

Solución taller sobre capítulo 4 parte 1

Santiago Franco

10/11/2022

```
library(ggplot2)
```

```
## Warning in register(): Can't find generic 'scale_type' in package ggplot2 to  
## register S3 method.
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

$$y_{ij} \sim \text{Bernoulli}(p_{ij}) \quad (1)$$

$$\text{logit}(p_{ij}) = -1.4 + b_{0i} + 0.33 x_{ij} \quad (2)$$

$$x \sim \text{Unif}(0, 1) \quad (3)$$

$$b_0 \sim N(0, 4) \quad (4)$$

1. Simular 100 observaciones para cada uno de los 200 grupos del siguiente modelo Logístico. Consulte la información de función logit en este enlace.

```
inverse_logit <- function(x){  
  return(exp(x)/(1+exp(x)))  
}  
logit <- function(x){  
  return(1/logit(x))  
}
```

```

set.seed(123456)
ni <- 100
G <- 200
nobs <- ni * G
grupo <- factor(rep(x=1:G, each=ni))
obs <- rep(x=1:ni, times=G)
x <- runif(n=nobs, min=0, max=1)
b0 <- rnorm(n=G, mean=0, sd=sqrt(4)) # Intercepto aleatorio
b0 <- rep(x=b0, each=ni)           # El mismo intercepto aleatorio pero repetido
p <- inverse_logit(-1.4 + 0.33 * x + b0) # Siempre utilizar función inversa a la planteada en el modelo
y <- rbinom(n=nobs, size=1, prob=p)
datos <- data.frame(obs, grupo, b0, x, p, y)

```

2. Vector de parámetros Θ del modelo:

$$\Theta = (\beta_0 = -1.4, \beta_1 = 0.33, \sigma_{b_0}^2 = 4)$$

3. Estimación de parámetros ajustados

```

mod_simulado <- glmer(formula= y ~ x + (1 | grupo),
                      family = binomial(link="logit"),
                      data = datos)

```

```
summary(mod_simulado)
```

```

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: y ~ x + (1 | grupo)
## Data: datos
##
##      AIC      BIC   logLik deviance df.resid
## 17165.6 17189.3 -8579.8 17159.6    19997
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -6.3039 -0.4878 -0.2047  0.4473  8.2991
##
## Random effects:
## Groups Name      Variance Std.Dev.
## grupo (Intercept) 4.052    2.013
## Number of obs: 20000, groups: grupo, 200
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.5064      0.1495 -10.076 < 2e-16 ***
## x            0.3482      0.0670   5.196 2.03e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```
## Correlation of Fixed Effects:
##   (Intr)
## x -0.230
```

$$\Theta = (\hat{\beta}_0 = -1.5064, \hat{\beta}_1 = 0.3482, \hat{\sigma}_{b_0}^2 = 4.052)$$

4. Modelo ajustado

$$y_{ij} \sim \text{Bernoulli}(p_{ij}) \quad (5)$$

$$\text{logit}(p_{ij}) = -1.5064 + b_{0i} + 0.3482 x_{ij} \quad (6)$$

$$b_0 \sim N(0, 4.052) \quad (7)$$

5. Predicciones de los b_0 para cada grupo

```
efectos <- ranef(mod_simulado)[[1]]
head(efectos)
```

```
##   (Intercept)
## 1    1.6427031
## 2    3.6638514
## 3   -2.7429095
## 4   -0.1755459
## 5    0.7423772
## 6    4.4733740
```

6. Modelo ajustado para el grupo $i = 3$

$$y_{ij} \sim \text{Bernoulli}(p_{3j}) \quad (8)$$

$$\text{logit}(p_{3j}) = -1.5064 - 2.74291 + 0.3482 x_{ij} \quad (9)$$

7. Calcule manualmente estimación para $Pr(y_{ij} = 1)$ cuando $x=0.7$ y del grupo $i = 3$.

```
inverse_logit(-1.5064 -2.74291 + 0.3482*0.7)
```

```
## [1] 0.01788809
```

8. Utilizar función predict para obtener estimación anterior.

```
new_data = data.frame(x=0.7, grupo=3)
predict(mod_simulado, newdata=new_data, type="response")
```

```
##           1
## 0.01788739
```

9. Gráfico de $\widehat{Pr}(y_{ij} = 1)$ versus x para todos los grupos $i = 1, 5, 10, 15$

```
mod_simulado
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: y ~ x + (1 | grupo)
##   Data: datos
##       AIC       BIC    logLik deviance df.resid
## 17165.637 17189.347 -8579.818 17159.637    19997
## Random effects:
##   Groups Name      Std.Dev.
##   grupo  (Intercept) 2.013
## Number of obs: 20000, groups: grupo, 200
## Fixed Effects:
## (Intercept)          x
##    -1.5064         0.3482
```

```
valores_ajustados <- fitted(mod_simulado)
datos['valores_ajustados'] <- valores_ajustados
```

```
grupos <- c(1, 5, 10, 15)
datos_grupos <- datos %>% filter(grupo %in% grupos)
```

```
ggplot(data = datos_grupos, aes(x=x, y=valores_ajustados, color=grupo)) +
  geom_point()
```

