

## *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 asintóticamente insesgado así como su estimador de varianza.

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, nuestra 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  $\text{binom}(N, k)$

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

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

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

Por otro lado, el estimador de Chapman, denotado con  $\tilde{N}$  tiene la siguiente expresión,

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

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

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

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

Ahora, para determinar el tamaño de  $n_2$  y 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

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

Donde,

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

A continuación 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 tamaño de población marcada  $n_1$  y el número de población marcada que se obtuvo en el segundo muestreo  $y$  dados.

##### PROBLEMA UNO #####

```

# 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 del estimador de máxima verosimilitud y de la varianza es
# 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.

```

```

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 <- número 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 cuántos
  # 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
  # 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),

```

```

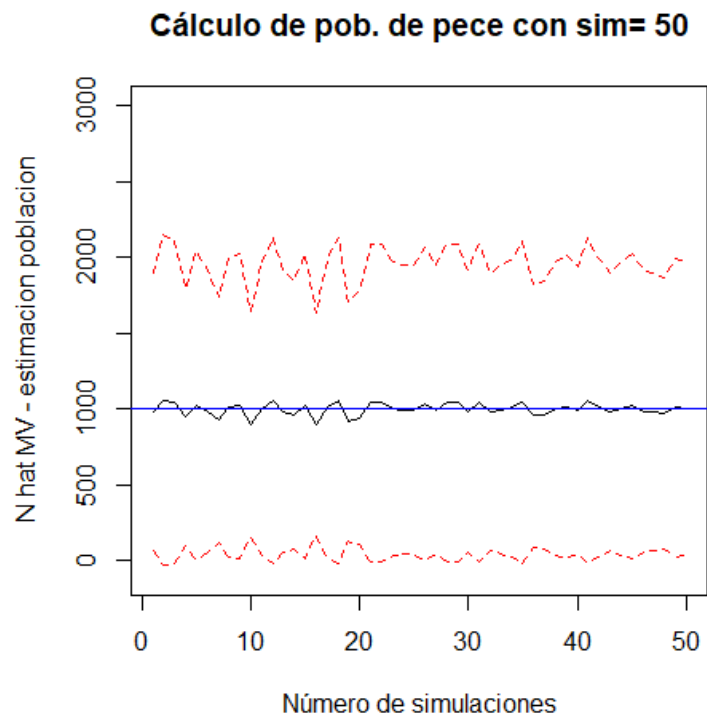
    main = paste('Cálculo de pob. de peces con sim=', S),
    xlab = 'Número de simulaciones',
    ylab = 'N hat MV - estimacion poblacion'
  )

  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)

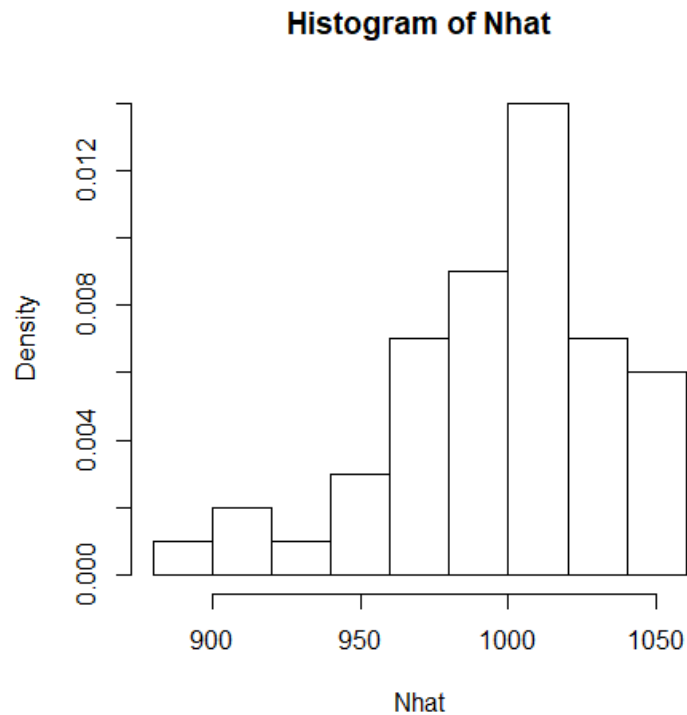
  return(grafica, h)
}

```



*# 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.*

*#la siguiente función usa el estimador de N propuesto por Chapman en 1982.  
 # Este estimador no se indefine y es asintóticamente insesgado así como  
 # su estimador de la varianza también es asintóticamente insesgado.*



```

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 <- número 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 cuántos
  # 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
  # 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)

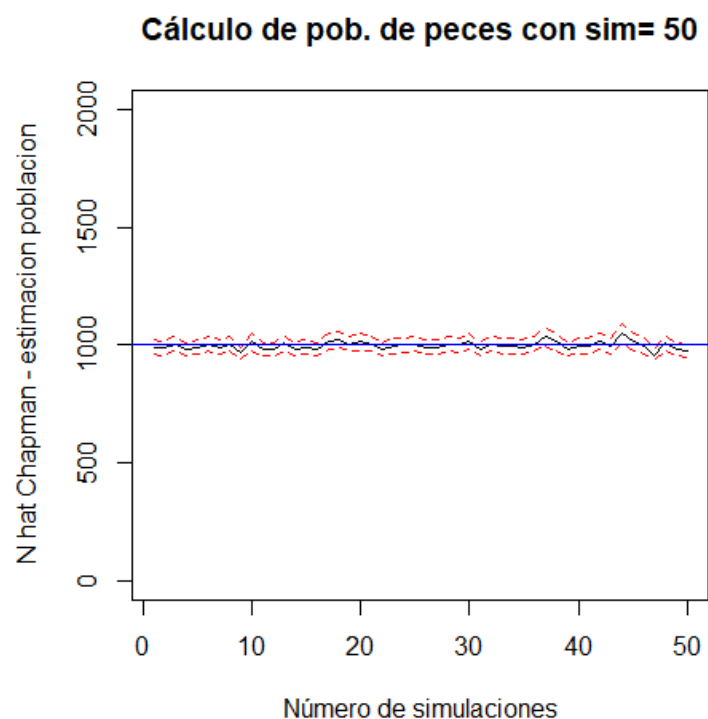
grafica <- plot(
  1:S,
  Nhat,
  type = 'l',
  ylim = c(0, 2 * N),
  main = paste('Cálculo de pob. de peces con sim=', S),
  xlab = 'Número de simulaciones',
  ylab = 'N hat Chapman - estimacion poblacion'
)

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)
}

```

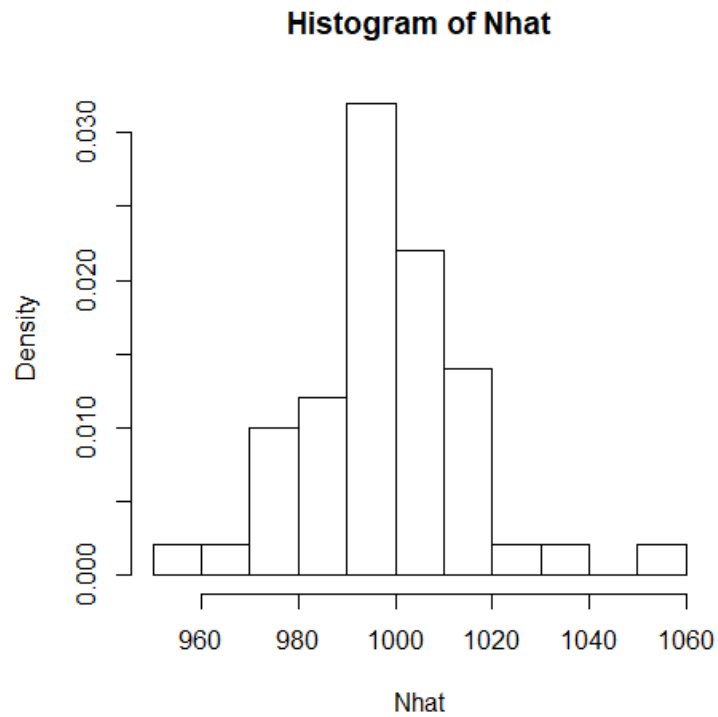


Notemos gráficamente como el estimador de Chapman nos proporciona mejores resultados para el intervalo de confianza y para la estimación de  $N$ .

```
# para calcular el tamaño de n2 para garantizar una buena aproximacion usaremos
# el estimador de Chapman. La siguiente función obtiene el tamaño de n2 usando
# la desigualdad de Chebyshev para calcular el tamaño de n2 para garantizar
# una buena aproximacion usaremos el estimador de Chapman.
```

```
tamano_n2 <- function(N, n1, y) {
  # ¿de qué tamaño debe ser n2 (segunda muestra) para alcanzar
  # una buena aproximacion de N (poblacion)?
  # ¿de qué tamaño debe ser n2 (segunda muestra) para alcanzar
  # una buena aproximacion de N (población)?
  #Dada la población N, los peces marcados n1 y el número de peces marcados
  # obtenidos de la segunda muestra aleatoria (variable) y
  # respondemos a la pregunta formulada arriba.

  # Para calcular el tamaño de n2, usaremos la desigualdad de Chebyshev
  # para alcanzar una buena aproximación a N
  # tomando un nivel de significancia de alfa =0.05
  # 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 calcular el tamaño de  $n_2$ , usaremos la desigualdad de Chebyshev  
# para alcanzar una buena aproximación a  $N$   
# tomando un nivel de significancia  $\alpha = 0.05$*

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

```
#vector de números consecutivos de 1 hasta la población,
# pues  $n_2$  puede ser una muestra aleatoria de 1 o  $N$  elementos
n2 <- c(1:N)
# vector de numeros consecutivos de 1 hasta la población pues  $n_2$ 
# 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 estándar de nuestro estimador
# (usamos la estimación de la desviación estandar de Chapman)
sdNhat <- sqrt(((y + 1) * (n2 + 1) * (n1 - y) * (n2 - y)) / ((y + 1) ^
                                                         2 * (y + 2)))

# 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)

#nos regresa el tamaño de n2 para alcanzar una buena aproximación de N
return(n2_size)
}

```

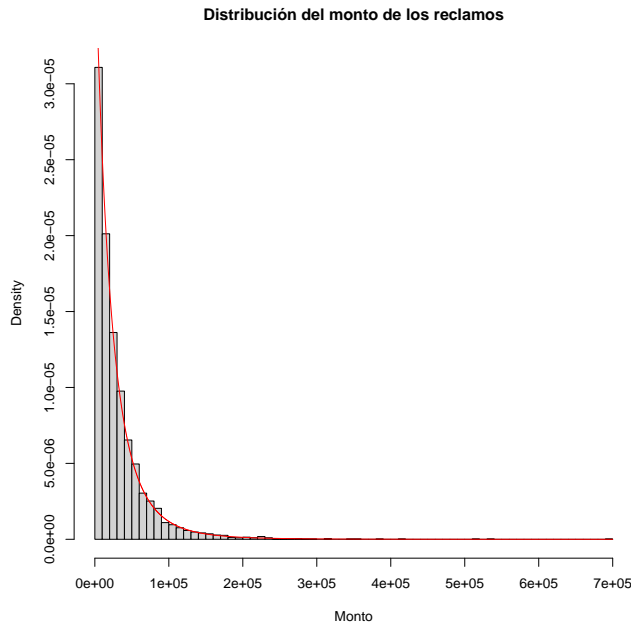
Notemos gráficamente como el estimador de Chapman nos proporciona mejores resultados para el intervalo de confianza y para la estimación de  $N$ .

### Problema 2

- a) Calcular la función de distribución  $F_X$ ,  $\mathbb{E}[X]$ , y  $\text{Var}[X]$ . Obtengan por simulación una muestra de  $X$  y su función de distribución, y comparen su estimado con la verdadera.

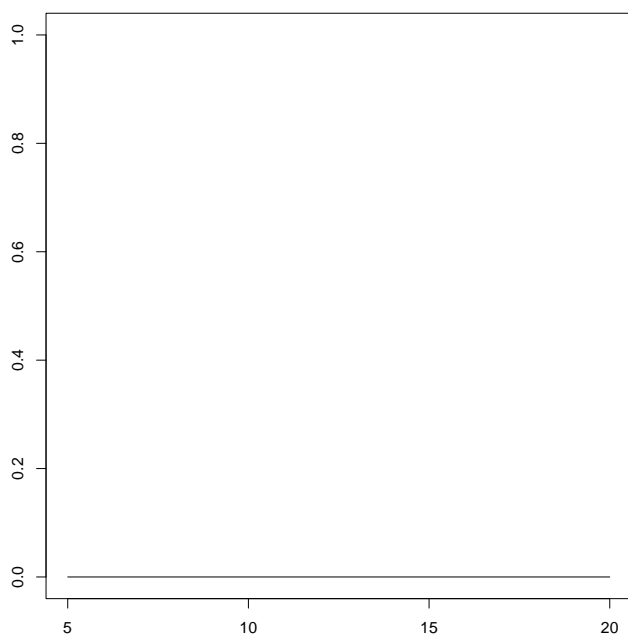
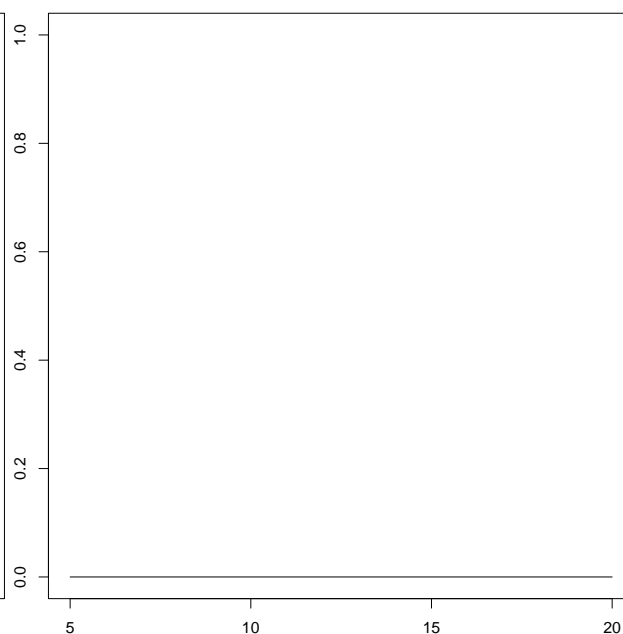
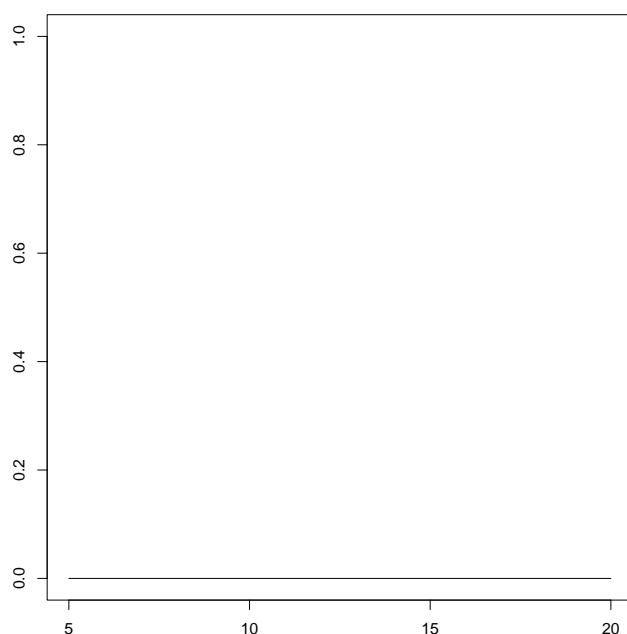
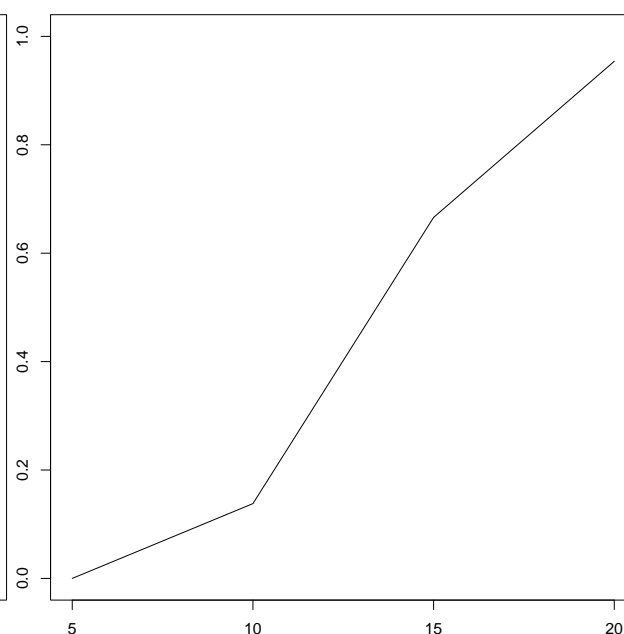
Como  $X$  sigue una distribución Pareto, la función de distribución teórica sería  $F_X = 1 - (\frac{\beta}{x+\beta})^\alpha$ , mientras que la esperanza es  $\mathbb{E}[X] = \frac{\alpha\beta}{\alpha-1}$  (para  $\alpha > 1$ ), y la varianza es  $\text{Var}[X] = \frac{\alpha\beta^2}{(\alpha-1)^2(\alpha-2)}$  (para  $\alpha > 2$ ). En nuestro caso,  $\mathbb{E}[X] = 156,250$  y  $\text{Var}[X] = 1,627,604,166.66$ .

Simulamos una muestra para  $X$  y comparamos la función de distribución con la verdadera.



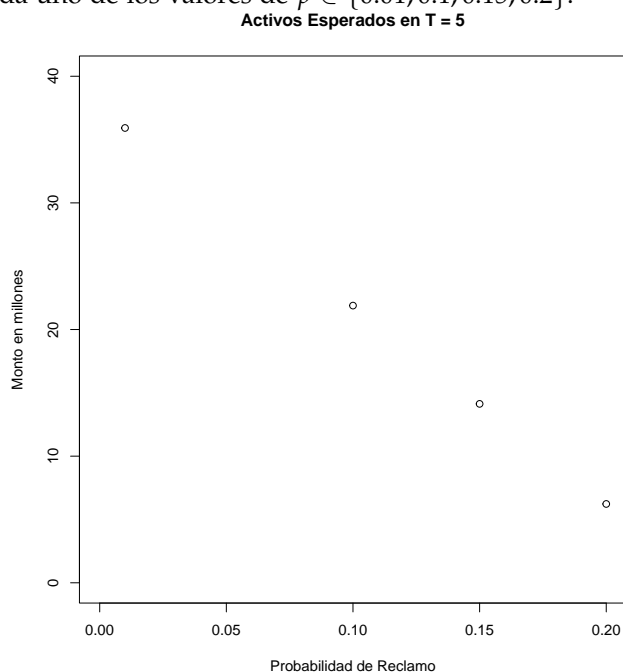
- b) Escriban una función para simular los activos de la compañía en el horizonte de tiempo  $T$  y estimen: (1) la probabilidad de que la compañía se vaya a bancarrota, y (2) Los activos esperados al final de cinco años.

Primero simulamos la probabilidad de bancarrota en el período  $T$  para cada uno de los valores de  $p \in \{0.01, 0.1, 0.15, 0.2\}$ .

Probabilidad Bancarrota con  $p = 0.01$ Probabilidad Bancarrota con  $p = 0.1$ Probabilidad Bancarrota con  $p = 0.15$ Probabilidad Bancarrota con  $p = 0.2$ 

Tal como podemos observar en las gráficas, la probabilidad de bancarrota es prácticamente cero en todo período de  $T = 1$  a  $T = 20$  cuando  $p \in \{0.01, 0.1, 0.15\}$ , pero aumenta rápidamente cuando  $p = 0.2$ , tal que la probabilidad de bancarrota es mayor que 0.95 en  $T = 20$ .

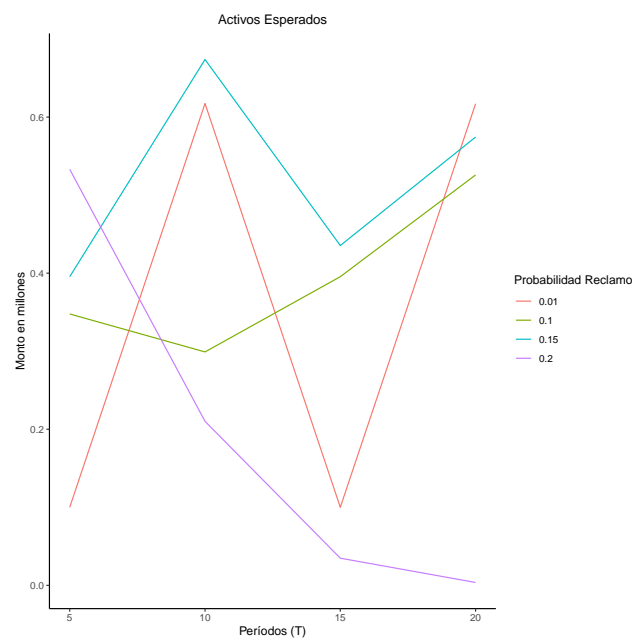
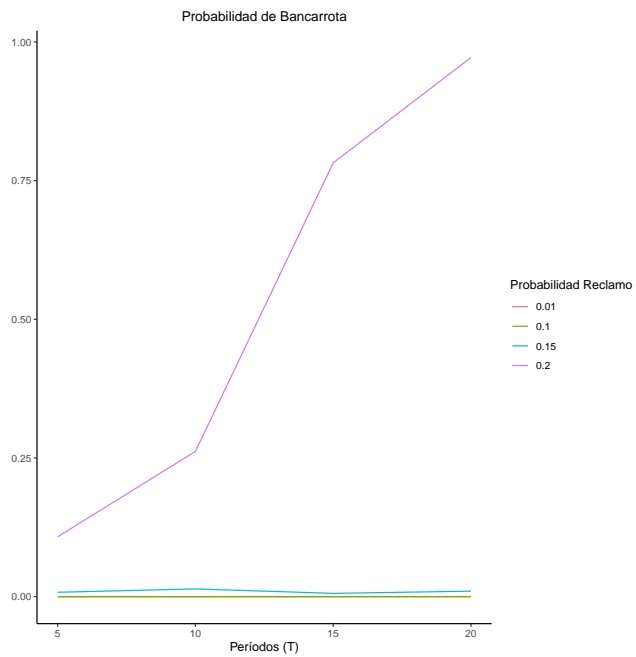
Ahora calculamos el valor esperado de los activos en  $T = 5$  para cada uno de los valores de  $p \in \{0.01, 0.1, 0.15, 0.2\}$ .

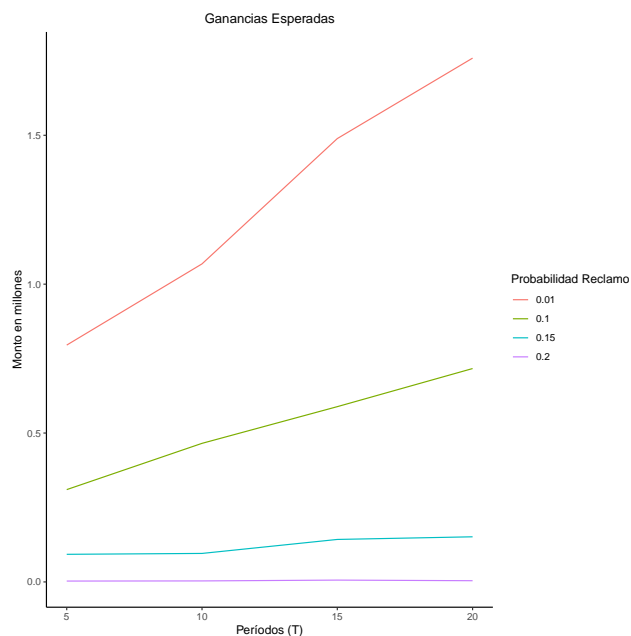


Como podemos observar en la gráfica, en  $T = 5$ , los activos esperados con las probabilidades  $p \in \{0.01, 0.1, 0.15\}$  son mayores que el capital inicial de 10 millones, mientras que en el caso  $p = 0.2$ , los activos esperados son menores que el capital inicial, lo cuál refleja la eventual bancarrota que ocurre en este caso.

- c) Toma de ganancias. Supongan ahora que la compañía toma ganancias al final de cada año. Usando este nuevo esquema, estimar: (1) la probabilidad de irse a la quiebra, (2) los activos esperados después de  $T$  años, y (3) Las ganancias totales esperadas después de  $T$  años.

Calculamos la probabilidad de bancarrota para cada valor de  $p$  con el nuevo esquema de toma de ganancias, el valor esperado de los activos y las ganancias para cada período  $T = 5, 10, 15, 20$ .





Como podemos observar en la gráficas, se mantiene que la probabilidad de bancarrota crece rápidamente para  $p = 0.2$ , mientras que es cero para  $p = 0.1$  y  $p = 0.01$ , y ligeramente positiva para  $p = 0.15$ . Asimismo, en la gráfica de los activos esperados, podemos ver que hay una clara tendencia decreciente para los activos cuando  $p = 0.2$ , que refleja la eventual alta probabilidad de bancarrota, mientras que los demás valor de  $p$  no siguen una tendencia fija. Si añadiéramos más puntos (períodos), las líneas que veríamos para  $p = 0.1$  y  $0.01$  serían ruido alrededor de un cierto nivel  $\bar{M}$ . Por último, para las ganancias esperadas es claro que la tendencia de las ganancias es creciente, y es mayor conforme la probabilidad de reclamo es menor.

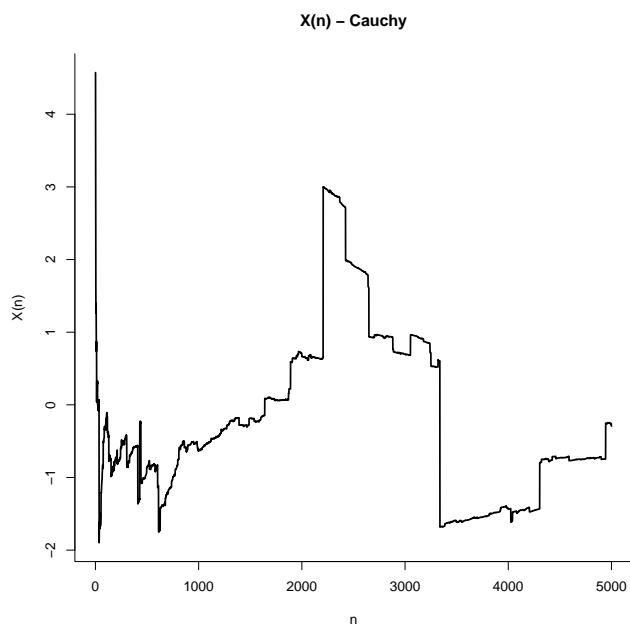
### Problema 3

#### Cauchy

Para simular la distribución Cauchy con función de densidad  $f(x) = \frac{1}{\pi\beta[1+(\frac{x-\gamma}{\beta})^2]}$ , usamos el hecho que su función de distribución  $F(x) = \frac{1}{\pi} \arctan(\frac{x-\gamma}{\beta}) + \frac{1}{2}$  es una función invertible. Usando el teorema de la función inversa, sea  $U \sim \text{Unif}[0,1]$ , entonces  $X = \beta \tan(\pi(U - \frac{1}{2})) + \gamma$  sigue una distribución Cauchy con parámetro de escala  $\beta > 0$  y de locación  $\gamma$ .

```
rcauchy <- function(n, param_gamma = 0, param_beta = 1){
  unif <- runif(n)
  cauchy <- param_beta*tan(pi*(unif-0.5))+param_gamma
  return(cauchy)
}
```

Ahora, tomamos una muestra aleatoria de 5000 observaciones y calculamos  $\bar{X}(n) = \sum_{i=1}^n X_i/n$  para  $n = 50, 100, 150, \dots, 5000$ .



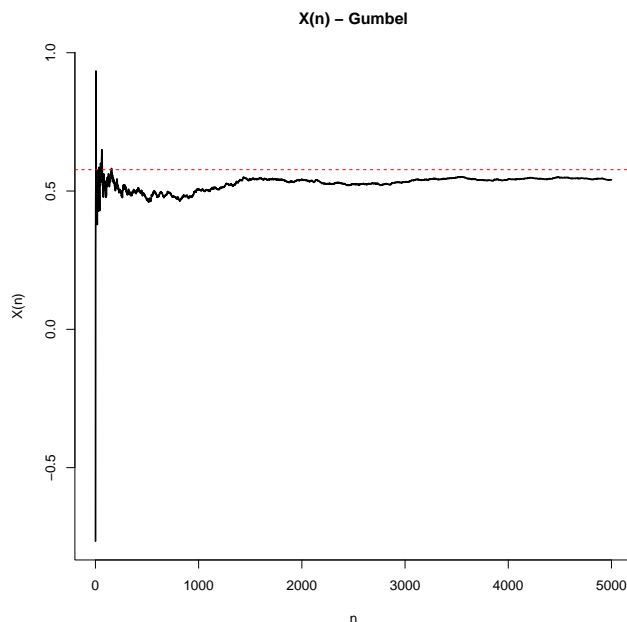
Podemos notar de la gráfica que  $\bar{X}(n)$  no converge a ningún valor, hecho consistente con que la distribución Cauchy no tiene media definida para ningún valor de sus parámetros.

### Gumbel

Para simular la distribución Gumbel con función de densidad  $f(x) = \frac{1}{\beta} \exp(-e^{-(x-\gamma)/\beta} - \frac{x-\gamma}{\beta})$ , usamos el hecho que su función de distribución  $F(x) = \exp(-e^{-(x-\gamma)/\beta})$  es una función invertible. Usando el teorema de la función inversa, sea  $U \sim \text{Unif}[0, 1]$ , entonces  $X = -\beta \ln(-\ln(U)) + \gamma$  sigue una distribución Gumbel con parámetro de escala  $\beta > 0$  y de locación  $\gamma$ .

```
rgumbel <- function(n, param_gamma = 0, param_beta = 1){
  unif <- runif(n)
  gumbel <- -param_beta*log(-log(unif)) + param_gamma
  return(gumbel)
}
```

Ahora, tomamos una muestra aleatoria de 5000 observaciones y calculamos  $\bar{X}(n) = \sum_{i=1}^n X_i/n$  para  $n = 50, 100, 150, \dots, 5000$ .



La media de la distribución Gumbel con  $\gamma = 0$  y  $\beta = 1$  es la constante de Euler-Mascheroni, que vale aproximadamente 0.5772157. Podemos ver en la gráfica que conforme aumenta  $n$ ,  $\bar{X}(n)$  converge al valor de la constante.



## Logística

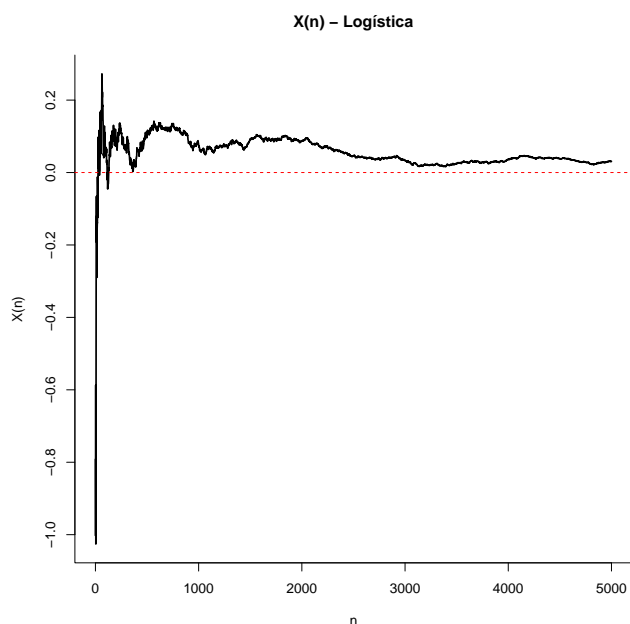
Para simular la distribución Logística con función de densidad

$f(x) = \frac{(1/\beta)e^{-(x-\gamma)/\beta}}{(1+e^{-(x-\gamma)/\beta})^2}$ , usamos el hecho que su función de distribución

$F(x) = \frac{1}{(1+e^{-(x-\gamma)/\beta})}$  es una función invertible. Usando el teorema de la función inversa, sea  $U \sim Unif[0,1]$ , entonces  $X = -\beta \ln\left(\frac{1-y}{y}\right) + \gamma$  sigue una distribución Logística con parámetro de escala  $\beta > 0$  y de locación  $\gamma$ .

```
rlogistic <- function(n, param_gamma = 0, param_beta = 1){
  unif <- runif(n)
  logistic <- -param_beta*log((1-unif)/unif) + param_gamma
  return(logistic)
}
```

Ahora, tomamos una muestra aleatoria de 5000 observaciones y calculamos  $\bar{X}(n) = \sum_{i=1}^n X_i/n$  para  $n = 50, 100, 150, \dots, 5000$ .



La media de la distribución Logística es  $\gamma = 0$ . Podemos ver en la gráfica que conforme aumenta  $n$ ,  $\bar{X}(n)$  converge precisamente a cero.

### Pareto

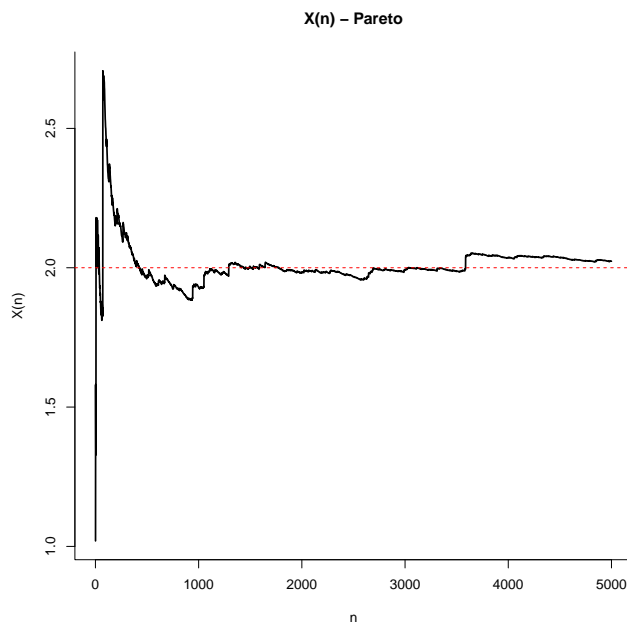
Para simular la distribución Pareto con función de densidad  $f(x) = \frac{\alpha_2 c^{\alpha_2}}{x^{\alpha_2+1}}$ , usamos el hecho que su función de distribución  $F(x) = 1 - (\frac{c}{x})^{\alpha_2}$  es una función invertible. Usando el teorema de la función inversa, sea  $U \sim Unif[0,1]$ , entonces  $X = \frac{c}{(1-U)^{1/\alpha_2}}$  o  $X = \frac{c}{(U)^{1/\alpha_2}}$  sigue una distribución Pareto con parámetro de escala  $c > 0$  y de forma  $\alpha_2 > 0$ .

```

rparet <- function(n, param_c = 1, param_alpha = 2){
  unif <- runif(n)
  pareto <- param_c/(unif)^(1/param_alpha)
  return(pareto)
}

```

Ahora, tomamos una muestra aleatoria de 5000 observaciones y calculamos  $\bar{X}(n) = \sum_{i=1}^n X_i/n$  para  $n = 50, 100, 150, \dots, 5000$ .



La media de la distribución Pareto es  $\frac{\alpha_2 c}{\alpha_2 - 1} = 2$ . Podemos ver en la gráfica que conforme aumenta  $n$ ,  $\bar{X}(n)$  converge a 2.

### Vasicek

Para simular la distribución Vasicek con función de densidad  $f(x) = \sqrt{\frac{1-\rho}{\rho}} \exp\left(\frac{1}{2}\left\{\Phi^{-1}(x)^2 - \left(\frac{\sqrt{1-\rho}\Phi^{-1}(\rho)}{\sqrt{\rho}}\right)^2\right\}\right)$ , usamos el hecho que

$X = \Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}Z}{\sqrt{1-\rho}}\right)$  sigue una distribución Vasicek con parámetros  $p$  y  $\rho$ , donde  $Z$  sigue una distribución normal estándar y  $\Phi$  es su función de distribución.

```
rvasicek <- function(n, param_rho, param_p){  
  normal <- rnorm(n)  
  vasicek <- pnorm((qnorm(p)-sqrt(rho)*normal)/sqrt(1-param_rho))  
  return(vasicek)  
}
```

## Problema 4

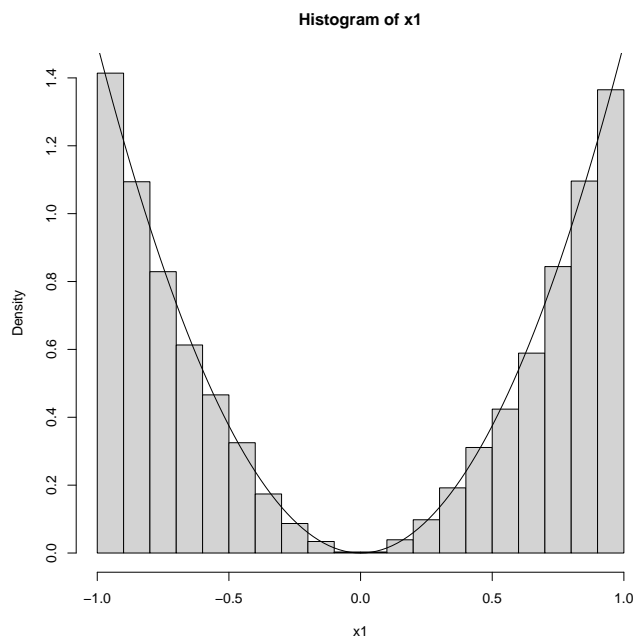
### Inciso (a)

Usamos primero el método de transformación con distribución inversa:

```
f1<-function(x){return(3/2*x^2)}
raiz<-function(x){
  sign(x)*abs(x)^(1/3)
}

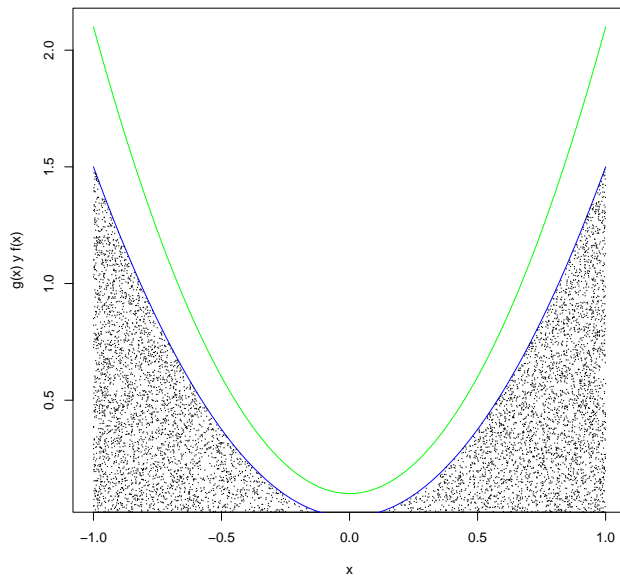
n<-10000
x1<-numeric(n)

for (i in 1:n){x1[i]=(raiz(2*runif(1)-1))}
```



También, logramos emplear el método de aceptación/rechazo para aproximar la densidad:

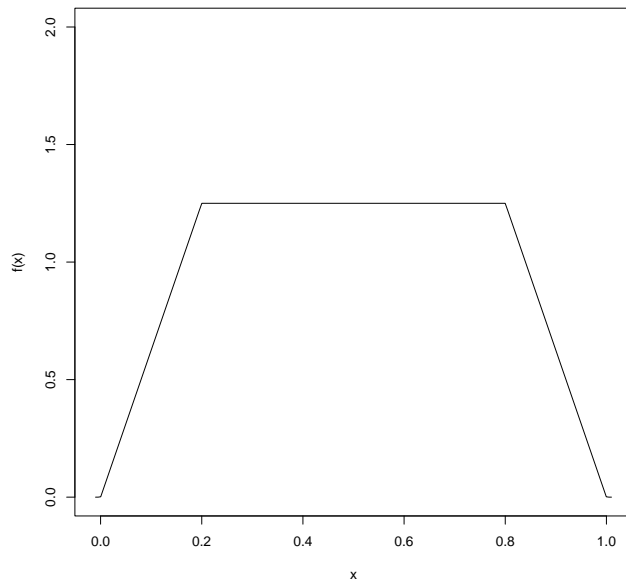
```
n<-10000
u1<-runif(n)
u2<-runif(n)
y1<-raiz(2*u2-1)
aceptados<- u1<=f1(y1)/(0.1+2*y1^2)
pts<-y1[aceptados]
```



### Inciso (b)

Para este inciso, utilizamos únicamente el método de transformación, que presentaba más dificultades al momento de calcular la solución de la distribución inversa.

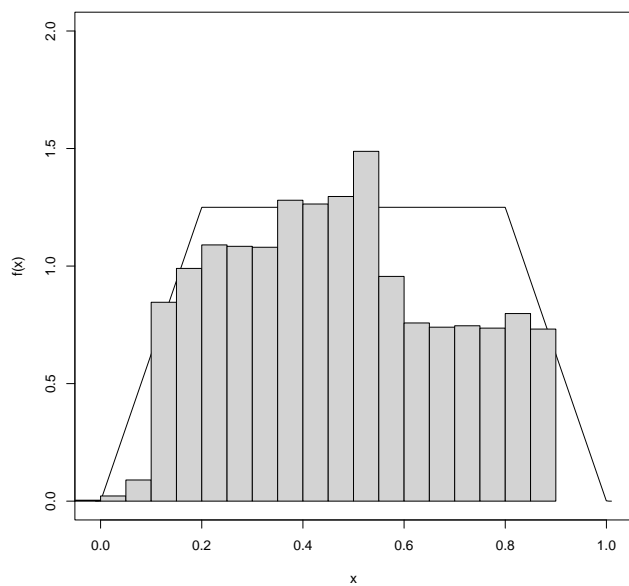
```
f2<-function(w,a){
  if (w>=0 && w<=a){return(w*(1/(a*(1-a))))}
  else{if(w>=a && w<=1-a){return(1/(1-a))}
    else{if(w>=1-a && w<=1){return((1-w)/(a*(1-a)))}
      else {return(0)}}}
}
a=0.2
len=200
x2 <- seq(-0.01,1.01,length=len)
y2<-numeric(len)
for (i in 1:len){
  y2[i]<-f2(x2[i],a)
}
plot(x2,y2,type="l",ylim=c(0,2),xlab="x",ylab="f(x)")
```



```

indicadora<-function(x,k1,k2){
  ifelse(x>=k1 & x<=k2,1,0)
}
fx<-function(x,r){
  indicadora(x,0,r)*sqrt(2*r*(1-r)*runif(1))+
  indicadora(x,r,1-r)*(runif(1)*(1-r)+r/2)+
  indicadora(x,1-r,1)*(1-sqrt(1-(runif(1)-(2+2*r-5*r^2)/(2*r*(1-r)))*2*r*(1-r)))
}
x<-seq(-0.0001,1.0001,length=10000)
y<-numeric(length(x))
for(i in 1:length(x)){
  y[i]<-fx(x[i],0.2)
}

```



*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 conocer 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)

```



*Problema 7*

```

n<-100000
p<-0.4
blli<-function(x,p){p^x*(1-p)^(1-x)}
tam<-100
muestra<-sample(1:tam,n,replace=T)
binomial1<-numeric(n)
for(i in 1:n){
  numexitos<-c(rep(1,muestra[i]),rep(0,tam-muestra[i]))
  binomial1[i]<-prod(blli(numexitos,p))*factorial(tam)/(factorial(muestra[i])*factorial(tam-muestra[i]))
}

binomial2<-dbinom(muestra,tam,p)

```

Como podemos ver, las entradas de ambos vectores (uno simulado y otro calculado con la función) son iguales sobre la muestra que extraímos:

```

binomial1[1:5]

## [1] 2.562323e-07 6.120845e-14 1.437301e-19 2.562714e-03 2.562714e-03

binomial2[1:5]

## [1] 2.562323e-07 6.120845e-14 1.437301e-19 2.562714e-03 2.562714e-03

```

Al evaluar el tiempo de respuesta del programa para simular una binomial mediante convolución, o mediante la función “rnorm”, obtuvimos que es más rápido simular la distribución con la función instalada en R.

```

system.time(
for(i in 1:n){
  numexitos<-c(rep(1,muestra[i]),rep(0,tam-muestra[i]))
  binomial1[i]<-prod(blli(numexitos,p))*factorial(tam)/(factorial(muestra[i])*factorial(tam-muestra[i]))
})

## user system elapsed
## 1.408 0.000 1.409

system.time(binomial2<-dbinom(muestra,tam,p))

## user system elapsed
## 0.02 0.00 0.02

```

### Problema 8

```
set.seed(1234)

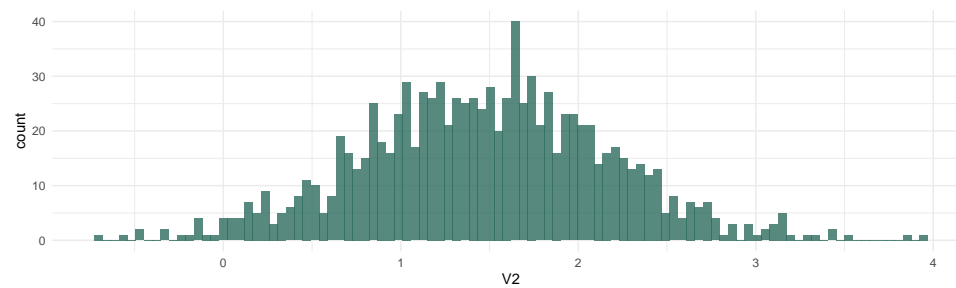
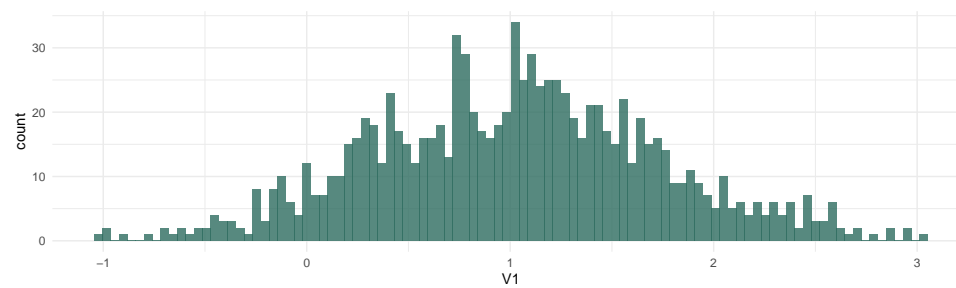
mezcla <- function(n, p, mu1, mu2, s1, s2) {
  p * rmvnorm(n, mean = mu1, sigma = s1) + (1 - p) * rmvnorm(n, mean = mu2, sigma = s2)
}

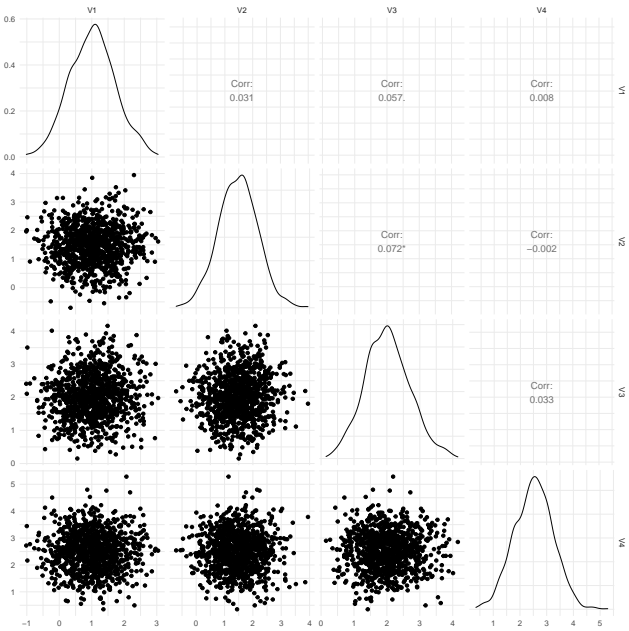
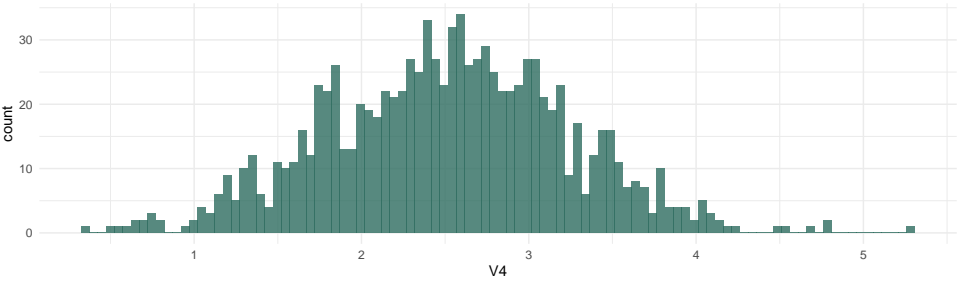
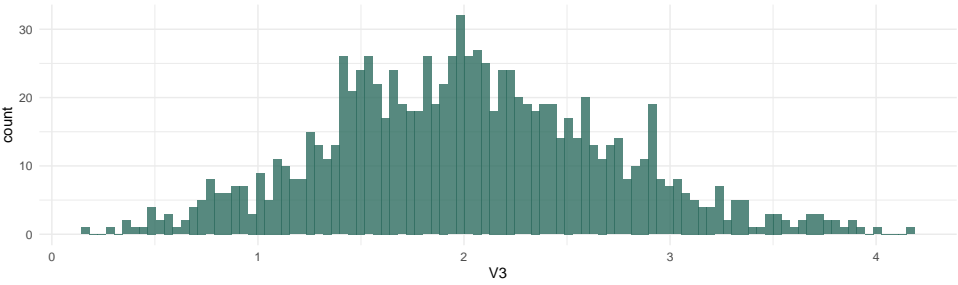
n <- 1000 # observaciones
p <- 0.5 # mezcla al 50%

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))
```





## 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: 4.474 sec elapsed

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

## Bartlett decomposition: 0.66 sec elapsed
```

### Problema 10

Después de generar las 500 observaciones de la distribución normal bivariada con su respectiva matriz de covarianza, se tomó el exponencial de la matriz. Al exponenciar una variable aleatoria distribuida normal, obtenemos una distribución lognormal. Lo que podemos observar en la gráfica, es que existen correlaciones que no pueden darse en una distribución lognormal, que terminan siendo muy cercanas a 0.

```
library(mvtnorm)
SIG2<-seq(from=0.1,to=20,length.out = 100)
RHO<-seq(from=-1,to=1,length.out=21)
matriz<-matrix(numeric(),ncol=2)
pts<-data.frame()
for (i in 1:length(SIG2)){
  for(j in 1:length(RHO)){
    cvz<-matrix(c(1,sqrt(SIG2[i])*RHO[j],sqrt(SIG2[i])*RHO[j],SIG2[i]),ncol=2)
    z<-rmvnorm(500,mean=c(0,0),sigma=cvz)
    expz<-exp(z)
    p<-cor(expz[,1],expz[,2])
    matriz<-rbind(matriz,c(SIG2[i],p))
  }
}
```

