Profesor: Héctor Bahamonde, PhD.

e:hector.bahamonde@uoh.cl

w:www.hectorbahamonde.com

Curso: MLE.

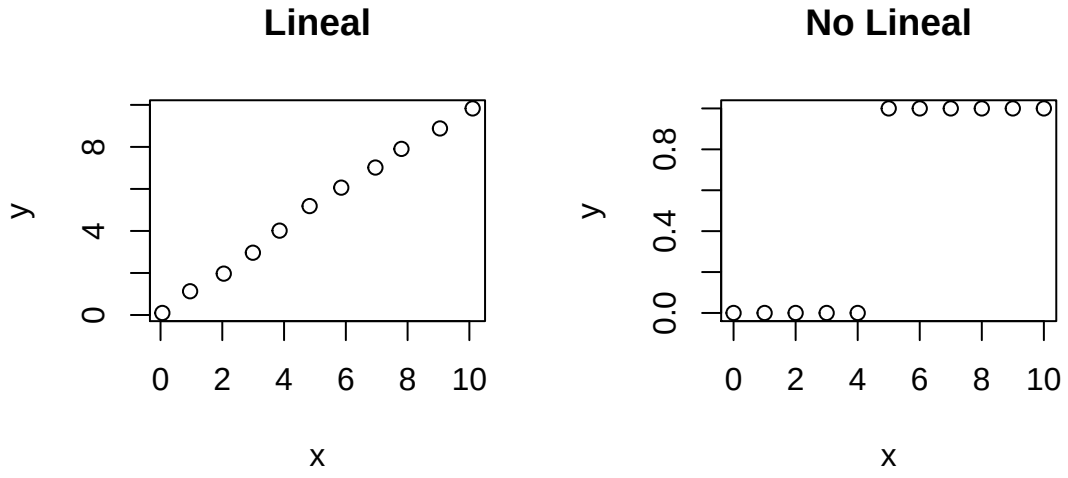
TA: Gonzalo Barría.

I. Derivando el Likelihood del Modelo Logit

El modelo lineal de probabilidad (OLS) asume que el valor de β es lineal. Esto significa que cada vez que avanzamos en nuestro eje x, avanzamos una unidad en y. Recuerda la interpretacion de β: "cuando subo X en una unidad, Y avanza β unidades".

Sin embargo, existen muchas instancias donde la variable dependiente (y) no es lineal. El primer ejemplo que revisaremos es cuando y es dicotómica, 0's y 1's (proceso "Bernoulli"). Si fijas en el panel derecho de la figura, cada vez que avanzamos x en una unidad, no avanzamos **nada** en nuestro y (entre 0 y 4). Sin embargo cuando $x \geq 5$, y es **siempre** 1.

```
par(mfrow=c(1,2))
plot(data.frame(x=jitter(0:10),y=jitter(0:10)), main = "Lineal")
plot(data.frame(x=0:10,y=c(rep(0,5), rep(1,6))), main = "No Lineal")
```



Al ser β no lineal, estaríamos violando varios supuestos:

1. La varianza de los errores ϵ_i son constantes (homoeskedasticidad).
2. La distribución de los errores ϵ_i es normal con promedio 0; $E(\epsilon_i) = 0$.
3. La forma funcional del modelo es lineal. Sin embargo, la forma funcional del panel de la derecha no es lineal, sino que con forma de "S".

Debido a que estos supuestos se violan, si aplicamos un modelo lineal a un y no lineal (como la y en el panel derecho), la interpretación de β no tiene sentido: "si avanzo una unidad en x, y sube β". Esto daría como resultado a que nuestra predicción diría que siempre nos pasaríamos de los límites del y observado.

I. Logit: Motivación Vía Modelos Latentes

Una manera de motivar los modelos binarios, es pensando en un modelo "latente". Esto es, en un modelo donde sólo observas ciertos *outcomes*. Piensa en un $y_i^\star$ continuo, que va del $\infty-$ al $\infty+$. Después, piensa que si cierto valor de $y_i^\star$ es grande, el outcome $y = 1$, pero si el valor de $y_i^\star$ es chico, el outcome $y = 0$. No observas lo que hay entre medio (por eso "latente"); sólo observas 0's y 1's. Después, piensa en lo que clasifica cada numero "grande" o "chico" es una especie de umbral τ. Más formalmente,

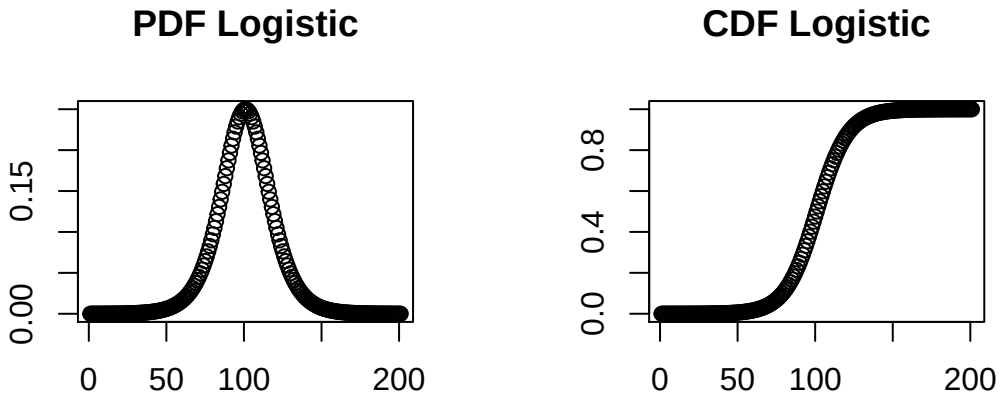
$$y_i = \begin{cases} 1 & \text{si } y_i^\star > \tau \\ 0 & \text{si } y_i^\star \leq \tau \end{cases} \tag{1}$$

Este ejemplo del modelo "latente" nos sirve sólo para motivar el problema. Debido a que $y_i^\star$ no es observado, no podemos estimar el modelo vía OLS. Lo tendremos que hacer vía MLE. Nuestro primer modelo es el modelo *logit*. Y proviene de la distribucion "logística".

☞ Es importante saber identificar las distribuciones. No precisamente la fórmula. Pero sí qué tipo de distribución está generando tu variable dependiente y.

Una distribución logística se ve así:

```r
par(mfrow=c(1,2))
plot(dlogis(seq(-10, 10, by = 0.1)), main = "PDF Logistic", ylab = "", xlab = "")
plot(plogis(seq(-10, 10, by = 0.1)), main = "CDF Logistic", ylab = "", xlab = "")
```



mientras que los errores están distribuidos de acuerdo a la *distribución logística estándard* con promedio 0 y varianza $\frac{\pi^2}{3} \approx 3.29$.

Ve que el panel izquierdo es el "PDF", o *probability density function*. Esta es la manera en que "se ordena" la distribución. Es muy parecida a la distribución normal, pero no lo es (ver varianza arriba).[1] El panel derecho es el "CDF" o *cumulative density function*. Al ver un "CDF" piensa en la intergral, y en como las probabilidades de que $y_i = 1$ se van sumando, hasta que éstas suman 1 (el máximo).

Formalmente, el modelo logit se ve así:

$$\pi_i = \frac{exp(x_i'\beta)}{1 + exp(x_i'\beta)} \tag{2}$$

donde π_i es la probabilidad de cada observación de ser 1 o 0 (así como en Equation 1), y x_i es una matriz de datos y β una matriz de estimaciones. Esto es lo que se llama el "link" logit.

[1] Sin embargo, la distribución *logística estandarizada* (con varianza 1) es casi identica a la normal. Lo que hace el modelo logístico distinto es también la forma de estimación maximización del likelihood de una variable dicotómica.

Link Function Una "función link" es una función que "junta" y con $\hat{y}$. En general, una función link se puede interpretar como una función que transforma un vector en otro. Por ejemplo, el logaritmo natural de un vector es una función link que "mapea" x en $log(x)$. En particular, tanto el modelo logit como el probit tienen *link functions* donde $\hat{y}$ es re-escalado y sólo puede tener dos valores: 0 o 1.

II. Probit

Desde el modelo logit se construyen varias extensiones. La primera es el modelo "probit". Tanto los modelos logit como los probit tienen variables dependientes dicótomicas (proceso Bernoulli). Pero su "link" es como sigue:

$$\pi_i = \Phi^{-1} exp(x_i'\beta) \tag{3}$$

donde Φ^{-1} es la inversa del CDF de la distribución normal estandard (promedio 0 y varianza 1). Además, en los modelos probit los errores tienen promedio 0 y están normalmente distribuidos (como en la regresión lineal; OLS). Las estimaciones del modelo probit y el modelo logit son en general iguales (estadísticamente iguales).

Qué modelo es mejor? Ambos son muy parecidos. No es poco frecuente estimar los dos. Cuando creas que los errores podrian ser homoeskedasticos, estima un probit. También hay asuntos disciplinares.

```
knitr::purl('Logit_Derivacion.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)
## if (!require("pacman")) install.packages("pacman"); library(pacman)

Stangle('Logit_Derivacion.Rnw')
```

```
## Writing to file Logit_Derivacion.R
```