Ampliación de Inferencia Estadística

Tercero Grado de Estadística

Universidad de Sevilla

Tema 3: Inferencia Bayesiana en el modelo Beta-Binomial

Práctica-R-3

Problema 1.

Supongamos que estamos interesados en el estudio de la proporción de estudiantes universitarios que duermen al menos ocho horas. Esta proporción viene representada por p el cual es un valor desconocido. Desde el punto de vista Bayesiano, una creencia personal sobre la incertidumbre de esta proporción se puede representar mediante una distribución de probabilidad. Esta distribución refleja la opinión subjetiva a priori sobre los valores plausibles de p. Se toma una muestra de estudiantes universitarios y se les pregunta sobre sus horas de sueño. Por ejemplo, en un artículo de internet se publica que una mayoría de estudiantes duermen solo seis horas. En otro informe, aparece la información de que en una muestra de 100 estudiantes, el 70% duermen entre 5 y 6 horas, el 28% entre 7 y 8, y solo el 2% duermen 9 horas. Por tanto, se piensa que más del 50% duermen menos horas de las aconsejables, es decir $p < 0.5$, incluso se puede creer plausible que $p = 0.3$. Posteriormente, se toma una muestra de tamaño 27, de los cuales solo 11 duermen al menos 8 horas diarias. El objetivo consiste en estimar p así como predecir el número de estudiantes con horas de sueño óptimas para una nueva muestra de tamaño 20.

SOLUCIÓN.

CASO DISCRETO

Supongamos que tenemos información acerca del conjunto de posibles valores de p. De tal forma que en este caso dichos valores son:

$$Dominio(p) = \{0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95\}.$$

Además, por experimentos previos, sabemos la credibilidad de esos valores expresada en términos absolutos por el vector:

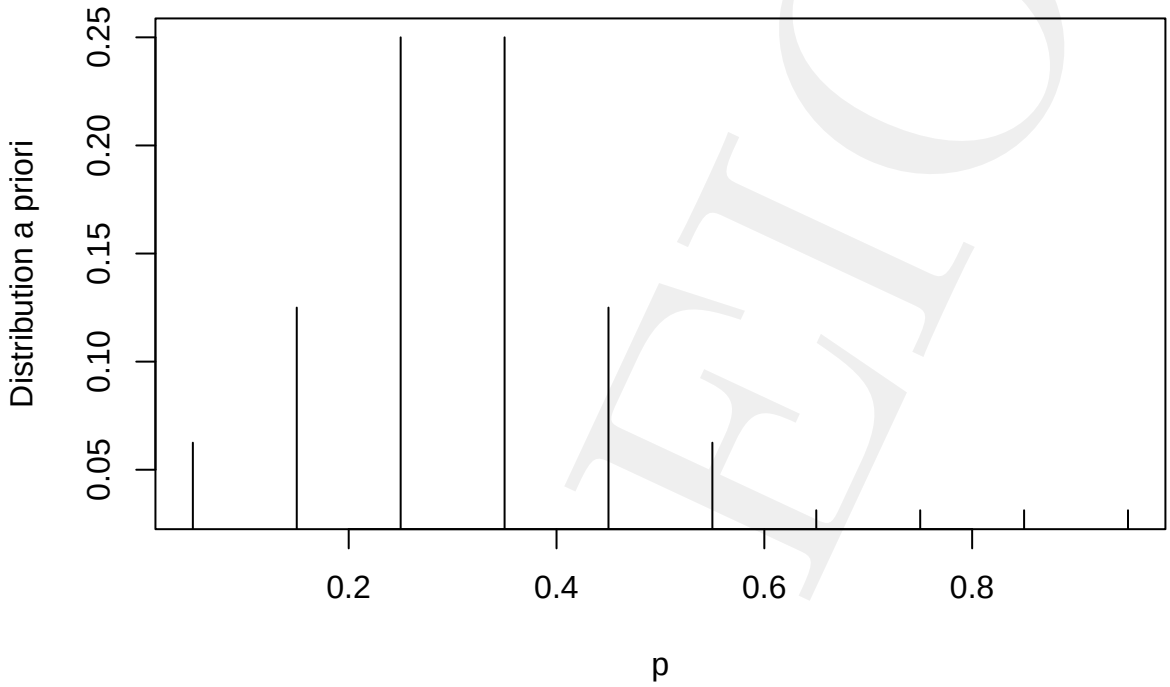
$$cred = (2, 4, 8, 8, 4, 2, 1, 1, 1, 1).$$

Es decir, la información disponible se puede modelar con la siguiente distribución de probabilidad a priori sobre p :

$$\pi(0.05) = 2/32, \pi(0.15) = 4/32, \cdots, \pi(0.95) = 1/32.$$

Por tanto, lo primero que tendremos que hacer en R es definir estos vectores y construir la distribución a priori a partir del vector de credibilidad. El conjunto de comandos para realizar estos cálculos es:

```
p=seq(0.05,0.95, by=0.1)
prior=c(2,4,8,8,4,2,1,1,1,1)
prior=prior/sum(prior)
plot(p,prior,type="h",ylab="Distribution a priori")
```



A partir de aquí se recoge una muestra que nos aporta la información de que hay 11 alumnos que duermen un número suficiente de horas y 16 que no duermen lo suficiente. Luego, asumiendo un modelo binomial (independencia e idénticamente distribuídas) la función de verosimilitud de la proporción es

$$l(p|datos) \propto p^{11}(1-p)^{16}$$

Por comodidad la información aportada por la muestra suele denotarse como *datos* aunque también es válida otra cualquiera que haga referencia a la misma. Para calcular la distribución a posteriori de p hay que aplicar el teorema de Bayes. Para ello será suficiente con calcular el producto de la a priori con la verosimilud ya que el denominador del teorema de Bayes juega el papel de normalizar el producto anterior para convertirlo en una función de probabilidad. En definitiva, tenemos que

$$\pi(p|Datos) \propto \pi(p) \times l(p|datos).$$

Así se obtiene

$$\pi(0.05|datos) \propto \pi(0.05) \times l(0.05|datos)$$

$$\vdots$$

$$\pi(0.95|datos) \propto \pi(0.95) \times l(0.95|datos).$$

Luego,

$$\pi(0.05|datos) \propto 2/32 \times 0.05^{11}0.95^{16}$$

$$\vdots$$

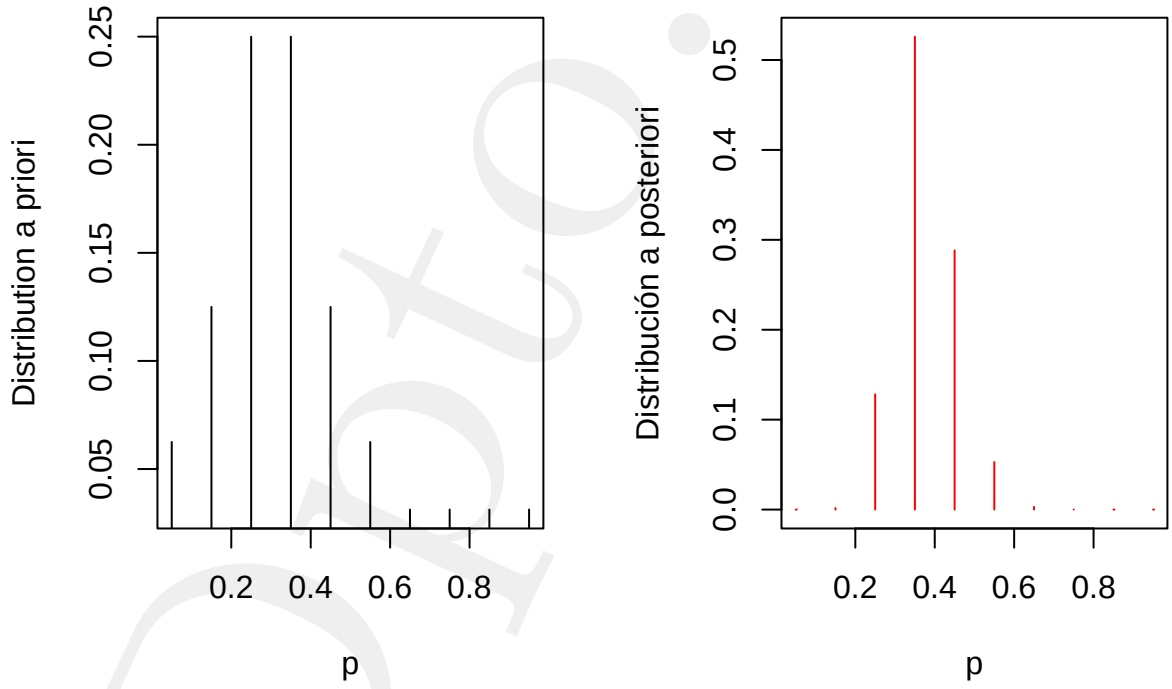
$$\pi(0.95|datos) \propto 1/32 \times 0.95^{11}0.05^{16}.$$

Recordemos que el símbolo $\propto$ significa *proporcional a*. La constante de proporcionalidad que falta en las expresiones anteriores es el cociente entre 1 y la suma de todos esos valores para convertirlos en una función de probabilidad. En primer lugar, es necesario instalar el paquete de R `LearnBayes`. Los comandos en R para realizar estos cálculos son los siguientes:

```
par(mfrow=c(1,2)) # Para que salgan dos gráficos en la misma altura
p=seq(0.05,0.95, by=0.1)
prior=c(2,4,8,8,4,2,1,1,1,1)
prior=prior/sum(prior)
plot(p,prior,type="h",ylab="Distribution a priori")
library("LearnBayes")
data=c(11,16)
post=pdisc(p,prior,data) # pdisc es un comando para calcular la distribución
# a posteriori de una discreta
cbind(p,prior,post) # cbind es un comando para unir vectores o matrices
```

```
##          p   prior         post
##  [1,] 0.05 0.06250 2.882642e-08
##  [2,] 0.15 0.12500 1.722978e-03
##  [3,] 0.25 0.25000 1.282104e-01
##  [4,] 0.35 0.25000 5.259751e-01
##  [5,] 0.45 0.12500 2.882131e-01
##  [6,] 0.55 0.06250 5.283635e-02
##  [7,] 0.65 0.03125 2.976107e-03
##  [8,] 0.75 0.03125 6.595185e-05
##  [9,] 0.85 0.03125 7.371932e-08
## [10,] 0.95 0.03125 5.820934e-15
```

```
plot(p,post, type="h", ylab="Distribución a posteriori",col="red")
```



Calculemos ahora el estimador bayesiano para la función de pérdida cuadrática.

```
pbayes=sum(p*post)
pbayes
```

```
## [1] 0.3771422
```

Para calcular la región de credibilidad al 95% tendremos que usar el paquete `tidyverse` para que resulta más cómodo el manejo de columnas (ver script-Practica-R-3).

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.1.8
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
lo=cbind(p,prior,post)
df=data.frame(lo)
a=df%>%arrange(desc(post)) #crea una nueva ordenada según post manteniendo los casos
a$postcum=cumsum(a$post)
sort(a$p[which(a$postcum<0.95)])
```

```
## [1] 0.25 0.35 0.45
```

Fijémonos que la muestra nos ha dado que 11 alumnos de 27 duermen las horas suficientes, luego nos saldría una primera estimacion de $\hat{p}_{MV} = 11/27 = 0.407$ por el método de máxima verosimilitud. Como se puede observar en la figura anterior la muestra origina que la distribución a posteriori se concentre aún más en torno a dicha estimación de 0.377. Sería como un efecto de mejora en la información pues la distribución a posterior tiene menos dispersión al desparecer las probabilidades de los valores inferiores y superiores de p.

CASO CONTINUO

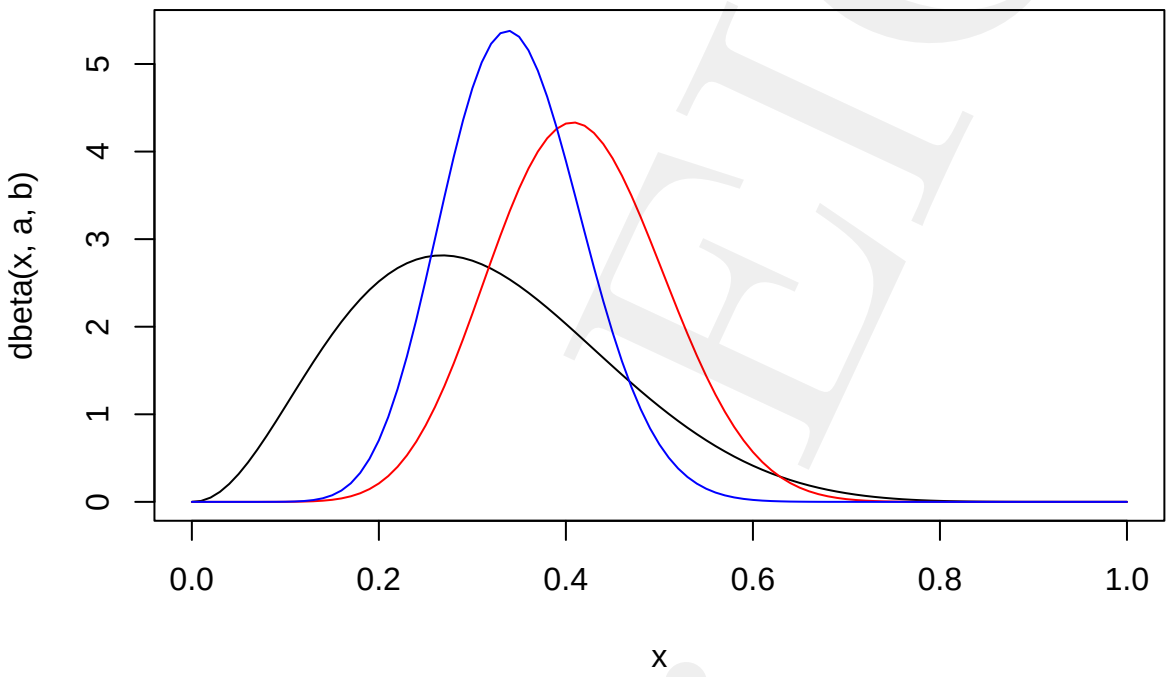
Resolvamos ahora un caso diferente al anterior en el sentido que el conjunto de posibles valores para el parámetro es el intervalo $[0, 1]$. Como hemos visto que la función de verosimilitud del parámetro tiene la *forma* de la distribución beta, la a priori adecuada según el Método de distribuciones conjugadas es la distribución beta. Se ha demostrado en teoría que si el parámetro p tiene una distribución beta de parámetros conocidos a y b y la verosimilitud es de la forma $l(p|datos) \propto p^k(1-p)^{n-k}$ entonces la distribución a posteriori es una beta con parámetros $a + k$ y $b + n - k$. En primer lugar vamos a determinar cuáles son los valores de la distribución beta que mejor se aproxima a la información que disponemos. En el paquete `LearnBayes` hay un comando que nos permite seleccionar una beta a priori a partir del conocimiento de los percentiles de la misma. En este caso se supone que la mediana es 0.3 y el percentil 90 es es 0.5. El comando para obtener los parámetros de la beta que mejor se ajusta a esa información es `beta.select()`:

```
s=11;f=16 #se supone que la a priori tiene de parámetros a y b, debido a la información
#existente.
q1=list(p=.5,x=0.3)
q2=list(p=.9,x=0.5)
beta.select(q1,q2)
```

```
## [1] 3.26 7.19
```

```
a=beta.select(q1,q2)[1]
b=beta.select(q1,q2)[2]
```

```
curve(dbeta(x,a,b),col="black",xlim=range(0,1),ylim=range(0,5.4)) #a priori
curve(dbeta(x,s+1,f+1),col="red",xlim=range(0,1),add=T) #verosimilitud
curve(dbeta(x,a+s,f+s),col="blue",xlim=range(0,1),add=T) #a posteriori
```



Analicemos la figura anterior. La distribución a priori posee unos parámetros de 3.26 y 7.19 lo cual da un gráfico dándole más probabilidad a los valores cercanos a 0, en particular se centra en 0.25. Es decir, la información que disponemos nos está diciendo que la moda está entorno a un 25% más o menos de la población duermen lo suficiente. Sin embargo, la muestra nos dice que esa moda ahora se sitúa en el 40%. Esta información aportada por la muestra origina que la distribución a priori se transforma en la a posteriori desplazándose hacia la derecha y centrándose en la moda que vale:

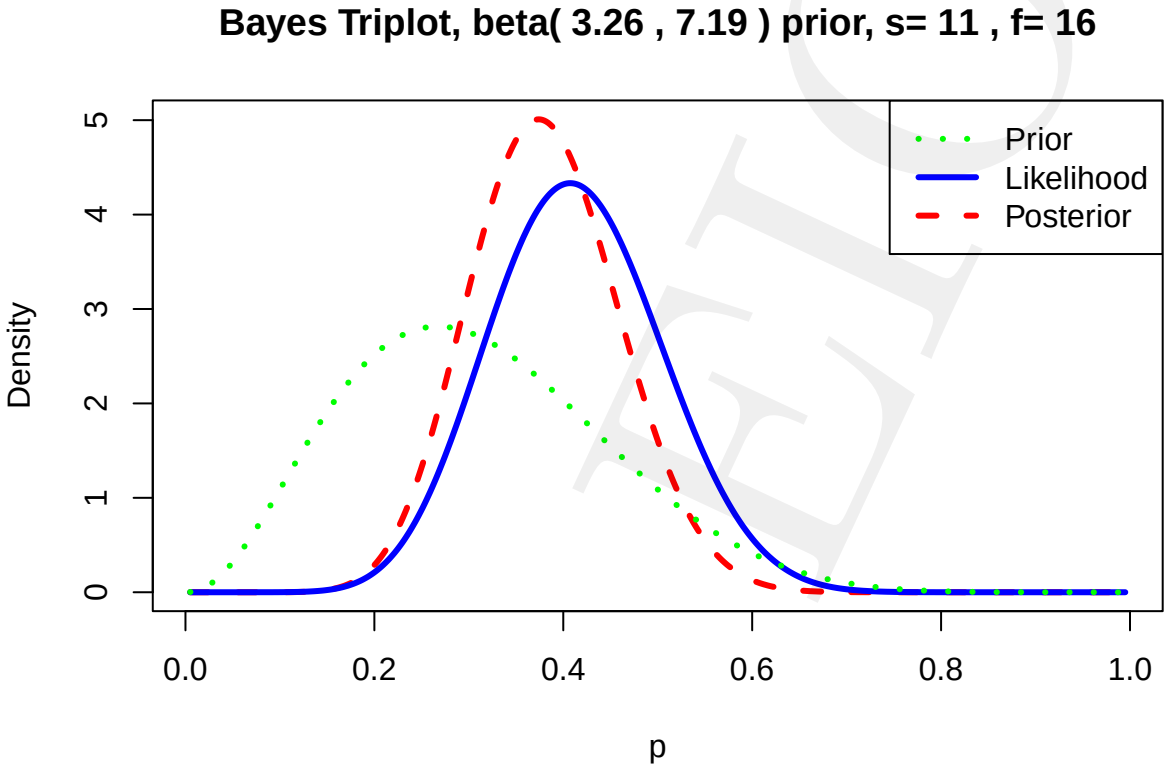
$$Mo = \frac{a+k-1}{a+b+n-2} = \frac{3.26+11-1}{3.26+7.19+27-2} = 0.374.$$

Otro efecto que puede verse es como con la nueva información se consigue disminuir la dispersión de la distribución a priori ya que la distribución a posteriori está más concentrada en el entorno de la moda.

A continución dibujamos en un mismo gráfico la a priori, la verosimilitud y la a posteriori.

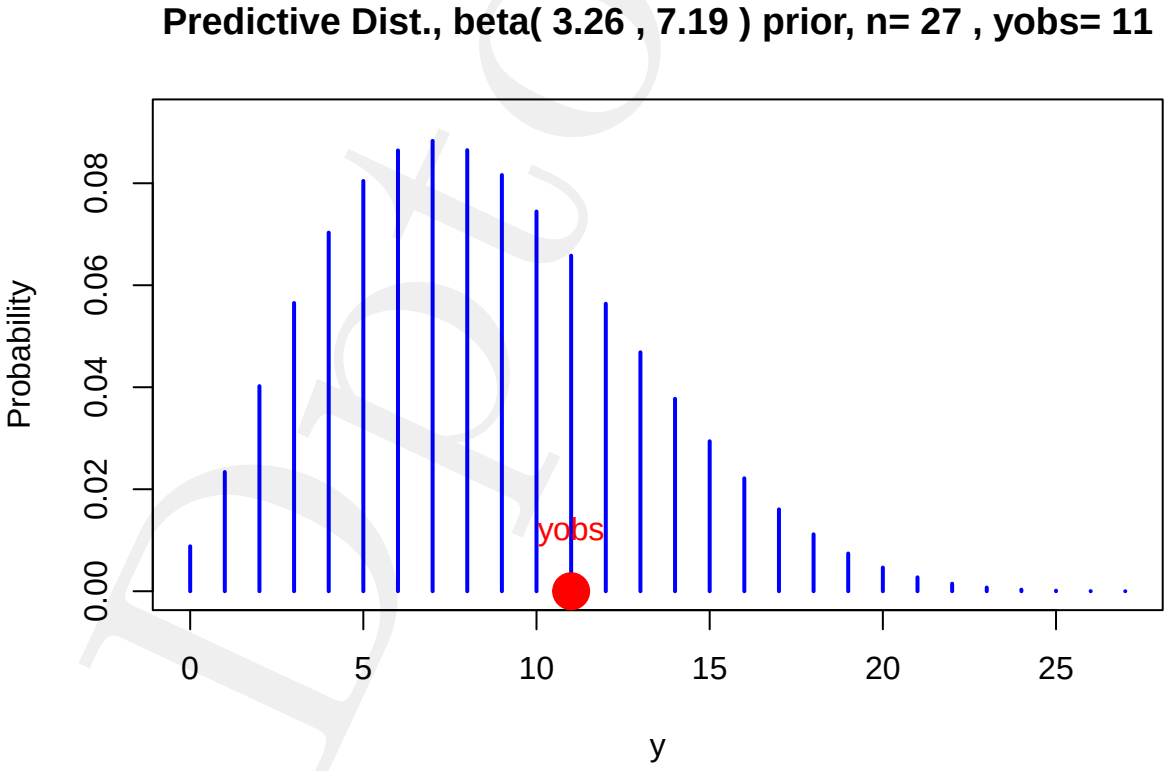
Los comandos siguientes hacen lo mismo

```
par=c(a,b)
triplot(par,c(s,f))
```



Dibuja un plot para ver el ajuste de la predictiva con lo ocurrido

```
predplot(c(a,b), 27, 11)
```



Calculemos diferentes aspectos de la a posteriori.

Por ejemplo, ¿es plausible que la proporción de estudiantes que duermen al menos ocho horas diarias es superior al 50%? Es decir, vamos a calcular el factor de Bayes para el contraste $H_0 : p < 0.5$ frente a $H_1 : p >= 0.5$

```
pp1<-1-pbeta(0.5,a+s,f+b)
pp0=pbeta(0.5,a+s,f+b)
p1=1-pbeta(0.5,a,f)
p0=pbeta(0.5,a,f)
FB01= (pp0/pp1)/(p1/p0)
print(FB01)
```

```
## [1] 14567.85
```

Usando la tabla de Jeffrey ese valor nos indica que hay una evidencia extrema para H_0.

Podemos obtener también la versión de los intervalos de confianza, que en el caso Bayesiano se llaman regiones creíbles si queremos una zona creíble al 95% calcularemos los cuantiles a posteriores al nivel del 5% y 95%

```
rc<-qbeta(c(0.05,0.95),a+s,b+f)
print(rc)
```

```
## [1] 0.2555267 0.5133608
```

Para calcular el intervalo de máxima densidad (HPDI) tenemos que recurrir a simular una muestra de la densidad a posteriori y luego usar el paquete `HDInterval`. En este caso tendríamos

```
library(HDInterval)
muestra=rbeta(10000,a+s,b+f)
dens <- density(muestra)
hdi(dens, credMass=0.90)
```

```
##     lower     upper
## 0.2524999 0.5101557
## attr(,"credMass")
## [1] 0.9
## attr(,"height")
## [1] 1.413922
```

Como se puede comprobar son prácticamente iguales y eso se debe a la casi simetría de la densidad beta alrededor de la moda. Vemos ahora como se usaría la distribución predicitiva en el modelo beta-binomial.

```
ab=c(11+a,16+b) # ojo que en realidad esta es la aposteriori, que es la apriori en el
                # modelo predictivo
m=20; ys=0:20
pred<-pbetap(ab,m,ys)
pred=round(pred,3)
pp=cbind(ys,pred)
pp
```

```
##       ys  pred
##  [1,]  0 0.001
##  [2,]  1 0.004
##  [3,]  2 0.014
##  [4,]  3 0.035
##  [5,]  4 0.066
##  [6,]  5 0.100
##  [7,]  6 0.130
##  [8,]  7 0.145
##  [9,]  8 0.143
```

```
## [10,]  9 0.124
## [11,] 10 0.096
## [12,] 11 0.065
## [13,] 12 0.040
## [14,] 13 0.021
## [15,] 14 0.010
## [16,] 15 0.004
## [17,] 16 0.001
## [18,] 17 0.000
## [19,] 18 0.000
## [20,] 19 0.000
## [21,] 20 0.000
```

```
predplot(ab,20,7)
```