086316 0.0005665505
## 42 0.9543118 0.04223456 0.003246963 0.0001959787
## 43 0.8759063 0.11440875 0.009086316 0.0005665505
```

Ahora hagamos un gráfico para cada cuenta de golpe de estado. Nota que tenemos una prediccion para cada cuenta (0, 1, 2, 3), y recuerda que la cuenta zero $y_i = 0$ fue estimada por separado. Finalmente, por simpleza estimaremos las *predicted probabilities* para el coeficiente `demonstrations`.

El proceso sigue siendo el mismo para el resto de los parámetros estimados.

```
par(mfrow=c(2,2)) # divide la consola de graf en 2 cols y 2 filas
# cuenta 0
plot(x=dat$demonstrations,y=zinb.y.p$X0,xlab="demonstrations",ylab="Predicted coups",main = "0")
lines(lowess(dat$demonstrations, zinb.y.p$X0), col = "blue")
# cuenta 1
plot(x=dat$demonstrations,y=zinb.y.p$X1,xlab="demonstrations",ylab="Predicted coups",main = "1")
lines(lowess(dat$demonstrations, zinb.y.p$X1), col = "blue")
# cuenta 2
plot(x=dat$demonstrations,y=zinb.y.p$X2,xlab="demonstrations",ylab="Predicted coups",main = "2")
lines(lowess(dat$demonstrations, zinb.y.p$X2), col = "blue")
# cuenta 3
plot(x=dat$demonstrations,y=zinb.y.p$X3,xlab="demonstrations",ylab="Predicted coups",main = "3")
lines(lowess(dat$demonstrations, zinb.y.p$X3), col = "blue")
```



Ahora veamos si la especificación ZIP tiene un mejor *fit* que la especificación ZINB. Para esto usaremos el Vuong test. Primero

```
p_load(MASS)
vuong(modelo.zip, modelo.zinb)

## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
```

```
##  null that the models are indistinguishible)
## ----------------------------------------------------------------
##                Vuong z-statistic             H_A p-value
## Raw                   -0.3531275 model2 > model1   0.362
## AIC-corrected         -0.3531275 model2 > model1   0.362
## BIC-corrected         -0.3531275 model2 > model1   0.362
```

Considerando que la hipótesis nula es que *the models are indistinguishible*, y que nuestro p-value no es significativo, no tenemos evidencia suficiente para determinar que la hipotesis alternativa ("the two models are **NOT** indistinguishible") se sostiene. Por lo tanto, los dos modelos son iguales (o "indistinguishible") y el segundo modelo ("ZINB") **NO** hace un mejor trabajo que el "ZIP".

II. Rare-event Logistic Regression

En esta sección estimaremos un `relogit`. El paquete de `R` que usaremos se llama `Zelig`.

Carguemos los datos:

```
install.packages("https://cran.r-project.org/src/contrib/Archive/Zelig/Zelig_5.0-12.tar.gz", repos=N
library(Zelig)
data(mid) # Militarized Interstate Disputes
mid = na.omit(mid) # Omitir NAs
```

Hagamos un resumen,

```
summary(mid)
```

```
##     conflict          major            contig           power
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0004201
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0945675
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.3030594
##  Mean   :0.3333   Mean   :0.1711   Mean   :0.2514   Mean   :0.3703210
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.6013527
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :0.9998207
```

```
##      maxdem            mindem            years
##  Min.   :-10.000   Min.   :-10.000   Min.   : 0.00
##  1st Qu.: -5.000   1st Qu.: -9.000   1st Qu.: 3.00
##  Median :  8.000   Median : -7.000   Median :11.00
##  Mean   :  3.689   Mean   : -5.282   Mean   :13.97
##  3rd Qu.: 10.000   3rd Qu.: -6.000   3rd Qu.:23.00
##  Max.   : 10.000   Max.   : 10.000   Max.   :46.00
```

En esta aplicación pensaremos en la variable `conflict`: dummy para conflictos armados.

```
table(mid$conflict)

##
##    0    1
## 2084 1042
```

Estimemos un `relogit`.

```
relogit.m <- zelig(conflict ~ major + contig + power + maxdem + mindem + years,
                   data = mid, model = "relogit")

## How to cite this model in Zelig:
##   Christine Choirat, James Honaker, Kosuke Imai, Gary King, and Olivia Lau. 2020.
##   relogit: Rare Events Logistic Regression for Dichotomous Dependent Variables
##   in Christine Choirat, James Honaker, Kosuke Imai, Gary King, and Olivia Lau,
##   "Zelig: Everyone's Statistical Software," http://zeligproject.org/
```

Veámos el modelo,

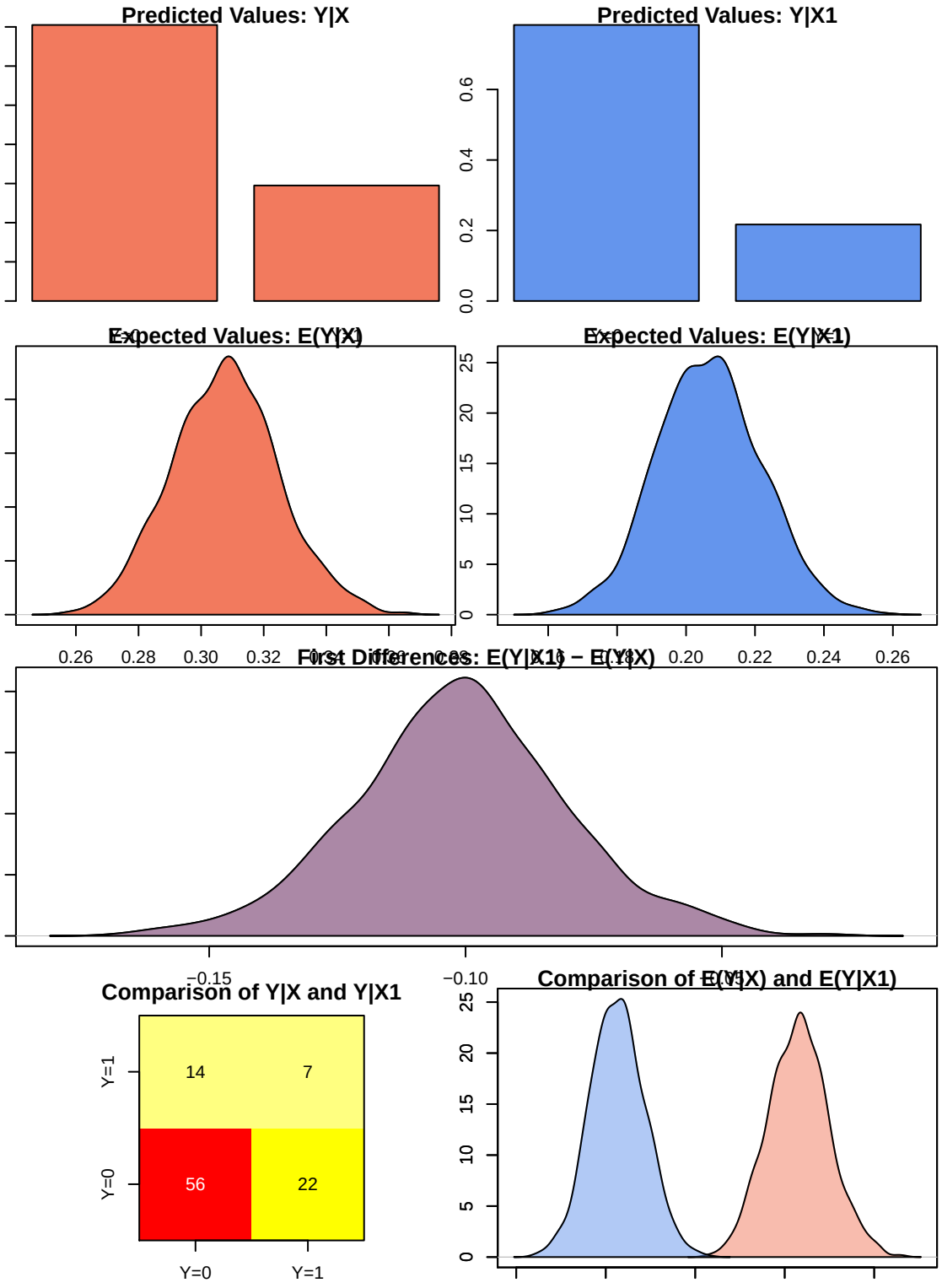
```
summary(relogit.m)

## Model:
##
## Call:
```

```
## z5$zelig(formula = conflict ~ major + contig + power + maxdem +
##     mindem + years, data = mid)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0742  -0.4444  -0.2772   0.3295   3.1556
##
## Coefficients:
##              Estimate Std. Error z value             Pr(>|z|)
## (Intercept) -2.535496   0.179685 -14.111 < 0.0000000000000002
## major        2.432525   0.157561  15.439 < 0.0000000000000002
## contig       4.121869   0.157650  26.146 < 0.0000000000000002
## power        1.053351   0.217243   4.849          0.000001243
## maxdem       0.048164   0.010065   4.785          0.000001708
## mindem      -0.064825   0.012802  -5.064          0.000000411
## years       -0.063197   0.005705 -11.078 < 0.0000000000000002
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3979.5  on 3125  degrees of freedom
## Residual deviance: 1868.5  on 3119  degrees of freedom
## AIC: 1882.5
##
## Number of Fisher Scoring iterations: 6
##
## Next step: Use 'setx' method
```

Usemos *predicted probabilities*. Este enfoque de interpretación mezcla la tradición Bayesiana y frecuentista. El estudiante interesado podrá referirse a King, Tomz, and Wittenberg (2000).

```
x.high <- setx(relogit.m, power = quantile(mid$power, prob = 0.75))
x.low <- setx(relogit.m, power = quantile(mid$power, prob = 0.25))
s.out2 <- sim(relogit.m, x = x.high, x1 = x.low)
par(mar=c(1,1,1,1))
plot(s.out2)
```


Predicted Values: Y|X
Predicted Values: Y|X1
0.0
0.2
0.4
0.6
Expected Values: E(Y|X)
Expected Values: E(Y|X1)
0
5
10
15
20
25
0.26
0.28
0.30
0.32
0.20
0.22
0.24
0.26
First Differences: E(Y|X1) − E(Y|X)
−0.15
−0.10
Comparison of Y|X and Y|X1
Y=1
Y=0
14
7
56
22
Y=0
Y=1
Comparison of E(Y|X) and E(Y|X1)
0
5
10
15
20
25

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

## Error in parse_block(g[-1], g[1], params.src, markdown_mode):  Duplicate chunk label
'setup', which has been used for the chunk:
## if (!require("pacman")) install.packages("pacman"); library(pacman)
## p_load(knitr)
## set.seed(2020)
## options(scipen=9999999)

Stangle('ZIP_ZINB_RELOGIT.Rnw')

## Writing to file ZIP_ZINB_RELOGIT.R

## Error in match.arg(options$results, c("verbatim", "tex", "hide")):  'arg' should
be one of "verbatim", "tex", "hide"
```

References

King, Gary, Michael Tomz, and Jason Wittenberg (Apr. 2000). "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." In: *American Journal of Political Science* 44.2, pp. 341–355.

King, Gary and Langche Zeng (Sept. 2001a). "Explaining Rare Events in International Relations." In: *International Organization* 55.3, pp. 693–715.

King, Gary and Langche Zeng (Jan. 2001b). "Logistic Regression in Rare Events Data." In: *Political Analysis* 9.2, pp. 137–163.