Respuestas Binarias (cont.)

MCP

22 de febrero de 2018

Intervalo de confianza de Clopper-Pearson

Este intervalo usa la relaci'on entre distribuci'on binomial y la distribuci'on beta.

El intervalo de confianza de $(1-\alpha)100\%$ est'a dado por

$$\text{beta}(\alpha/2; w, n-w+1) < \pi < \text{beta}(1-\alpha/2; w+1, n-w)$$

donde

$$\alpha = \int_0^{v_\alpha} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} v^{a-1}(1-v)^{b-1}\, dv; \quad a > 0 \wedge b > 0$$

```
# Para w=4, n=10, 95% confianza.
w<-4
n<-10
alpha<-0.05
pi.hat<-w/n
library(binom)
round(qbeta(p = c(alpha/2, 1-alpha/2), shape1 = c(w, w+1), shape2 = c(n-w+1, n-w)),4)
```

```
## [1] 0.1216 0.7376
```

```
binom.confint(x = w, n = n, conf.level = 1-alpha, methods = "exact")
```

```
##   method x  n mean     lower     upper
## 1  exact 4 10  0.4 0.1215523 0.7376219
```

Pruebas

Si sólo un parámetro es de nuestro interés, por lo general se prefiere usar intervalos por sobre prueba de hipótesis. Sin embargo puede haber situaciones donde un valor fijo de π, digamos π_0 es de interés, lo que lleva a la prueba de hipótesis $H_0: \pi = \pi_0$ contra $H_a: \pi \neq \pi_0$.

Con respecto al intervalo de Wilson, cuando la hipótesis nula es verdadera, Z_0 debería tener una distribución aproximadamente normal. La hipótesis nula es rechazada cuando un valor inusual de Z_0 es observado con relación a su distribución,a saber, algo menor a $-Z_{1-\alpha/2}$ o mayor que $Z_{1-\alpha/2}$.

El valor p es una medida de cuan extremo es el valor del estadístico de prueba con respecto a lo que se espera cuando H_0 es verdadera. Este valor se calcula como $2P(Z > |Z_0|)$.

Existen procedimientos alternativos, como la prueba de razón de verosimilitud (LRT). De manera informal, el estadístico LRT es

$$\Lambda = \frac{\text{Valor máximo de la función de verosimilitud bajo } H_0}{\text{Valor máximo de la función de verosimilitud bajo } H_0 \text{ o } H_a}$$

Parala prueba $H_0: \pi = \pi_0$ contra $H_a: \pi \neq \pi_0$, el valor del denominador es $\hat{\pi}^w(1-\hat{\pi})^{n-w}$ y el numerador es $\pi_0^w(1-\pi_0)^{n-w}$. El estadístico transformado $-2\log(\Lambda)$ resulta ser aproximadamente χ_1^2 en muestras grandes si la hipótesis nula es verdadera. Se rechaza la hipótesis si $-2\log(\Lambda) > \chi^2_{1,1-\alpha/2}$, donde $\chi^2_{1,1-\alpha/2}$ es el $1-\alpha/2$-ésimo cuantil de una distribución ji-cuadrada con 1 grado de libertad.

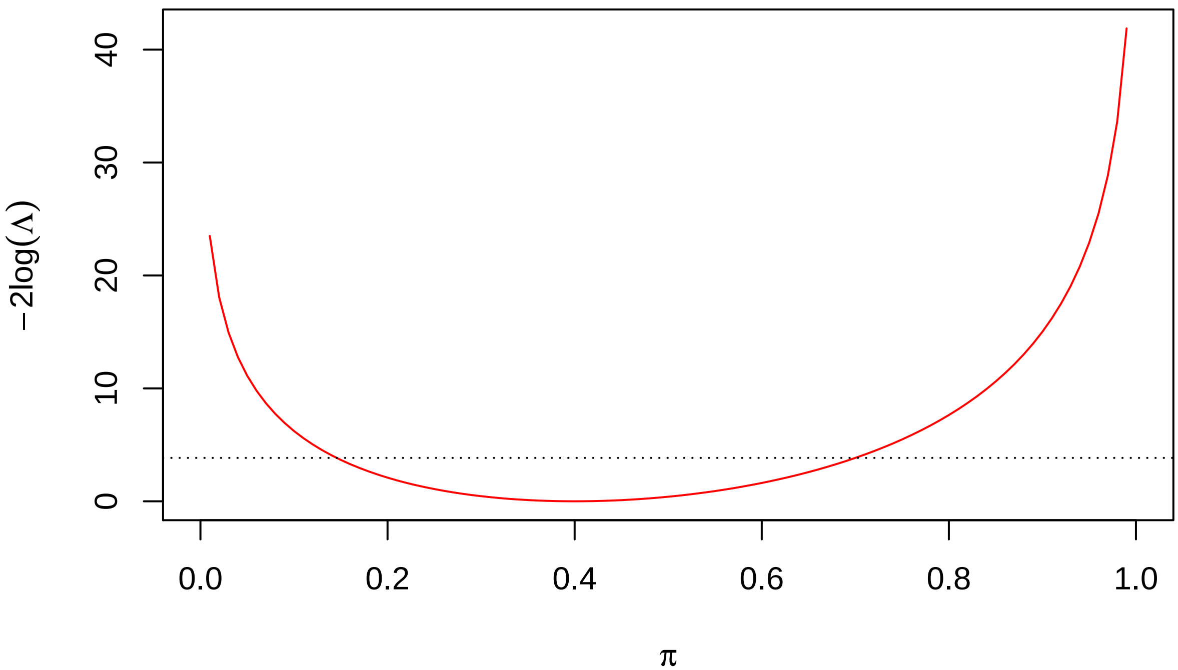
```r
# intervalo LRT

LRT.int<-binom.confint(x = w, n = n, conf.level = 1-alpha, methods = "lrt")
LRT.int
```

```
##   method x  n mean     lower     upper
## 1    lrt 4 10  0.4 0.1456425 0.7000216
```

```r
# -2log(Lambda)
LRT<-function(pi.0, w, n) {
pi.hat<-w/n
-2*(w*log(pi.0/pi.hat) + (n-w)*log((1-pi.0)/(1-pi.hat)))     }

# Gráfica de -2log(Lambda) - Los valores de pi por debajo de la recta horizontal ji^2_1,1-alpha corresp
curve(expr = LRT(pi.0 = x, w = w, n = n), from = 0, to = 1, col = "red", xlab =expression(pi),
            ylab = expression(-2*log(Lambda)))
abline(h = qchisq(p = 1-alpha, df = 1), lty = "dotted")
```




```r
# Puntos de intersección de -2log(Lambda) y chi^2_1,1-alpha
LRT(pi.0 = LRT.int$lower, w = w, n = n)
```

```
## [1] 3.841437
```

```r
LRT(pi.0 = LRT.int$upper, w = w, n = n)
```

```
## [1] 3.841458
```

```r
# hacemos -2log(Lambda) - chi^2_1,1-alpha = 0 y resolvemos para encontrar los límites superior # e infe
LRT2<-function(pi.0, w, n, alpha) {
pi.hat<-w/n
-2*(w*log(pi.0/pi.hat) + (n-w)*log((1-pi.0)/(1-pi.hat))) - qchisq(p = 1-alpha, df = 1)}

# Intervalo de confianza
uniroot(f = LRT2, lower = 0, upper = w/n, w = w, n = n, alpha = alpha)  # Límite inferior
```

```
## $root
## [1] 0.1456271
##
## $f.root
## [1] 0.0006063786
##
## $iter
## [1] 6
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

```
uniroot(f = LRT2, lower = w/n, upper = 1, w = w, n = n, alpha = alpha)  # Límite superior
```

```
## $root
## [1] 0.6999695
##
## $f.root
## [1] -0.001490819
##
## $iter
## [1] 3
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 0.000116483
```

```
# Intervalo logit
binom.confint(x = w, n = n, conf.level = 1-alpha, methods = "logit")
```

```
##   method x  n mean    lower     upper
## 1  logit 4 10  0.4 0.158342 0.7025951
```

```
# Intervalo de Blaker
library(BlakerCI)
binom.blaker.limits(x = w, n = n, level = 1-alpha)
```

```
## [1] 0.1500282 0.7170653
```

```
# Mid-p
library(package = PropCIs)
midPci(x = w, n = n, conf.level = 1 - alpha)
```

```
##
##
##
## data:
##
## 95 percent confidence interval:
##  0.1426 0.7086
```

Niveles de confianza verdaderos para intervalos de confianza

Las inferencias para π del intervalo de confianza de Wald se basan en la aproximación a la distribución normal para el estimador de máxima verosimilitud. Para que esta aproximación sea buena, se necesita una muestra grande. Más aún, en ocasiones $\hat{\pi}$ puede, a veces, tomar un número finito de valores, pero la distribución normal es una función continua. Estos problemas llevan al intervalo de Wald a ser aproximado en el sentido de que la probabilidad de que el intervalo cubra el parámetro (covertura o nivel de confianza verdadero) no sea necesariamente el nivel $1 - \alpha$.

Consideremos un ejemplo donde $n = 40$ con un nivel de confianza de 0.95 ($\alpha = 0.05$):

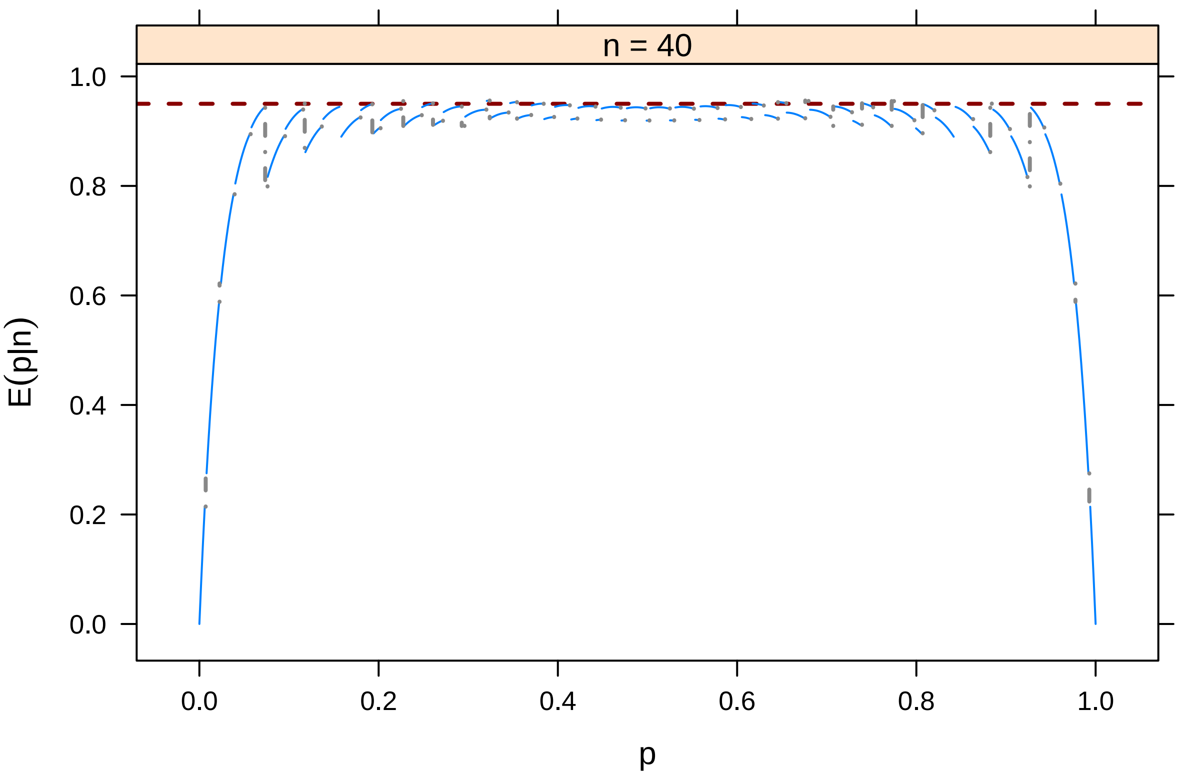
```
binom.coverage(p = 0.16, n = n, conf.level = 0.95, method = c("asymptotic", "agresti-coull",
      "wilson", "exact"))
```

```
## Warning in if (method %in% binom.methods) {: la condición tiene longitud >
## 1 y sólo el primer elemento será usado
```

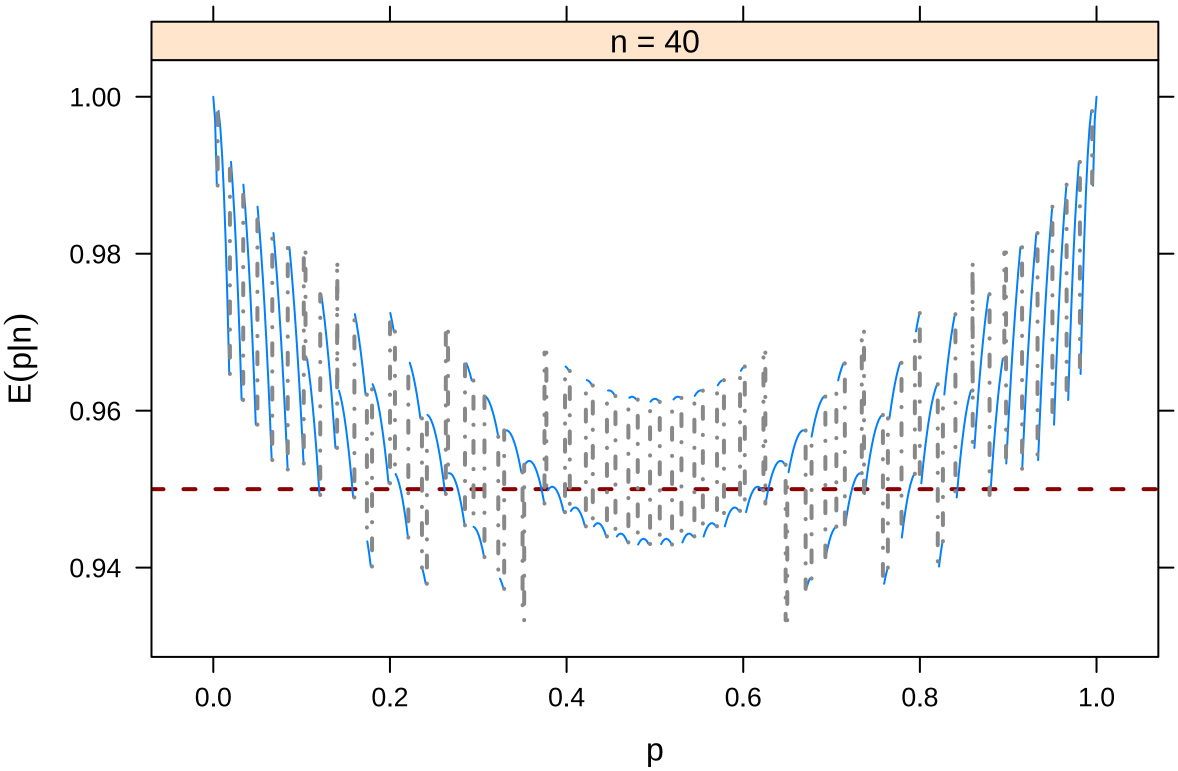
```
##          method    p  n  coverage
## 1 agresti-coull 0.16 10 0.9386423
## 2    asymptotic 0.16 10 0.8120886
## 3         exact 0.16 10 0.9869899
## 4        wilson 0.16 10 0.9386423
```

```
binom.plot(n = 40, method = binom.asymp, np = 500, conf.level = 0.95)
```

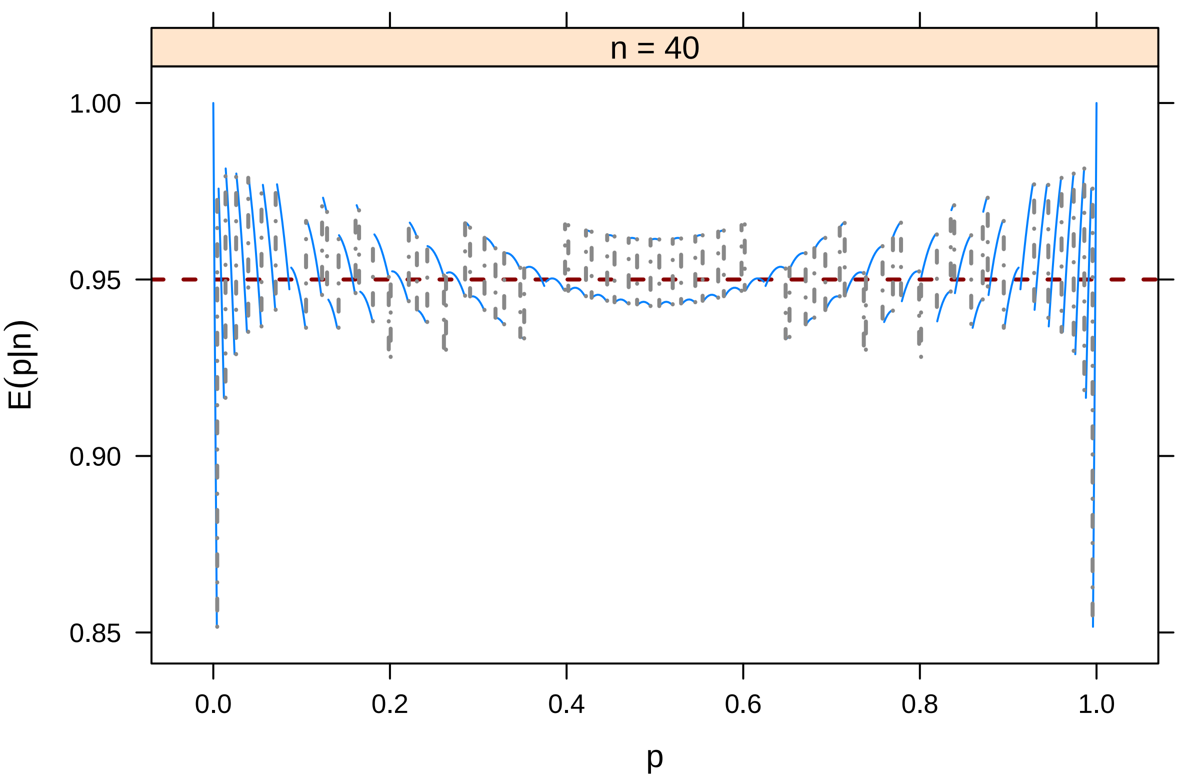
```
## Loading required package: lattice
```



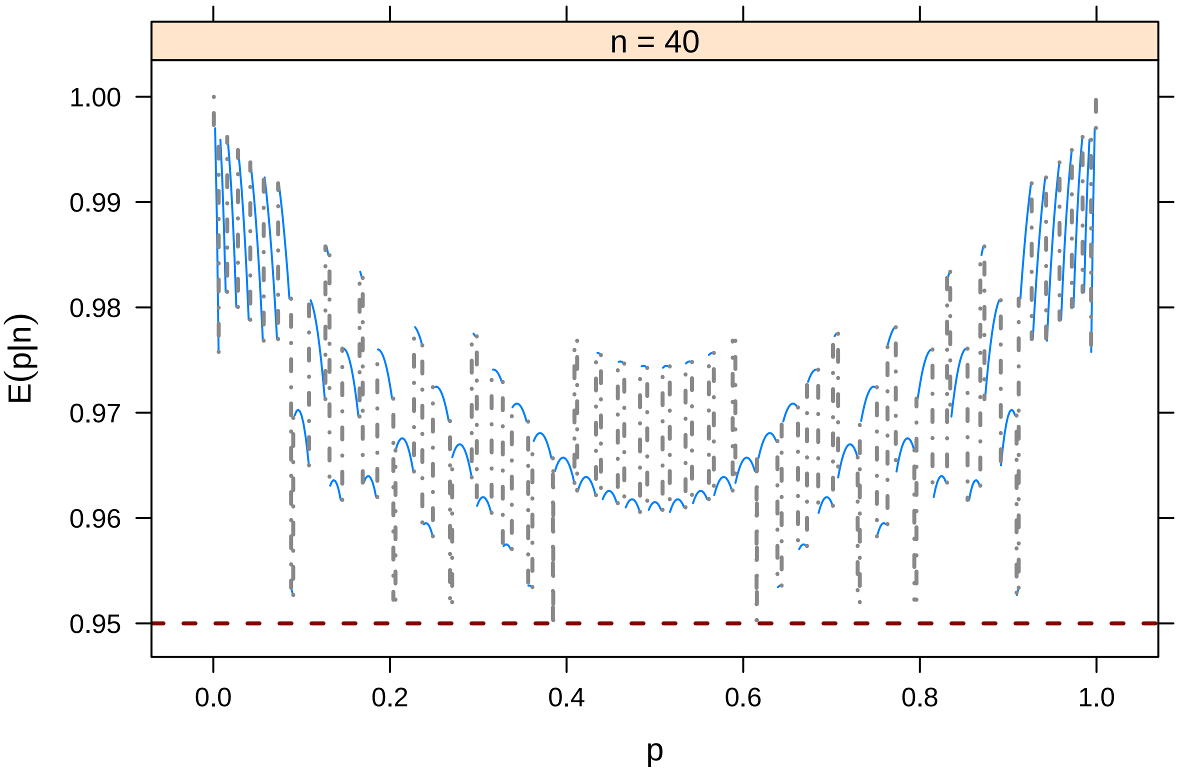
```
binom.plot(n = 40, method = binom.agresti.coull, np = 500, conf.level = 0.95)
```




```
binom.plot(n = 40, method = binom.wilson, np = 500, conf.level = 0.95)
```



```r
binom.plot(n = 40, method = binom.exact, np = 500, conf.level = 0.95)
```



Nivel de confianza verdadero para el intervalo de Wald

Calcularemos el nivel de confianza verdadero para el intervalo de Wald con $n = 40$, $\pi = 0.157$ y $\alpha = 0.05$ en 3 pasos. 1. Encontrar la probabilidad de obtener cada posible valor de w usando la función "dbinom()" con $n = 40$ y $\pi = 0.157$. 2. Calcular el intervalo de confianza de 95% de Wald. 3. Sumar las probabilidades correspondientes a aquellos intervalos que contienen a $\pi = 0.157$; éste es el nivel de confianza verdadero.

```r
alpha<-0.05
pi<-0.157
# pi<-0.156
n<-40
w<-0:n
pi.hat<-w/n
pmf<-dbinom(x = w, size = n, prob = pi)
var.wald<-pi.hat*(1-pi.hat)/n
lower<-pi.hat - qnorm(p = 1-alpha/2) * sqrt(var.wald)
upper<-pi.hat + qnorm(p = 1-alpha/2) * sqrt(var.wald)
save<-ifelse(test = pi>lower, yes = ifelse(test = pi<upper, yes = 1, no = 0), no = 0)
data.frame(w, pi.hat, round(data.frame(pmf, lower, upper),4), save)[1:13,]
```

```
##      w pi.hat    pmf   lower  upper save
## 1    0  0.000 0.0011  0.0000 0.0000    0
## 2    1  0.025 0.0080 -0.0234 0.0734    0
## 3    2  0.050 0.0292 -0.0175 0.1175    0
## 4    3  0.075 0.0689 -0.0066 0.1566    0
## 5    4  0.100 0.1187  0.0070 0.1930    1
## 6    5  0.125 0.1591  0.0225 0.2275    1
```

```
## 7    6   0.150 0.1729  0.0393 0.2607    1
## 8    7   0.175 0.1564  0.0572 0.2928    1
## 9    8   0.200 0.1201  0.0760 0.3240    1
## 10   9   0.225 0.0795  0.0956 0.3544    1
## 11 10   0.250 0.0459  0.1158 0.3842    1
## 12 11   0.275 0.0233  0.1366 0.4134    1
## 13 12   0.300 0.0105  0.1580 0.4420    0
```

```
sum(save*pmf)
```

```
## [1] 0.875905
```

```
sum(dbinom(x = 4:11, size = n, prob = pi)) # Nivel de confianza real
```

```
## [1] 0.875905
```

Estimación del verdadero nivel de confianza para el intervalo de Wald

Supongamos de nuevo que $n = 40, \pi = 0.157, \alpha = 0.05$. El proceso para estimar el nivel de confianza verdadero se pueden seguir los siguientes pasos:

1. Simular 1000 muestras usando *rbinom()*.
2. Calcular el intervalo de Wald de 95% para cada muesta.
3. Calcular la proporción de intervalos que contienen $\pi = 0.157$; éste es el valor del estimado del nivel de confianza verdadero.

```
numb.bin.samples<-1000  # Número de muestras binomiales de tamaño n

# Para calcular el número de veces que cada "w" ocurre:
set.seed(4516)
w<-rbinom(n = numb.bin.samples, size = n, prob = pi)
counts<-table(x = w)
counts
```

```
## x
##   1   2   3   4   5   6   7   8   9  10  11  12  13
##   8  35  64 123 147 165 172 123  76  46  26  11   4
```

```
sum(counts[4:11])/numb.bin.samples
```

```
## [1] 0.878
```

```
# Para calcular el estimado del nivel de confianza verdadero:
pi.hat<-w/n
pi.hat[1:10]
```

```
##  [1] 0.150 0.150 0.175 0.200 0.200 0.150 0.200 0.075 0.125 0.100
```

```
var.wald<-pi.hat*(1-pi.hat)/n
lower<-pi.hat - qnorm(p = 1-alpha/2) * sqrt(var.wald)
upper<-pi.hat + qnorm(p = 1-alpha/2) * sqrt(var.wald)
data.frame(w, pi.hat, lower, upper)[1:10,]
```

```
##    w pi.hat       lower     upper
## 1  6  0.150  0.039344453 0.2606555
## 2  6  0.150  0.039344453 0.2606555
## 3  7  0.175  0.057249138 0.2927509
```

```
## 4   8   0.200   0.076040994 0.3239590
## 5   8   0.200   0.076040994 0.3239590
## 6   6   0.150   0.039344453 0.2606555
## 7   8   0.200   0.076040994 0.3239590
## 8   3   0.075 -0.006624323 0.1566243
## 9   5   0.125   0.022511030 0.2274890
## 10  4   0.100   0.007030745 0.1929693
```

```r
save<-ifelse(test = pi>lower, yes = ifelse(test = pi<upper, yes = 1, no = 0), no = 0)
save[1:10]
```

```
##  [1] 1 1 1 1 1 1 1 0 1 1
```

```r
mean(save)
```

```
## [1] 0.878
```

```r
true.conf<-mean(save)
cat("Un estimado del verdadero nivel de confianza es:", round(true.conf,4), "\n")
```

```
## Un estimado del verdadero nivel de confianza es: 0.878
```