

Profesor: Héctor Bahamonde, PhD.

e: hector.bahamonde@uoh.cl

w: www.hectorbahamonde.com

Curso: MLE.

TA: Gonzalo Barria.

I. DIAGNÓSTICOS

Al igual que en el “mundo” OLS, existen formas para evaluar cuán bueno (o malo) es nuestro modelo.

Qué tipo de diagnósticos existen en el “mundo” OLS?

Recuerda que en OLS, el residuo ϵ_i es la diferencia entre lo que predecimos y lo que observamos, o $\epsilon_i = y_i - x_{ij}\beta_j$ (donde y_i son los valores de la variable dependiente para observación i , x_{ij} son los j variables dependientes para cada una de las observaciones i , y β_j son los j parámetros estimados).

Si recuerdas bien, $E(\epsilon_i) = 0$ y homoesquedástico (varianza constante).

En MLE, es bastante similar, “pero ni tanto”. En vez de un β_j que se multiplica por cada x_{ij} , hablamos de la probabilidad π_i de que observemos la realización del evento en el sujeto i . O más formalmente, $\pi_i = E(y_i|\mathbf{x}_i) = Pr(y_i = 1|\mathbf{x}_i)$. Nota que \mathbf{x}_i es una matriz (**por qué?**).

Debido a que y_i es una variable bimodal, la distribución de $Pr(y_i = 1|\mathbf{x}_i)$ es sigmoideal. En consecuencia, las desviaciones $y_i - \pi_i$ son heteroesquedásticos (no constantes). En general, en MLE trabajamos con los Residuos Pearson r_i :

$$r_i = \frac{y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1 - \hat{\pi}_i)}} \quad (1)$$

donde $\sqrt{\hat{\pi}_i(1 - \hat{\pi}_i)}$ es la varianza. Cuando r_i es grande, eso indica que existe un mal “fit”, i.e. nuestra línea de regresión pasa lejos de las observaciones. Si te fijas, cada observación i tiene su contribución al error total del modelo: hay un r_i para cada observación i . Veamos ahora un “*index plot*” donde graficamos todos los r_i de manera ordenada (o “por índice”: el primer r_1 , después el segundo r_2 , etc.).

Carguemos los datos.

```
mydata <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
```

```
head(mydata)
```

```
##   admit gre  gpa rank
## 1     0 380 3.61   3
## 2     1 660 3.67   3
## 3     1 800 4.00   1
## 4     1 640 3.19   4
## 5     0 520 2.93   4
## 6     1 760 3.00   2
```

```
summary(mydata)
```

```
##      admit           gre           gpa           rank
## Min.   :0.0000  Min.   :220.0  Min.   :2.260  Min.   :1.000
## 1st Qu.:0.0000  1st Qu.:520.0  1st Qu.:3.130  1st Qu.:2.000
## Median :0.0000  Median :580.0  Median :3.395  Median :2.000
## Mean   :0.3175  Mean   :587.7  Mean   :3.390  Mean   :2.485
## 3rd Qu.:1.0000  3rd Qu.:660.0  3rd Qu.:3.670  3rd Qu.:3.000
## Max.   :1.0000  Max.   :800.0  Max.   :4.000  Max.   :4.000
```

Ahora estimemos el modelo:

```
logit.1 <- glm(admit ~ gre + gpa, data = mydata, family = binomial(link = "logit"))
```

```
summary(logit.1)
```

```
##
```

```
## Call:
```

```
## glm(formula = admit ~ gre + gpa, family = binomial(link = "logit"),
```

```
##   data = mydata)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.2730  -0.8988  -0.7206   1.3013   2.0620
```

```
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.949378   1.075093  -4.604 0.00000415 ***
## gre          0.002691   0.001057   2.544   0.0109 *
## gpa          0.754687   0.319586   2.361   0.0182 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 480.34  on 397  degrees of freedom
## AIC: 486.34
##
## Number of Fisher Scoring iterations: 4
```

Y usando la función `rstandard` calcularemos r_i

```
r = rstandard(logit.1)
head(r)

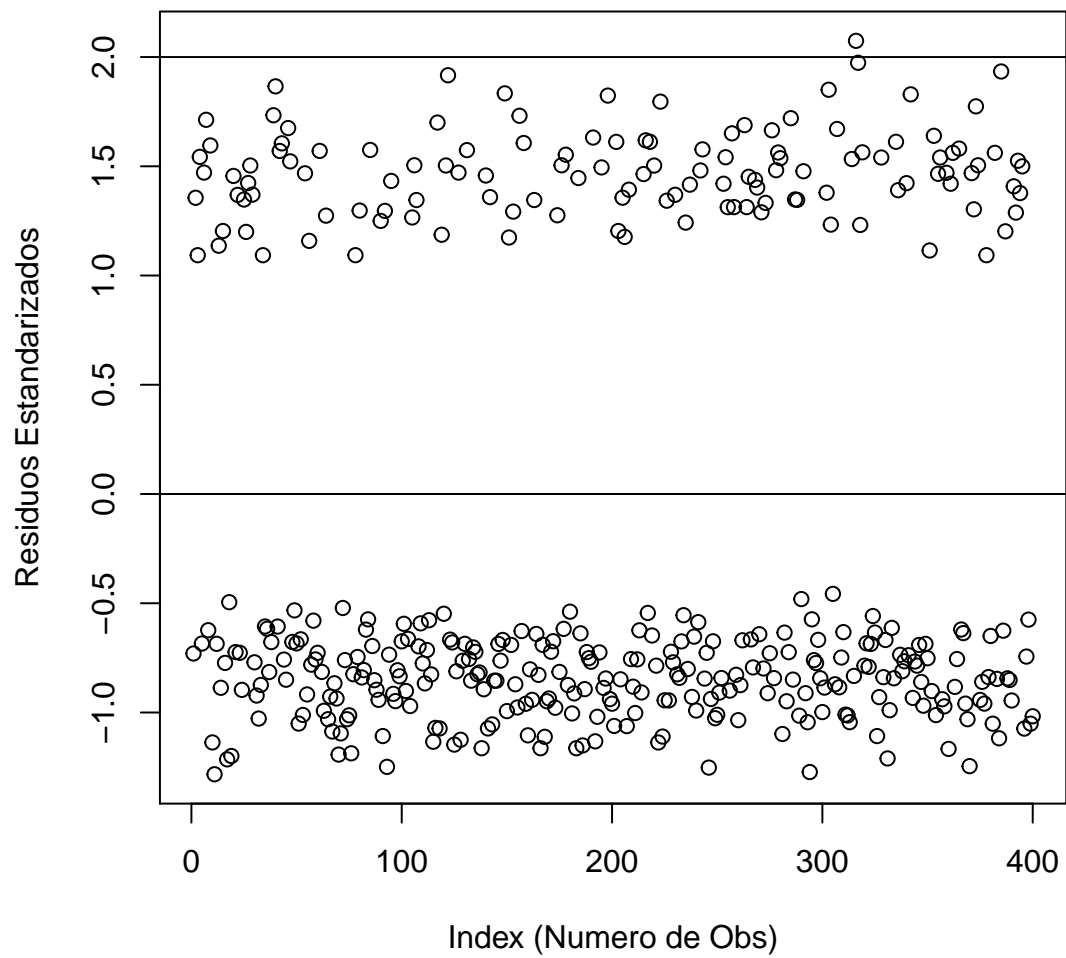
##           1           2           3           4           5           6
## -0.7300511  1.3559165  1.0930693  1.5430303 -0.6844243  1.4711335
```

Como ves, cada observación i tiene su propio error (distancia entre el parámetro y la observación).

Hagamos el “index plot”:

```
plot(1:nrow(mydata), # Numero de Obs
     r, # y
     ylab="Residuos Estandarizados",
     xlab="Index (Numero de Obs)")
abline(0, 0)
```

```
abline(2, 0)
abline(-2, 0)
```



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

## [1] "Diagnosticos.R"

Stangle('Diagnosticos.Rnw')
```

```
## Writing to file Diagnosticos.R
```