

# Parcial II

Alejandro Salazar Mejía

23/8/2021

```
library('MCMCpack')
```

```
## Warning: package 'MCMCpack' was built under R version 4.0.5
```

```
## Loading required package: coda
```

```
## Warning: package 'coda' was built under R version 4.0.5
```

```
## Loading required package: MASS
```

```
## ##
```

```
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2021 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
```

```
## ## Support provided by the U.S. National Science Foundation
```

```
## ## (Grants SES-0350646 and SES-0350613)
```

```
## ##
```

```
library(hdrcde)
```

```
## Warning: package 'hdrcde' was built under R version 4.0.5
```

```
## This is hdrcde 3.4
```

```
library(KernSmooth)
```

```
## Warning: package 'KernSmooth' was built under R version 4.0.5
```

```
## KernSmooth 2.23 loaded
```

```
## Copyright M. P. Wand 1997-2009
```

```
library(knitr)
```

```
library(readr)
```

```
library(plotrix)
```

```
## Warning: package 'plotrix' was built under R version 4.0.3
```

```
library(HH)
```

```
## Warning: package 'HH' was built under R version 4.0.4

## Loading required package: lattice

## Loading required package: grid

## Loading required package: latticeExtra

## Warning: package 'latticeExtra' was built under R version 4.0.2

## Loading required package: multcomp

## Warning: package 'multcomp' was built under R version 4.0.4

## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 4.0.3

## Loading required package: survival

## Loading required package: TH.data

## Warning: package 'TH.data' was built under R version 4.0.4

##
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
##
##     geyser

## Loading required package: gridExtra

## Warning: package 'gridExtra' was built under R version 4.0.2

##
## Attaching package: 'HH'

## The following object is masked from 'package:coda':
##
##     acfplot

source('funcionesAux.R')
```

Como se explicó en la sección 3.1, la metodología de elicitación consiste básicamente en elegir al menos dos puntos de diseño, y elicitar la distribución normal de la respuesta en dichos puntos. En esta sección se muestra dicho proceso.

Se escogieron tres puntos de diseño: los modelos 2013, 2015 y 2017. Una vez obtenidos los intervalos en cada año y la frecuencia de la muestra hipotéticas en estos, se hace uso de la función `estimaNormal`, programada por el profesor Juan Carlos Correa Morales y presentada en el curso de estadística bayesiana en las diapositivas de la clase 10. Esta función nos permite encontrar una distribución normal que mejor ajuste sus probabilidades a las frecuencias relativas correspondientes a cada subintervalo por medio de un proceso de optimización.

A continuación se muestra de manera gráfica el resultado de la elicitación del precio del auto en cada año-modelo, junto con la distribución normal ajustada y su tabla de parámetros estimados.

**2013:**

```
limites2013 <- c(25, 26, 27, 28, 29, 30)
frecus2013 <- c(20, 40, 80, 100, 100)
```

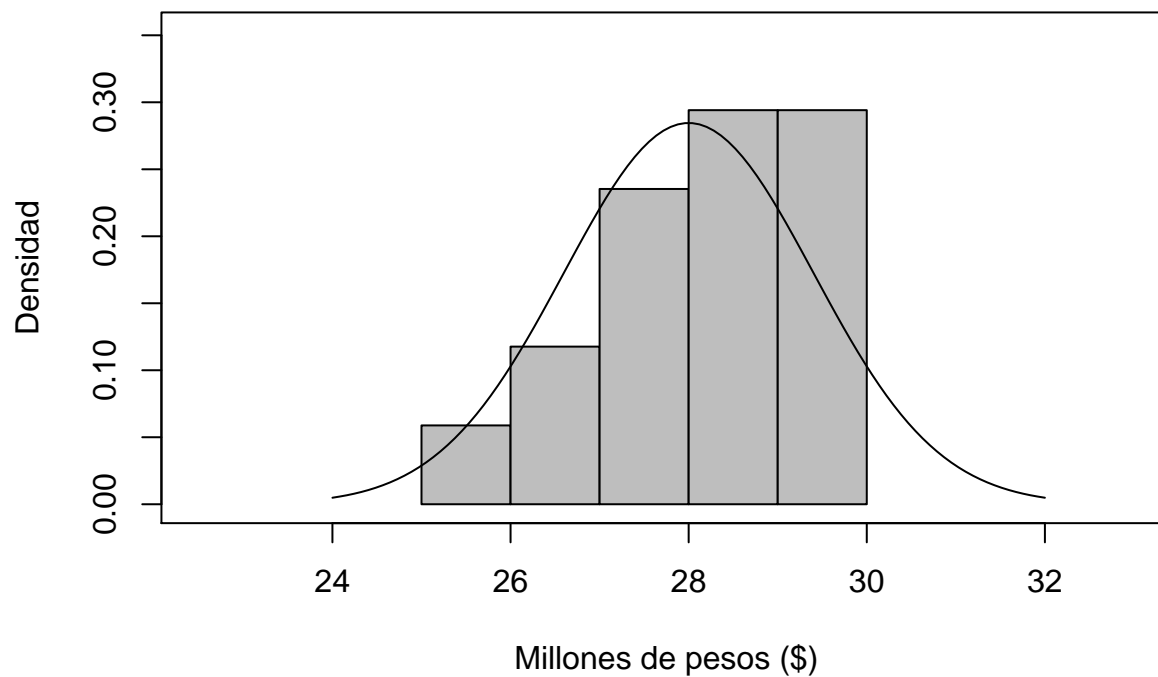
```
(resu2013 <- optim(c(26,5),estimaNormal,method='L-BFGS-B',
  lower=c(25,0.01),upper=c(28,50),
  limites=limites2013 ,frecu=frecus2013))
```

```
## $par
## [1] 28.000000 1.965467
##
## $value
## [1] 0.0213493
##
## $counts
## function gradient
##      16      16
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```
param.opt2013<-resu2013$par
```

```
histograma(frecus2013,limites2013,
  main = "Distribución del precio \nen Millones de pesos del modelo 2013",
  xlab = "Millones de pesos ($)",
  ylab = "Densidad")
points(xx<-seq(min(limites2013)-1,max(limites2013)+2,length=100),
  dnorm(xx,mean=param.opt2013[1],sd=sqrt(param.opt2013[2])),type='l')
```

## Distribución del precio en Millones de pesos del modelo 2013



```
kable(t(param.opt2013),
      col.names = c("$\\mu_{2013}$", "$\\sigma^2_{2013}$"),
      caption = "Parámetros de la distribución en el año-modelo 2013")
```

Table 1: Parámetros de la distribución en el año-modelo 2013

$\mu_{2013}$	$\sigma^2_{2013}$
28	1.965467

**2015:**

```
limites2015 <- c(30, 31, 32, 33)
frecus2015 <- c(20, 50, 50)
```

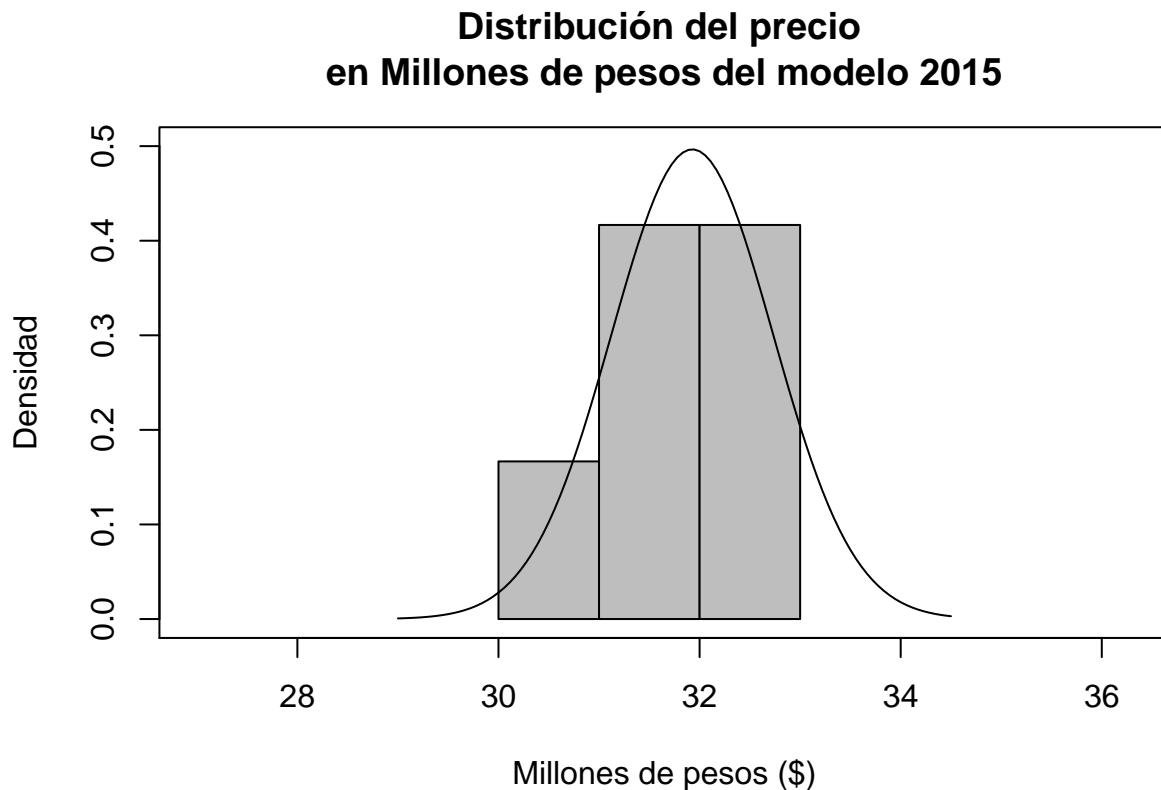
```
(resu2015 <- optim(c(30, 5), estimaNormal, method='L-BFGS-B',
                  lower=c(29, 0.02), upper=c(32, 50),
                  limites=limites2015, frecu=frecus2015))
```

```
## $par
## [1] 31.9274190 0.6453334
##
## $value
```

```
## [1] 0.004495004
##
## $counts
## function gradient
##      29      29
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```
param.opt2015<-resu2015$par
```

```
histograma(frecus2015,limites2015,
            main = "Distribución del precio \nen Millones de pesos del modelo 2015",
            xlab = "Millones de pesos ($)",
            ylab = "Densidad")
points(xx<-seq(min(limites2015)-1,max(limites2015)+1.5,length=100),
       dnorm(xx,mean=param.opt2015[1],sd=sqrt(param.opt2015[2])),type='l')
```



```
kable(t(param.opt2015),
      col.names = c("$\\mu_{2015}$", "$\\sigma^2_{2015}$"),
      caption = "Parámetros de la distribución en el año-modelo 2015")
```

Table 2: Parámetros de la distribución en el año-modelo 2015

$\mu_{2015}$	$\sigma_{2015}^2$
31.92742	0.6453334

**2017:**

```
limites2017 <- c(35, 36, 37, 38)
frecus2017 <- c(20, 30, 50)
```

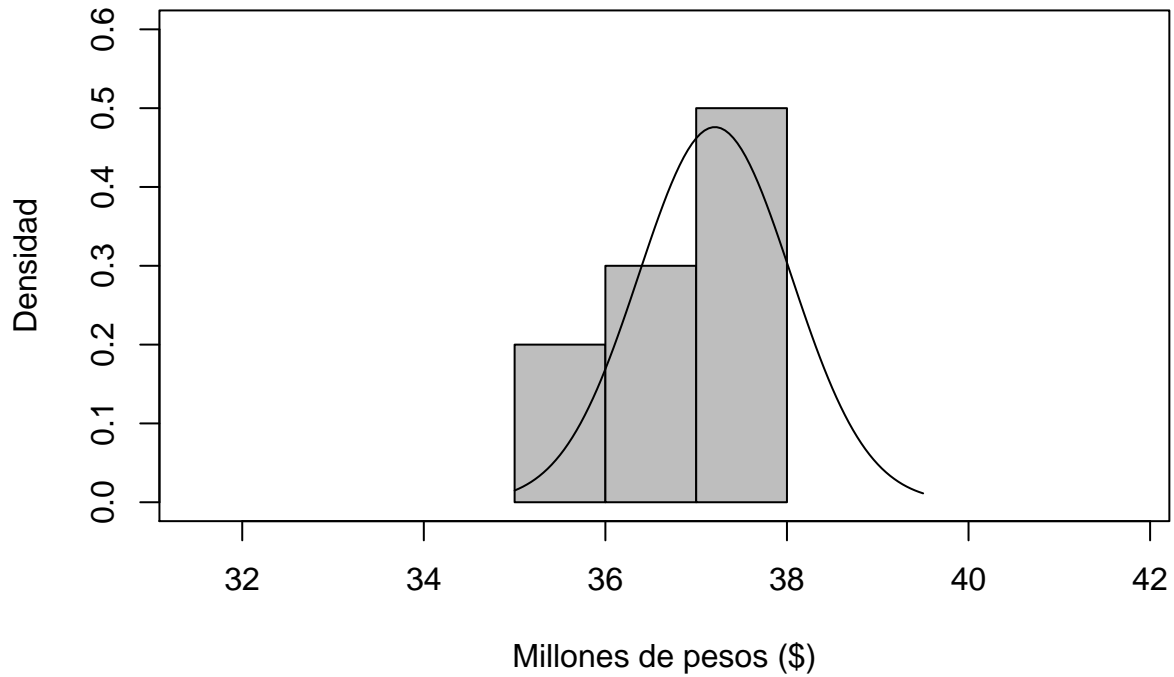
```
(resu2017 <- optim(c(36,5),estimaNormal,method='L-BFGS-B',
  lower=c(35,0.01),upper=c(38,50),
  limites=limites2017 ,frecu=frecus2017))
```

```
## $par
## [1] 37.2062508 0.7023848
##
## $value
## [1] 0.02302974
##
## $counts
## function gradient
##      22      22
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```
param.opt2017<-resu2017$par
```

```
histograma(frecus2017,limites2017,
  main = "Distribución del precio \nen Millones de pesos del modelo 2017",
  xlab = "Millones de pesos ($)",
  ylab = "Densidad")
points(xx<-seq(min(limites2017),max(limites2017)+1.5,length=100),
  dnorm(xx,mean=param.opt2017[1],sd=sqrt(param.opt2017[2])),type='l')
```

## Distribución del precio en Millones de pesos del modelo 2017



```
kable(t(param.opt2017),
      col.names = c("$\\mu_{2017}$", "$\\sigma^2_{2017}$"),
      caption = "Parámetros de la distribución en el año-modelo 2017")
```

Table 3: Parámetros de la distribución en el año-modelo 2017

$\mu_{2017}$	$\sigma^2_{2017}$
37.20625	0.7023848

Una vez obtenidas las tres normales, se simularon 20 datos de cada una (este número busca reflejar el conocimiento que tiene el experto sobre el tema el cual es alto). A dicha muestra se le ajustó un modelo de regresión lineal simple y se guardaron los valores de los tres parámetros de interés. Este proceso se repitió 1000 veces, lo que resulta en una muestra simulada de tamaño 1000 de  $\beta_0$ ,  $\beta_1$  y  $\sigma^2$ .

### Apriori:

```
betas0 <- c()
betas1 <- c()
sigmas2 <- c()

x <- rep(c(13,15,17), each= 20)
```

```

for (i in 1:1000) {
  y2013 <- rnorm(20, mean = param.opt2013[1], sd = param.opt2013[2])
  y2015 <- rnorm(20, mean = param.opt2015[1], sd = param.opt2015[2])
  y2017 <- rnorm(20, mean = param.opt2017[1], sd = param.opt2017[2])
  y <- c(y2013, y2015, y2017)
  modelo <- lm(y~x)

  betas0 <- c(betas0, modelo$coefficients[1])
  betas1 <- c(betas1, modelo$coefficients[2])
  sigmas2 <- c(sigmas2, (summary(modelo)$sigma)**2)
}

taus <- 1/sigmas2

```

```

# Estimación de parás. de normal bivariada de Beta0 y Beta1
betas <- data.frame(betas0, betas1)
b0 <- colMeans(betas)
B0 <- solve(var(betas))

# Estimación de parás. de Gamma inversa por MM
m <- mean(sigmas2)
v <- var(sigmas2)
alpha <- m^2/v+2
beta <- m*(m^2/v+1)

```

Se busca contruir una distribución apriori con la siguiente p.d.f:

$$\xi(\beta_0, \beta_1, \sigma^2) = \mathcal{N}(\beta_0, \beta_1) \times \mathcal{GI}(\sigma^2)$$

Para esto, vamos a aproximar la distribución apriori del experto al modelo teórico anterior. Se estima entonces el vector de medias y la matriz de var-cov de  $\mathcal{N}(\beta_0, \beta_1)$ , y los parámetros de escala y forma para  $\mathcal{GI}(\sigma^2)$  usando el método de los momentos.

A continuación se muestra de manera gráfica las distribuciones de  $\beta_0, \beta_1$ , y  $\sigma^2$ , y su tabla de parámetros estimados necesarios para ajustar el modelo teórico.

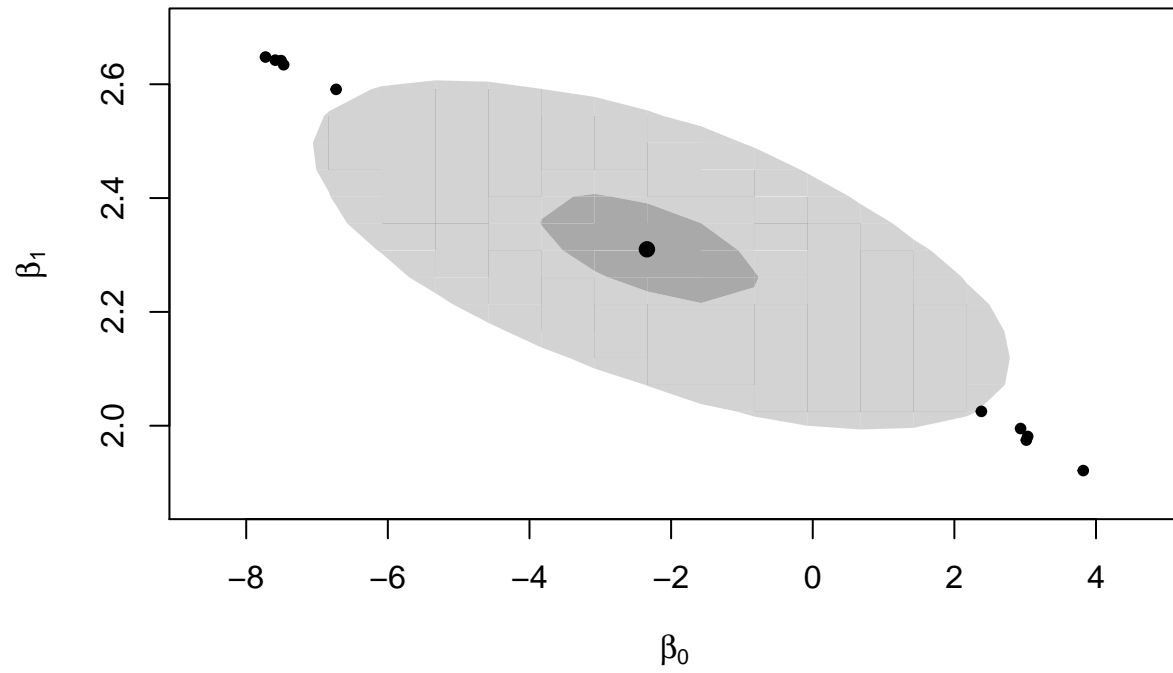
```

# Distribución Bivariada de beta0 y beta1
hdr.boxplot.2d(betas0, betas1,
  main = expression("Distribución conjunta de " ~beta[0] ~y~beta[1]),
  xlab = expression(beta[0]),
  ylab = expression(beta[1]))

```

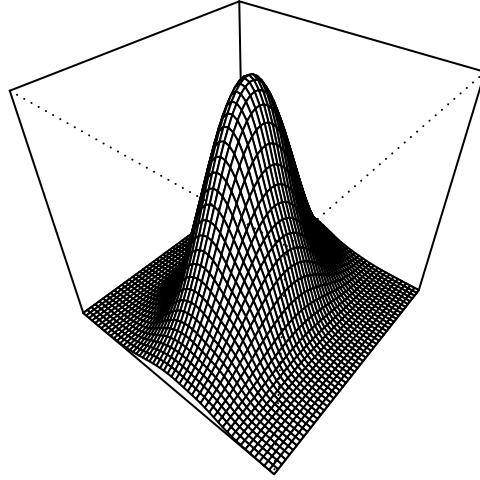


Distribución conjunta de  $\beta_0$  y  $\beta_1$

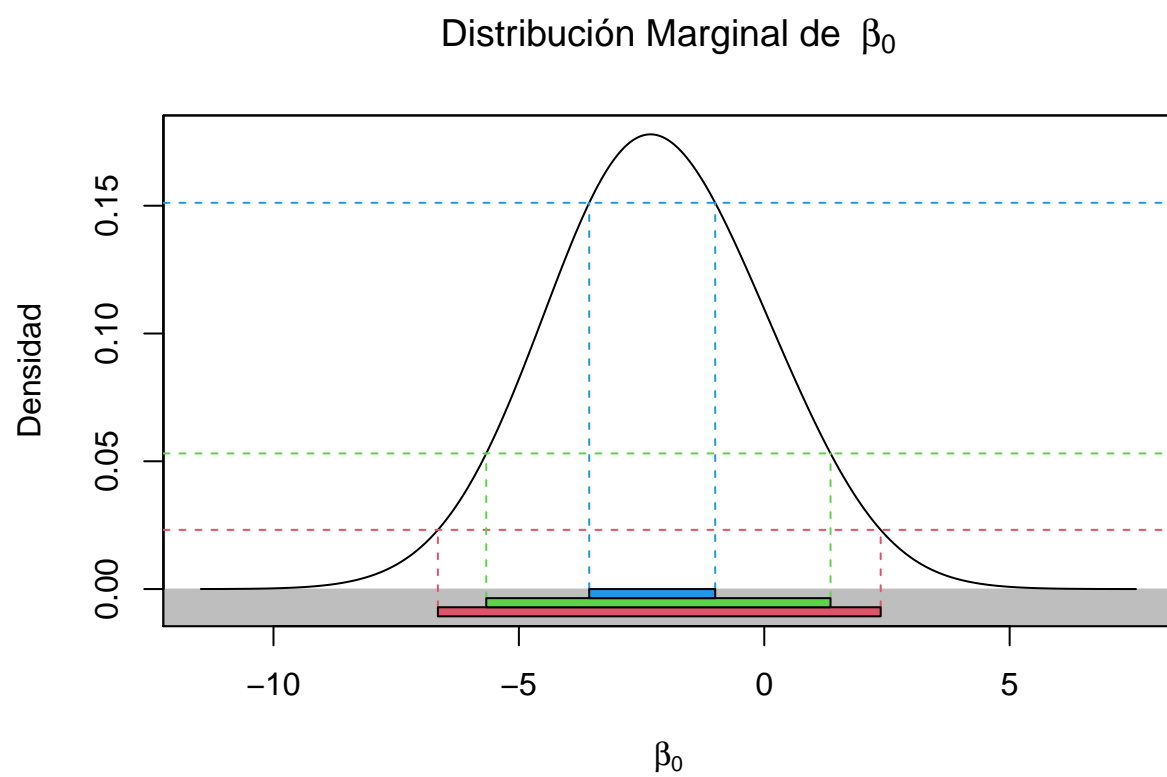


```
est <- bkde2D(betas, bandwidth = c(0.5,0.5))
persp(est$fhat,
      phi = 40, theta = 40, axes = F,
      main = expression("Distribución conjunta de " ~ beta[0] ~ y ~ beta[1]))
```

## Distribución conjunta de $\beta_0$ y $\beta_1$

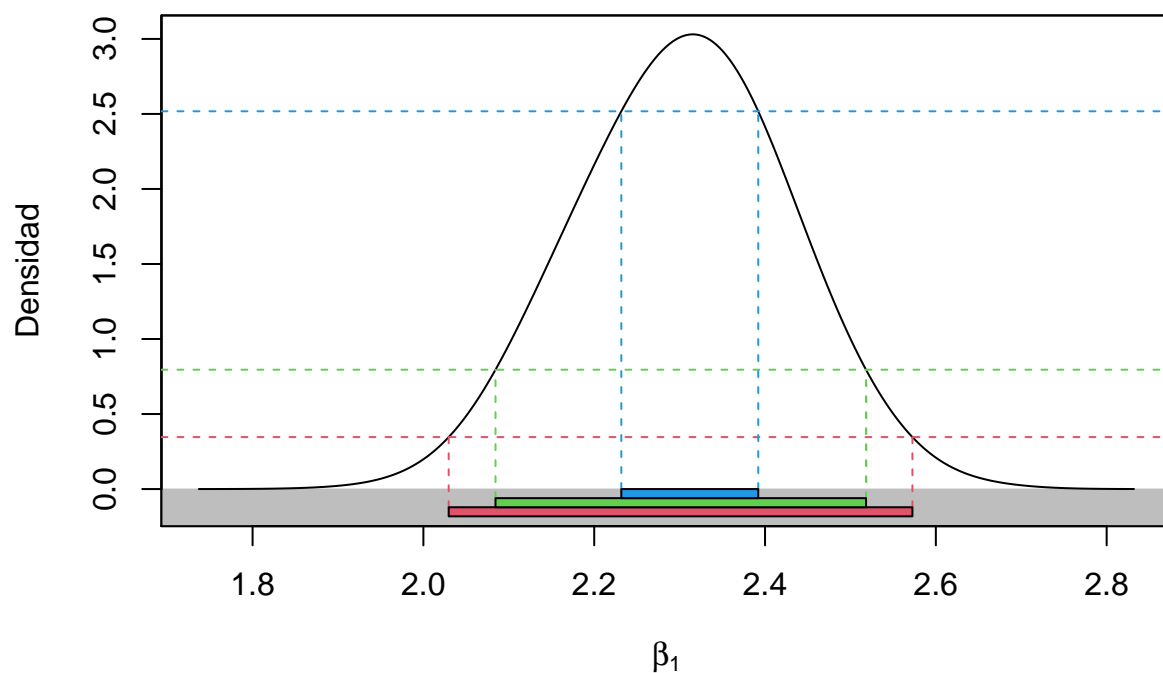


```
hdrBeta0 <- hdr.den(betas0, xlab = expression(beta[0]),  
  ylab = "Densidad",  
  main = expression("Distribución Marginal de " ~ beta[0]))
```



```
hdrBeta1 <- hdr.den(betas1, xlab = expression(beta[1]),  
  ylab = "Densidad",  
  main = expression("Distribución Marginal de " ~ beta[1]))
```

## Distribución Marginal de $\beta_1$



```
kable(t(b0),
      col.names = c("$\\mu_{\\beta_0}$", "$\\mu_{\\beta_1}$"),
      caption = "Vector de Medias")
```

Table 4: Vector de Medias

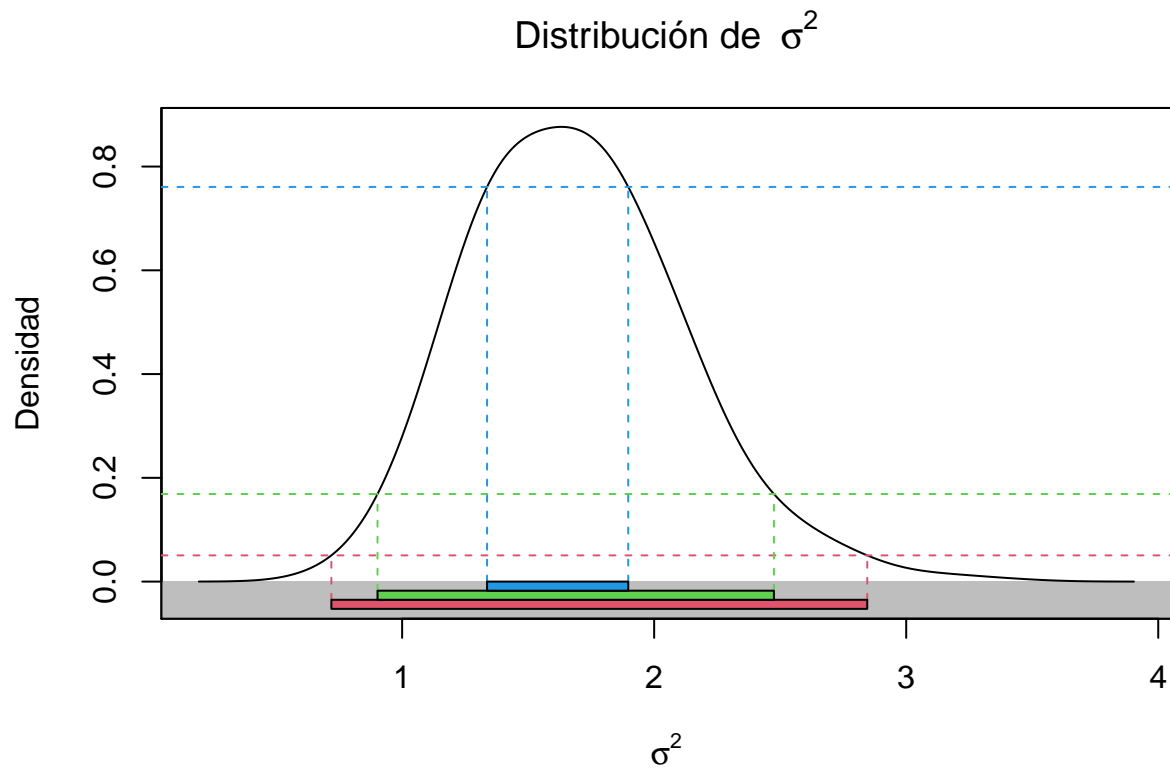
$\mu_{\beta_0}$	$\mu_{\beta_1}$
-2.180951	2.303826

```
rownames(B0) <- c("$\\beta_0$", "$\\beta_1$")
kable(B0,
      col.names = c("$\\beta_0$", "$\\beta_1$"),
      caption = "Matriz de var-cov")
```

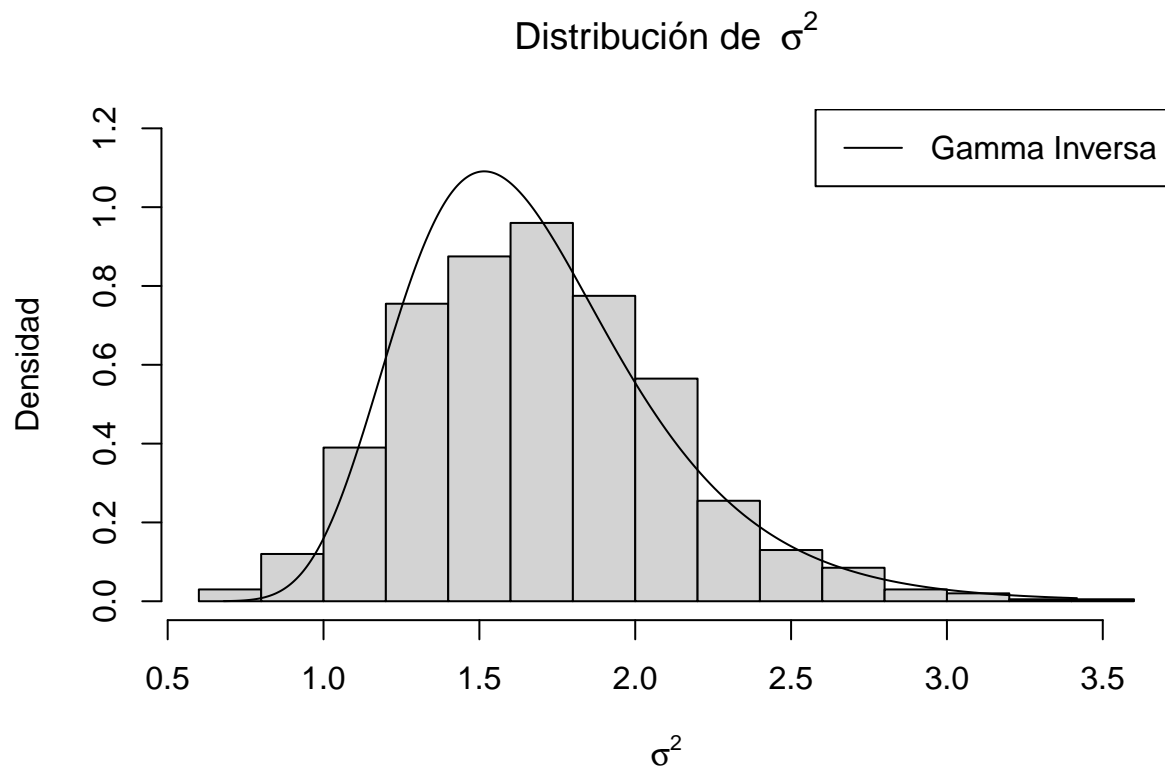
Table 5: Matriz de var-cov

	$\beta_0$	$\beta_1$
$\beta_0$	83.83875	1341.963
$\beta_1$	1341.96265	21556.291

```
hdrSigma2 <- hdr.den(sigmatas2, xlab = expression(sigma^2),
  ylab = "Densidad",
  main = expression("Distribución de "~sigma^2))
```



```
xxx <- seq(from = min(sigmatas2), to = max(sigmatas2), length.out = 1000)
hist(sigmatas2, freq = F, ylim = c(0, 1.2),
  xlab = expression(sigma^2),
  ylab = "Densidad",
  main = expression("Distribución de "~sigma^2))
lines(xxx, invGamma(alpha, beta, xxx))
legend("topright", legend="Gamma Inversa", col=1, lwd=1)
```



```
kable(t(c(alpha, beta)),
      col.names = c("Forma", "Escala"),
      caption = "Parámetros de G.I")
```

Table 6: Parámetros de G.I

Forma	Escala
18.27191	29.20815

### Datos Muestrales:

Obtenidos de: <https://carros.tucarro.com.co/>  
<https://carros.mercadolibre.com.co/>  
<https://www.olx.com.co/>

En particular se buscaron autos cuyo modelo concordaran con los puntos de diseño escogidos, de forma que, más adelante, podamos comparar la información del experto con la muestral.

```
DatosCarros <- read_csv("DatosCarros.csv", col_names = c("modeloAuto", "precio"))
```

```
## Parsed with column specification:
## cols(
##   modeloAuto = col_double(),
```

```
## precio = col_double()
## )
```

```
DatosCarros$modeloAuto <- as.numeric(sub("20","",as.character(DatosCarros$modeloAuto)))
attach(DatosCarros)
```

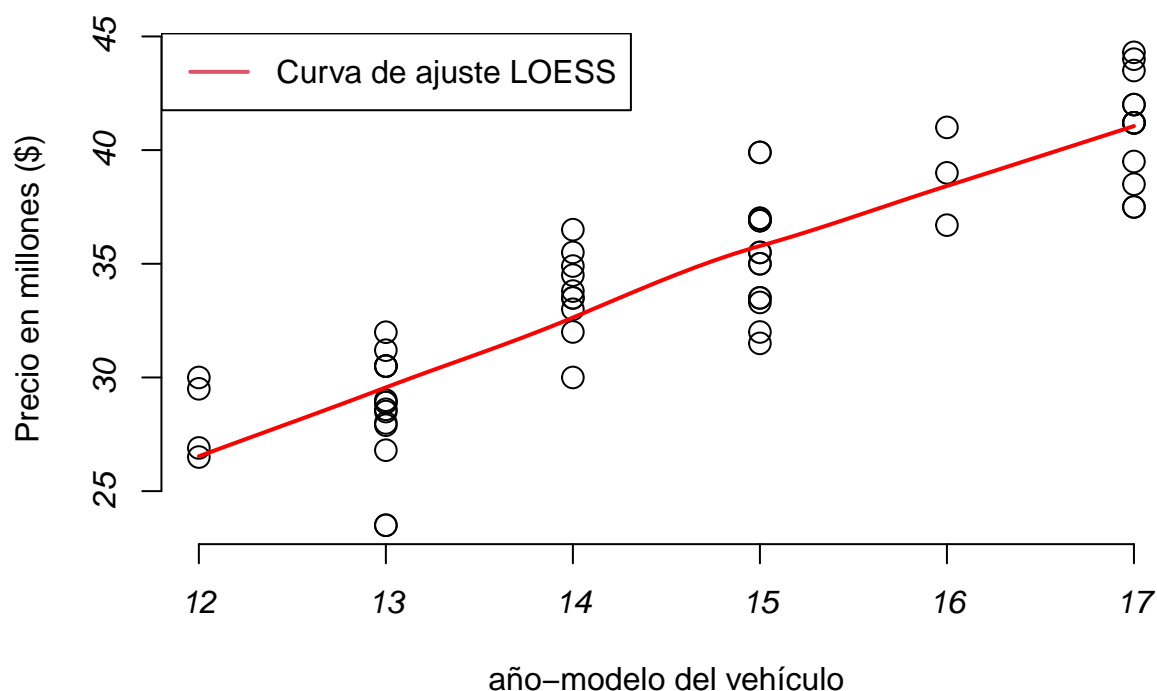
```
# Table <- data.frame(round(matrix(c(summary(precio),sd(precio)),ncol=7),2))
# names(Table)=c(names(summary(precio)),"sd")
# win.graph(width = 10)
# layout(rbind(c(1,1),c(2,3)),heights=c(1,3))
# plot.new()
# addtable2plot(x="top",table=Table, xpad=1, ypad=1, bty='o',
#               display.rownames = F, hlines = TRUE, vlines = TRUE,
#               title = "Estadísticos de resumen")
#
# den <- density(precio)
# hist(precio,breaks="FD",freq=F,xlim=c(min(den$x),max(den$x)),
#      main = paste("Histograma del precio en Millones de pesos"),
#      xlab="Millones de pesos ($)",col="white")
# lines(den,col=2,lwd=2)
# boxplot(precio,boxwex=0.4,xlab="Precio en Millones ($)",
#         col="white")
```

```
kable(table(modeloAuto))
```

modeloAuto	Freq
12	4
13	16
14	10
15	16
16	3
17	13

```
plot(modeloAuto, precio,
     main="Gráfico de Dispersión",
     xlab="año-modelo del vehículo",ylab="Precio en millones ($)",cex=1.5,bty="n",
     font=3,font.main=4)
lines(loess.smooth(modeloAuto, precio,family="gaussian",span=0.8),
      lty=1,lwd=2,col="red")
legend("topleft",legend="Curva de ajuste LOESS",col=2,lwd=2)
```

## Gráfico de Dispersión



```
modeloVero <- lm(precio~modeloAuto)
anovaModeloVero <- data.frame(anova(modeloVero))
anovaModeloVero[2,c(4,5)] <- c("", "")
kable(anovaModeloVero)
```

	Df	Sum.Sq	Mean.Sq	F.value	Pr..F.
modeloAuto	1	1299.4866	1299.486589	236.85028740771	1.68555407073729e-22
Residuals	60	329.1919	5.486532		

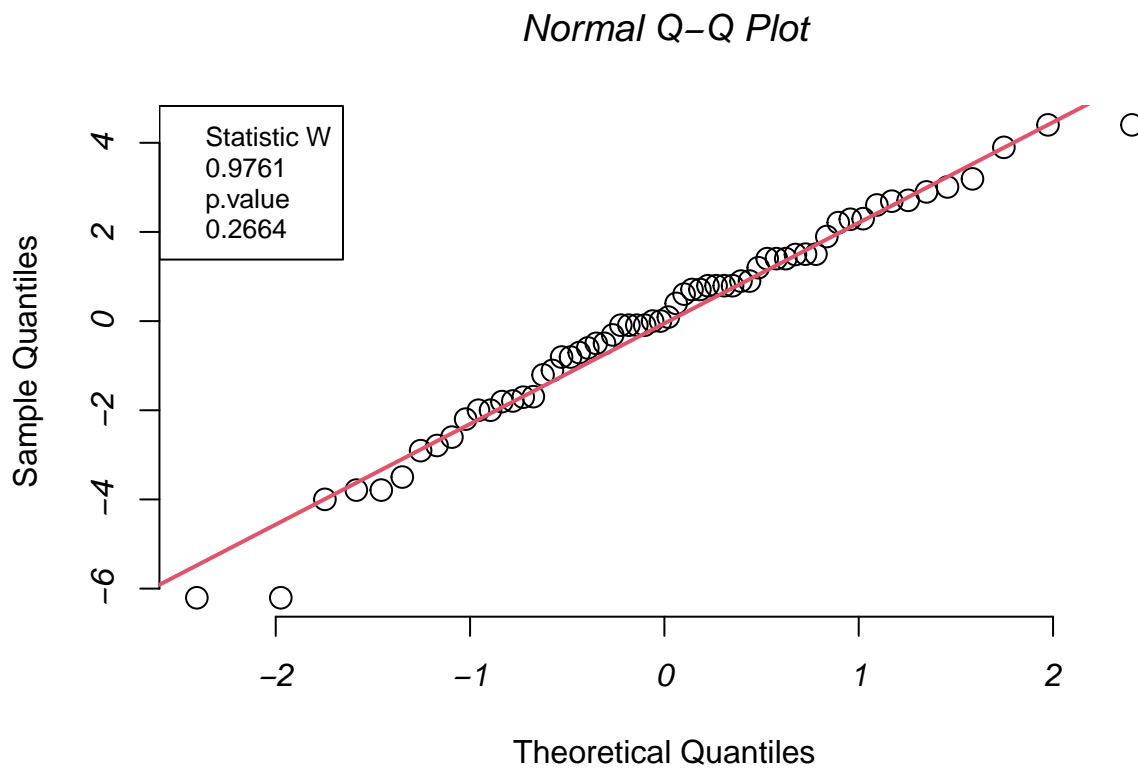
```
estimaciones <- rbind(summary(modeloVero)$coefficients,
  sigma2=summary(modeloVero)$sigma**2)
rownames(estimaciones) <- c("$\\beta_0$", "$\\beta_1$", "$\\sigma^2$")
kable(estimaciones)
```

	Estimate	Std. Error	t value	Pr(> t )
$\beta_0$	-7.945133	2.7630245	-2.875520	0.0055751
$\beta_1$	2.896230	0.1881898	15.389941	0.0000000
$\sigma^2$	5.486532	5.4865316	5.486532	5.4865316

```
test <- shapiro.test(residuals(modeloVero))
qqnorm(residuals(modeloVero), cex=1.5, bty="n", font=3, font.main=3)
```

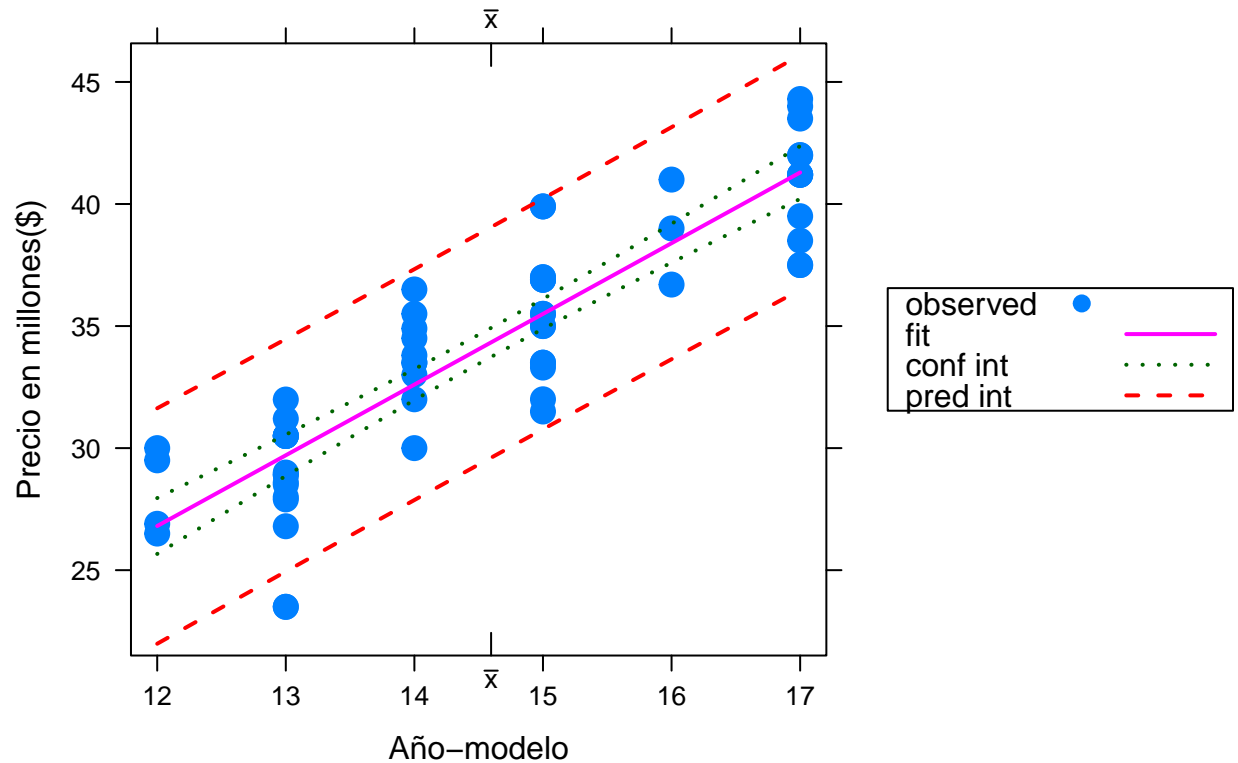


```
qqline(residuals(modeloVero),lty=1,lwd=2,col=2)
legend("topleft",legend=rbind(c("Statistic W","p.value"),
                                round(c(test$Statistic,test$p.value),digits=4)),cex=0.8)
```



```
# win.graph(width=8,height=6)
ci.plot(modeloVero,xlab="Año-modelo",ylab="Precio en millones($)",
        lty=c(2,1,3), cex=1.5,conf.level=.95)
```

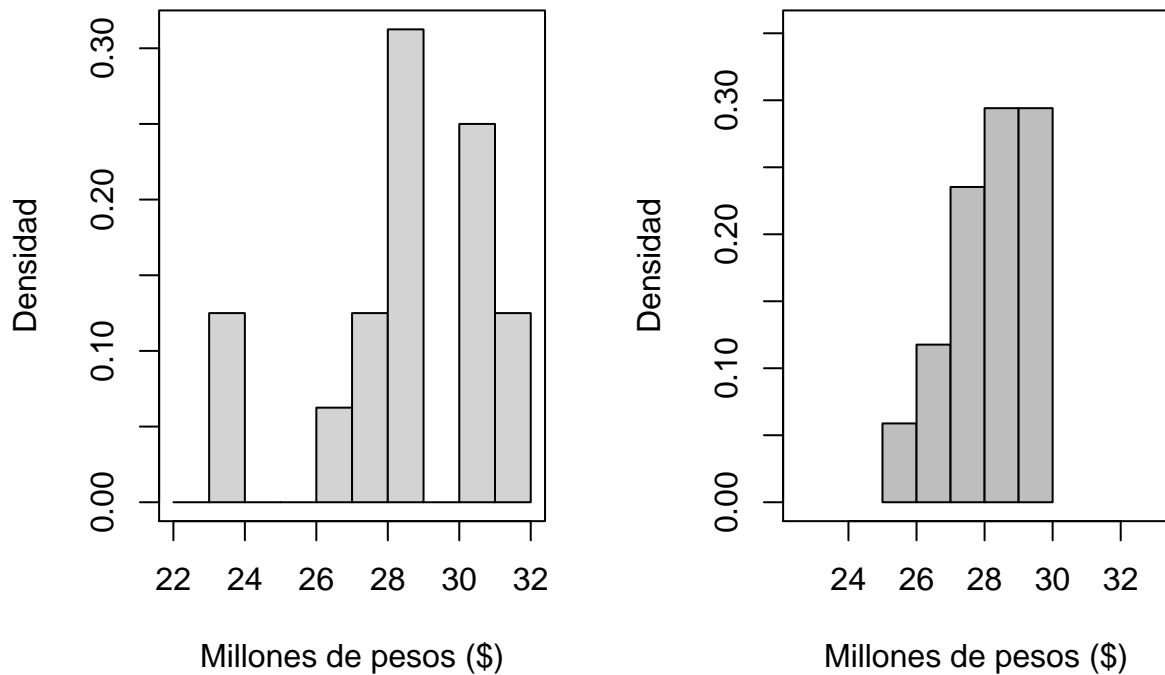
### 95% confidence and prediction intervals for modeloVero



### Datos Muestrales vs. Experto

```
#win.graph()
par(mfrow = c(1,2))
datos2013 <- precio[modeloAuto == 13]
hist(datos2013, breaks = 22:32, freq = F,
      xlab = "Millones de pesos ($)",
      ylab = "Densidad", main = NA)
box()
histograma(frecus2013, limites2013,
           xlab = "Millones de pesos ($)",
           ylab = "Densidad")
title("Datos precios vs. elicitación: 2013", line = -1, outer = T)
```

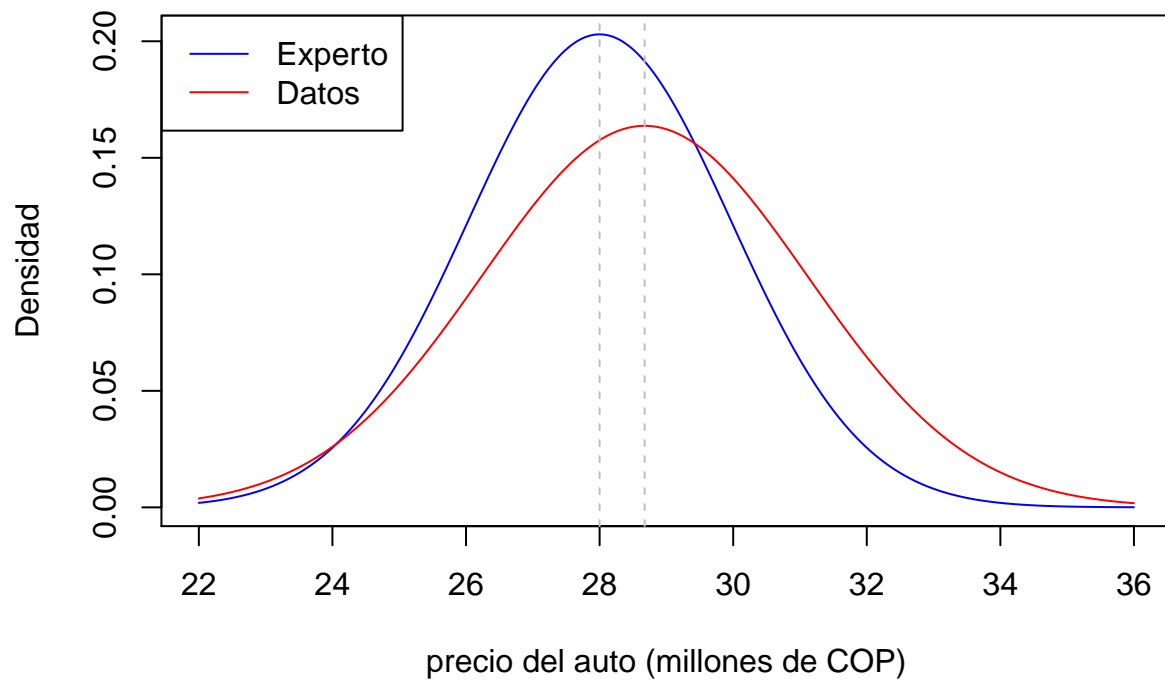
## Datos precios vs. elicitación: 2013



2013

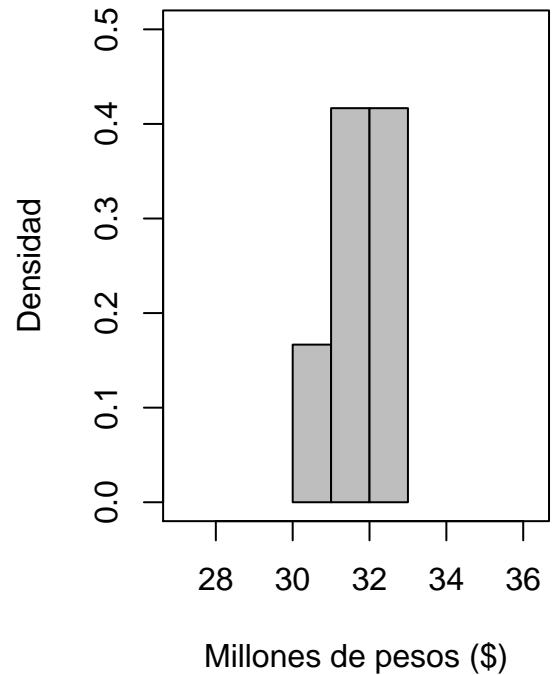
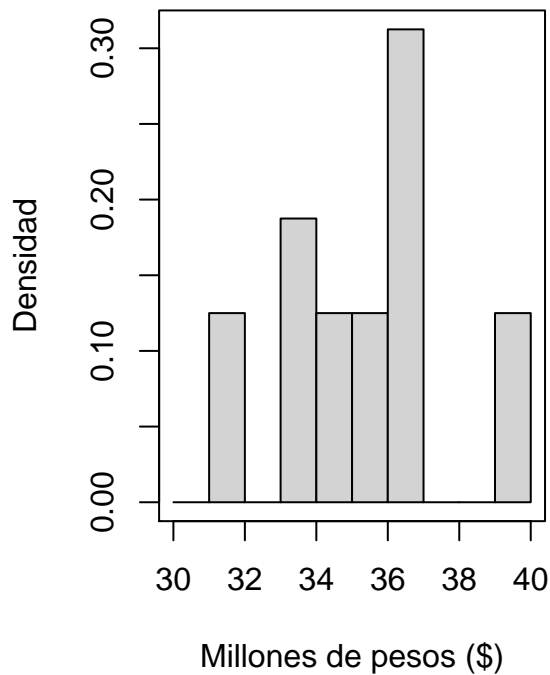
```
xxx <- seq(22,36, length.out = 1000)
plot(xxx, dnorm(xxx, param.opt2013[1], param.opt2013[2]), type = "l", col = "blue",
     main = "Distribución del precio según experto y datos del modelo 2013",
     xlab = "precio del auto (millones de COP)",
     ylab = "Densidad")
points(xxx, dnorm(xxx, mean(datos2013), sd(datos2013)), type = "l", col = "red")
legend("topleft", legend=c("Experto", "Datos"), col=c("blue", "red"), lty = 1)
abline(v = param.opt2013[1], lty = 2, col = "grey")
abline(v = mean(datos2013), lty = 2, col = "grey")
```

## Distribución del precio según experto y datos del modelo 2013



```
#win.graph()
par(mfrow = c(1,2))
datos2015 <- precio[modeloAuto == 15]
hist(datos2015, breaks = 30:40,freq = F,
      xlab = "Millones de pesos ($)",
      ylab = "Densidad", main = NA)
box()
histograma(frecus2015,limites2015,
           xlab = "Millones de pesos ($)",
           ylab = "Densidad")
title("Datos precios vs. elicitación: 2015", line = -1, outer = T)
```

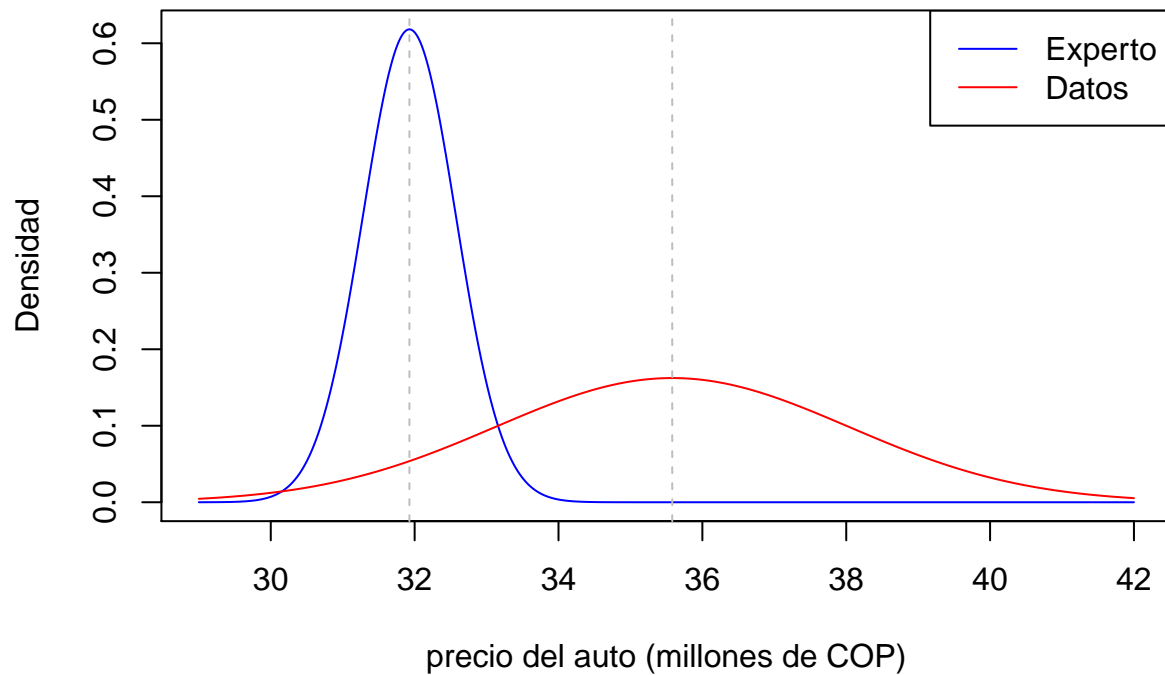
## Datos precios vs. elicitación: 2015



2015

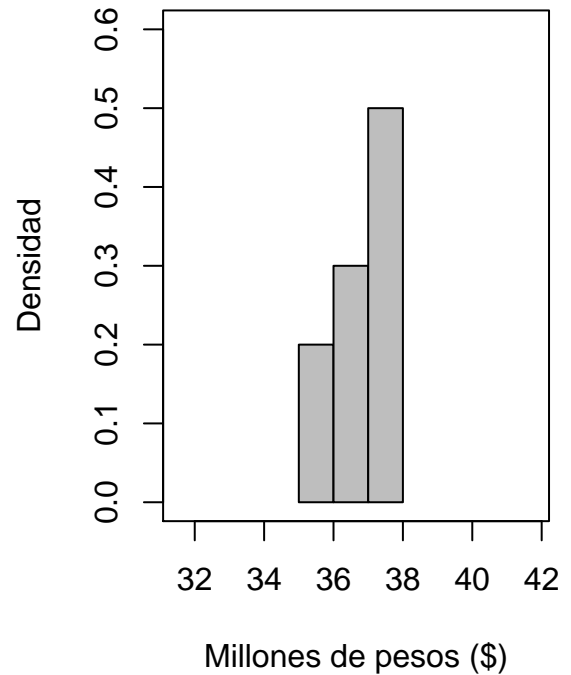
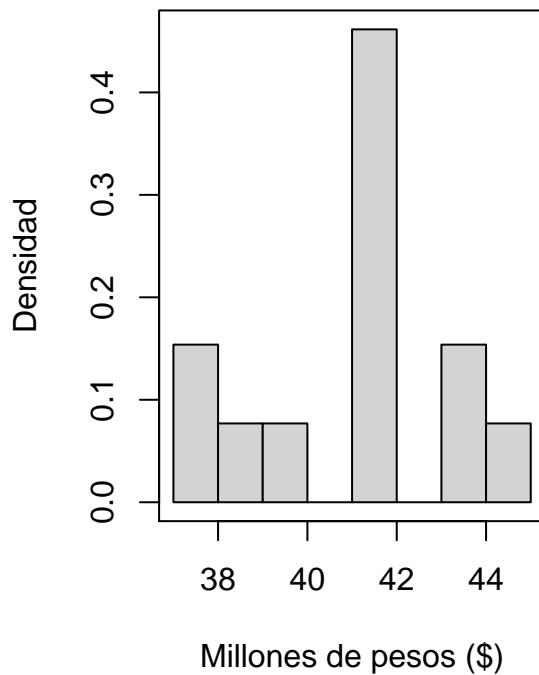
```
xxx <- seq(29,42, length.out = 1000)
plot(xxx, dnorm(xxx, param.opt2015[1], param.opt2015[2]), type = "l", col = "blue",
     main = "Distribución del precio según experto y datos del modelo 2015",
     xlab = "precio del auto (millones de COP)",
     ylab = "Densidad")
points(xxx, dnorm(xxx, mean(datos2015), sd(datos2015)), type = "l", col = "red")
legend("topright", legend=c("Experto", "Datos"), col=c("blue", "red"), lty = 1)
abline(v = param.opt2015[1], lty = 2, col = "grey")
abline(v = mean(datos2015), lty = 2, col = "grey")
```

## Distribución del precio según experto y datos del modelo 2015



```
#win.graph()
par(mfrow = c(1,2))
datos2017 <- precio[modeloAuto == 17]
hist(datos2017,freq = F, breaks = 37:45,
      xlab = "Millones de pesos ($)",
      ylab = "Densidad", main = NA)
box()
histograma(frecus2017,limites2017,
           xlab = "Millones de pesos ($)",
           ylab = "Densidad")
title("Datos precios vs. elicitación: 2017", line = -1, outer = T)
```

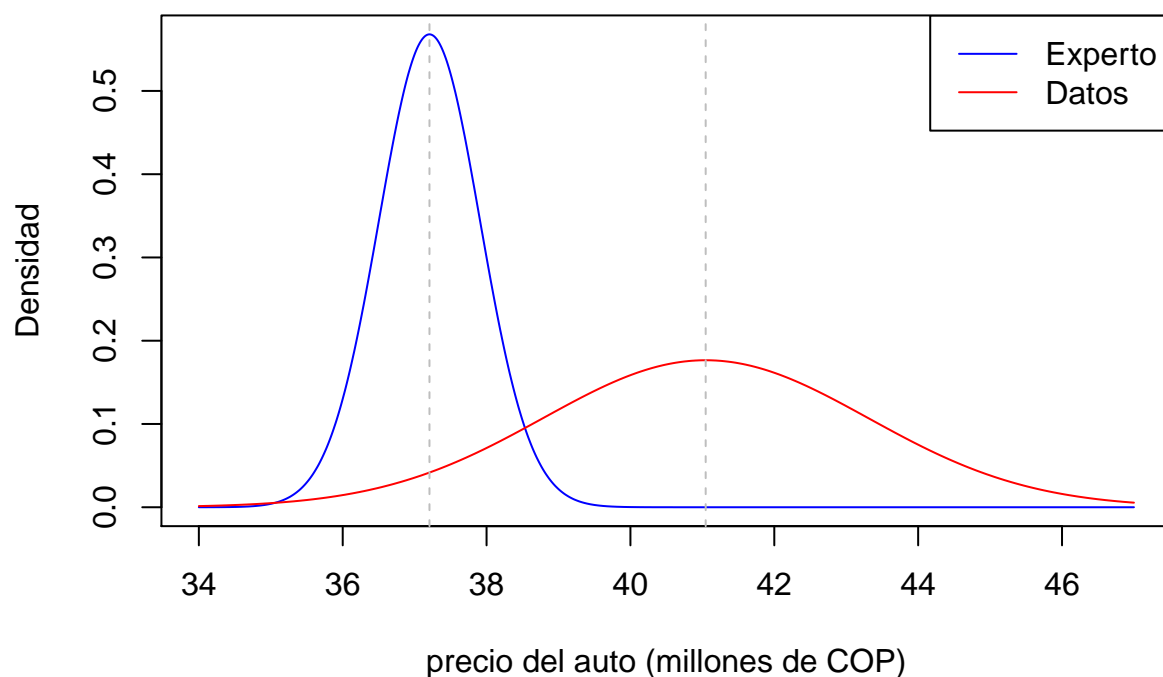
## Datos precios vs. elicitación: 2017



2017

```
xxx <- seq(34,47, length.out = 1000)
plot(xxx, dnorm(xxx, param.opt2017[1], param.opt2017[2]), type = "l", col = "blue",
     main = "Distribución del precio según experto y datos del modelo 2017",
     xlab = "precio del auto (millones de COP)",
     ylab = "Densidad")
points(xxx, dnorm(xxx, mean(datos2017), sd(datos2017)), type = "l", col = "red")
legend("topright", legend=c("Experto", "Datos"), col=c("blue", "red"), lty = 1)
abline(v = param.opt2017[1], lty = 2, col = "grey")
abline(v = mean(datos2017), lty = 2, col = "grey")
```

## Distribución del precio según experto y datos del modelo 2017



Posterior

```
resPoste <- MCMCregress(precio ~ modeloAuto,
  b0 = b0, B0 = round(B0,3),
  sigma.mu = mean(sigmas2), sigma.var = var(sigmas2))
```

```
suPoste <- summary(resPoste)
kable(suPoste$statistics, col.names = colnames(suPoste$statistics))
```

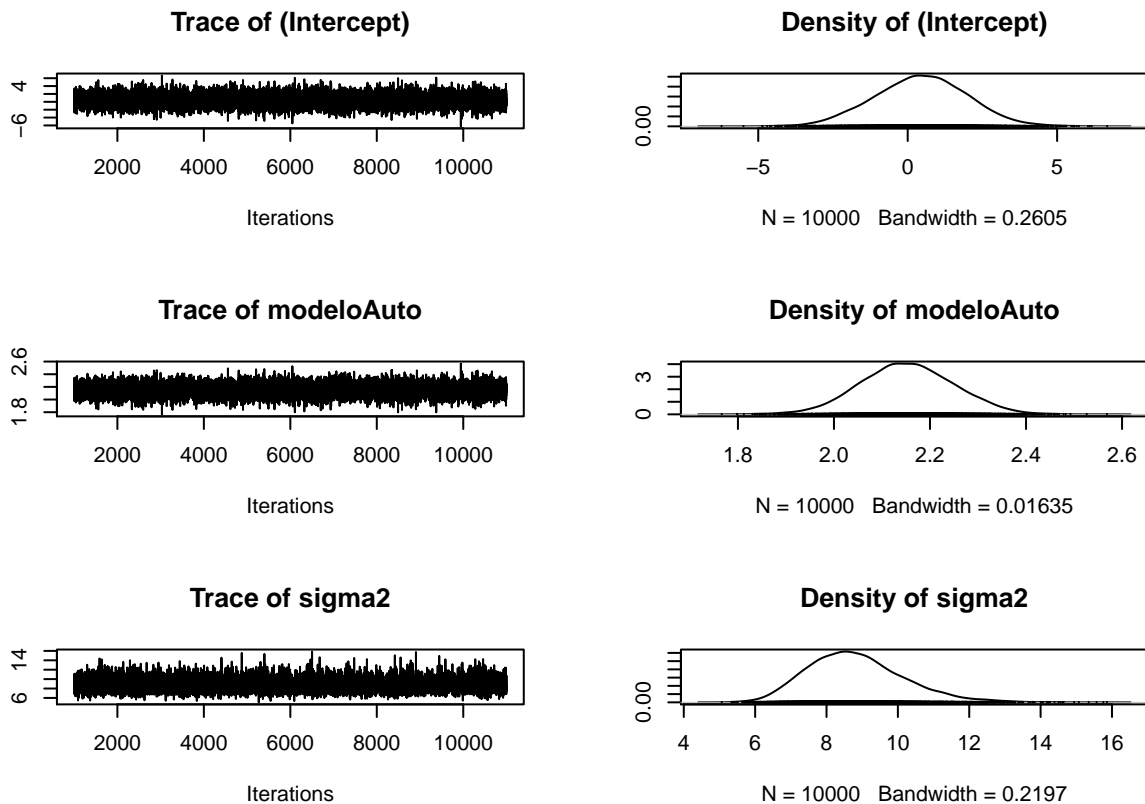
	Mean	SD	Naive SE	Time-series SE
(Intercept)	0.4466519	1.5529481	0.0155295	0.0153135
modeloAuto	2.1528019	0.0973066	0.0009731	0.0009536
sigma2	8.8325715	1.3500479	0.0135005	0.0148292

```
kable(suPoste$quantiles, col.names = colnames(suPoste$quantiles))
```

	2.5%	25%	50%	75%	97.5%
(Intercept)	-2.604465	-0.5890419	0.4618165	1.488456	3.462205
modeloAuto	1.963473	2.0867092	2.1515699	2.217531	2.343329
sigma2	6.584194	7.8767489	8.7080555	9.629242	11.853601



```
plot(resPoste)
```



```
cor(resPoste)
```

```
##           (Intercept) modeloAuto    sigma2
## (Intercept)  1.0000000 -0.9975114 -0.1428914
## modeloAuto  -0.9975114  1.0000000  0.1241324
## sigma2      -0.1428914  0.1241324  1.0000000
```

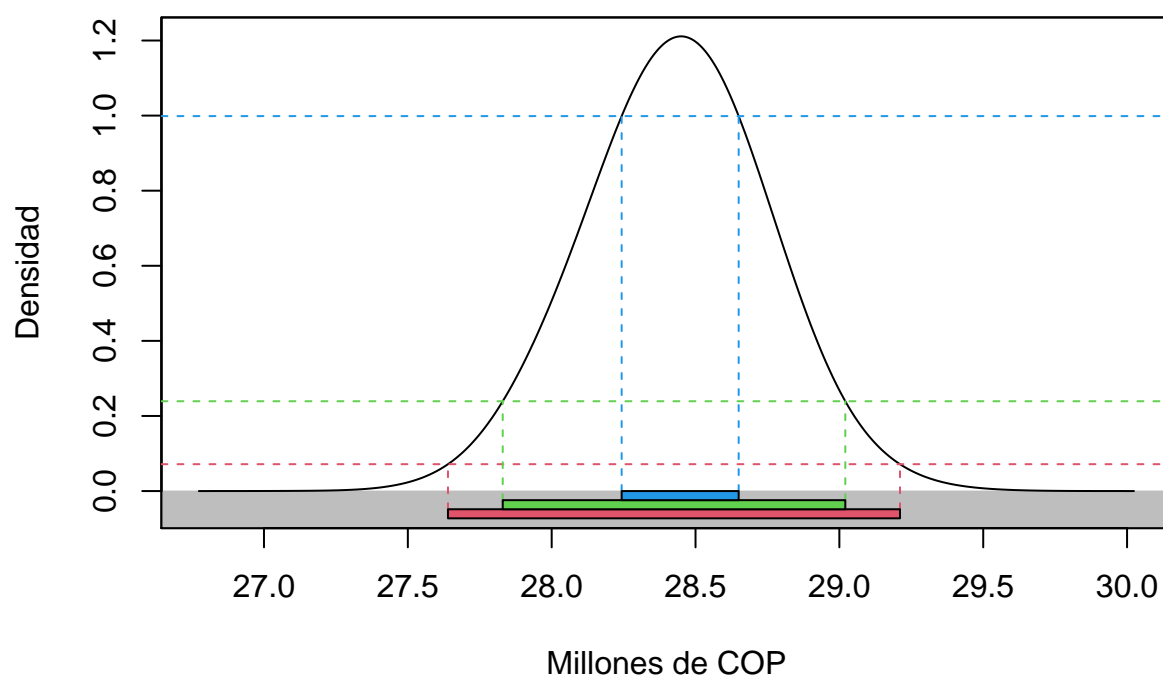
## Precio del modelo 2013

### Distribución del precio medio del modelo 2013

```
preciosMedios2013 <- resPoste[,1] + resPoste[,2] * 13
```

```
hdrPreciosMedios2013 <- hdr.den(preciosMedios2013, xlab = "Millones de COP",
                                ylab = "Densidad",
                                main = expression("Distribución del precio medio del modelo 2013"))
```

## Distribución del precio medio del modelo 2013



```
suM2013 <- summary(preciosMedios2013)
kable(suM2013$statistics, col.names = colnames(suM2013$statistics))
```

Mean	28.4330767
SD	0.3044668
Naive SE	0.0030447
Time-series SE	0.0031036

```
kable(suM2013$quantiles, col.names = colnames(suM2013$quantiles))
```

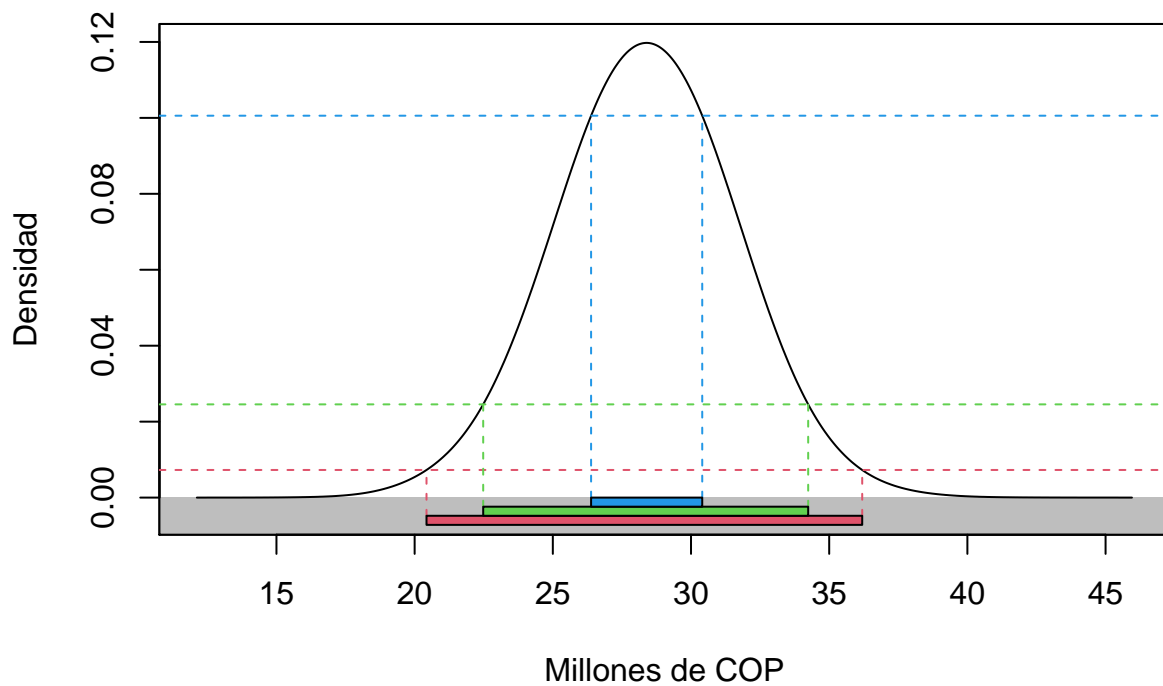
2.5%	27.83028
25%	28.23036
50%	28.43807
75%	28.63710
97.5%	29.02216

## Distribución predictiva del precio del modelo 2013

```
errores <- sapply(resPoste[,3], genera.error)
preciosPredic2013 <- preciosMedios2013 + errores
```

```
hdrPreciosPredic2013 <- hdr.den(preciosPredic2013, xlab = "Millones de COP",
                                ylab = "Densidad",
                                main = expression("Distribución predictiva del precio del modelo 2013"))
```

### Distribución predictiva del precio del modelo 2013



```
suP2013 <- summary(preciosPredic2013)
kable(suP2013$statistics, col.names = colnames(suP2013$statistics))
```

Mean	28.3697892
SD	2.9970633
Naive SE	0.0299706
Time-series SE	0.0308643

```
kable(suP2013$quantiles, col.names = colnames(suP2013$quantiles))
```

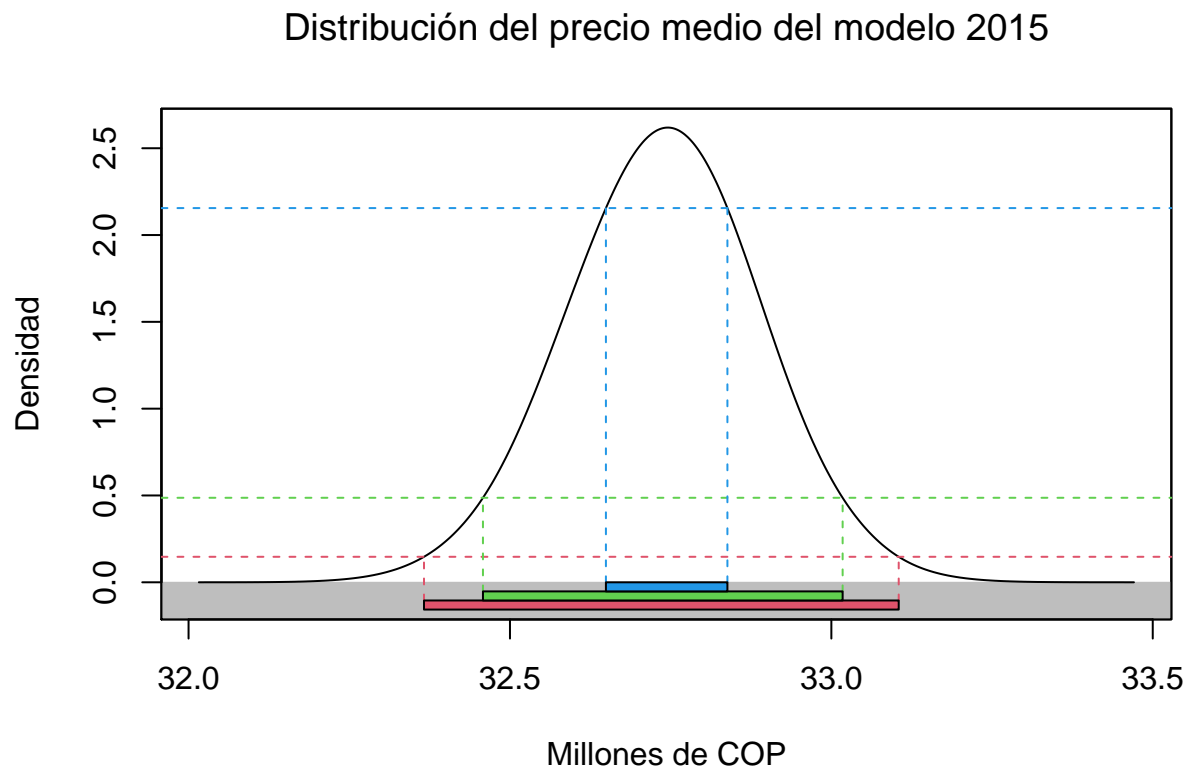
2.5%	22.46036
25%	26.37350
50%	28.38800
75%	30.39281
97.5%	34.22801

## Precio del modelo 2015

### Distribución del precio medio del modelo 2015

```
preciosMedios2015 <- resPoste[,1] + resPoste[,2] * 15
```

```
hdrPreciosMedios2015 <- hdr.den(preciosMedios2015, xlab = "Millones de COP",  
                                ylab = "Densidad",  
                                main = expression("Distribución del precio medio del modelo 2015"))
```



```
suM2015 <- summary(preciosMedios2015)  
kable(suM2015$statistics, col.names = colnames(suM2015$statistics))
```

Mean	32.7386805
SD	0.1414064
Naive SE	0.0014141
Time-series SE	0.0015226

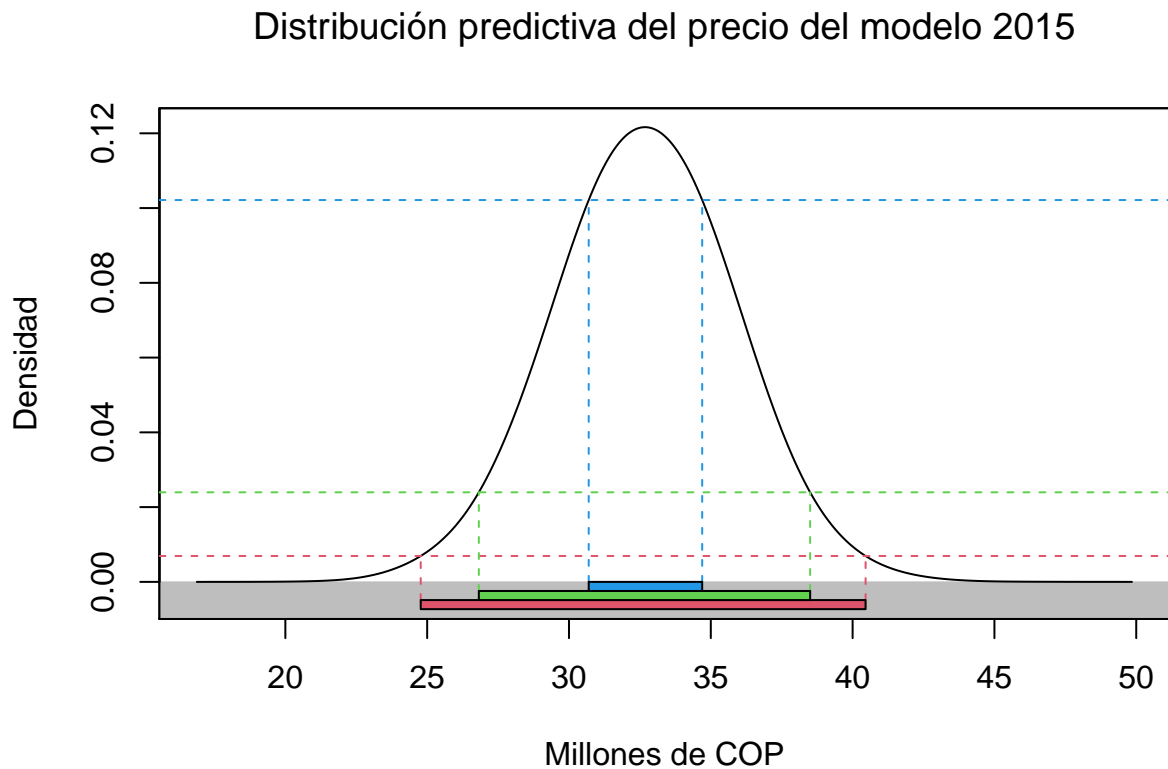
```
kable(suM2015$quantiles, col.names = colnames(suM2015$quantiles))
```

2.5%	32.45697
25%	32.64418
50%	32.74013
75%	32.83338
97.5%	33.01537

### Distribución predictiva del precio del modelo 2015

```
preciosPredic2015 <- preciosMedios2015 + errores
```

```
hdrPreciosPredic2015 <- hdr.den(preciosPredic2015, xlab = "Millones de COP",  
                                ylab = "Densidad",  
                                main = expression("Distribución predictiva del precio del modelo 2015"))
```



```
suP2015 <- summary(preciosPredic2015)  
kable(suP2015$statistics, col.names = colnames(suP2015$statistics))
```

Mean	32.6753931
SD	2.9846612
Naive SE	0.0298466
Time-series SE	0.0307561

```
kable(suP2015$quantiles, col.names = colnames(suP2015$quantiles))
```

2.5%	26.80926
25%	30.68312
50%	32.68747
75%	34.69018
97.5%	38.48212

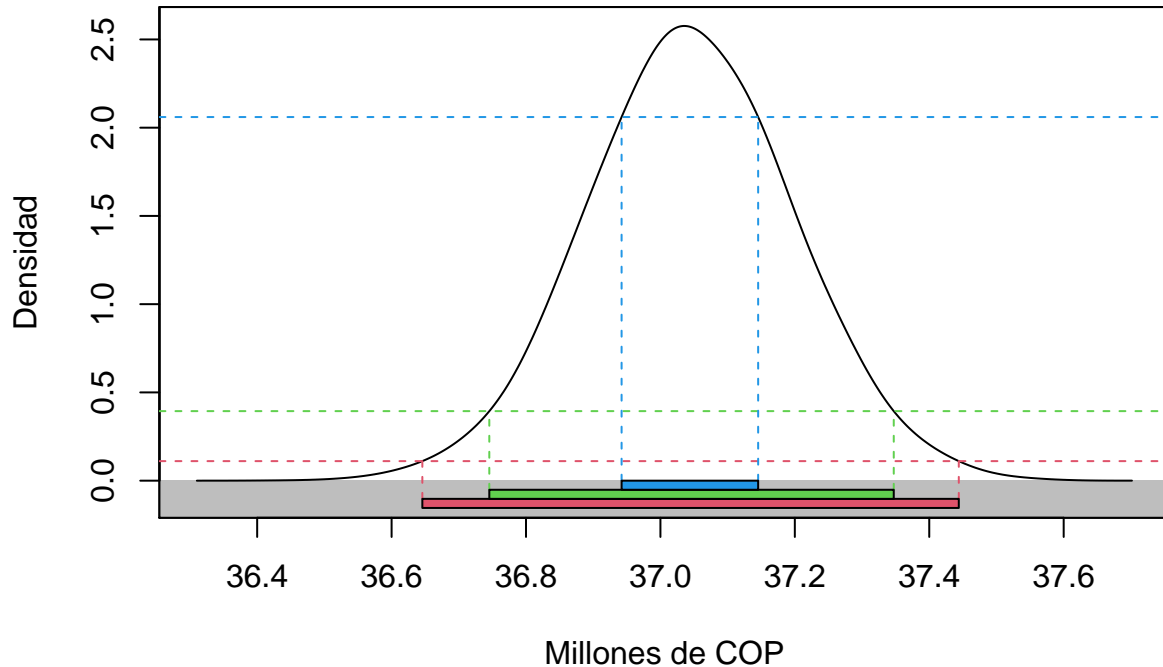
## Precio del modelo 2017

### Distribución del precio medio del modelo 2017

```
preciosMedios2017 <- resPoste[,1] + resPoste[,2] * 17
```

```
hdrPreciosMedios2017 <- hdr.den(preciosMedios2017, xlab = "Millones de COP",  
                                ylab = "Densidad",  
                                main = expression("Distribución del precio medio del modelo 2017"))
```

### Distribución del precio medio del modelo 2017



```
suM2017 <- summary(preciosMedios2017)  
kable(suM2017$statistics, col.names = colnames(suM2017$statistics))
```

Mean	37.0442843
SD	0.1517895
Naive SE	0.0015179
Time-series SE	0.0015179

```
kable(suM2017$quantiles, col.names = colnames(suM2017$quantiles))
```

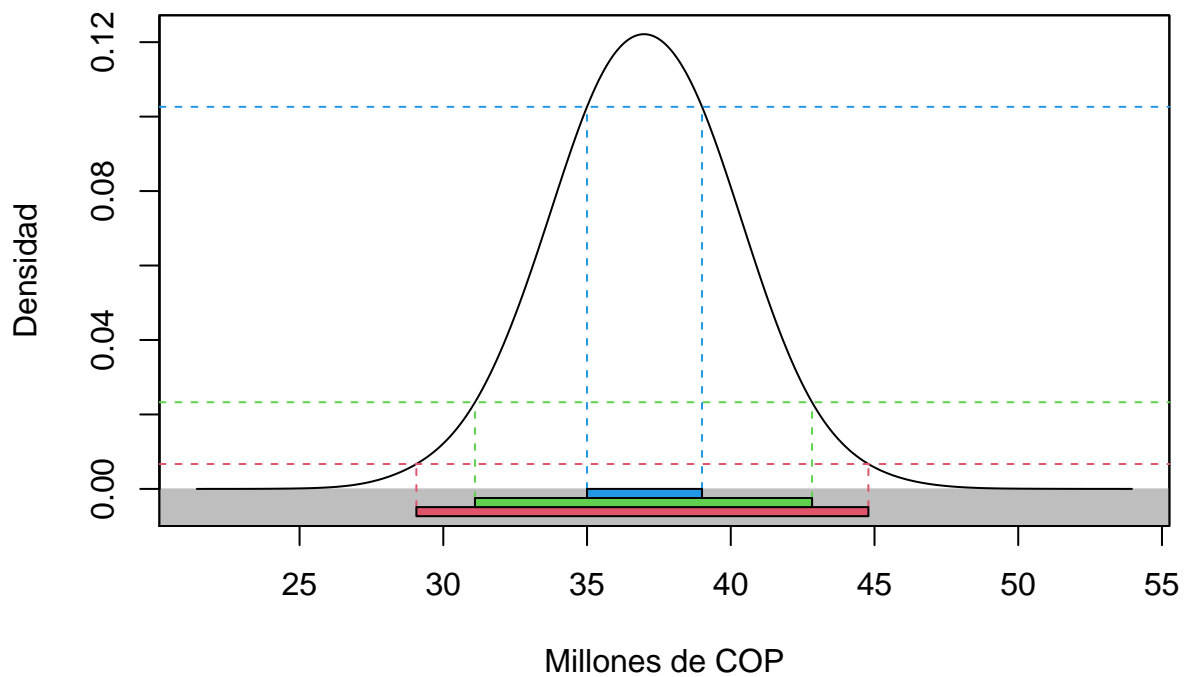
2.5%	36.74423
25%	36.94307
50%	37.04256
75%	37.14640
97.5%	37.34642

### Distribución predictiva del precio del modelo 2017

```
preciosPredic2017 <- preciosMedios2017 + errores
```

```
hdrPreciosPredic2017 <- hdr.den(preciosPredic2017, xlab = "Millones de COP",  
                                ylab = "Densidad",  
                                main = expression("Distribución predictiva del precio del modelo 2017"))
```

### Distribución predictiva del precio del modelo 2017



```
suP2017 <- summary(preciosPredic2017)
kable(suP2017$statistics, col.names = colnames(suP2017$statistics))
```

Mean	36.9809969
SD	2.9849229
Naive SE	0.0298492
Time-series SE	0.0307652

```
kable(suP2017$quantiles, col.names = colnames(suP2017$quantiles))
```

2.5%	31.09711
25%	34.99406
50%	36.98867
75%	38.99287
97.5%	42.82485