

Introducción a los Modelos Bayesianos con R

R-Ladies

Carolina Montoya Rivera

Actuaria, Instituto Nacional de Seguros

22/05/2019

Stio web: R-LADIES GLOBAL

Rladies xaringan theme



Copyrighted Material

Use R!

Jim Albert

Bayesian Computation with R

 Springer

Copyrighted Material

Índice

- Capítulo 1: Introducción a R.
- **Capítulo 2: Introducción al pensamiento bayesiano.**
- **Capítulo 3: Modelos Univariados.**
- Capítulo 4: Modelos Multivariados.
- Capítulo 5: Introducción a la computación bayesiana.
- Capítulo 6: Métodos MCMC
- Capítulo 7: Modelos Jerárquicos.
- Capítulo 8: Comparación de modelos.
- Capítulo 9: Modelos de regresión.
- Capítulo 10: Muestreo de Gibbs.
- Capítulo 11: R como interfaz de WinBUGS

Capítulo 2

Introducción al pensamiento Bayesiano

- La distribución de probabilidad de una variable aleatoria X , es una función que asigna la probabilidad de ocurrencia a cada realización de X .
- La idea básica de la estadística bayesiana, consiste en considerar tanto los datos como los parámetros (θ) asociados a su distribución como variables aleatorias.
- Supongamos que D contiene la información sobre el parámetro θ . Suponga que tenemos un vector de n observaciones \mathbf{x} . La verosimilitud se denota como $L(\mathbf{x}|\theta)$ y describe la probabilidad de obtener los valores de \mathbf{x} dado que θ es el parámetro verdadero. Por el teorema de Bayes, la distribución posterior de θ es:

- $p(\theta)$ es la distribución previa y $p(\theta|\mathbf{x})$ es la distribución posterior del parámetro

Ejemplo: Estimación de una proporción.

- A una persona se le asigna un estudio donde tiene que determinar:
 ¿Qué proporción de estudiantes universitarios duerme al menos ocho horas?
 - Supongamos que la persona cree que la proporción es igualmente probable que sea menor o mayor que 0.5 . Además, tiene una confianza del 90% de que es menor que 0.5 .
 - La distribución beta es útil para modelar proporciones puesto que su dominio está en el intervalo $(0,1)$.
-
- ¿Cómo podemos modelar este conocimiento previo del parámetro mediante una distribución beta en R?

Ajuste de la previa

- Ocupamos una distribución beta cuyo cuantil al 50% sea 0.3 y al 90% sea 0.5:

```
Param_previa<- function(q,p,init, dist) {  
  previa_optim <- function(param) {  
    (dist(p[1],param[1],param[2])-q[1])^2 + (dist(p[2],param[1],param[2])-q[2])^2  
  }  
  r <- optim(init,previa_optim)  
  r <- unlist(r)  
  r <- c(r[1],r[2])  
  return(data.frame(r))  
}  
  
res <- Param_previa(c(0.3,0.5), c(0.5,0.9),c(1,1), qbeta)  
kable(res, format='html')
```

	r
par1	3.262585
par2	7.183773

De aquí vemos que la previa de la distribución es

.

- Se toma una muestra de 27 estudiantes universitarios, de los cuales se observa que 11 reportaron haber dormido al menos 8 horas la noche anterior. ¿Cómo podemos incorporar esta información con nuestro conocimiento previo?
- La distribución binomial modela eventos de acierto-desacierto en un experimentos independientes. Supongamos que vemos como éxito el dormir al menos 8 horas. Tomamos una muestra aleatoria de estudiantes con éxitos y fallos, entonces la verosimilitud es:

- Recordemos que _____ y _____ luego:

- Es decir, _____ . Este es un ejemplo de un análisis conjugado, donde la distribución previa y posterior del parámetro, tienen la misma forma funcional.

```

p <- seq(0, 1, length = 500)
a <- res$r[1]
b <- res$r[2]
s = 11
f = 16

df <- data.frame(p = seq(0, 1, length = 500),
  Previa= dbeta(p,a,b),
  Verosimilitud= dbeta(p,s+1,f+1),
  Posterior= dbeta(p,a+s,b+f))
df[1:3,]

```

```

##           p      Previa Verosimilitud   Posterior
## 1 0.000000000 0.000000000 0.000000e+00 0.000000e+00
## 2 0.002004008 0.0002956486 7.401509e-22 1.186151e-25
## 3 0.004008016 0.0014011469 1.467854e-18 1.114834e-21

```

```

df <- gather(df,
  key = "Dist",
  value = "Densidad", -p)

df[1:3,]

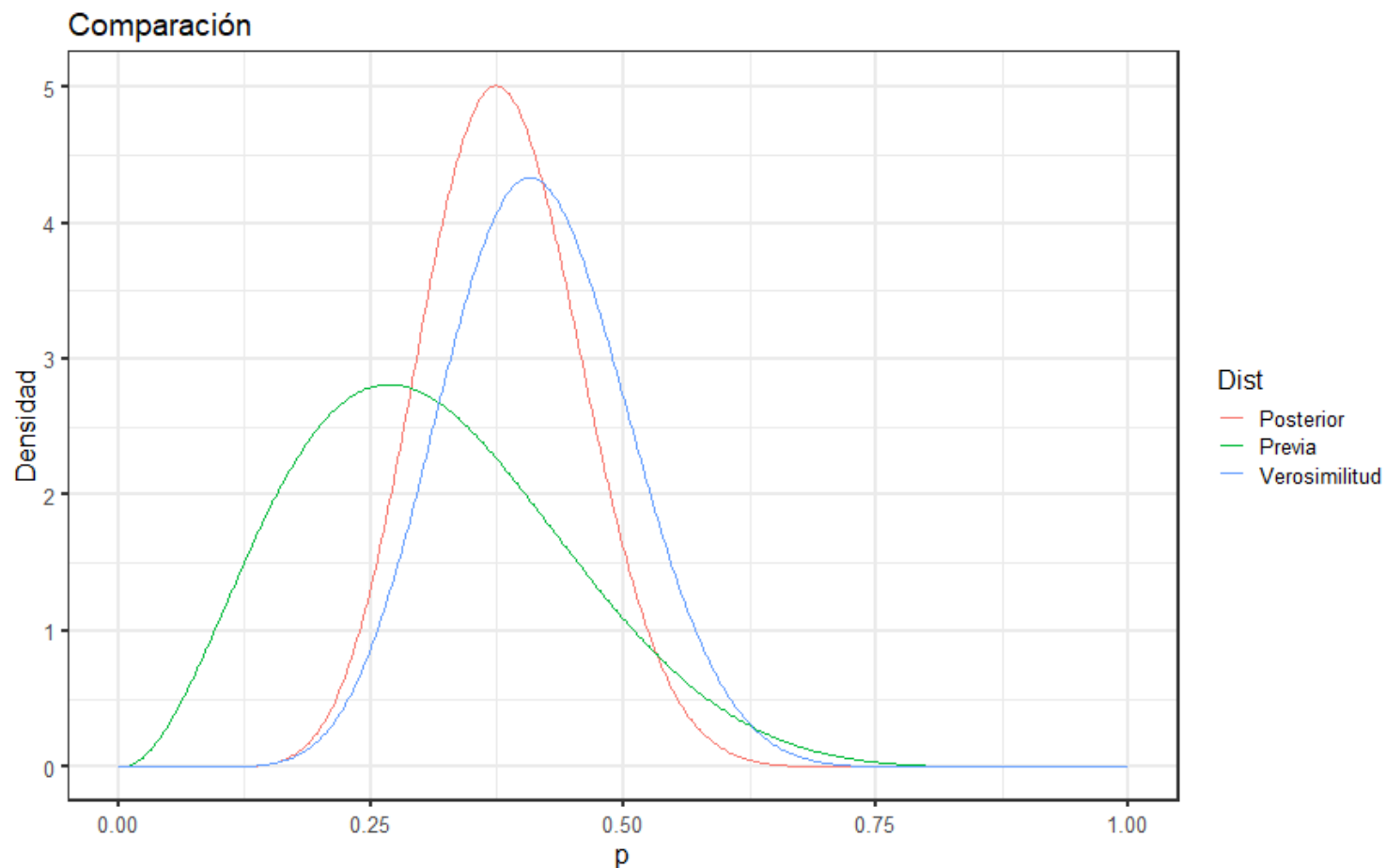
```

```

##           p   Dist      Densidad
## 1 0.000000000 Previa 0.0000000000
## 2 0.002004008 Previa 0.0002956486
## 3 0.004008016 Previa 0.0014011469

```

```
ggplot(data=df, aes(x=p,y=Densidad,colour=Dist))+  
  geom_line()+  
  theme_bw(base_size=16)+  
  ggtitle("Comparación")
```



Inferencia sobre el parámetro

- ¿

```
#Usando la distribución teórica  
1-pbeta(0.5, a+s, b+f)
```

```
## [1] 0.0692092
```

```
#Usando simulaciones  
ps <- rbeta(1000,a+s,b+f)  
sum(ps>=0.5)/1000
```

```
## [1] 0.058
```

```
#Aumentando el número de simulaciones  
ps0 <- rbeta(100000,a+s,b+f)  
sum(ps0>=0.5)/100000
```

```
## [1] 0.06925
```

- ¿Intervalo de confianza?

```
#Usando la distribución teórica  
qbeta(c(0.05, 0.95), a + s, b + f)
```

```
## [1] 0.2556166 0.5134772
```

```
#Usando simulaciones  
quantile(ps, c(0.05,0.95))
```

```
##           5%           95%  
## 0.2601120 0.5096245
```

```
#Aumentando el número de simulaciones  
quantile(ps0, c(0.05,0.95))
```

```
##           5%           95%  
## 0.2556724 0.5134142
```

Inferencia sobre el parámetro

- ¿Media?

```
#Usando la distribución teórica  
(a+s)/((a+s)+(b+f))
```

```
## [1] 0.3808804
```

```
#Usando simulaciones  
sum(ps)/length(ps)
```

```
## [1] 0.3804161
```

```
#Aumentando el número de simulaciones  
sum(ps0)/length(ps0)
```

```
## [1] 0.3807846
```

- ¿Varianza?

```
#Usando la distribución teórica  
(a+s)*(b+f)/(((a+s)+(b+f))^2*((a+s)+(b
```

```
## [1] 0.006133495
```

```
#Usando simulaciones  
var(ps)
```

```
## [1] 0.005907259
```

```
#Aumentando el número de simulaciones  
var(ps0)
```

```
## [1] 0.006113915
```

Inferencia sobre una función del parámetro

- **Monte Carlo.** Queremos calcular:

Suponga que se tiene simulada una muestra _____ . Entonces:

- ¿Media de _____ ?

```
est=mean(ps0^2)
est
```

```
## [1] 0.1511108
```

El error estándar asociado a este parámetro es:

- ¿Varianza?

```
se=sd(ps0^2)/sqrt(1000)
se
```

```
## [1] 0.001931197
```

Predicción

- También nos interesa conocer la cantidad de estudiantes dormilones en una muestra futura. La densidad predictiva de y está dada por:

Si π es la dist. previa, entonces nos referimos a $\pi(y)$ como la densidad predictiva previa, mientras que si π es la dist. posterior, $\pi(y)$ es la densidad predictiva posterior.

- En este caso particular, se puede calcular analíticamente la siguiente expresión cerrada para la densidad posterior:

Donde $\pi(y)$, $\pi(y)$ y $\pi(y)$.

```
ab=c(a, b)
m=20; xs=0:20
pred=pbetap(ab, m, xs)
kable(data.frame(xs=0:9, pred=pred[1:10
```

xs	pred
0	0.0180389
1	0.0449541
2	0.0722845
3	0.0943781
4	0.1083502
5	0.1135104
6	0.1106848
7	0.1015893
8	0.0883129
9	0.0729318

```
kable(data.frame(xs=10:20,
  pred=pred[11:21]),
  format='html')
```

xs	pred
10	0.0572497
11	0.0426510
12	0.0300475
13	0.0198972
14	0.0122719
15	0.0069550
16	0.0035491
17	0.0015796
18	0.0005808
19	0.0001589
20	0.0000246

- De acuerdo a esta distribución predictiva, lo más probable es que al tomar una muestra de 20 estudiantes, hayan 5 o 6 dormilones.

Usando simulaciones

```
p=rbeta(10000,a, b)
x = rbinom(10000, 20, p)
table(x)
```

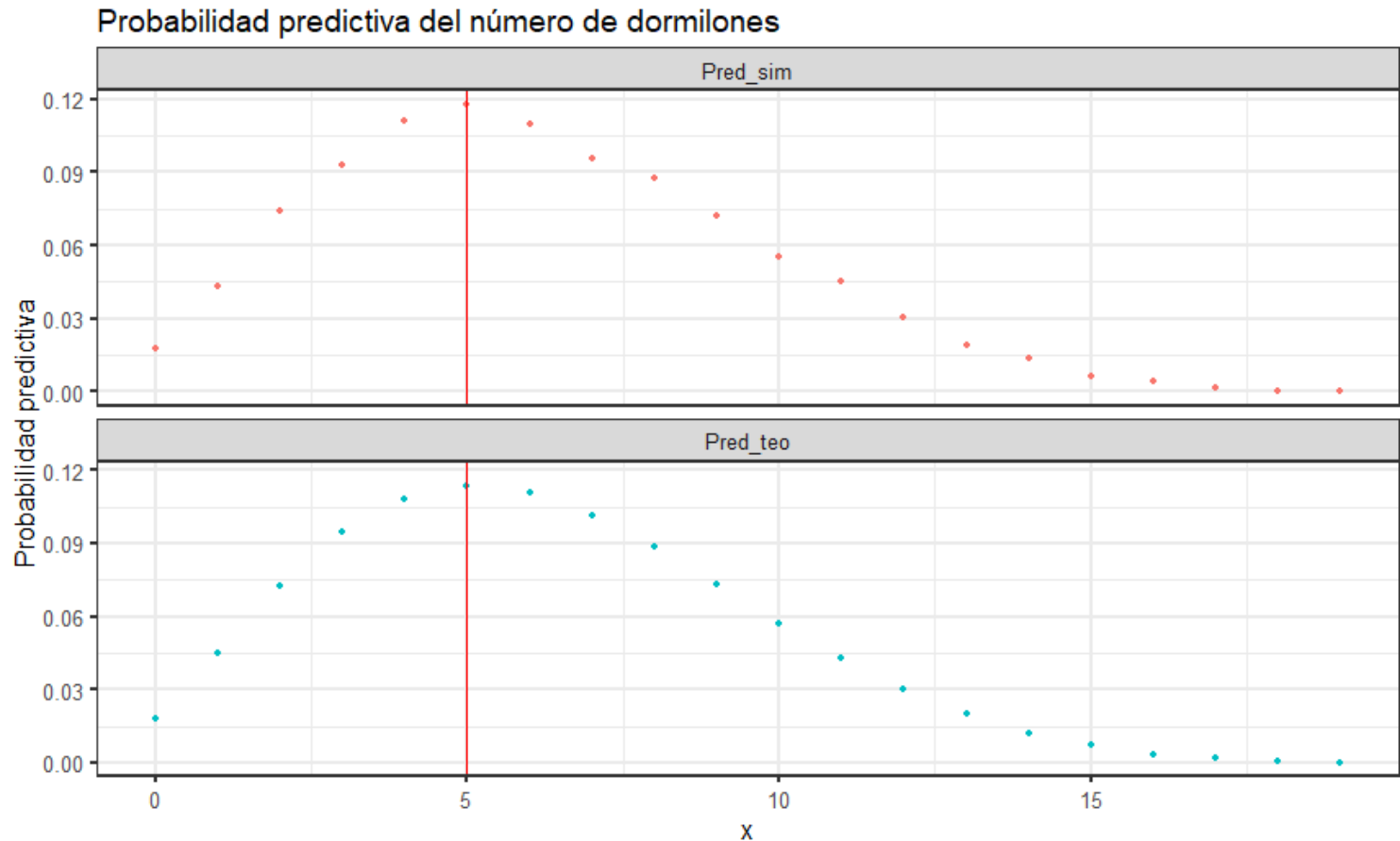
```
## x
##   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
## 177  434  744  932 1112 1179 1100  954  873  723  557  453  307  191  138
##   15   16   17   18   19
##   66   41   13    3    3
```

```
freq=table(x)
xs=c(0:max(x))
predprob=freq/sum(freq)
predprob
```

```
## x
##   0    1    2    3    4    5    6    7    8    9
## 0.0177 0.0434 0.0744 0.0932 0.1112 0.1179 0.1100 0.0954 0.0873 0.0723
##   10   11   12   13   14   15   16   17   18   19
## 0.0557 0.0453 0.0307 0.0191 0.0138 0.0066 0.0041 0.0013 0.0003 0.0003
```

```
df <- data.frame(x=0:(length(predprob)-1),Pred_sim=as.vector(predprob), Pred_teo=p)
df <- gather(df,"Densidad", "Probabilidad_predictiva",-x)
```

```
ggplot(data=df, aes(x=x, y=Probabilidad_predictiva, color=Densidad))+
  geom_point(lwd=1.5)+
  labs(x="x", y="Probabilidad predictiva", title=("Probabilidad predictiva del número
  geom_vline(xintercept = 5, colour="red")+
  facet_wrap(~ Densidad, dir = "v")+
  theme_bw(base_size=16)+
  theme(legend.position="none")
```



Links de interés (teórico)

Familias exponenciales

Tabla con previas conjugadas

Capítulo 3

Modelos univariados:

- Distribución Normal con media conocida y varianza desconocida:

```
data(footballscores)
DT::datatable(
  head(footballscores, 20),
  fillContainer = FALSE, options = list(pageLength = 3, scrollX = TRUE)
)
```

Show entries

Search:

	year	home	favorite	underdog	spread	favorite.name	under
1	1981	1	21	13	2	TB	MIN
2	1981	1	27	0	9.5	ATL	NO
3	1981	1	31	0	4	BUF	NYJ

Showing 1 to 3 of 20 entries

Previous

1

2

3

4

5

6

7

Next

- Estamos interesados en estimar la diferencia

Si se asume que estas diferencias son $\mu_1 - \mu_2$, entonces la verosimilitud de $\mu_1 - \mu_2$ está dada por:

—

- Suponemos la previa no informativa: $\mu_1 - \mu_2 \sim N(0, \sigma^2)$. Entonces:

Donde $\sigma^2 = \sigma_1^2 + \sigma_2^2$.

- Si definimos $\mu = \mu_1 - \mu_2$ se puede mostrar que:

```
attach(footballscores)

d <- favorite - underdog - spread
n <- length(d)
v <- sum(d^2)
```

Para hacer inferencia sobre μ usamos la transformación $\frac{\bar{d} - \mu}{s/\sqrt{n}}$:

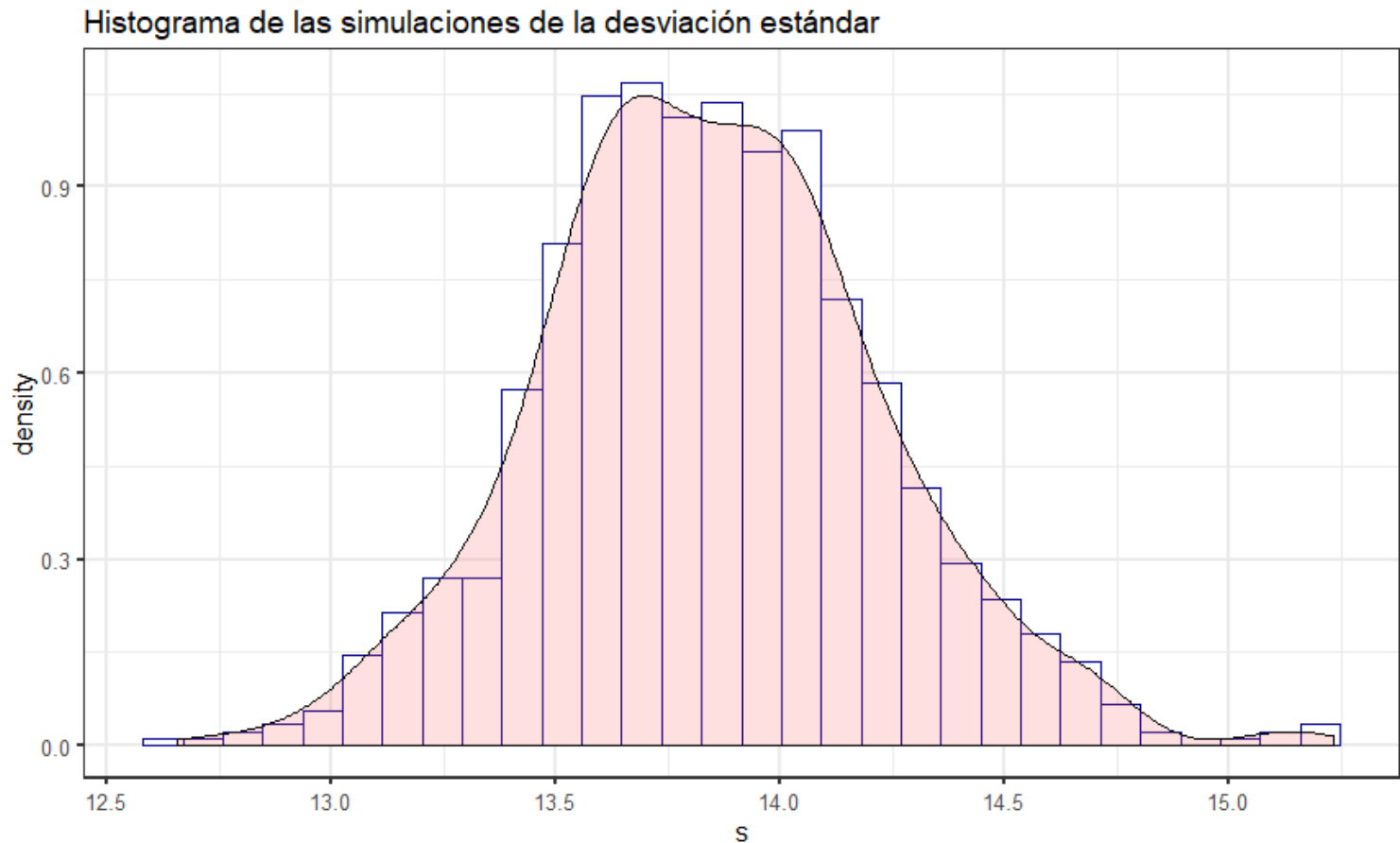
```
P <- rchisq(1000, n)/v
s <- sqrt(1/P)

kable(quantile(s, probs = c(0.025, 0.5, 0.975)), col.names=c("Cuantil"), format='h'
```

	Cuantil
2.5%	13.12307
50%	13.83335
97.5%	14.64179

Los percentiles al 2.5% y 97.5% forman un intervalo de confianza al 95% para μ .

```
ggplot(data.frame(s), aes(x=s))+  
geom_histogram(aes(y=..density..),color="darkblue", fill="white")+  
geom_density(alpha=.2, fill="#FF6666")+  
theme_bw(base_size=16)+  
labs(title="Histograma de las simulaciones de la desviación estándar")
```



Estimación de la tasa de mortalidad por trasplantes de corazón.

- n : Cantidad de cirugías de trasplante cardíaco en el hospital.
- d : Número de muertes observadas.
- x : Exposición.

Donde λ es la tasa de mortalidad por unidad de exposición. El objetivo es estimar λ .

- El estimador estándar de MLE para λ es $\frac{d}{x}$. Problema: cuando x es cercano a 0, no es un buen estimador.
- Bajo este contexto, intentaremos un ajuste bayesiano.

- Una escogencia conveniente para la distribución de la previa es una distribución miembro de la familia de la forma:

- Se observan n muertes, con exposición t , (son 10 hospitales). Donde:

Tomando la previa no informativa, la dist. actualizada de λ dada esta data es:

- Recordemos que para un v.a Poisson:

- Luego _____.

- Si el número de muertes observado de un hospital x par a un hospital con exposición e , es $x \sim \text{Poisson}(\lambda)$ y se asigna la previa $\lambda \sim \text{Gamma}(\alpha, \beta)$, entonces la distribución posterior tiene la forma:

Es decir, es una

- También podemos calcular la distribución predictiva de x :

```
alpha<-16;beta<-15174
yobs<-1; ex<-66; y<-0:10; lam<-alpha/beta
py<-dpois(y, lam*ex)*dgamma(lam, shape = alpha,rate = beta)/dgamma(lam, shape= alp
kable(cbind(y, round(py, 3)),format='html')
```

y	
0	0.933
1	0.065
2	0.002
3	0.000
4	0.000
5	0.000
6	0.000
7	0.000
8	0.000
9	0.000
10	0.000

```
lambdaA <- rgamma(1000, shape = alpha + yobs, rate = beta + ex)
```

```
ex <- 1767; yobs<-4
y <- 0:10
py <- dpois(y, lam * ex) * dgamma(lam, shape = alpha, rate = beta)/dgamma(lam, shape = alpha, rate = beta)
kable(cbind(y, round(py, 3)),format='html')
```

y	
0	0.172
1	0.286
2	0.254
3	0.159
4	0.079
5	0.033
6	0.012
7	0.004
8	0.001
9	0.000
10	0.000

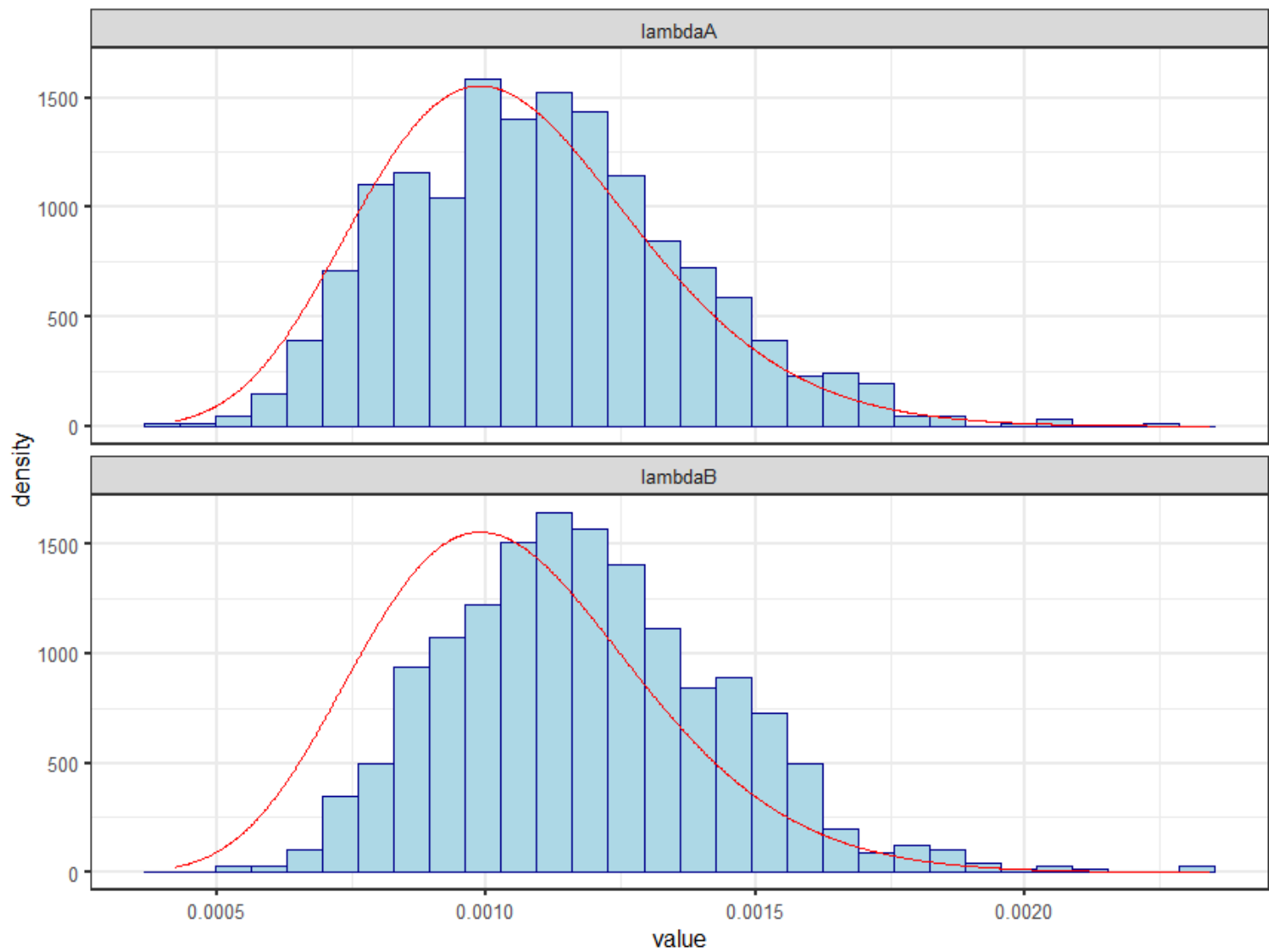
```
lambdaB <- rgamma(1000, shape = alpha + yobs, rate = beta + ex)
```

```
lambda <- data.frame(lambdaA, lambdaB)
lambda <- gather(lambda)

grid <- with(lambda, seq(min(value), max(value), length = 1000))

gammad <- lambda %>%group_by(key) %>% mutate(
  value = grid,
  density = dgamma(grid, alpha, beta))

ggplot(lambda, aes(x=value)) +
  geom_histogram(aes(y=..density..), color="darkblue", fill="lightblue") +
  geom_line(aes(y = density),data=gammad, colour = "red") +
  facet_wrap(~ key,dir = "v")+
  theme_bw()
```



Robustez bayesiana

- Hay muchas distribuciones previas que podrían calzar con nuestra información inicial.
- Un análisis bayesiano se dice robusto si la inferencia es insensible a la elección de la previa.
- Supongamos que queremos estimar el IQ de una persona llamada Joe. Creemos que Joe tiene una inteligencia promedio, la media de nuestra previa será 100. Además estamos 90% seguros que su IQ caerá entre 80 y 120.

```
res <- Param_previa(c(80,120),c(0.05,0.95),c(1,1),qnorm);kable(res,format='html')
```

	r
par1	100.00364
par2	12.15611

Luego

- Joe toma 4 tests. Sus notas son y_1, y_2, y_3, y_4 . Asumiendo que μ es la media desconocida y σ^2 es la varianza desconocida, la nota promedio \bar{y} es una estimación de μ con sd $\frac{\sigma}{\sqrt{4}}$.

- Tomando la previa normal, la posterior también será normal con los siguientes parámetros:

Vamos a calcular la distribución posterior bajo 3 escenarios de nota promedio observada:

```
mu <- res$r[1]; tau <- res$r[2]
sigma <- 15
n<-4
se <- sigma/sqrt(4)
ybar <- c(110, 125, 140)
tau1 <- 1/sqrt(1/se^2 + 1/tau^2)
mu1 <- (ybar/se^2 + mu/tau^2)*tau1^2
summ1 <- cbind(ybar, mu1, tau1)
kable(summ1, format='html')
```

ybar	mu1	tau1
110	107.2439	6.382905
125	118.1083	6.382905
140	128.9727	6.382905

- Consideremos ahora otra previa. Una t-student con parámetro de localización y escala

$$\frac{\frac{1}{\sigma} \left(\frac{1}{\mu} - \frac{1}{\sigma} \right)}{\frac{1}{\sigma} \left(\frac{1}{\mu} - \frac{1}{\sigma} \right) + \frac{1}{\sigma} \left(\frac{1}{\mu} - \frac{1}{\sigma} \right)}$$

```
Param_previa_ts <- function(q,p,init) {
  previa_optim <- function(param) {
    (qst(p=p[1],mu=param[1],sigma=param[2],nu=2)-q[1])^2 + (qst(p=p[2],mu=param[1]
  )
  }
  r <- optim(init,previa_optim)
  r <- unlist(r)
  r <- c(r[1],r[2])
  return(data.frame(r))
}

rest <- Param_previa_ts(c(80,120),c(0.05,0.95),c(1,1))
kable(rest, format='html')
```

	r
par1	99.993578
par2	6.850288

```

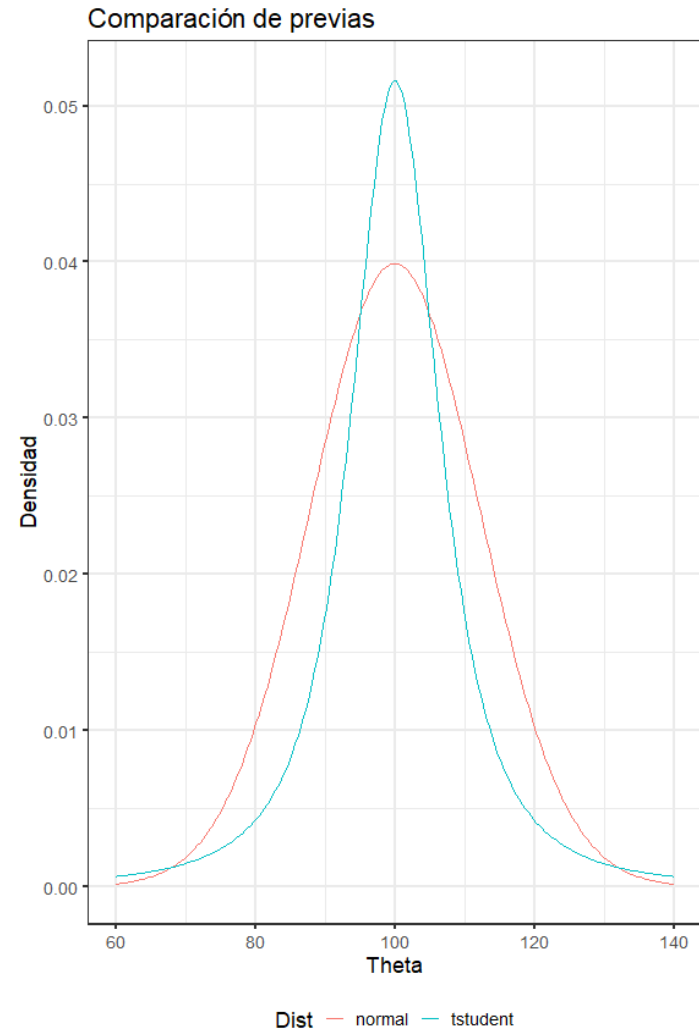
theta <- seq(60, 140, length = 200)

mu <- rest$r[1]
tscale <- rest$r[2]

df <- data.frame(Theta=seq(60, 140, length = 200),
                 tstudent=1/tscale*dt(
                   normal=1/10*dnorm((theta-mu)/tscale),
                   tstudent=1/dt(
                     theta-mu, df=rest$df,
                     scale=tscale)
                 ),
                 Dist = "normal")
df <- gather(df, key = "Dist",
             value = "Densidad",
             ~Theta)

ggplot(data=df, aes(x=Theta, y=Densidad)) +
  geom_line() +
  theme_bw(base_size=18) +
  ggtitle("Comparación de previas") +
  theme(legend.position = "bottom")

```



Cálculo de la distribución posterior

- Haremos el cálculo de la posterior usando las dos previas:

—

```
summ2 = c()
postts <- matrix(nrow=500,ncol=3)
for(i in 1:3){
  theta = seq(60, 180, length = 500)
  like = dnorm((theta - ybar[i])/7.5)
  prior = dt((theta - mu)/tscale, 2)
  post = prior * like
  post = post/sum(post)
  postts[,i] <-post
  m = sum(theta * post)
  s = sqrt(sum(theta^2 * post) - m^2)
  summ2 = rbind(summ2, c(ybar[i], m, s))
}

postts <- cbind(c(rep(110,500),rep(125,500),rep(140,500)),c(postts[,1:3]))

normpost = matrix(nrow=500,ncol=3)
normpost = sapply(1:3, function(i) normpost[,i] = dnorm(theta, mu1[i], tau1)/sum(d
normpost <- cbind(c(rep(110,500),rep(125,500),rep(140,500)),c(normpost[,1:3]))
```

```

moments <- data.frame(round(summ1,2),round(summ2[, -1],2))
names(moments) <- c("y", "E_n", "Sd_n", "E_ts", "Sd_ts")
kable(moments,format='html')

```

y	E_n	Sd_n	E_ts	Sd_ts
110	107.24	6.38	105.29	5.84
125	118.11	6.38	118.09	7.89
140	128.97	6.38	135.41	7.97

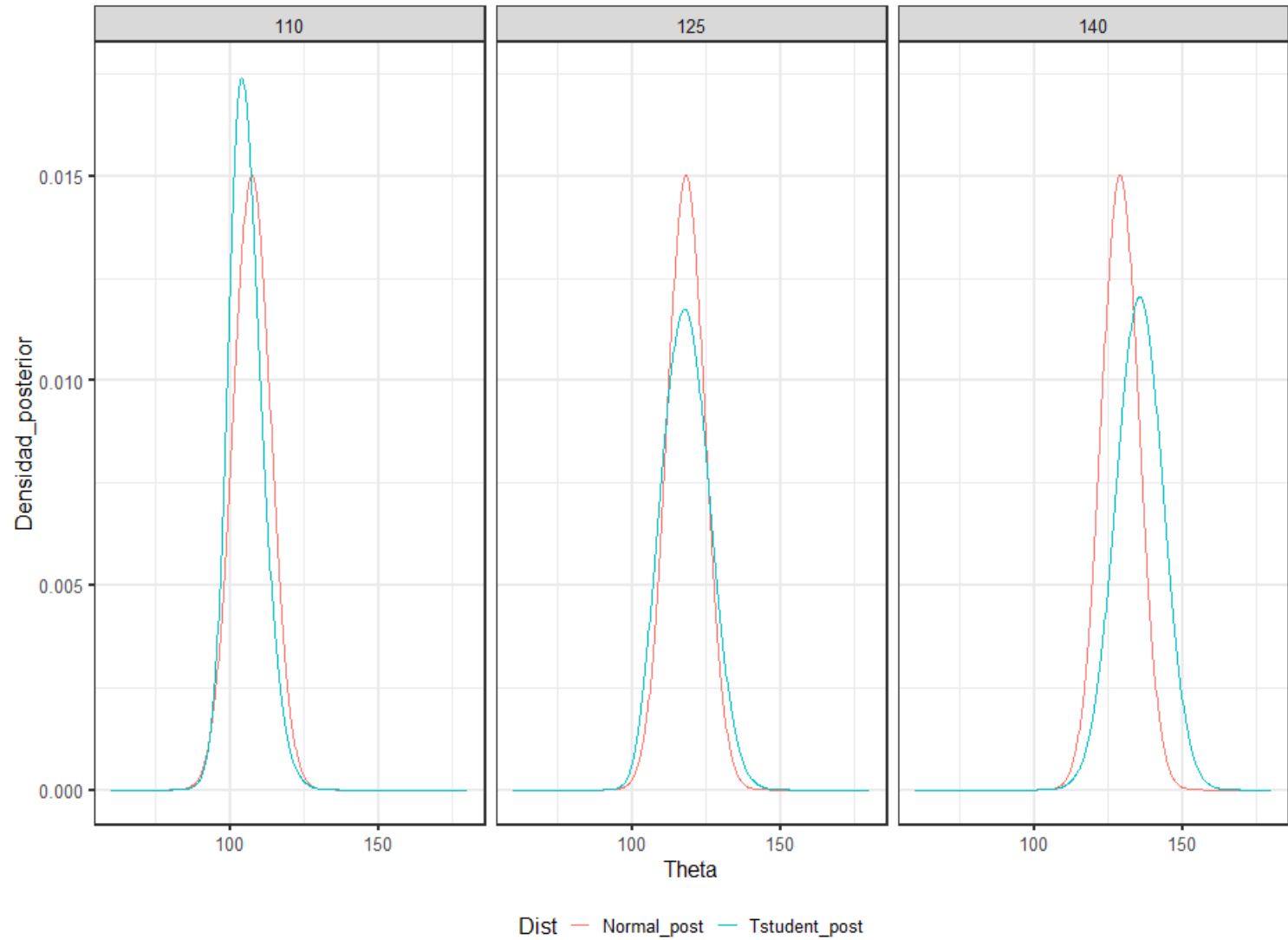
```

df <- data.frame(Theta=rep(seq(60, 180, length = 500),6), Tstudent_post=postts[,2])
df <- gather(df, "Dist", "Densidad_posterior", -Theta,-y)

ggplot(data=df, aes(x=Theta, y=Densidad_posterior, colour=Dist))+
  geom_line()+
  theme_bw(base_size=14)+
  ggtitle("Comparación de posteriores")+
  theme(legend.position = "bottom")+
  facet_wrap(~ y,dir = "h")

```

Comparación de posteriores



Bibliografia:

- *Bayesian computation with R*, Jim Albert (2007)
- *Pricing new Risks or Insurance Products in a Bayesian Framework*, Dhruv D. Haria (2007).

¡Gracias!

Esta presentación fue creada mediante **xaringan**.