Introducción a los Modelos Bayesianos con R

R-Ladies

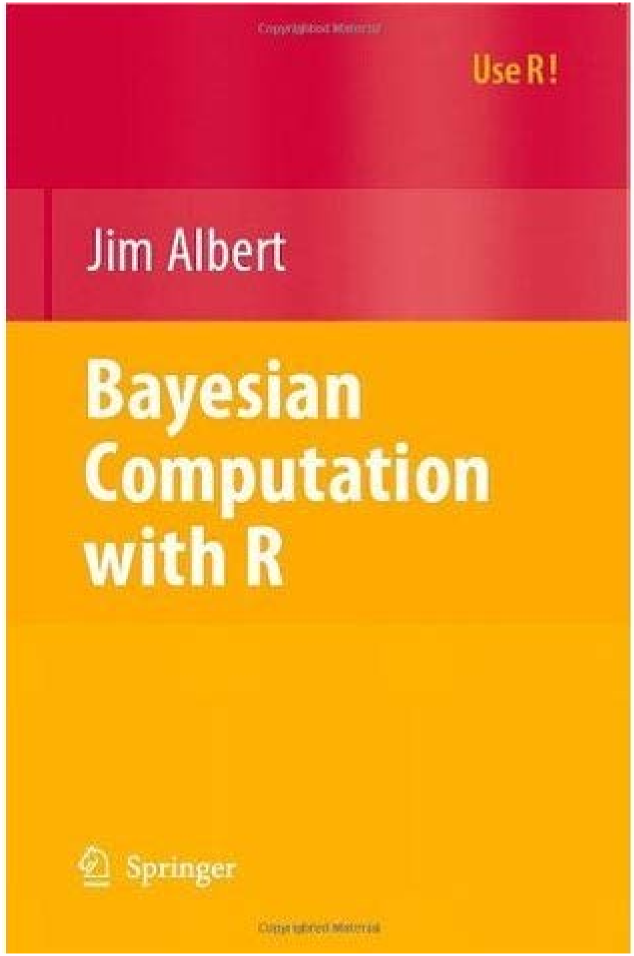
Carolina Montoya Rivera

Actuaria, Instituto Nacional de Seguros

22/05/2019

Stio web: R-LADIES GLOBAL

Rladies xaringan theme


Use R!
Jim Albert
Bayesian Computation with R
Springer

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 X como los parámetros (p) asociados a su distribución como variables aleatorias.

- Supongamos que $g(p)$ contiene la información sobre el parámetro p. Suponga que tenemos un vector de n observaciones $\bar{x} = (x_1, x_2, \cdots, x_n)^T$. La verosimilitud se denota como $f(\bar{x}|p)$ y describe la probabilidad de obtener los valores de $\bar{x}$ dado que p es el parámetro verdadero. Por el teorema de Bayes, la distribución posterior de p es:

$$g(p|\bar{x}) = \frac{f(\bar{x}|p)g(p)}{f(\bar{x})}$$

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

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 $p = 0.3$. Además, tiene una confianza del 90% de que p es menor que 0.5.

- La distribución beta es útil para modelar proporciones puesto que su dominio está en el intervalo (0,1).

$$g(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \implies g(p) \propto p^{\alpha-1}(1-p)^{\beta-1}$$

- ¿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 $Beta(\alpha = 3.26, \beta = 7.18)$.

- 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 s éxitos y f fallos, entonces la verosimilitud es:

$$p^s(1-p)^f$$

- Recordemos que $g(p|\bar{x}) = \dfrac{f(\bar{x}|p)g(p)}{f(\bar{x})}$ y $g(p) \propto p^{\alpha-1}(1-p)^{\beta-1}$ luego:

$$g(p|\bar{x}) \propto p^{\alpha+s-1}(1-p)^{b+f-1}, \text{ donde: } 0 < p < 1$$

- Es decir, $g(p|\bar{x}) \sim Beta(\hat{\alpha} = \alpha + s, \hat{\beta} = \beta + f)$. Este es un ejemplo de un análisis conjugado, donde la distribución previa y posterior del parámetro, tienen la misma forma funcional.

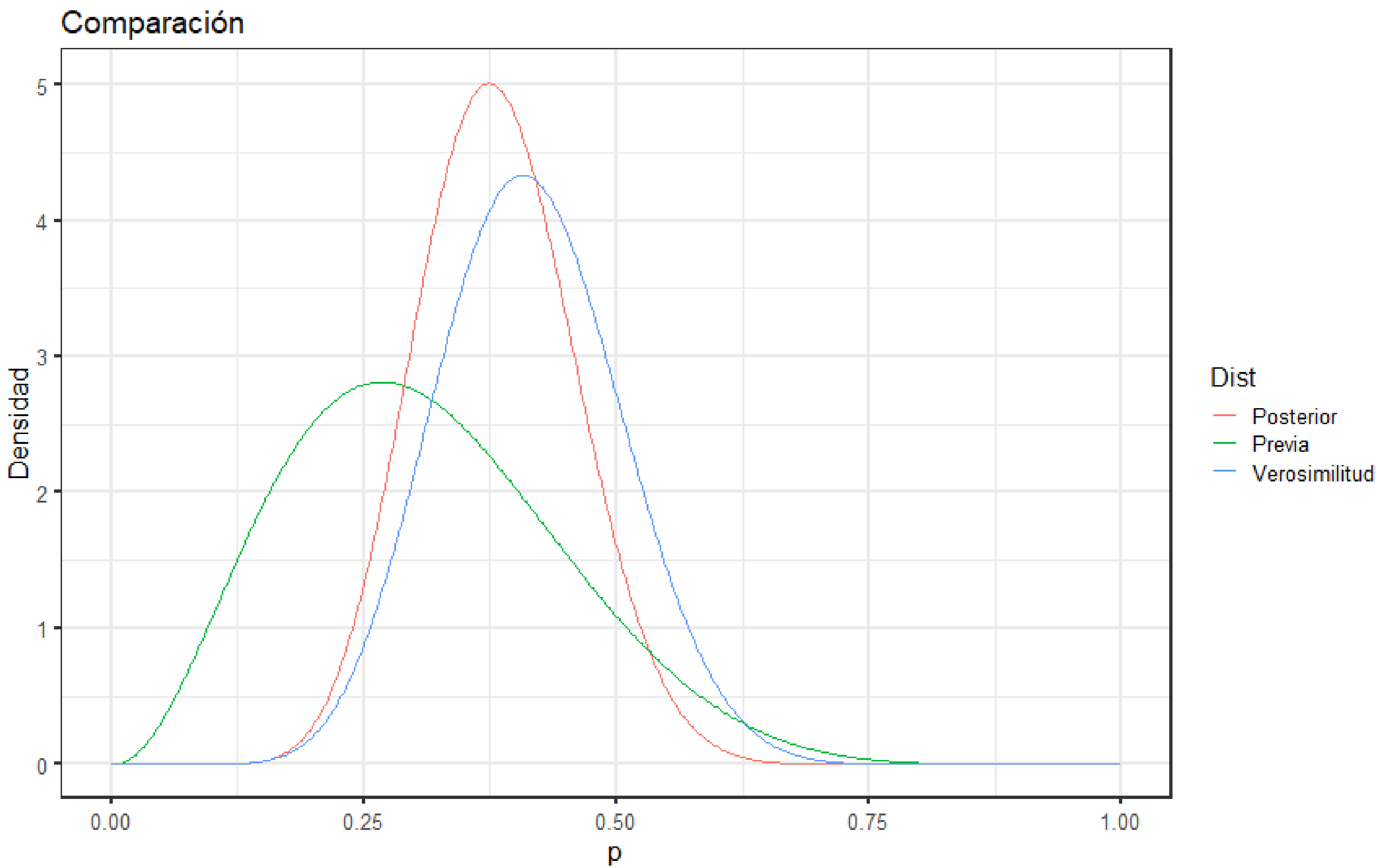
```r
  p <-  seq(0, 1, length = 500)
  a <-  res$r[1]
  b <-  res$r[2]
  s = 11
  f = 16

  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 0.0000000000  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
```

```r
  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
```

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



Inferencia sobre el parámetro

- ¿$P(p >= 0.5)$?

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

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

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

```
## [1] 0.06
```

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

```
## [1] 0.06751
```

- ¿Intervalo de confianza?

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

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

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

```
##        5%       95%
## 0.2482984 0.5075222
```

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

```
##        5%       95%
## 0.2556319 0.5121198
```

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] 0.3779749
```

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

```
## [1] 0.3809634
```

- ¿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] 0.006384494
```

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

```
## [1] 0.006099881
```

Inferencia sobre una función del parámetro

- **Monte Carlo**. Queremos calcular:

$$E(h(p)|x) = \int h(p)g(p|x)dp$$

Suponga que se tiene simulada una muestra $p^1, p^2, \cdots, p^m$. Entonces:

$$\bar{h} = \frac{\sum_{j=1}^{m} h(p^j)}{m}$$

- ¿Media de p^2?

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

```
## [1] 0.1512329
```

El error estándar asociado a este parámetro es:

$$se_{\bar{h}} = \sqrt{\frac{\sum_{j=1}^{m} (h(p^j) - \bar{h})}{m(m-1)}}$$

- ¿Varianza?

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

```
## [1] 0.00192995
```

Predicción

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

$$f(\tilde{x}) = \int f(\tilde{x}|p)g(p)dp$$

Si g es la dist. previa, entonces nos referimos a f como la densidad predictiva previa, mientras que si g es la dist. posterior, f es la densidad predictiva posterior.

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

$$f(\tilde{x}) = \int f_B(\tilde{y}|m,p)g(p)dp = \binom{m}{\tilde{x}} \frac{B(a+\tilde{x}, b+m-\tilde{x})}{B(a,b)}$$

Donde $g \sim Beta(3.26, 7.18)$, $f \sim Binomial(m,p)$ y $\tilde{x} = 0, \cdots, m$.

```r
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

```r
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)
```

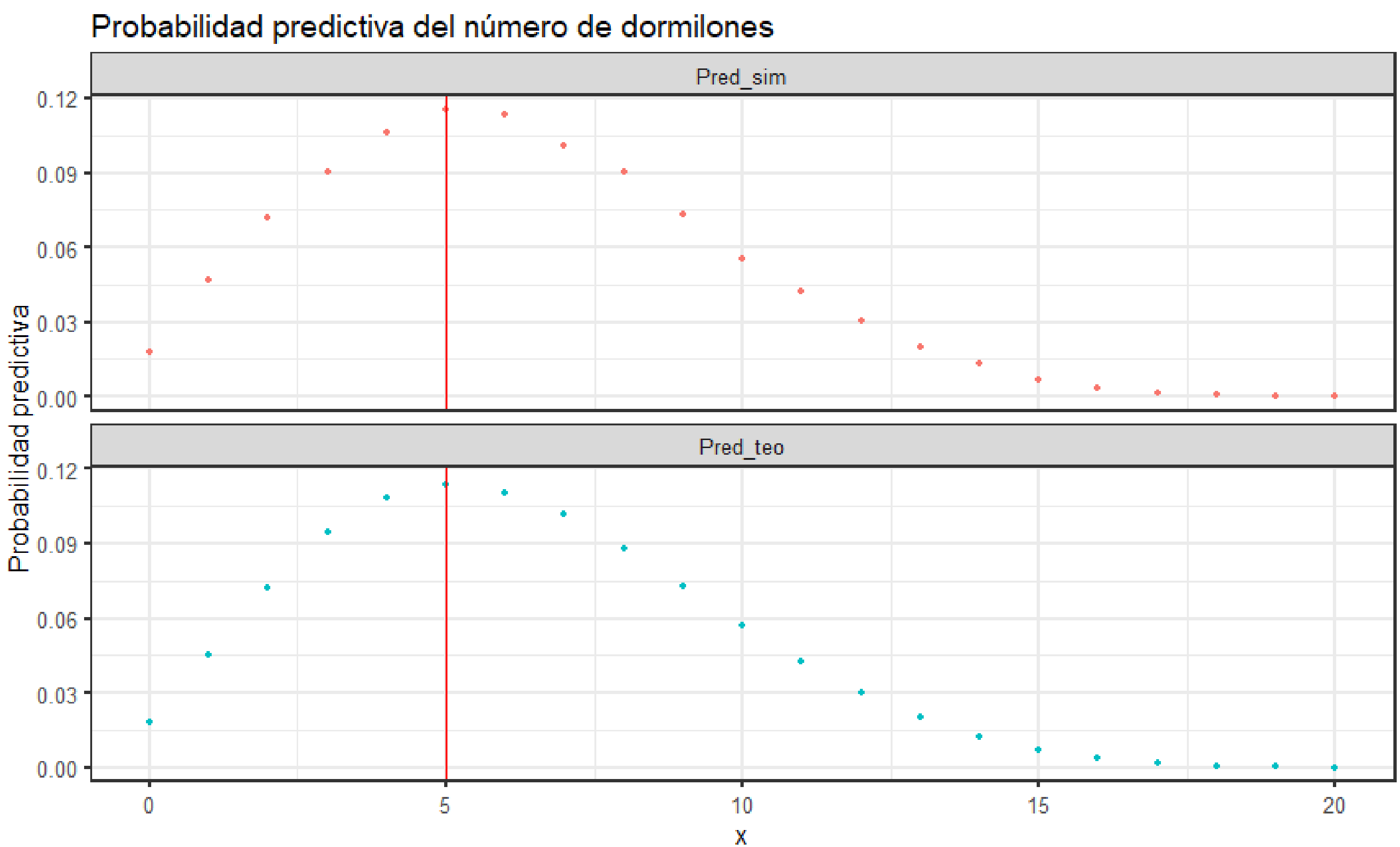
```
## x
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
##  177  468  721  906 1059 1154 1135 1007  904  734  552  420  306  200  134
##   15   16   17   18   19   20
##   65   33   16    5    2    2
```

```
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 0.0468 0.0721 0.0906 0.1059 0.1154 0.1135 0.1007 0.0904 0.0734
##     10     11     12     13     14     15     16     17     18     19
## 0.0552 0.0420 0.0306 0.0200 0.0134 0.0065 0.0033 0.0016 0.0005 0.0002
##     20
## 0.0002
```

```
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 núme
  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 3 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

$$d = favorite - underdog - spread$$

Si se asume que estas diferencias son $N(0, \sigma^2)$, entonces la verosimilitud de d está dada por:

$$(\sigma^2)^{\overline{\frac{-n}{2}}} exp\{-\sum_{i=1}^{n} d_i^2/(2\sigma^2)\}, \qquad \sigma^2 > 0$$

- Suponemos la previa no informativa: $p(\sigma^2) = \frac{1}{\sigma^2}$. Entonces:

$$g(\sigma^2|data) \propto (\sigma^2)^{-n/2-1} exp\{-v/(2\sigma^2)\}$$

Donde $v = \sum_{i=1}^{n} d_i^2$.

- Si definimos $P = 1/\sigma^2$ se puede mostrar que:

$$P \sim \tilde{\chi}^n/v$$

```r
attach(footballscores)

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

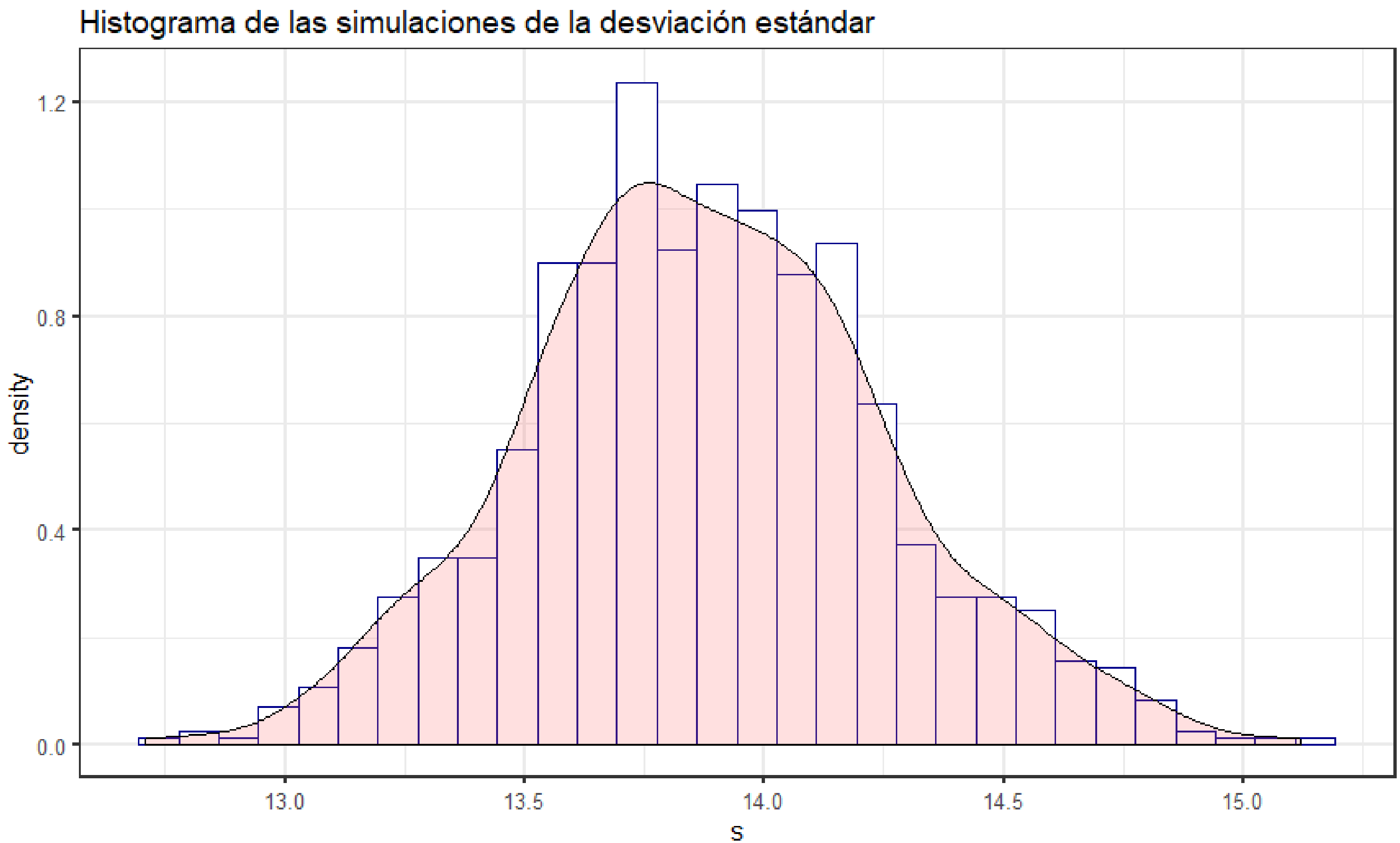
Para hacer inferencia sobre σ usamos la transformación $\sigma = \sqrt{1/P}$:

```r
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%	13.16493
50%	13.86854
97.5%	14.67879

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

```
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 transplantes de corazón.

- n: Cantidad de cirugías de transplante cardíaco en el hospital X.
- y: Número de muertes observadas.
- e: Exposición.

  $$y \sim Poisson(e\lambda)$$

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

- El estimador estándar de MLE para λ es $\hat{\lambda} = y/e$. Problema: cuando y es cercano a 0, no es un buen estimador.
- Bajo este contexto, intentaremos una ajuste bayesiano.

- Una escogencia conveniente para la distribución de la previa es una distribución miembro de la familia $gamma(\alpha, \beta)$ de la forma:

$$p(\lambda) \propto \lambda^{\alpha-1} exp(-\beta\lambda), \qquad \lambda > 0$$

- Se observan z_j muertes, con exposición $o_j, j = 1, \cdots, 10$ (son 10 hospitales). Donde:

$$z_j \sim Poisson(o_j \lambda)$$

Tomando la previa no informativa $p(\lambda) = \lambda^{-1}$, la dist. actualizada de λ dada esta data es:

$$p(\lambda) \propto \lambda^{\sum_{j=1}^{10} z_j - 1} exp\Bigg(- \Big(\sum_{j=1}^{10} o_j \Big) \lambda \Bigg)$$

- Recordemos que para un v.a Z Poisson:

$$f(k, \lambda) = P(Z = k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

- Luego $\lambda \sim Gamma(\alpha = \sum_{j=1}^{10} z_j = 16, \beta = \sum_{j=1}^{10} o_j = 15174)$.

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

$$p(\lambda) \propto \lambda^{\sum_{j=1}^{10} z_j} exp\Bigg(- \Big(\sum_{j=1}^{10} o_j \Big) \lambda \Bigg) \cdot \lambda^{\alpha-1} exp(-\beta\lambda)$$

$$p(\lambda) \propto \lambda^{\sum_{j=1}^{10} z_j + \alpha - 1} exp\Bigg(- \Big(\sum_{j=1}^{10} o_j + \beta \Big) \lambda \Bigg)$$

Es decir, es una $Gamma(\alpha + y_{obs}, \beta + e)$

- También podemos calcular la distribución predictiva de y:

$$f(y) = \frac{f(y|\lambda)g(\lambda)}{g(\lambda|y)}$$

```r
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

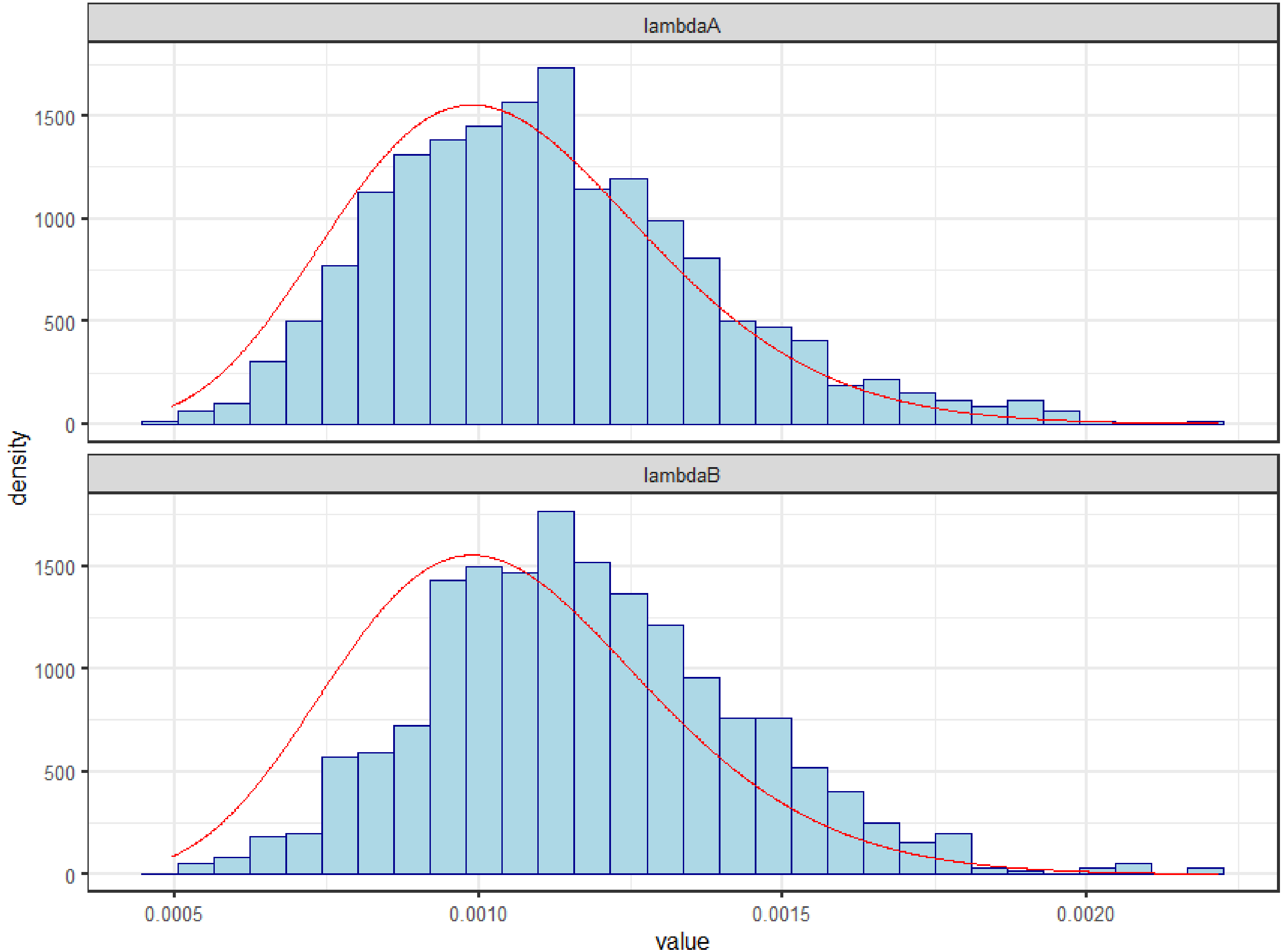
```r
lambdaA <- rgamma(1000, shape = alpha + yobs, rate = beta + ex)
```

```
ex <- 1767; yobs<-4
y <- 0:10
py <- dpois(y, lam * ex) * dgamma(lam, shape = alpha,rate = beta)/dgamma(lam, shap
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)
```

```r
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()
```


lambdaA
lambdaB
density
value
0
500
1000
1500
0.0005
0.0010
0.0015
0.0020

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 inteligecia 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 $\theta \sim N(100, 12.16)$

- Joe toma 4 tests. Sus notas son y_1, y_2, y_3, y_4. Asumiendo que $y \sim N(\theta, \sigma)$ con sd conocida $\sigma = 15$, la nota promedio $\bar{y} \sim N(\theta, \sigma/\sqrt{4})$.

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

$$\tau_1 = 1/\sqrt{4/\sigma^2 + 1/\tau^2} \qquad \mu_1 = \frac{\bar{y}(4/\sigma^2) + \mu(1/\tau^2)}{4/\sigma^2 + 1/\tau^2}$$

Vamos a calcular la distribución posterior bajo 3 escenarios de nota promedio observada: $\bar{y} = 110, \bar{y} = 125, \bar{y} = 140$.

```
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

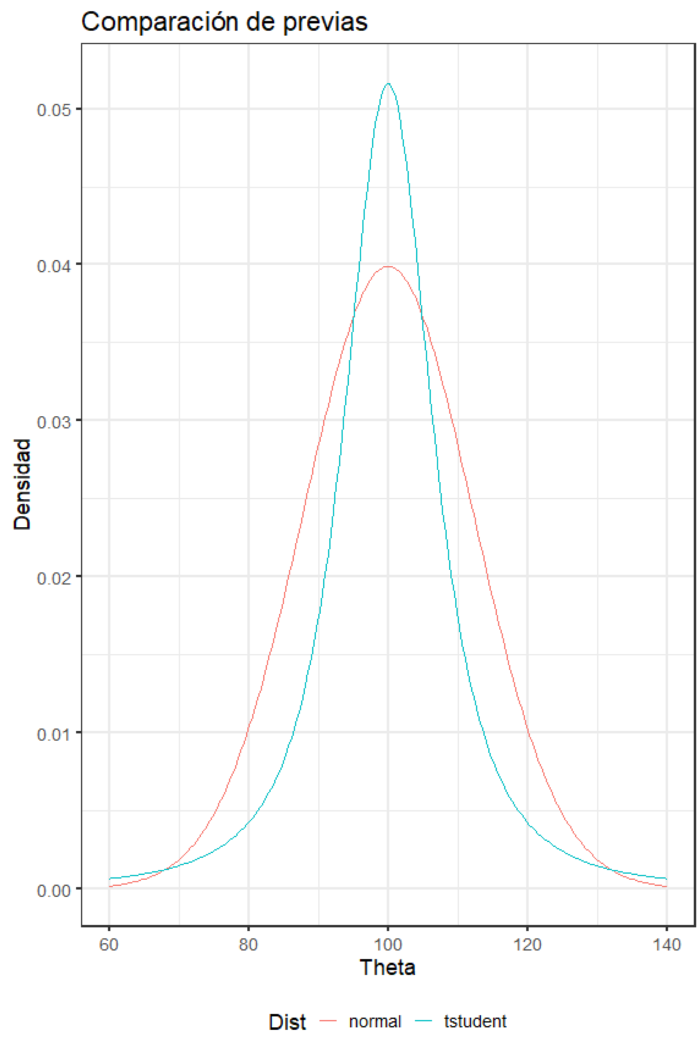
- Consideremos ahora otra previa. Una t-student con parámetro de localización y escala (μ, τ)

$$p(\theta|v,\mu,\tau) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\Gamma\left(\frac{v}{2}\right)\cdot\sqrt{\pi v \tau}}\left(1+\frac{1}{v}\left(\frac{\theta-\mu}{\tau}\right)^2\right)^{-(v+1)/2}$$

```
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

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

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

df <- data.frame(Theta=seq(60, 140, le
                 tstudent=1/tscale*dt(
                 normal=1/10*dnorm((th
df <- gather(df,key = "Dist",
             value = "Densidad",
             -Theta)

ggplot(data=df, aes(x=Theta, y=Densida
  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:

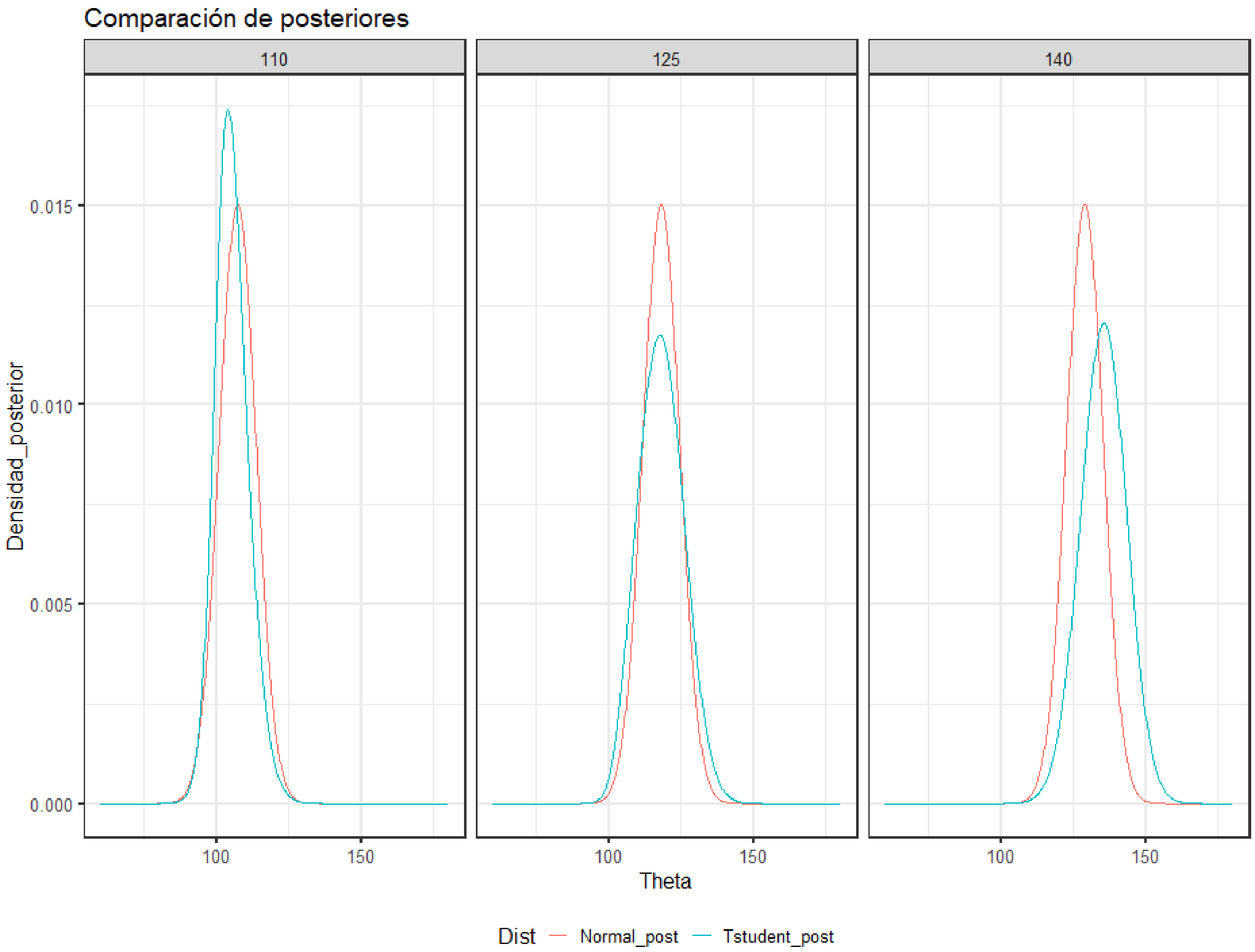
$$g(\theta|data) \propto \phi(\bar{y}|\theta, \sigma/\sqrt{n}) g_T(\theta|v, \nu, \tau)$$

```
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
110
125
140
0.015
0.010
0.005
0.000
Densidad_posterior
100
150
Theta
Dist
Normal_post
Tstudent_post

¡Gracias!

Esta presentación fue creada mediante **xaringan**.