

# Regresión Logística

Lizbeth Naranjo Albarrán y Yanink Neried Caro Vega

26/agosto/2022

## Regresión Logística

- Simular una respuesta de un modelo de regresión logística.

Se tiene la variable de interés  $Y$  binaria (variable respuesta o dependiente),  $Y \in \{0, 1\}$ , y se relaciona con un conjunto de covariables  $x_1, \dots, x_k$ , a través de un modelo de regresión logística:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

o equivalentemente

$$p = \frac{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k)}$$

que es igual a la función de distribución acumulada (fda) de una distribución logística.

Si  $X \sim \text{Logistica}(0, 1)$  su fda es:  $F_X(x) = \frac{\exp(x)}{1 + \exp(x)} = \frac{1}{1 + \exp(-x)}$ .

## Regresión logística usando coeficientes para la probabilidad de éxito

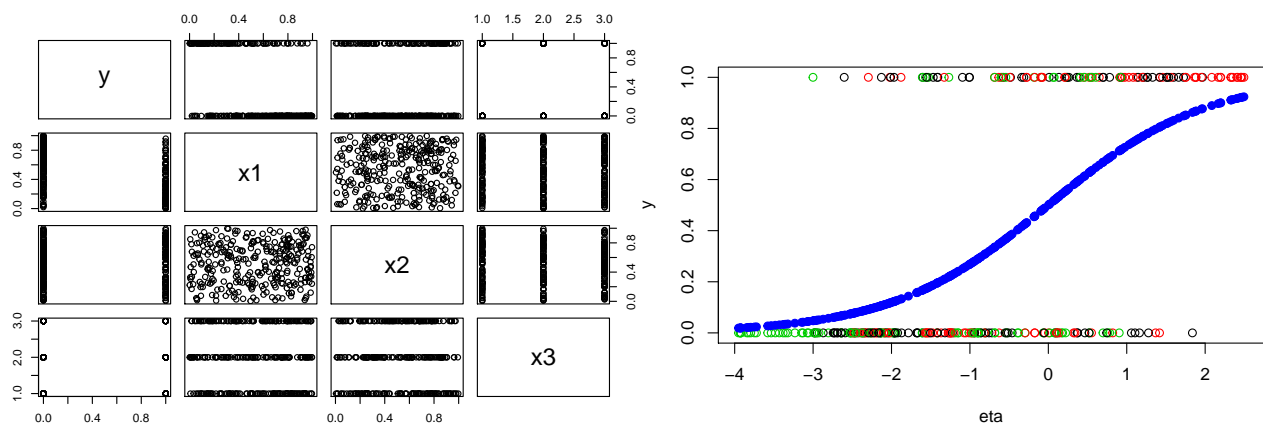
```
n = 300    ### numero de sujetos
K = 3     ### numero parametros de regresion
betas = as.vector(c(3,-2,0.5,+1))    ### coeficientes de regresion
set.seed(12345)
x1 = runif(n)    ### covariable x1
x2 = runif(n)    ### covariable x2
x3 = factor(rep(c("A","B","C"),n/3))    ### variable categorica con categorias A, B, C
dummyB = ifelse(x3==levels(x3)[2],1,0)
dummyC = ifelse(x3==levels(x3)[3],1,0)
#eta = betas[2] + betas[1]*x1 ####+ betas[3]*dummyB+ betas[4]*dummyC
eta = 2 - 5*x1 + 0.5*dummyB - 1*dummyC ####+ betas[2]*x2 ####combinacion lineal
p = plogis(eta)    ### se calcula p=P[Y=1]
u = runif(n)
y = ifelse(p>u,1,0)    ### simular la variable respuesta Y de manera aleatoria
head(cbind(y,x1,x2,x3,eta,p,u),4)
```

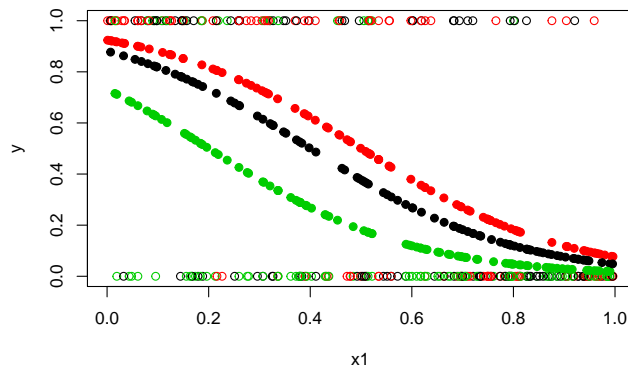
```
##      y      x1      x2 x3      eta      p      u
## [1,] 0 0.7209039 0.05268832 1 -1.604519 0.16735090 0.69926306
```

```
## [2,] 1 0.8757732 0.59865728 2 -1.878866 0.13251919 0.01040697
## [3,] 0 0.7609823 0.70829065 3 -2.804912 0.05705934 0.50390708
## [4,] 0 0.8861246 0.65344066 1 -2.430623 0.08086716 0.73105921
```

```
pairs(cbind(y,x1,x2,x3))
plot(eta,y,col=x3)
points(eta,p,col="blue",pch=19)
plot(x1,y,col=x3)
points(x1,p,col=x3,pch=19)
modelo <- glm(y ~ x1 + factor(x3), family=binomial(link="logit")) ### estimar parametros
summary(modelo)
```

```
##
## Call:
## glm(formula = y ~ x1 + factor(x3), family = binomial(link = "logit"))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0210  -0.7051  -0.3810   0.6933   2.5777
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.0576     0.3907   5.266 1.40e-07 ***
## x1            -4.7524     0.5984  -7.942 1.99e-15 ***
## factor(x3)B    0.4477     0.3470   1.290  0.197
## factor(x3)C   -1.5400     0.3941  -3.908 9.30e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 393.19  on 299  degrees of freedom
## Residual deviance: 279.48  on 296  degrees of freedom
## AIC: 287.48
##
## Number of Fisher Scoring iterations: 5
```





```
library(AUC)
pred <- predict(modelo) ### Predecir P[Y=1]
Ypred <- ifelse(pred>0.5,1,0) ### Predecir Y, considerando como punto de corte 0.5
table(y,Ypred) ### Tabla de observaciones vs. predicciones
```

```
##      Ypred
## y      0   1
## 0 177  14
## 1  55  54
```

```
sens <- sensitivity(pred,factor(y)) ### Calcula la sensibilidad
head(sens$cutoffs) ### puntos de corte para calcular la sensibilidad
```

```
##              14      32      83      119      257
## 1.0000000 1.0000000 0.9966667 0.9933333 0.9900000 0.9866667
```

```
head(sens$measure) ### valores de la sensibilidad
```

```
## [1] 0.000000000 0.009174312 0.018348624 0.027522936 0.036697248 0.045871560
```

```
spec <- specificity(pred,factor(y)) ### Calcula la especificidad
head(spec$cutoffs) ### puntos de corte para calcular la especificidad
```

```
##              14      32      83      119      257
## 1.0000000 1.0000000 0.9966667 0.9933333 0.9900000 0.9866667
```

```
head(spec$measure) ### valores de la especificidad
```

```
## [1] 1 1 1 1 1 1
```

```
plot(roc(pred,factor(y)),main="Curva ROC")
auc(roc(pred,factor(y))) ### Area bajo la curva ROC
```

```
## [1] 0.8401941
```

