

Examen final

Probabilidad y simulación

Juan Cantero Jimenez

2 de diciembre, 2021

Ejercicio 1

Para obtener las probabilidades pedidas se deberá de aplicar el Teorema de Bayes, de tal forma que podamos "invertir" las probabilidades condicionadas dadas en el enunciado.

```
p_positivo_enfermo <- 0.7
p_negativo_sano <- 0.98
p_enfermo <- 0.015
```

a Valor predictivo positivo o $P(E|+)$:

```
p_enfermo_positivo <- (p_positivo_enfermo*p_enfermo)/(p_positivo_enfermo*p_enfermo +
(1-p_negativo_sano)*(1-p_enfermo))
p_enfermo_positivo

## [1] 0.3476821
```

b Valor predictivo negativo o $P(S|-)$:

```
p_sano_negativo <- (p_negativo_sano*(1-p_enfermo))/(p_negativo_sano*(1-p_enfermo)+
(1-p_positivo_enfermo)*p_enfermo)
p_sano_negativo

## [1] 0.9953599
```

c El valor predictivo negativo cambiará poco al disminuirse la incidencia de la población en la enfermedad. Sin embargo, esto no ocurrirá con el valor predictivo positivo que si cambia de forma más apreciable, reduciéndose.

```
p_positivo_enfermo <- 0.7
p_negativo_sano <- 0.98
p_enfermo <- 0.005

##ac Valor predictivo positivo o p_enfermo_positivo

p_enfermo_positivo <- (p_positivo_enfermo*p_enfermo)/(p_positivo_enfermo*p_enfermo +
(1-p_negativo_sano)*(1-p_enfermo))
p_enfermo_positivo

## [1] 0.1495726
```

```
##bc Valor predictivo negativo o p_sano_negativo

p_sano_negativo <- (p_negativo_sano*(1-p_enfermo))/(p_negativo_sano*(1-p_enfermo)
+(1-p_positivo_enfermo)*p_enfermo)
p_sano_negativo

## [1] 0.9984641
```

Ejercicio 2

- a Las distribuciones condicionales pedidas se pueden obtener mediante el cociente de la función de probabilidad conjunta partido la función de probabilidad marginal de la variable que esta condicionando, en este caso X. Así en R, las distribuciones pedidas pueden calcularse como.

```
fx <- function(x,y){
  p <- matrix(c(0.1, 0.2, 0.1,
                0.2, 0.3, 0.1),byrow = TRUE, nrow=2)
  return(p[x,y+1])
}
fx <- function(x){
  p <- matrix(c(0.1, 0.2, 0.1,
                0.2, 0.3, 0.1),byrow = TRUE, nrow=2)
  pp <- apply(p, 1, sum)
  pp[x]
}
fy <- function(y){
  p <- matrix(c(0.1, 0.2, 0.1,
                0.2, 0.3, 0.1),byrow = TRUE, nrow=2)
  pp <- apply(p, 2, sum)

  pp[y+1]
}

### a
y <- 0:2
py_dado_x1 <- sapply(y, function(y){
  fxy(1,y)/fx(1)
})
py_dado_x1

## [1] 0.25 0.50 0.25

py_dado_x2 <- sapply(y, function(y){
  fxy(2, y)/fx(2)
})
py_dado_x2

## [1] 0.3333333 0.5000000 0.1666667
```

- b Para el calculo de la función distribución de U simplemente se deben de sumar las probabilidades de aquellos valores de X e Y que den como resultado algún valor de la distribución de U. Así en R, la función de distribución pedida puede obtenerse:

```
fu <- function(u){  
  p <- c(fxy(1,1)+fxy(2,2), fxy(1,0)+fxy(2,1)+fxy(1,2), fxy(2,0))  
  return(p[u+1])  
}  
U = 0:2  
fu(0:2) # Para U = 0, 1, 2  
  
## [1] 0.3 0.5 0.2
```

Ejercicio 3

a

```
n = 20  
p = 0.2  
nm = 25  
# P (x < 80) = P(x <= 79)  
pnbinom(79, n, p)  
  
## [1] 0.5199786  
  
sum(dnbinom(0:79,n,p))  
  
## [1] 0.5199786
```

- b La probabilidad de que la media muestral sea inferior a 75, se puede calcular mediante el teorema central del límite, pues la media muestral seguirá una distribución normal con la misma media que la distribución binomial negativa y desviación típica la de esta distribución dividida por la raíz cuadrada del espacio muestral. Así:

```
media = n*(1-p)/p  
varianza = n*(1-p)/p^2  
p2 = pnorm(75, media, sqrt(varianza)/sqrt(nm))  
p2  
  
## [1] 0.1056498
```

Esta probabilidad podría obtenerse por simulación:

```
N = 10000  
nm = 25  
data <- matrix(rnbinom(N*nm,n,p), ncol=nm)  
data_mean <- apply(data, 1, mean)  
mean(data_mean < 75)  
  
## [1] 0.1026
```

- c El evento descrito puede modelizarse mediante una distribución binomial de probabilidad, la obtenida en el apartado anterior y tamaño 5. Así:

```
#P(x >= 1) = 1 - P(x <= 0)
n2 = 5
1-pbinom(0, n2, p2)

## [1] 0.4278128

1-dbinom(0, n2, p2)

## [1] 0.4278128
```

Ejercicio 4

a

```
set.seed(123)
N = 10000
n = 5
alfa = 20
beta = 4
data <- matrix(rgamma(n*N, alfa, beta), ncol=n)
p_values <- apply(data, 1, function(x){
  t.test(x, mu = 5, alternative = "two.sided")$p.value
})
mean(p_values < 0.05)

## [1] 0.0533
```

La hipótesis nula se rechazará en el 0.0533 de los casos. Este porcentaje representa el error de tipo 1, es decir rechazar la hipótesis nula cuando no se debería de rechazar. Puesto que este viene determinado por alfa, en un principio no se deberían de obtener porcentajes muy superiores a alfa.

b

```
set.seed(123)
N = 10000
n = 50
alfa = 20
beta = 4
data <- matrix(rgamma(n*N, alfa, beta), ncol=n)
p_values <- apply(data, 1, function(x){
  t.test(x, mu = 5, alternative = "two.sided")$p.value
})
mean(p_values < 0.05)

## [1] 0.0511
```

La hipótesis nula se rechazará en el 0.0511 de los casos. Como en el apartado anterior no se deberían de obtener valores de rechazo de la hipótesis nula superiores a alfa.