EST–24107: Tarea 2

Carlos Lezama, Marco Medina,
Emiliano Ramírez y Santiago Villarreal

Lunes, 13 de septiembre de 2021

Problema 1

Para este problema se impondrán los siguientes supuestos:

- La población es cerrada. Durante el tiempo de estudio asumimos que el tamaño poblacional N permanece constante o, si se produce una variación en dicho tamaño, esta es insignificante respecto al verdadero tamaño poblacional.
- Todos los individuos de la población tienen la misma probabilidad de ser capturados en la primera muestra.
- La segunda muestra es también una muestra aleatoria. Para la validez de esta hipótesis, necesitamos que los métodos con los que se marquen a los individuos no afecten a la capturabilidad, de modo que en la recaptura, un individuo que no está marcado tenga la misma probabilidad de ser capturado que uno que no lo está.

Usaremos dos estimadores para el número total de la población de peces N. El primero corresponde al estimador conseguido por el método de Máxima Verosimilitud. El segundo corresponde al estimador propuesto por Chapman. El motivo del uso de los dos estimadores es que el estimador de MV puede indefinirse si en la segunda muestra aleatoria no se captura ningún pez marcado y que además no es insesgado ni tiene un estimador de varianza insesgado.

Notemos que el problema puede modelarse con una distribución hipergeométrica, ya que tenemos a la población N particionada en dos conjuntos: marcados y no marcados. Así pues, nuesta función de verosimilitud es la función masa de probabilidad de la distribución hipergeométrica con argumento N. Para obtener el estimador de máxima verosimilitud maximizamos discretamente nuestra función de verosimilitud. Denotemos con $\hat{N}$ el estimador MV y al coeficiente binomial con $binom(N,k)$

$$\hat{N} \text{ maximiza } \mathcal{L} \iff \frac{\mathcal{L}(N)}{\mathcal{L}(N-1)} \geq 1 \Rightarrow$$

$$\frac{(N-n_1)(N-n_2)}{N(N-n_1-n_2+y)} > 1 \Rightarrow$$

$$N < \frac{n_1 * n_2}{y} \Rightarrow \hat{N} = ceil\left(\frac{n_1 * n_2}{y}\right).$$

Por otro lado, el estimador de Chapman, denotado con tiene la siguiente expresión,

$$= ceil\left(\frac{(n_1+1)(n_2+1)}{y+1} - 1.\right)$$

Por tanto con los datos del problema tenemos que la estimación puntual de N es,

$$\hat{N} = ceil\left(\frac{250 * 174}{50}\right) = 870$$

$$= ceil\left(\frac{(250+1)(174+1)}{50+1} - 1 = 861.\right)$$

Ahora, para determinar el tamaño de n_2 para conseguir una buena aproximación de N usaremos un nivel $\alpha = 0.05$ de significancia, la desigualdad de Chebyshev y como parámetros dados a y y n_1. Para obtener el tamaño usaremos el estimador de Chapman y el estimador de la varianza de Chapman que tiene la siguiente expresión

Desigualdad de Chebyshev: $\mathbb{P}\left(|\tilde{N} - \mu| \geq z_{1-\frac{\alpha}{2}} * \sqrt{var(\tilde{N})}\right) < \frac{var(\tilde{N})}{z_{1-\frac{\alpha}{2}} * \sqrt{var(\tilde{N})}} = 0.95$

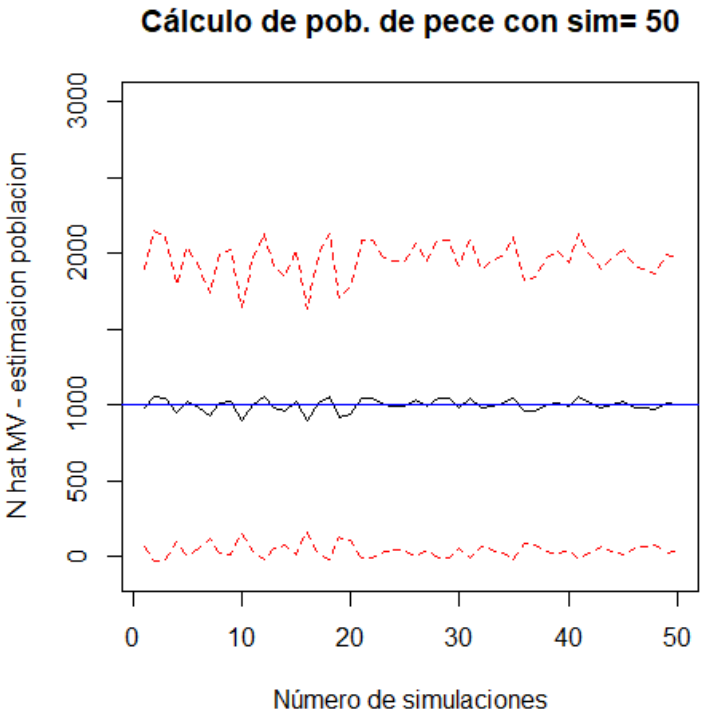
Donde,

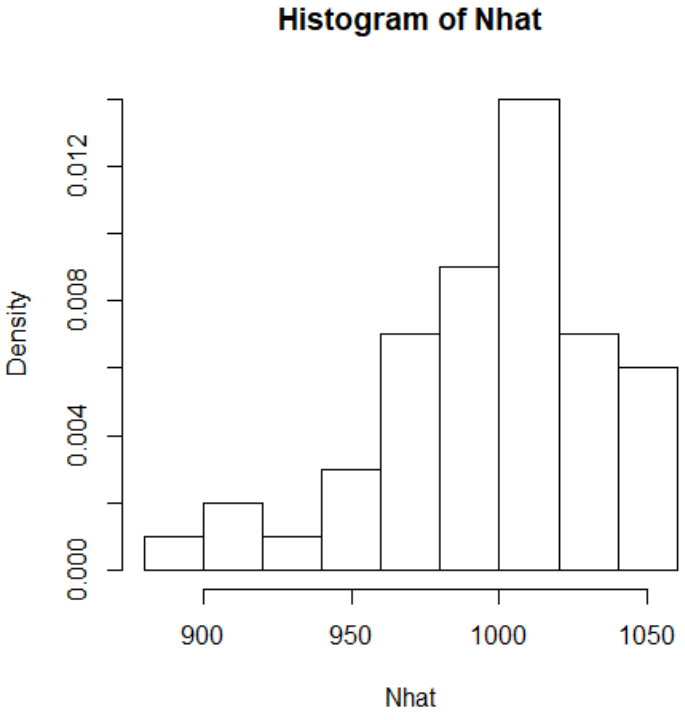
$$var(\tilde{N}) = \frac{(y+1)(n_2+1)(n_1-y)(n_2-y)}{(y+1)^2(y+2)}.$$

A continuacion se presentará el código en R que calcula con una función la estimación de N con simulación usando un intervalo de confianza al 95% tanto con el estimador de máxima verosimilitud como con el de CHapman. Y también una función que calcula el tamaño de n_2 para el tamañoñ de población marcada n_1 y el número de población marcada que se obtuvo en el segundo muestreo y dados.

```
# Esta primera función usa el estimador de la varianza de Sekar y Deming y
# el estimador de N que se obtiene del método de máxima verosimilitud.
# El problema de el estimador de maxima verosimilitud y de la varianza es
# que se indefine cuando el número de peces capturados marcados es cero.
```

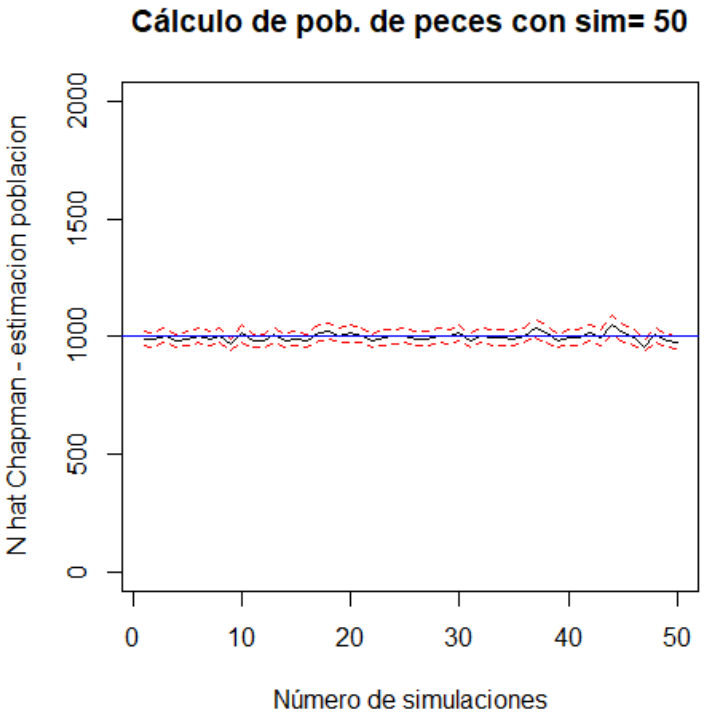
```r
captura_recaptura <- function(N, n1, S) {
  # N <- tamaño población de peces en el estanque
  # n1 <- peces marcados y devueltos de la primera muestra
  # n2 <- tamaño de la muestra después de regresar al estanque
  # S <- numero de simulaciones

  # simulamos la muestra aleatoria del segundo muestreo
  n2 <- sample(0:N, 1)

  alfa <- 0.05 #nivel de significancia
  z <- qnorm(1 - alfa / 2) #cuantil normal

  # obtenemos una simulación de hipergeométrica para simular cuantos
  # peces marcados obtenemos de la segunda muestra
  y <- rhyper(S, n1, N - n1, n2)
  # usamos nuestro estimador de máxima verosimilitud
  Nhat <- ceiling((n1 * n2) / y) #vector de estimaciones

  # calculamos la varianza muestral de nuestro estimador
  varNhat <- (n1 * n2 * (n1 - y) * (n2 - y)) / y ^ 2

  # calculamos los intervalos de confianza para nuestro estimador
  limInf <- Nhat - z * sqrt(varNhat)
  limSup <- Nhat + z * sqrt(varNhat)

  grafica <- plot(
    1:S,
    Nhat,
    type = "l",
    ylim = c(-100, 3 * N),
    main = paste("Cálculo de pob. de peces con sim=", S),
    xlab = "Número de simulaciones",
    ylab = "N hat MV - estimacion poblacion"
  )

  lines(1:S, limInf, lty = 2, col = "red")
  lines(1:S, limSup, lty = 2, col = "red")
  abline(h = N, col = "blue")

  h <- hist(Nhat, prob = T)

  return(grafica, h)
}
```

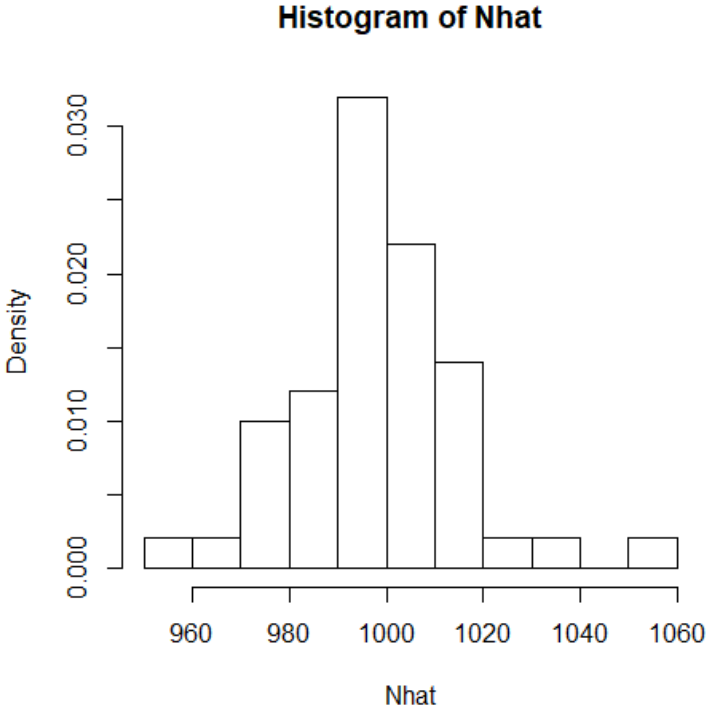

Cálculo de pob. de pece con sim= 50
N hat MV - estimacion poblacion
0
500
1000
2000
3000
0
10
20
30
40
50
Número de simulaciones


Histogram of Nhat
Density
0.000
0.004
0.008
0.012
900
950
1000
1050
Nhat

```r
# La siguiente funcion usa el estimador de N propuesto por Chapman en 1982.
# Este estimador no se indefine y es asintoticamente insesgado así como
# su estimacion de varianza también es asintoticamente insesgada.

captura_recaptura2 <- function(N, n1, S) {
  # N <- tamaño población de peces en el estanque
  # n1 <- peces marcados y devueltos de la primera muestra
  # n2 <- tamaño de la muestra después de regresar al estanque
  # S <- numero de simulaciones

  # simulamos la muestra aleatoria del segundo muestreo
  n2 <- sample(0:N, 1)

  alfa <- 0.05 #nivel de significancia
  z <- qnorm(1 - alfa / 2) #cuantil normal

  # obtenemos una simulación de hipergeométrica para simular cuantos
  # peces marcados obtenemos de la segunda muestra
  y <- rhyper(S, n1, N - n1, n2)
  # usamos nuestro estimador de máxima verosimilitud
  Nhat <- ceiling(((n1 + 1) * (n2 + 1)) / (y + 1) - 1) #vector de estimaciones

  # calculamos la varianza muestral de nuestro estimador
  varNhat <- ((y + 1) * (n2 + 1) * (n1 - y) * (n2 - y)) / ((y + 1) ^ 2 *
                                                             (y + 2))

  # calculamos los intervalos de confianza para nuestro estimador
  limInf <- Nhat - z * sqrt(varNhat)
  limSup <- Nhat + z * sqrt(varNhat)

  grafica <- plot(
    1:S,
    Nhat,
    type = "l",
    ylim = c(0, 2 * N),
    main = paste("Cálculo de pob. de peces con sim=", S),
    xlab = "Número de simulaciones",
    ylab = "N hat Chapman - estimacion poblacion"
  )

  lines(1:S, limInf, lty = 2, col = "red")
  lines(1:S, limSup, lty = 2, col = "red")
  abline(h = N, col = "blue")
```

```
  h <- hist(Nhat, prob = T)

  return(grafica, h)
}

# para calcular el tamanio de n2 para garantizar una buena aproximacion
# usaremos el estimador de Chapman.

tamanio_n2 <- function(N, n1, y) {
  # ¿de qué tamanio debe ser n2 (segunda muestra) para alcanzar
  # una buena aproximacion de N (poblacion)?

  # Dada la población N, los peces marcados n1 y el número de peces marcados
  # obtenidos de la segunda muestra aleatoria y
  # respondemos a la pregunta formulada arriba.

  # Para caluclar el tamanio de n2, usaremos la desigualdad de Chebyshev
  # para alcanzar una buena aproximacion a N
  # tomando un nivel de significacnia alfa =0.05


  alfa <- 0.05 #nivel de significancia
  z <- qnorm(1 - alfa / 2) #cuantil normal

  # vector de numeros consecutivos de 1 hasta la población pues n2
  # puede ser una muestra aleatoria de 1 o N elementos
  n2 <- c(1:N)

  # calculamos constante de chebysev
  c <- z * (1 - alfa)

  # calculemos el error estandar de nuestro estimador
  # usamos la estimacion de la desviacion estandar de Chapman
  sdNhat <- sqrt(((y + 1) * (n2 + 1) * (n1 - y) * (n2 - y)) / ((y + 1) ^
                                                                 2 * (y + 2)))

  N2 <- ifelse(c <= sdNhat, 0, 1)

  n2_size <- sum(N2)

  return(n2_size)

}
```


Cálculo de pob. de peces con sim= 50
N hat Chapman - estimacion poblacion
0
500
1000
1500
2000
0
10
20
30
40
50
Número de simulaciones


Histogram of Nhat
Density
0.000
0.010
0.020
0.030
960
980
1000
1020
1040
1060
Nhat

Notemos gráficamente como el estimador de Chapman nos proporciona mejores resultados para el intervalo de confianza y para la estimamción de N.

Problema 2

Problema 3

Problema 4

Problema 5

Tenemos $V_i \sim \mathcal{U}(-1,1)$, $\forall i \in \{1,2\}$, tal que *no* generamos nuevas uniformes si caemos dentro del círculo unitario — nuestra zona de aceptación — cuya área es $(0.5)^2\pi = \pi/4 \approx 0.7853982$.

Basta ver que $X \sim \text{Geo}(\pi/4)$ para concer el número de rechazos antes de aceptar. Por lo tanto, $\mathbb{E}[X] = 1/(\pi/4) = 4/\pi \approx 1.2732395$.

Problema 6

Obtendremos la muestra de 1,000 números de este problema usando el método de la inversa generalizada para una función discreta.

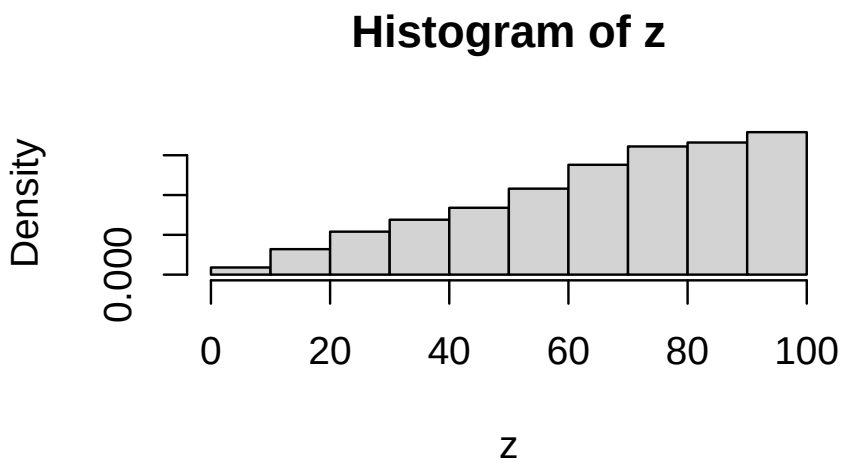
- Generamos $u \sim \mathcal{U}(0,1)$
- Tomamos $x = x_{(i)}$, donde $F(x_{(i-1)}) < u \leq F(x_{(i)})$

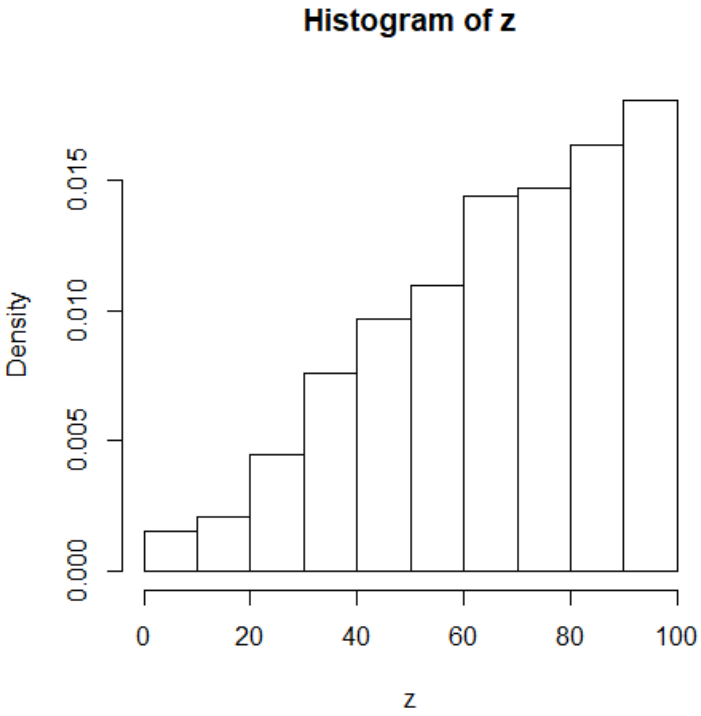
Usaremos una forma recursiva de la función acumulada de probabilidad de la distribución del problema.

$$F(x+1) = F(x) + f(x+1) = F(x) + f(x) + \frac{2}{k(k+1)}$$

Nuestro algoritmo consiste en almacenar una tabla para los valores de F y buscar el valor de u entre esos valores para así producir la muestra de la distribución dada.

```
dproblema6 <- function(k, n){
  k<-100
  n<-1000
  # crear una tabla
  f <- Fn <- numeric(k)
  # generamos constante particular de la distribución
  const <- 2/(k*(k+1))
  f[1] <- const
  Fn[1] <- f[1]
  for (i in 2:k){ # hasta k pues el dominio de la distribución llega hasta k
    f[i] <- i*const
    Fn[i] <- Fn[i-1]+f[i]
  }
  # generamos la muestra de la distribución
  u <- runif(n)
  x <- NULL
  for(i in 1:n-1) x <- append(x, sum(Fn < u[i+1]))
  return(x)
}


z <- dproblema6(100, 1000)

hist(z, prob=T)
```

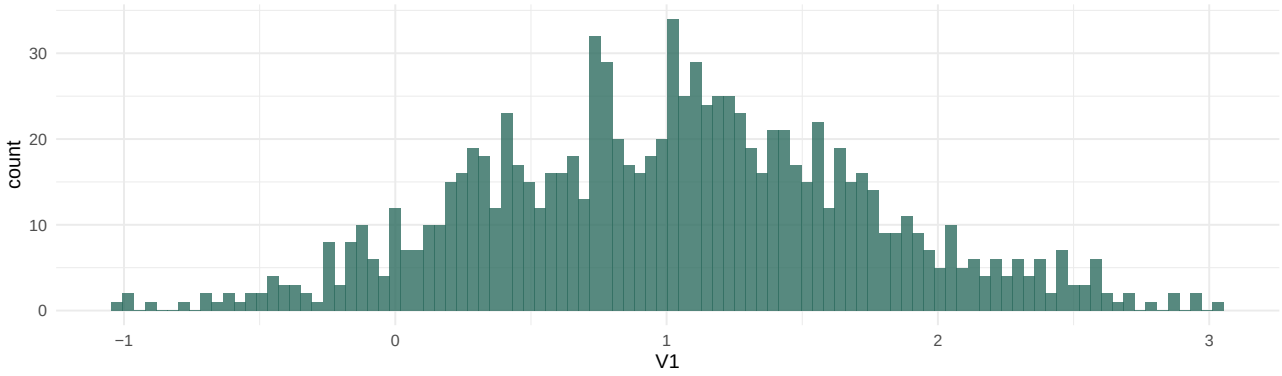

Histogram of z
Density
0.000
0
20
40
60
80
100
z

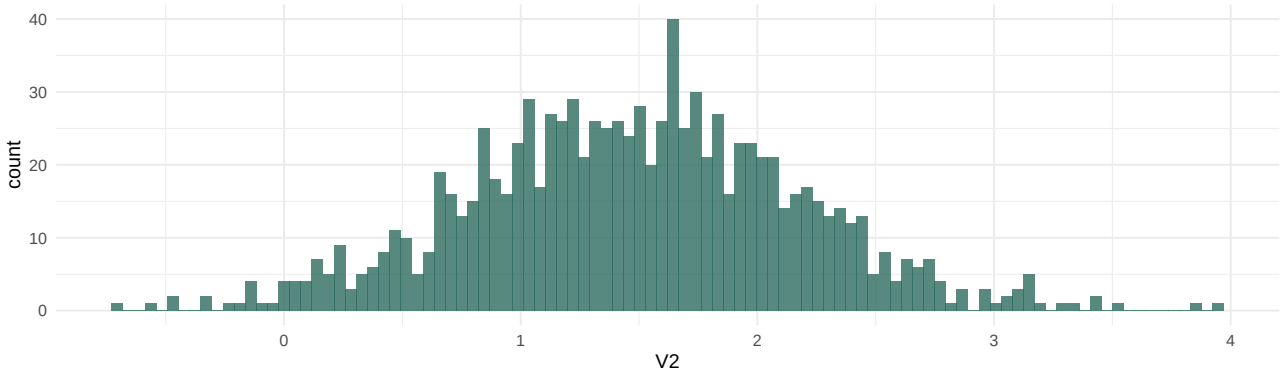

Histogram of z
Density
0.000
0.005
0.010
0.015
0
20
40
60
80
100
z

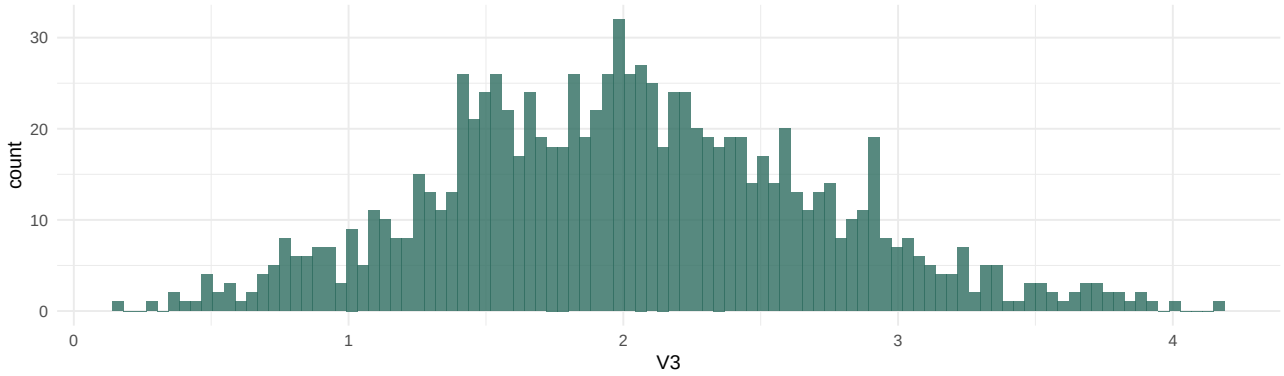
Problema 7

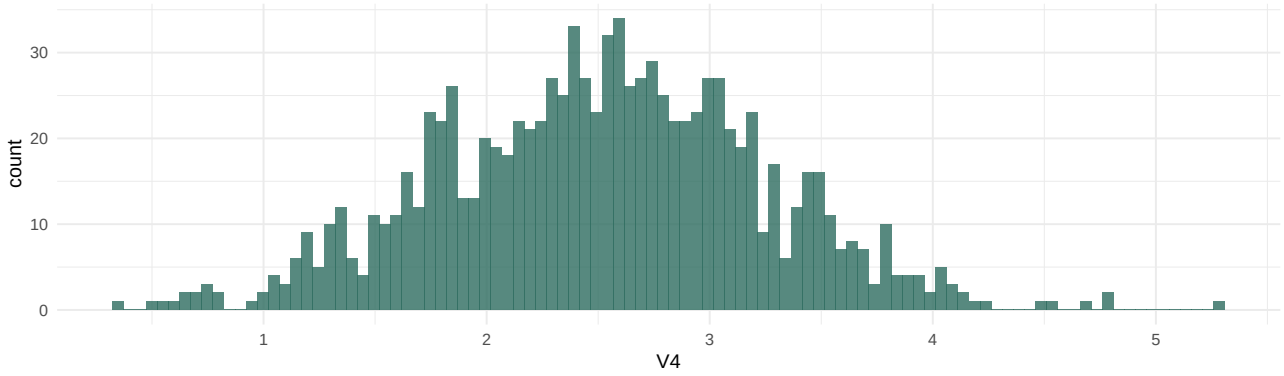
Problema 8

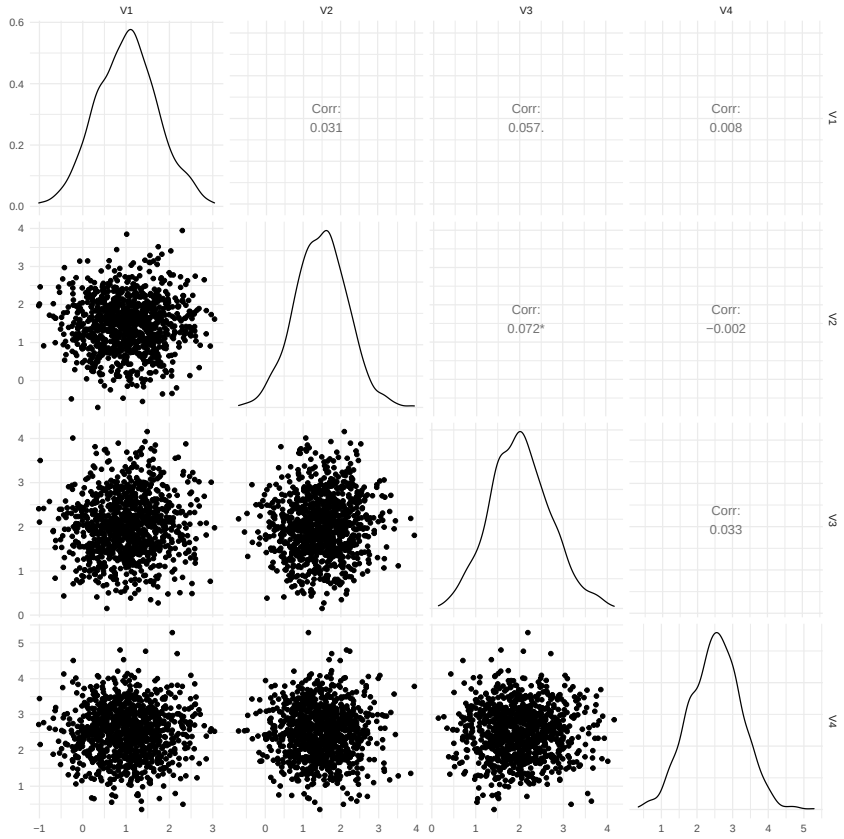
```
set.seed(1234)

mezcla <- function(n, p, mu1, mu2, s1, s2) {
    p * rmvnorm(n, mean = mu1, sigma = s1) + (1 - p) * rmvnorm(n, mean = mu2, sigma = s2)
}

n <- 1000 # observaciones
p <- 0.5  # mezcla al 50%

mu1 <- c(0, 0, 0, 0)
mu2 <- c(2, 3, 4, 5)

s1 <- s2 <- diag(4)

z <- as.data.frame(mezcla(n, p, mu1, mu2, s1, s2))
```






count
V3


count
V4


V1
V2
V3
V4
Corr:
0.031
Corr:
0.057.
Corr:
0.008
Corr:
0.072*
Corr:
−0.002
Corr:
0.033

Problema 9

Primer método

```
wishart1 <- function(k, n, mean, sigma) {
    require(mvtnorm)
    W <- list(NULL)
    for (i in 1:k) {
        X <- rmvnorm(n, mean = mean, sigma = sigma)
        W[[i]] <- t(X) %*% X
    }
    return(W)
}
```

Método de Bartlett

```
wishart2 <- function(k, n, mean, sigma) {
    W <- list(NULL)
    d <- length(mean)
    A <- matrix(0, nrow = d, ncol = d)
    for (i in 1:k) {
        A[lower.tri(matrix(0, nrow = d, ncol = d))] <-
            rnorm(d * (d + 1) / 2 - d)
        diag(A) <- sqrt(rchisq(d, n - (1:d) + 1))
        L <- chol(sigma)
        W[[i]] <- L %*% A %*% t(A) %*% t(L)
    }
    return(W)
}
```

Comparación

```
k <- 10000; n <- 10
mean <- c(0, 0, 0, 0, 0)
sigma <- diag(5)

tic('First method')
test1 <- wishart1(k, n, mean, sigma)
toc()
```

```
## First method: 2.386 sec elapsed
```

```
tic('Bartlett decomposition')
test2 <- wishart2(k, n, mean, sigma)
toc()
```

```
## Bartlett decomposition: 0.421 sec elapsed
```

Problema 10