Prácticas de Inferencia Estadística II

Grado en Estadística y doble Grado en Ingeniería Informática y Estadística (INdat),
Universidad de Valladolid

Yolanda Larriba y Bonifacio Salvador

Índice general

Prólogo

Este libro contiene material de apoyo para las lecciones prácticas de la asignatura Inferencia de Estadística II (https://apps.stic.uva.es/guias_docentes/uploads/2023/549/47091/1/Documento.pdf). Esta asignatura se imparte en el Grado en Estadística (http://www.eio.uva.es/nuestra-docencia/) y en el doble Grado en Ingeniería Informática y Estadística (https://www.inf.uva.es/indat/) de la Universidad de Valladolid (https://www.uva.es/export/sites/uva/).

Para facilitar su uso en la práctica, y de acuerdo con el software estadístico empleado en la asignatura, este libro ha sido escrito en R-Markdown empleando el paquete `bookdown`. Así mismo, con el propósito de garantizar el mantenimiento y actualización de los contenidos, también está disponible en el repositorio https://github.com/yolandalago/IEII de Github, desde donde se mantentrán actualizados las versiones y contenidos.

La asignatura de Inferencia Estadística II (6 ECTS) está diseñada para dedicar aproximadamente un 25% del total de ECTS a sesiones prácticas. En general, se intenta realizar una sesión práctica de dos horas cada cinco horas de teoría. Cada una de las prácticas incluidas en este libro se corresponde con una de esas sesiones prácticas y está pensada para su realización en una sesión de dos horas. Las prácticas están guionizas incluyendo notas sobre teoría, ejercicios resueltos, en ocasiones tomados de las listas de ejercicios disponibles para los temas de teoría, y simulaciones. La colección de prácticas propuestas cubre los bloques estudiados previamente en las sesiones de teoría. El horario se ajusta a las restricciones de tiempo de cada curso académico, lo que inevitablemente hará que los contenidos se tengan que reajustar curso a curso. Sin embargo, los guiones de estas prácticas y los ejercicios propuestos están pensados para ofrecer una visión global desde el punto de vista aplicado de los resultados estudiados en las clases de teoría. Este libro está estructurado de la siguiente manera:

- Práctica 1: Inferencias de Wald e inferencias basadas en el Test Razón de Verosimilitudes (RV). Caso uiniparamétrico.
- Práctica 2: Inferencias de Wald. Caso multiparamétrico.
- Práctica 3: Inferencias basadas en el estadístico RV. Caso multiparamétrico.
- Práctica 4: Algoritmos de maximización de la verosmilitud.
- Práctica 5: Bootstrap.
- Práctica 6: Tests de bondad de ajuste.
- Práctica 7: Test basados en rangos. CAMBIAR

La notación y terminología empleada en este libro se despresde de lo expuesto en las clases y materiales teóricos de la asignatura. Las referencias utilizadas para la elaboración de este libro coinciden con las propuestas en la guía docente de la asignatura, y pueden consultarse tamnién al final de este libro.

 Cualquier comentario o errata detectado será bienvenido, para ello basta con enviar un correo electrónico a yolanda.larriba@uva.es.

1 Práctica 1: Inferencias de Wald e inferencias basadas en el Test Razón de Verosimilitudes (RV). Caso uniparamétrico

1.1 Inferencias de Wald: IC y CH

Sean $X_1, \dots, X_n$ v.a.i.i.d. $P_\theta, \theta \in \Theta \subset \mathbb{R}; f(x, \theta)$.

Como hemos visto, en situaciones regulares el estimador máximo verosimil (EMV) ($\hat{\theta}$) para θ es consistente asintóticamente normal (CAN) y asintóticamente eficiente (AE):

$$\sqrt{n}(\theta - \hat{\theta}) \xrightarrow{\mathcal{L}} N(0, V^2(\theta)), \quad (Eq.1) \tag{1.1}$$

donde $V^2(\theta) = \frac{1}{I_1(\theta)}$.

Las inferencias acerca de θ basadas en esta distribución asintótica se conocen como inferencias de Wald. Por un lado, para obtener un intervalo de confianza (IC) para θ, se necesita estimar $I_1(\theta)$. Tal y como vimos la información de Fisher (IF) observada, que definimos como $\widehat{I_1(\theta)} = -\frac{\partial^2}{\partial \theta^2} \log f(X, \theta)|_{\theta=\hat{\theta}}$, nos permite estimar esta cantidad, y concretamente:

$$\widehat{V^2(\theta)} = \hat{V}^2(\theta) = \frac{1}{\widehat{I_1(\theta)}},$$

es decir, $V^2(\theta)$ se estima como la inversa de la IF observada, y de donde también se sigue que:

$$\sqrt{n}(\theta - \hat{\theta}) \xrightarrow{\mathcal{L}} N(0, \widehat{V^2(\theta)}), \text{ o equivalentemente, } \hat{\theta} \cong N(0, \frac{\widehat{V^2(\theta)}}{n})$$

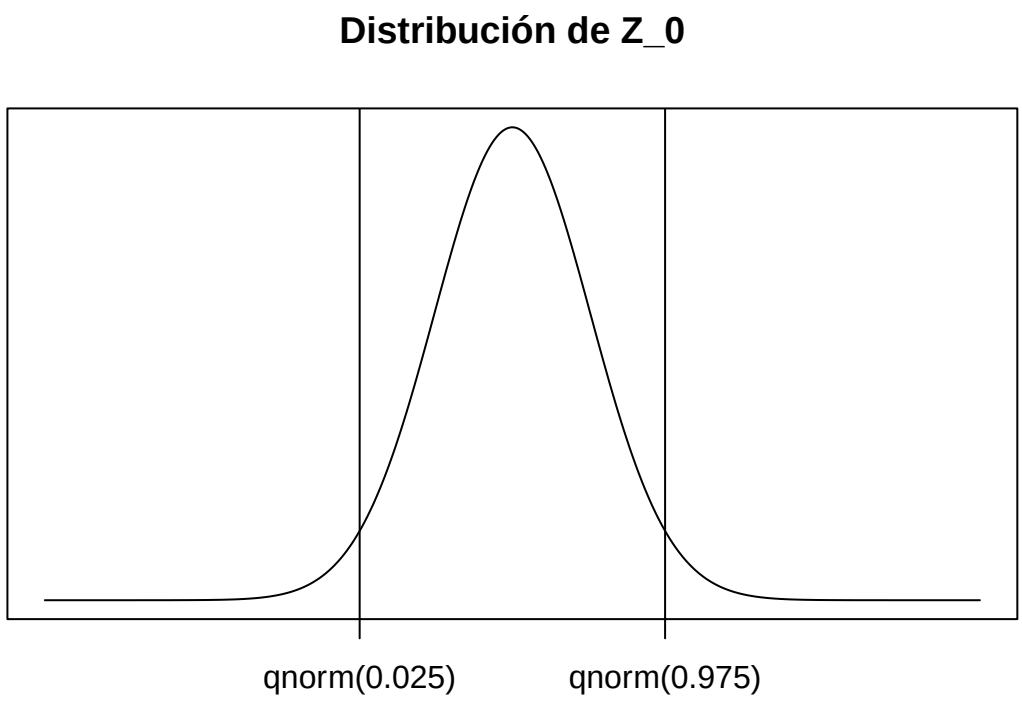
Por tanto, $Z_0 = \frac{\sqrt{n}|(\hat{\theta}-\theta)|}{\sqrt{\hat{V}^2(\theta)}} \cong N(0,1)$. Así, fijado α, un IC un Wald $(1-\alpha)\%$ para θ queda definido por:

$$P_\theta(\frac{\sqrt{n}|(\hat{\theta} - \theta)|}{\sqrt{\hat{V}^2(\theta)}} \leq \lambda_{1-\alpha/2}) \simeq 1 - \alpha,$$

donde $\lambda_{1-\alpha/2}$ es el cuantil $1 - \alpha/2$ de una normal estandar ($qnorm(1 - \alpha/2)$ en R). O equivalentemente, despejando θ en la expresión anterior se obtiene el IC de Wald pedido:

$$P_\theta(\theta \in [\hat{\theta} \pm qnorm(1 - \alpha/2)\frac{\sqrt{\hat{V}^2(\theta)}}{\sqrt{n}}]) \simeq 1 - \alpha$$

Gráficamente:



El resultado anterior de la Eq. 1 también es clave para resolver contrastes de hipótesis (CH) para θ:

$$H_0 : \theta = \theta_0 \quad vs \quad H_1 : \theta \neq \theta_0. \tag{1.2}$$

.

Tal y como vimos el estadístico de Wald:

$$Q_W = \frac{n(\hat{\theta} - \theta_0)^2}{\widehat{V^2(\theta)}} \simeq n(\hat{\theta} - \theta_0)^2 \widehat{I_1(\theta)} \overset{H_0}{\cong} \chi^2_1$$

por tratarse del cuadrado de una distribución normal estándar. Esta distribución nos permitirá calcular el p-valor como $P_{\theta_0}(Q_W > q_{W,obs})$.

1.2 Inferencias basadas en el estadístico RV: IC y CH

Sean $X_1, \dots, X_n$ v.a.i.i.d. $P_\theta, \theta \in \Theta \subset \mathbb{R}; f(x, \theta)$. Dado un nivel de significación α se quiere resolver el CH para θ:

$$H_0 : \theta = \theta_0 \quad vs \quad H_1 : \theta \neq \theta_0. \tag{1.3}$$

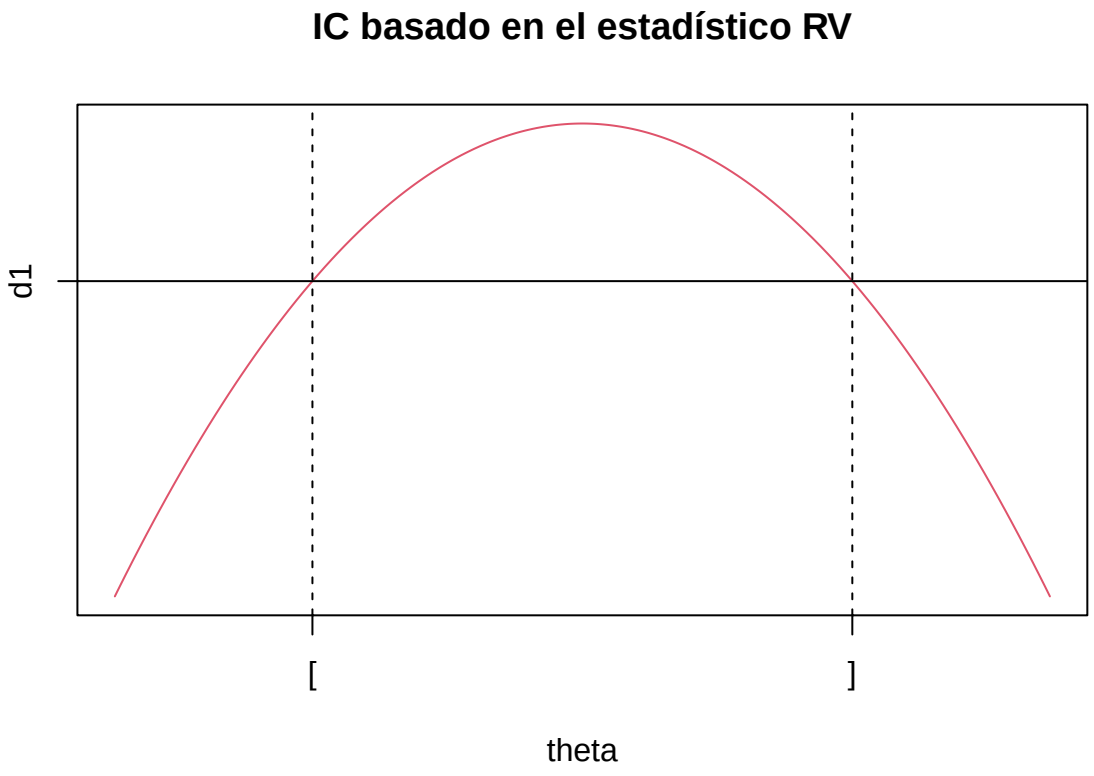
Según lo estudiado, el estadístico de la RV se define como:

$$Q_L(X) = 2[\log L(\hat{\theta}, X) - \log L(\theta_0, X)]$$

y es asintóticamente equivalente a Q_w. Luego el test de RV de nivel α para resolver CH para θ tiene una región crítica de tamaño α aproximadamente que viene dada por $(Q_L(X) > \chi^2_{1,1-\alpha}) \simeq (Q_L(X) > qchisq(1, 1-\alpha))$, donde $\chi^2_{1-\alpha,1}$ es el cuantil $1-\alpha$ de una χ^2_1. El p-valor del test vendrá dado por $P_{\theta_0}(Q_L > q_{L,obs})$.

De la relación entre test de nivel α e IC $(1-\alpha)$, se tiene que el IC basado en el estadístico de la RV (ICRV$(1-\alpha)$) se corresponde con el conjunto de valores de θ para los que el test de la RV de nivel α no rechaza la H_0 es decir:

$$ICRV(1-\alpha) = \{\theta_0 : Q_L(X) \leq qchisq(1-\alpha,1)\} \tag{1.4}$$
$$= \{\theta_0 : 2[\log L(\hat{\theta}, X) - \log L(\theta_0, X)] \leq qchisq(1-\alpha,1)\} \tag{1.5}$$
$$= \{\theta_0 : \log L(\theta_0, X) \geq \log L(\hat{\theta}, X) - \frac{qchisq(1-\alpha,1)}{2}\} \tag{1.6}$$
$$= \{\theta_0 : \log L(\theta_0, X) \geq d_1\} \tag{1.7}$$



1.3 Ejercicio 1: Modelo de Poisson

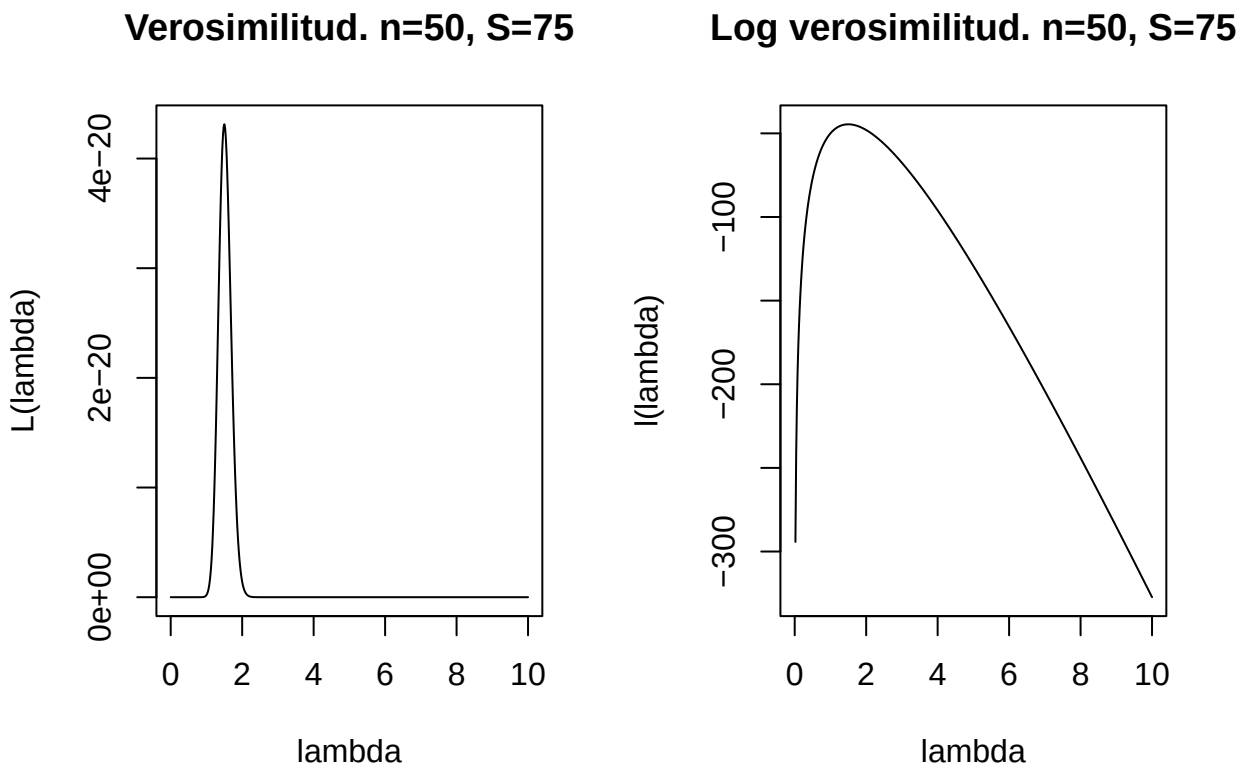
Considere $X_1, \ldots, X_{50}$ m.a.s. de $\mathcal{P}(\lambda)$, $f(x,\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}, x = 0,1,2,\ldots$. Para los datos observados se tiene $S = \sum_{i=1}^{50} P_\lambda(X_i = x_i) = 75$.

a. Obtener y representar la función de verosimilitud y log verosimilitud.

Verosimilitud: $L(\lambda, X_1, \ldots, X_n) \propto e^{-n\lambda}\frac{\lambda^{\sum_{i=1}^n X_i}}{\prod_{i=1}^n X_i!}$

Log verosimilitud: $\log L(\lambda, X_1, \ldots, X_n) = -n\lambda + \sum_{i=1}^n X_i \log\lambda$

```r
#Datos
n<-50
S<-75
#Verosimilitud X1,...x_n ~ P(lambda)
#Todo aquello que no depende de lambda paodemos suprimirlo
v<-function(lambda){
  return(exp(-n*lambda)*lambda^S)
}
#Log-verosimilitud
lv<-function(lambda){
  return(-n*lambda+S*log(lambda))
}
par(mfrow=c(1,2))
plot(seq(0,10,length.out=500),v(seq(0,10,length.out=500)),xlab="lambda",ylab="L(lambda)",
     main="Verosimilitud. n=50, S=75",type="l")
plot(seq(0,10,length.out=500),lv(seq(0,10,length.out=500)),xlab="lambda",ylab="l(lambda)",
     main="Log verosimilitud. n=50, S=75",type="l")
```

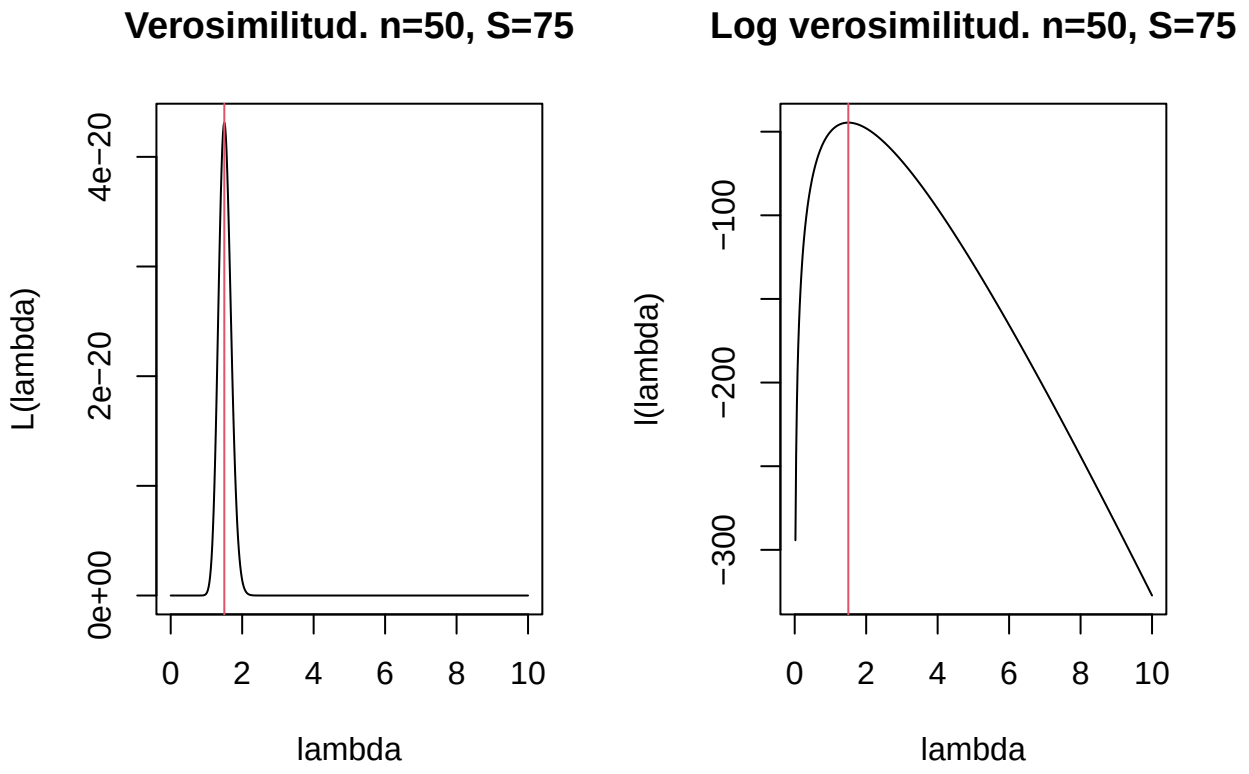


b. Calcular el EMV de forma analítica. Ecuación de verosimilitud: $\frac{\partial}{\partial\lambda}\log L(\lambda, X_1, \dots, X_n) = -n + \frac{\sum_{i=1}^{n} X_i}{\lambda} = 0$, de donde se sigue $\hat{\lambda} = \overline{X} = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{75}{50} = 1.5$

```r
#EMV
emv<-S/n
emv
```

```
## [1] 1.5
```

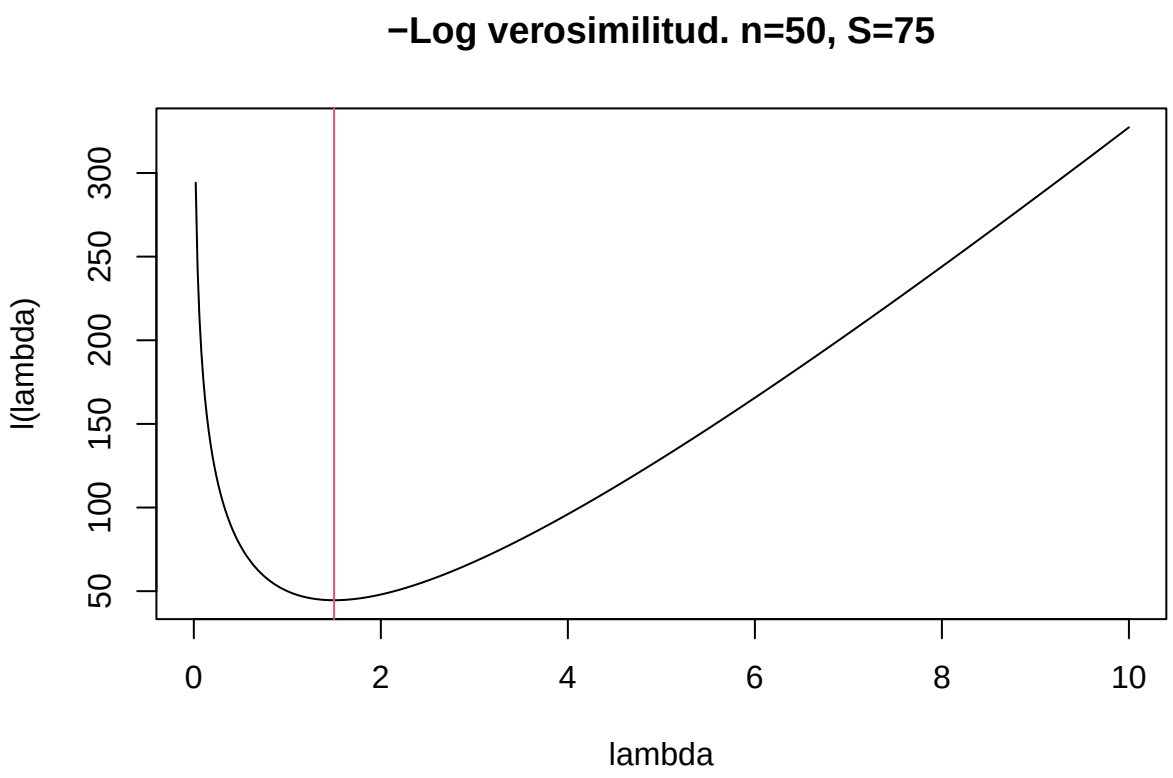
```
par(mfrow=c(1,2))
plot(seq(0,10,length.out=500),v(seq(0,10,length.out=500)),xlab="lambda",ylab="L(lambda)",
     main="Verosimilitud. n=50, S=75",type="l")
abline(v=emv,col=2)
plot(seq(0,10,length.out=500),lv(seq(0,10,length.out=500)),xlab="lambda",ylab="l(lambda)",
     main="Log verosimilitud. n=50, S=75",type="l")
abline(v=emv,col=2)
```



c. Obetener el EMV de forma numérica (aproximada) mediante la función optim. No siempre es posible calcular EMV de forma analítica y es común recurrir a métodos numéricos (por ejemplo Newton Raphson). En {R} podemos usar la función optim que minimiza funciones lineales y no lineales usando Nelder–Mead, quasi-Newton o de descenso de gradiente.

```
#Leer https://cran.r-project.org/doc/contrib/Optimizacion_Matematica_con_R_Volumen_I.pdf

#optim(par, fn, gr = NULL, ...,
#      method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN",
#                 "Brent"),
#      lower = -Inf, upper = Inf,
#      control = list(), hessian = FALSE)
#By default optim performs minimization, but it will maximize negative functions.
```

```r
#par: Initial values for the parameters to be optimized over.

#fn: A function to be minimized, with first argument the vector
#of parameters over which minimization is to take place. It should return a scalar result.

#method: The default method is an implementation of that of Nelder and Mead.
#Method "Brent" is for one-dimensional problems only

#lower, upper: Bounds bounds in which to search for method "Brent".

#hessian: Logical. Should a numerically differentiated Hessian matrix
#be returned? Valor de la segunda derivada de fn

#Caso uniparamétrico basta con:
mlv<-function(lambda){
  return(-lv(lambda))
}
minimo<-optim(par=3, fn=mlv, method = "Brent",
      lower = 0.1, upper = 3,  hessian = TRUE)

#> minimo$par
#> [1] 1.5
#>
#> minimo$value
#> [1] 44.59012
#>
#> minimo$hessian
#> [1] 33.33336
par(mfrow=c(1,1))
plot(seq(0,10,length.out=500),mlv(seq(0,10,length.out=500)),xlab="lambda",ylab="l(lambda)",
     main="-Log verosimilitud. n=50, S=75",type="l")
abline(v=emv,col=2)
```




```
emv<- minimo$par
emv
```

```
## [1] 1.5
```

NOTA: Estamos trabajando con mlv, de forma que la inversa de {minimo$hessian} es un estimador de la varianza asintótica.

d. Calcular IC de Wald para λ con confianza aproximada del 95%.

Podemos llegar de dos maneras diferentes.

Distribución asintótica del EMV (TCL): $\sqrt{n}(\overline{X}-\lambda) \xrightarrow{\mathcal{L}} N(0,\lambda)$, puesto que $E_\lambda(X_i)=\lambda$ y $Var_\lambda(X_i)=\lambda \forall i$.

Distribución asintótica del EMV (Eq. 1): $\sqrt{n}(\overline{X}-\lambda) \xrightarrow{\mathcal{L}} N(0,\frac{1}{I_1(\lambda)})$, donde $I_1(\lambda) = -E_\lambda(\frac{\partial^2}{\partial\lambda^2}\log f(X,\lambda)) = -E_\lambda(\frac{-X}{\lambda^2}) = \frac{1}{\lambda}$.

Para poder hacer inferencias, necesitamos estimar $I_n(\lambda)$ mediante la IF observada: Calculamos $\frac{\partial^2}{\partial\lambda^2}\log L(\lambda, X_1, \dots, X_n) = -\frac{\sum_{i=1}^n X_i}{\lambda^2}$, $\widehat{I_n(\lambda)} = -\frac{\partial^2}{\partial\lambda}\log f(X,\lambda)|_{\theta=\hat{\lambda}} = \frac{\sum_{i=1}^n X_i}{\lambda^2}|_{\lambda=\overline{X}} = \frac{n^2}{\sum_{i=1}^n X_i} = \frac{2500}{75}$.

Y también se tiene: $(\overline{X}-\lambda) \xrightarrow{\mathcal{L}} N(0,\frac{75}{2500})$, donde $se(\hat{\lambda}) = \sqrt{\frac{75}{2500}}$.

Por tanto, el IC $(1-\alpha)\%$ de Wald para λ es:

$$(\overline{X} \pm qnorm(0.975)\sqrt{\frac{75}{2500}}) \simeq (1.5 \pm qnorm(0.975)\sqrt{0.03}) \simeq (1.16, 1.84)$$

```r
#var asintotica del emv
emv.var<-1/minimo$hessian # la inversa de la salida Hessian
emv.var
```

```
##             [,1]
## [1,] 0.02999997
```

```r
#IC Wald con confianza 0.95
emv-qnorm(0.975)*sqrt(emv.var)
```

```
##          [,1]
## [1,] 1.160524
```

```r
emv+qnorm(0.975)*sqrt(emv.var)
```

```
##          [,1]
## [1,] 1.839476
```

e. Calcular el estadístico de Wald para contrastar $H_0 : \lambda = 1 \quad vs \quad H_1 : \lambda \neq 1$. ¿Cuál es el p-valor del test?

$Q_W = \frac{n(\hat{\lambda}-\lambda_0)^2}{\widehat{V^2(\lambda)}} = n(\hat{\lambda}-\lambda_0)^2\widehat{I_1(\lambda)} = \frac{n(\overline{X}-1)^2}{\overline{X}} \overset{H_0}{\cong} \chi^2_1$

Región crítica: $(Q_W > C_{0.05})$, donde $C_{0.05} = qchisq(0.95, 1) = 3.841$, pues $Q_W \sim_{H_0} \chi^2_1$.

P-valor: $P_{\lambda=1}(Q_W > q_{W,obs}) \cong 1 - P_{\lambda=1}(\chi^2_1 \leq q_{W,obs}) = 1 - pchisq(q_{W,obs}, 1)$, donde $q_{W,obs} = \frac{n(\overline{X}-1)^2}{\overline{X}} = \frac{50(1.5-1)}{1.5^2}$

```r
#Estadistico Wald al tipicar la distribuciaon asintótica
Q_Wobs<-((emv-1)^2)/emv.var
Q_Wobs
```

```
##          [,1]
## [1,] 8.333341
```

```r
#Region crítica
qchisq(0.95,1)
```

```
## [1] 3.841459
```

```r
#P-valor
pvalorW<-1-pchisq(Q_Wobs,1)
pvalorW # Se rechaza H_0
```

```
##              [,1]
## [1,] 0.003892401
```

f. Calcular el estadístico test de la RV para contrastar $H_0 : \lambda = 1 \quad vs \quad H_1 : \lambda \neq 1$. ¿Cuál es el p-valor del test? Estadístico RV:

$$Q_L = 2[\log L(\hat{\lambda}, X_1, \dots, X_n) - \log L(\lambda_0, X_1, \dots, X_n)] \tag{1.8}$$
$$= 2[\log L(\overline{X}, X_1, \dots, X_n) - \log L(1, X_1, \dots, X_n)] \tag{1.9}$$
$$= 2[-n\overline{X} + n\overline{X}\log\overline{X} + n] \overset{H_0}{\cong} \chi^2_1 \tag{1.10}$$

P-valor: $P_{\lambda=1}(Q_L > q_{L,obs}) \cong 1 - P_{\lambda=1}(\chi^2_1 \leq q_{L,obs}) = 1 - pchisq(q_{L,obs}, 1)$, donde $q_{L,obs} = 2[-75 + 75\log(1.5) + 50]$

```
#Estadistico RV
Q_Lobs<-2*(-minimo$value+n)#donde toma el valor minimo
Q_Lobs#DISTINTO DE Q_W
```

```
## [1] 10.81977
```

```
#P-valor
pvalorL<-1-pchisq(Q_Lobs,1)
pvalorL # Se rechaza H_0, si est Wald ha rechazado, este también
```

```
## [1] 0.001004222
```

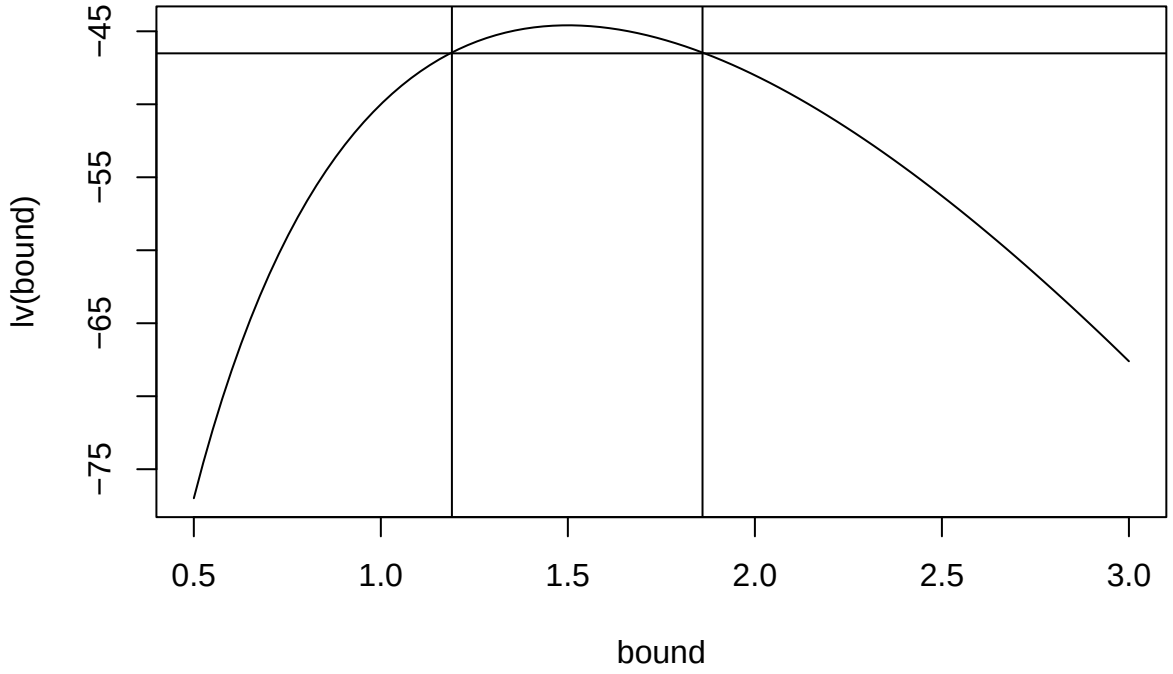
g. Calcular IC basado en la RV para λ con confianza aproximada del 95%.

$$ICRV(0.95) = \{\lambda_0 : 2[\log L(\overline{X}, X) - \log L(\lambda_0, X)] \leq qchisq(0.95, 1)\} \tag{1.11}$$
$$= \{\lambda_0 : \log L(\lambda_0, X) \geq \log L(\overline{X}, X) - \frac{qchisq(0.95, 1)}{2}\} \tag{1.12}$$
$$= \{\lambda_0 : \log L(\lambda_0, X) \geq d_1\} \tag{1.13}$$

```
#Calcular la constante d_1
d1<--minimo$value-qchisq(0.95,1)/2
d1
```

```
## [1] -46.51085
```

```
#Representacion de log verosimilitud en un entorno de emv, p.e. 0.5 y 3
bound<-seq(0.5,3,length.out=100)
plot(bound,lv(bound),type="l")
#añado la constante en d
abline(h=d1)

#ahora queda buscar esos puntos
#n un par de intentos se puede encontrar
#abline(v=1.1)
#abline(v=1.2)
abline(v=1.19)

#abline(v=1.8)
#abline(v=1.9)
abline(v=1.86)
```




```
#[1.86,1.19]
```

h. Calcular IC de Wald para $g(\lambda) = P_\lambda(X = 0) = e^{-\lambda}$ con confianza aproximada del 95%. Por un lado, el EMV es invariante para cualquier función g (con inversa), se tiene que $\widehat{g(\lambda)} = g(\hat{\lambda}) = g(\overline{X}) = e^{-\overline{X}}$. Por otro lado, el Delta Método permite a partir de la distribución asintótica del EMV calcular la distribución de cualquier función g (derivable) del EMV como:

$$\sqrt{n}(g(\hat{\lambda}) - g(\lambda)) \xrightarrow{\mathcal{L}} N(0, \widehat{V^2(\lambda)}(g^{'}(\hat{\lambda}))^2,$$

donde $g(\hat{\lambda}) = e^{-\overline{X}}$, $g(\lambda) = e^{-\lambda}$, $\widehat{V^2(\lambda)} = \overline{X}$ y $(g^{'}(\hat{\lambda}))^2 = (-e^{-\hat{\lambda}})^2 = e^{-2\overline{X}}$. De donde se sigue:

$$\frac{\sqrt{n}(e^{-\overline{X}} - e^{-\lambda})}{\sqrt{\overline{X}e^{-2\overline{X}}}} \cong N(0, 1).$$

El IC $(1-\alpha)\%$ de Wald viene dado por: $P_\lambda(\frac{\sqrt{n}|e^{-\overline{X}} - e^{-\lambda}|}{\sqrt{\overline{X}e^{-2\overline{X}}}} \leq qnorm(1-\alpha/2)) \simeq 1-\alpha$. De forma que despejando λ de la expresión anterior se llega al resultado:

$$(e^{-\overline{X}} \pm qnorm(1-\alpha/2)\sqrt{\frac{\overline{X}e^{-2\overline{X}}}{n}})$$

```
g<-function(x){
  return(exp(-x))
}
g_prima<-function(x){
```

```
  return(-exp(-x))
}
emv_g<- g(emv)
emv_g
```

```
## [1] 0.2231302
```

```
emv.var_g<-emv.var*(g_prima(emv))^2
emv.var_g
```

```
##              [,1]
## [1,] 0.001493611
```

```
emv_g-qnorm(0.975)*sqrt(emv.var_g)
```

```
##           [,1]
## [1,] 0.1473829
```

```
emv_g+qnorm(0.975)*sqrt(emv.var_g)
```

```
##           [,1]
## [1,] 0.2988774
```

i. Calcular el IC basado en la RV para $g(\lambda) = P_\lambda(X = 0)$ con confianza aproximada del 95%.

Los IC basados en el estadístico de la RV son invariantes frente a transformaciones. Para calcular un IC para una función del parámetro bastará con aplicar dicha función a los extremos del intervalo RV obtenidos para el parámetro. En este caso el IC pedido es:

$$(e^{-1.19}, e^{-1.86}) \simeq (0.16, 0.30)$$

1.4 Ejercicio Propuesto 1: Modelo exponencial

Considere $X_1, \ldots, X_{30}$ m.a.s. de $\mathcal{E}xp(\lambda)$, $f(x, \lambda) = \lambda e^{-\lambda x}, x > 0$, $\lambda > 0$. Para los datos observados se tiene $\overline{X} = 0.9$.

a. Obtener y representar la función de verosimilitud y log verosimilitud.
b. Calcular el EMV de forma analítica.
c. Obetener el EMV de forma numérica (aproximada) mediante la función optim.
d. Calcular IC de Wald para λ con confianza aproximada del 95%.
e. Calcular el estadístico test de Wald para contrastar $H_0 : \lambda = 1 \quad vs \quad H_1 : \lambda \neq 1$. ¿Cuál es el p-valor del test?
f. Calcular el estadístico test de la RV para contrastar $H_0 : \lambda = 1 \quad vs \quad H_1 : \lambda \neq 1$. ¿Cuál es el p-valor del test?
g. Calcular IC basado en la RV para λ con confianza aproximada del 95%.

1.5 Ejercicio Propuesto 2: Muestreo por lotes

Se seleccionan aleatoriamente 1000 muestras de suelo que se combinan en lotes de tamaño 10 muestras. Los lotes son examinados para evaluar la presencia de una toxina, es decir, las muestras individuales de suelo no se examinan. Concretamente, en 100 lotes, de 10 muestras de suelo cada uno, se detecto la toxina en 12 lotes.

a. Calcular el EMV para p = probabilidad de que la toxina esté presente en una muestra de suelo individual.
b. Detaerminar la distribución asintótica del EMV de p y calcular IC aproximados de Wald para p con confianza 0.95.
c. Calcular ICRV con confianza aproximada de 0.95 para p.

2 Práctica 2: Inferencias de Wald. Caso multiparamétrico

2.1 Ejercicio1: Repaso, Muestreo por lotes (uniparamétrico)

Se seleccionan aleatoriamente 1000 muestras de suelo que se combinan en lotes de tamagno 10 muestras. Los lotes son examinados para evaluar la presencia de una toxina, es decir, las muestras individuales de suelo no se examinan. Concretamente, en 100 lotes, de 10 muestras de suelo cada uno, se detecto la toxina en 12 lotes.

a. Calcular el EMV para p = probabilidad de que la toxina esté presente en una muestra de suelo individual.

¿Qué datos observamos? ¿Cuál es el modelo estadístico que subyace en estos datos? $Y_1, \dots, Y_{100} \sim Modelo(\theta)$

¿Cómo se relaciona θ con p? ¿Si se conoce p, cuanto vale θ (o al revés)? ¿$P(Y_1 = 1)$ o $P(Y_1 = 0)$?

```
# EMV y de theta

sY<-12
n<-100

mlv<-function(theta){
  return(-sY*log(theta)-(100-sY)*log(1-theta))
}
minimoTheta<-optim(0.5,mlv,method="Brent",lower=0.01,upper=0.99,hessian=TRUE)
emvTheta<-minimoTheta$par
emvTheta #sY/n
```

```
## [1] 0.12
```

```
# EMV de p
g<-function(theta){
  return(1-(1-theta)^0.1)
}

#p<-=g(theta)=1-(1-theta)^(1/10) es invariante por transformacion
emvP<-g(emvTheta)
emvP
```

```
## [1] 0.01270198
```

b. Determinar la distribución asintótica del EMV de p y calcular IC aproximados de Wald para p con confianza 0.95.

```
# Informacion de Fisher observada
infObsTheta<-minimoTheta$hessian
infObsTheta # 1/((emvTheta*(1-emvTheta))/n)
```

```
##          [,1]
## [1,] 947.0857
```

```
# Delta metodo

gPrima<-function(theta){
  return(0.1*(1-theta)^-0.9)
}

emvVarP<-(1/infObsTheta)*gPrima(emvTheta)^2
emvVarP
```

```
##              [,1]
## [1,] 1.329052e-05
```

```
#IC Wald 95% no son invariantes por transformacion
seP<-sqrt(emvVarP)
emvP-qnorm(0.975)*seP
```

```
##             [,1]
## [1,] 0.005556701
```

```
emvP+qnorm(0.975)*seP
```

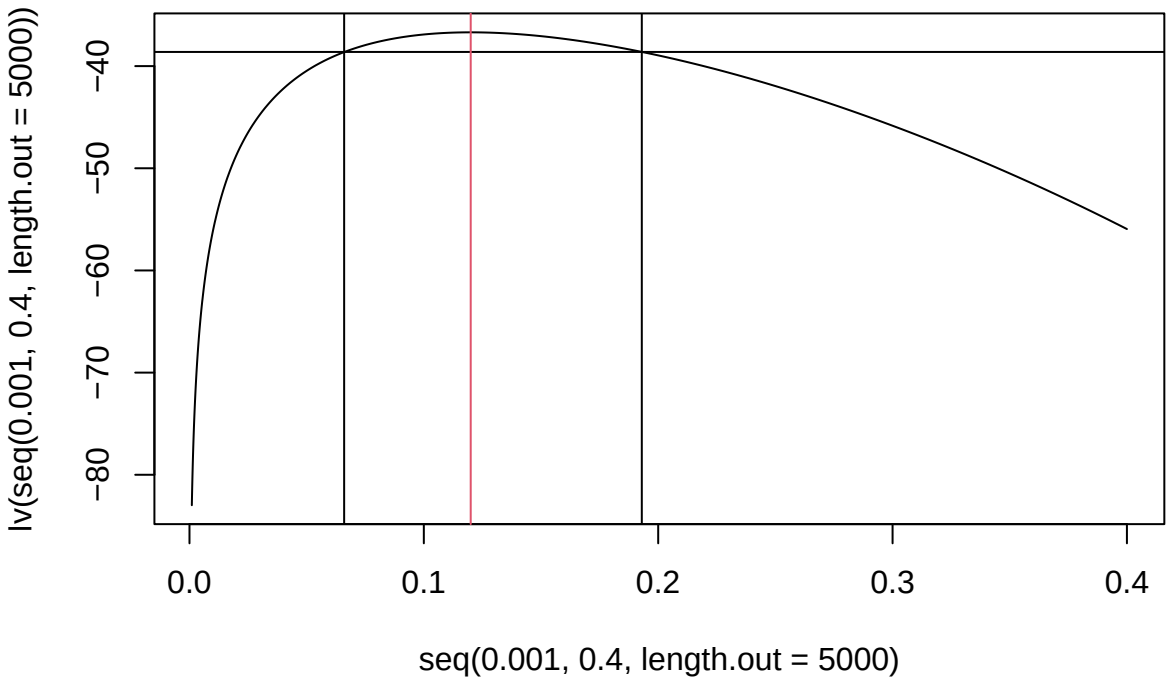
```
##            [,1]
## [1,] 0.01984725
```

c. Calcular ICRV con confianza aproximada de 0.95 para p.

```
#IC RV para theta

lv<-function(theta){
  return(sY*log(theta)+(100-sY)*log(1-theta))
}
d<--minimoTheta$value-qchisq(0.95,1)/2
d
```

```
## [1] -38.61323
```

```
# ICs para theta
plot(seq(0.001,0.4,length.out=5000),lv(seq(0.001,0.4,length.out=5000)),type="l")
abline(h=d)
abline(v=0.12,col=2) # emv
abline(v=c(0.066,0.193),lty=1) # IC RV
```




```
#IC RV para p 95% son invariantes por transformacion
g(0.066);g(0.193)
```

```
## [1] 0.006804627
```

```
## [1] 0.02121489
```

```
# Comparacion de los intervalos. Ventajas e inconvenientes
# c(0.0056,0.0198),lty=2) # IC Wald, siempre simetrico
# c(0.0068,0.0212),lty=1) # IC RV
```

2.2 Ejercicio 2: Genética de Fisher. Adaptación Ejercicio 9 (multiparamétrico)

Se cuenta con una muestra de tamaño n=3839 de una población que toma 4 valores (fenotipos ab, Ab, aB, AB) distintos con probabilidades p_1, p_2, p_3, y p_4 tales que $\sum_{r=1}^{4} p_r = 1$. Concretamente se observan las siguientes frecuencias para las tres primeras clases: 1997, 904 y 906. La hipótesis de Fisher (en contra de lo propuesto por Mendel) es que las probabilidades dependen de un solo parámetros θ de la forma siguiente:

$$H_0 : p_1 = \frac{2+\theta}{4}, p_2 = \frac{1-\theta}{4} = p_3, p_4 = \frac{\theta}{4}.$$

a. Calcular el EMV para $p = (p_1, p_2, p_3)^{'}$ e IC de Wald con confianza aproximada del 95% para p_1, p_2 yp_3.

```r
n<-3839
fr<-c(1997,904,906)
mlv<-function(p){
  return(-sum(fr*log(p))-(n-sum(fr))*log(1-sum(p)))
}

# EMV para p con optim y Nelder-Mead (por defecto)

# optim minimiza -logver ~ maximiza -logver. Argumentos:
# par: valores iniciales de los parametros
# fn: funcion a minimizar
# hessian=TRUE calcula la matriz de derivadas segundas de fn ene l EMV
# method: por defecto en el caso multiparamétrico Nelder-Mead.
# Uniparamétrico: Brent (lower,upper)
vi<-rep(0.3,3)
minimo<-optim(par=vi,fn=mlv,hessian=TRUE)
```

```
## Warning in log(1 - sum(p)): Se han producido NaNs

## Warning in log(1 - sum(p)): Se han producido NaNs

## Warning in log(1 - sum(p)): Se han producido NaNs

## Warning in log(1 - sum(p)): Se han producido NaNs

## Warning in log(1 - sum(p)): Se han producido NaNs

## Warning in log(1 - sum(p)): Se han producido NaNs

## Warning in log(1 - sum(p)): Se han producido NaNs

## Warning in log(1 - sum(p)): Se han producido NaNs
```

```r
# EMV

emv<-minimo$par# EMV
emv
```

```
## [1] 0.5202219 0.2354914 0.2359584
```

```r
p1Hat<-fr[1]/n;p2Hat<-fr[2]/n;p3Hat<-fr[3]/n;p4Hat<-1-p1Hat-p2Hat-p3Hat
p1Hat;p2Hat;p3Hat
```

```
## [1] 0.5201875

## [1] 0.235478

## [1] 0.235999
```

```r
# Estimador de la varianza

infObs<-minimo$hessian # Matriz IF observada. -Hesianno ya en mlv. 3x3
emvVar<-solve(infObs) # Inversa de IF observada. Caso uniparametrico 1/infObs
emvVar
```

```
##               [,1]          [,2]          [,3]
## [1,]  6.499573e-05 -3.192230e-05 -3.197831e-05
## [2,] -3.192230e-05  4.689324e-05 -1.447523e-05
## [3,] -3.197831e-05 -1.447523e-05  4.695013e-05
```

```r
# Error estandar

se<-sqrt(diag(emvVar)) # Errores estandar de EMV
se
```

```
## [1] 0.008061993 0.006847864 0.006852016
```

```r
# IC para p1, p2 y p3
tabla.Wald<-cbind(emv,se,extremo.inferior=emv-qnorm(0.975)*se,
    extremo.superior=emv+qnorm(0.975)*se)
nombres.par<-c("p1","p2", "p3")
rownames(tabla.Wald)<-nombres.par
round(tabla.Wald,4) #usar 4 decimales
```

```
##       emv     se extremo.inferior extremo.superior
## p1 0.5202 0.0081           0.5044           0.5360
## p2 0.2355 0.0068           0.2221           0.2489
## p3 0.2360 0.0069           0.2225           0.2494
```

b. Calcular p-valor del test de Wald para contrastar H_0.

```r
# Hipótesis compuesta: H0: p2-p3=0; p1+p2-3/4=0
#g1(p)=p2-p3=0
#g2(p)=p1+p2-3/4=0

g1<-function(p){
  p1<-p[1];p2<-p[2];p3<-p[3]
  return(p2-p3)
}

g2<-function(p){
  p1<-p[1];p2<-p[2];p3<-p[3]
  return(p1+p2-3/4)
}


# Delta método
G<-matrix(c(0,1,-1,1,1,0),2,3,byrow=TRUE)
varG<-G%*%emvVar%*%t(G)
Wobs<-cbind(g1(emv),g2(emv))%*%solve(varG)%*%t(cbind(g1(emv),g2(emv)))
Wobs
```

```
##          [,1]
## [1,] 2.043841
```

```
pvalW<-1-pchisq(Wobs,2)
pvalW # No se rechaza H0
```

```
##           [,1]
## [1,] 0.3599031
```

c. Calcular IC para θ con confianza aproximada de 0.95 basado en el estadístico de Wald.

```
# Bajo H0 se trata IC Wald para theta uniparamétrico:
# EMV de theta bajo H0 y su distribucion asintótica

#mlv bajo H0, solo depende de theta
mlvH0<-function(theta){
  pH0<-c((2+theta)/4,(1-theta)/4,(1-theta)/4)
  return(-sum(fr*log(pH0))-(n-sum(fr))*log(1-sum(pH0)))
}
viH0<-0.1
minimoH0<-optim(viH0,mlvH0,method="Brent",lower=0.01,upper=0.99,hessian=TRUE)
emvH0<-minimoH0$par
emvH0
```

```
## [1] 0.0357123
```

```
infObsH0<-minimoH0$hessian
emvVarH0<-1/infObsH0

emvH0-qnorm(0.975)*sqrt(emvVarH0)
```

```
##            [,1]
## [1,] 0.02390586
```

```
emvH0+qnorm(0.975)*sqrt(emvVarH0)
```

```
##            [,1]
## [1,] 0.04751874
```

d. Calcular p-valor del test de RV para contrastar H_0.

```
Qobs<-2*(-minimo$value+minimoH0$value)
Qobs
```

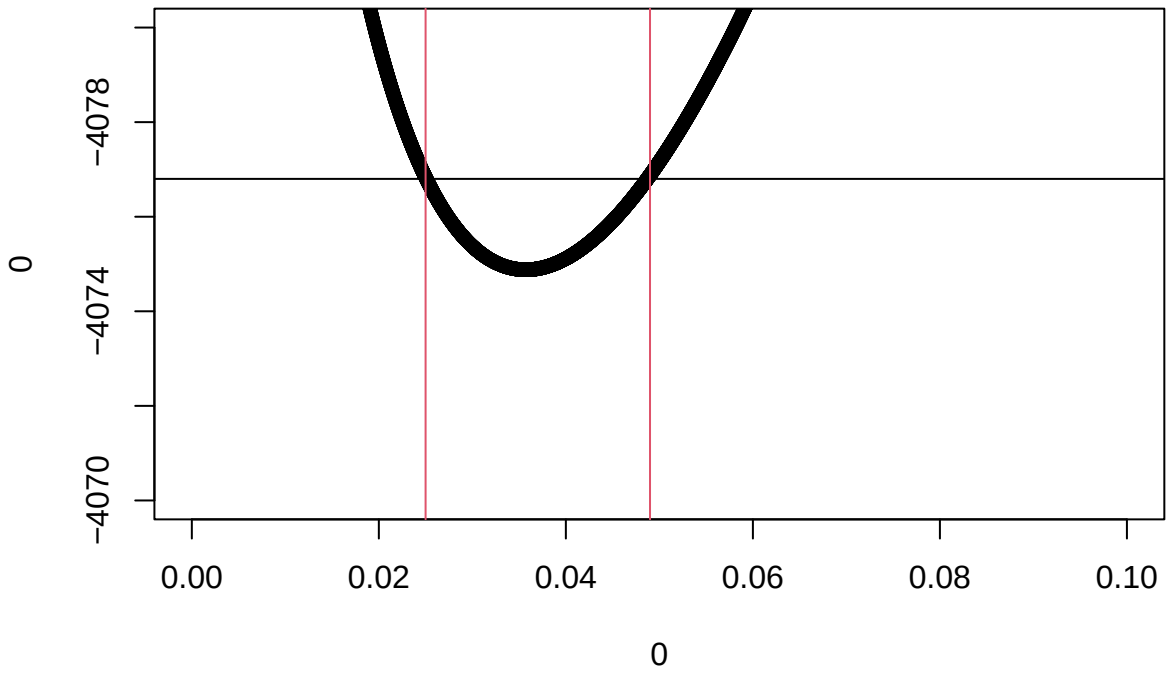
```
## [1] 2.018659
```

```
pvalQ<-1-pchisq(Qobs,2)
pvalQ # Misma conclusion pvalor Wald, aunque no tienen por que coincidir
```

```
## [1] 0.3644632
```

e. Calcular IC para θ con confianza aproximada de 0.95 basado en el estadístico de RV.

```
lvH0<-function(theta){
  pH0<-c((2+theta)/4,(1-theta)/4,(1-theta)/4)
  return(sum(fr*log(pH0))+(n-sum(fr))*log(1-sum(pH0)))
}
d<--minimoH0$value-qchisq(0.95,1)/2
plot(0,0,type="n",xlim=c(0,0.1),ylim=c(-4070,-4080))
ss<-seq(0,0.1,length.out=10000)
for(i in 1:10000){
points(ss[i],lvH0(ss[i]),pch=16,lwd=0.5)
}
abline(h=d)
abline(v=c(0.025,0.049),col=2)
```



2.3 Ejercicio Propuesto 1: Modelo multinomial

Sean $X_1, \dots, X_{50}$ v.a.i.i.d. con distribución discreta que toma los valores 1, 2, 3, 4 y 5 con probabilidades p_1, p_2, p_3, p_4 y p_5 respectivamente, con $p_r \geq 0, r = 1, \dots, 5$ y $\sum_{r=1}^{5} p_r = 1$. Sea $Y_r = \sum_{i=1}^{50} \mathbb{I}_{(X_i=r)}$, $r = 1, \dots, 5$. Si se observa $Y_1 = 7, Y_2 = 6, Y_3 = 8, Y_4 = 16$ e $Y_5 = 13$.

a. Obtener el p-valor del test de Wald para contrastar la hipótesis $H_0 : p_1 = p_2 = p_3, p_4 = p_5$ contra H_1 : el resto.
b. Obtener el p-valor del test de la razón de verosimilitudes para contrastar la hipótesis $H_0 : p_1 = p_2 = p_3, p_4 = p_5$ contra H_1 : el resto.

c. Obtener el p-valor del test de Wald para contrastar la hipótesis $H_0 : p_2 - 2p_1 = 0, p_3 = p_4$ contra $H_1 :$ el resto.
d. Obtener el p-valor del test de la razón de verosimilitudes para contrastar la hipótesis $H_0 : p_2 - 2p_1 = 0, p_3 = p_4$ contra $H_1 :$ el resto.

3 Práctica 3: Inferencias basadas en el estadístico RV. Caso multiparamétrico

3.1 Ejercicio 1: Modelo de mixturas

En un estudio de micropropagación de raices, se anotó el número de raices de 40 variedades de manzana. Los datos observados fueron:

Nº raices	0	1	2	3	4	5	6	7	8	9
Frecuencia	19	2	2	4	3	1	4	3	0	2

Se ha establecido como modelo para describir este fenómeno una mixtura de dos distribuciones f_1 y f_2 con proporciones p y $1-p$ respectivamente. En concreto, f_1 toma el valor cero con probabilidad 1 ($f_1(0)=1$), y f_2 es la función de masa de probabilidad de una distribución de Poisson(λ).

a. Obtener el EMV para los parámetros e IC de Wald con confianza aproximada de 95% para p y para l.

```
# Datos observados
 y<-c(rep(0,19),rep(1,2),rep(2,2),rep(3,4),rep(4,3),rep(5,1),rep(6,4),rep(7,3),rep(9,2))

# mlv
mlv<-function(theta){
  p<-theta[1]
  lambda<-theta[2]
  return(-(19*log(p+(1-p)*exp(-lambda))+21*log(1-p)-21*lambda+sum(y)*log(lambda)))
  #NOTA: sum(y), empieza en 0
}

vi<-c(0.5,1) # valores coherentes con p y lambda, respectivamente
minimo<-optim(vi,mlv,hessian=TRUE)
```

```
## Warning in log(1 - p): Se han producido NaNs
```

```
emv<-minimo$par
emv
```

```
## [1] 0.4698821 4.6207927
```

```
infObs<-minimo$hessian
emvVar<-solve(infObs)
emvVar
```

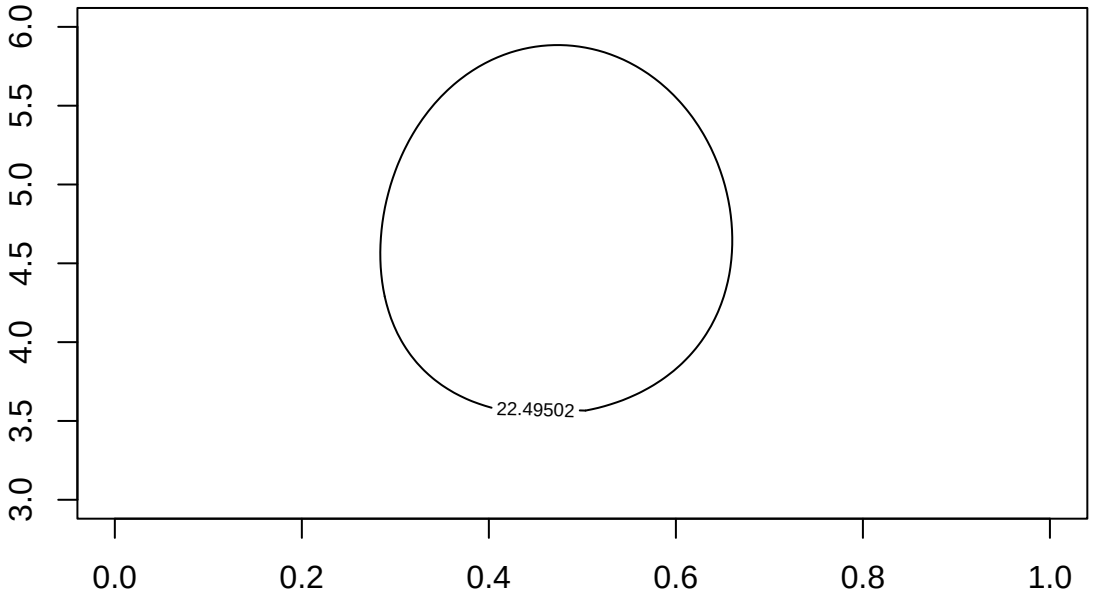
```
##              [,1]        [,2]
## [1,] 0.006365515 0.001203439
## [2,] 0.001203439 0.228362471
```

```r
# Errores estandar e IC de Wald para los dos parámetros
se<-sqrt(diag(emvVar))
tabla.Wald<-cbind(emv,se,extremo.inferior=emv-qnorm(0.975)*se,
    extremo.superior=emv+qnorm(0.975)*se)
nombres.par<-c("p","l")
rownames(tabla.Wald)<-nombres.par
round(tabla.Wald,4) #usar 4 decimales
```

```
##      emv     se extremo.inferior extremo.superior
## p 0.4699 0.0798           0.3135           0.6263
## l 4.6208 0.4779           3.6842           5.5574
```

b. Obtener una región de confianza simultánea para (p,l) al 95%, basada en la razón de verosimilitud.

```r
lv<-function(p,lambda){
  return(19*log(p+(1-p)*exp(-lambda))+21*log(1-p)-21*lambda+sum(y)*log(lambda))
}

d2<--minimo$value-qchisq(0.95,2)/2
pSeq<-seq(0,1,length.out=150) # conocemos EMV, para definir mejor este grid
lSeq<-seq(1,8,length.out=150)
#g<-outer(pSeq,lSeq,lv)
g<-outer(pSeq,lSeq,lv)
contour(pSeq,lSeq,g,levels=c(d2),ylim=c(3,6))
```

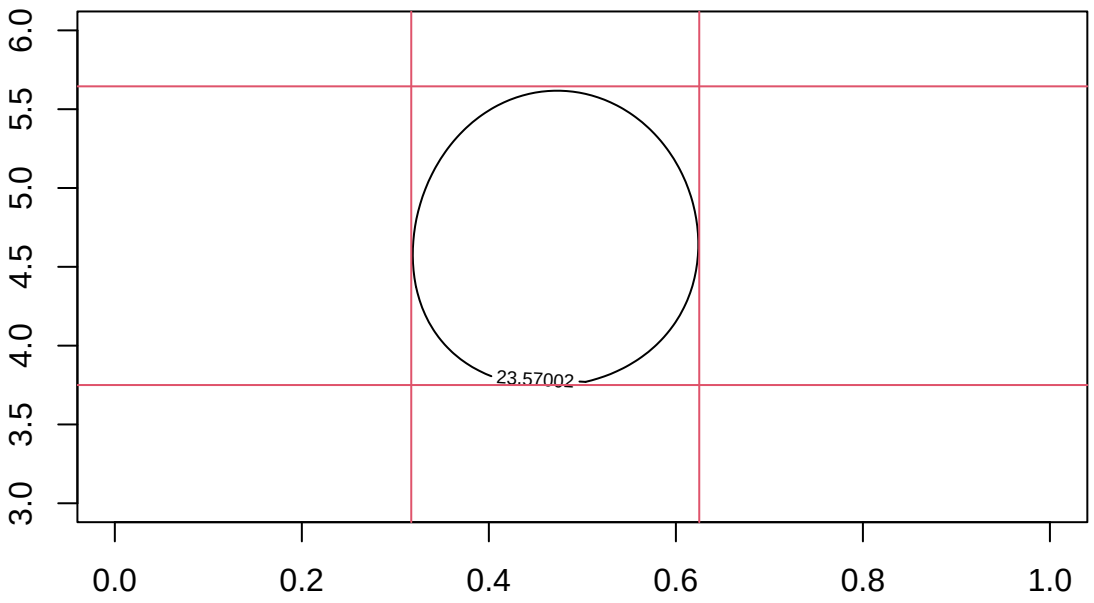


```
emv
```

```
## [1] 0.4698821 4.6207927
```

c. Obtener IC al 95% para p y l basados en el test de la razón de verosimilitudes.

```
# Manera 1
# Similar a ejecicio 6 hecho en clase. Expresión compleja
#h<-function(p){
  #return()
#}
#d1<--minimo$value-qchisq(0.95,1)/2
#pSeq<-seq(0,1,length.out=150)
#plot(pSeq,h(pSeq),type="l")
#abline(h=d1)


# Manera 2
d1<--minimo$value-qchisq(0.95,1)/2
pSeq<-seq(0,1,length.out=150)
lSeq<-seq(1,8,length.out=150)
o<-outer(pSeq,lSeq,lv)
contour(pSeq,lSeq,o,levels=c(d1),ylim=c(3,6))# está contenida en la anterior
abline(v=c(0.317,0.625),col=2) # Comparar con los IC calculados para Wald
abline(h=c(3.75,5.645),col=2)
```



d. Obtener el p-valor del test basado en el estadístico RV para contrastar la hipótesis $H_0 : \lambda = 2p^2 + 4$ contra $H_1 : \lambda \neq 2p^2 + 4$.

```
mlvH0<-function(p){
  lambda<-2*p^2+4
  return(-(19*log(p+(1-p)*exp(-lambda))+21*log(1-p)-21*lambda+sum(y)*log(lambda)))
}
minimoH0<-optim(par=0.5,fn=mlvH0,method="Brent",lower=0.01,upper=0.99,hessian=TRUE)
# Uniparamétrico usamos Brent
emvH0<-minimoH0$par
emvH0
```

```
## [1] 0.4780091
```

```
Qobs<-2*(-minimo$value+minimoH0$value)
Qobs
```

```
## [1] 0.1332147
```

```
pvalRV<-1-pchisq(Qobs,1)
pvalRV
```

```
## [1] 0.7151219
```

e. Obtener el p-valor del test basado en el estadístico de Wald para contrastar la hipótesis $H_0 : \lambda = 2p^2+4$ contra $H_1 : \lambda \neq 2p^2 + 4$. (Repaso)

```
g<-function(theta){
  p<-theta[1]
  lambda<-theta[2]
  return(lambda-2*p^2-4)
}
G<-cbind(-4*emv[1],1)
uve<-G%*%emvVar%*%t(G)
Wobs<-(g(emv)-0)^2/uve
pvalW<-1-pchisq(Wobs,1)
pvalW
```

```
##           [,1]
## [1,] 0.7180305
```

3.2 Ejercicio 2: Modelo beta

En una muestra aleatoria de tamaño 30 de una población $B(a, b)$ de parámetros $a = b = 2$: (https://mathworld.wolfram.com/BetaDistribution.html https://www.wolframalpha.com/input/?i=beta+distribution)

a. Obtener EMV de a y b, su distribución asintótica e IC Wals con confienza aproximada 95% para a y b.

```r
n<-30
set.seed(1234)
x<-rbeta(n,2,2) # Datos

A<-sum(log(x))
B<-sum(log(1-x))

mlv<-function(tita){
    a<-tita[1]
    b<-tita[2]
    return(30*lbeta(a,b)-A*(a-1)-B*(b-1)) #lbeta(a,b) calcula log(beta(a,b))
}


vi<-c(1,1) # ya que en la muetra proviene de B(2,2)
minimo<-optim(vi,mlv,hessian=TRUE)
emv<-minimo$par
emv
```

```
## [1] 2.25584 3.18622
```

```r
infobs<-minimo$hessian
emvVar<-solve(infobs)
emvVar
```

```
##           [,1]      [,2]
## [1,] 0.3040330 0.3682515
## [2,] 0.3682515 0.6462389
```

```r
# Errores estandar e IC de Wald para los dos parámetros
se<-sqrt(diag(emvVar))
tabla.Wald<-cbind(emv,se,extremo.inferior=emv-qnorm(0.975)*se,
    extremo.superior=emv+qnorm(0.975)*se)
nombres.par<-c("a","b")
rownames(tabla.Wald)<-nombres.par
round(tabla.Wald,4) #usar 4 decimales
```

```
##      emv     se extremo.inferior extremo.superior
## a 2.2558 0.5514           1.1751           3.3365
## b 3.1862 0.8039           1.6106           4.7618
```

b. Obtener el p-valor basado en el estadístico de RV para contrastar la hipótesis (compuesta) $H_0 : a = 2$ vs $H_1 : a \neq 2$

```r
mlvH0<-function(b){
    return(30*lbeta(2,b)-A*(2-1)-B*(b-1))
}
minimoH0<-optim(1,mlvH0,method="Brent",hessian=TRUE,lower=0.1,upper=4)
emvH0<-minimoH0$par
emvH0
```

```
## [1] 2.876028
```

```
T1<- 2*(-minimo$value+minimoH0$value)
T1
```

```
## [1] 0.2340296
```

```
pvalorT1<-1-pchisq(T1,1)
pvalorT1
```

```
## [1] 0.6285519
```

c. Obtener el p-valor basado en el estadístico de RV para contrastar la hipótesis (compuesta) $H_0 : a = b$ vs $H_1 : a \neq b$

```
mlvH2<-function(b){
    return(30*lbeta(b,b)-A*(b-1)-B*(b-1))
}
minimoH02<-optim(1,mlvH2,method="Brent",hessian=TRUE,lower=0.1,upper=4)
emvH02<-minimoH02$par
emvH02
```

```
## [1] 2.277675
```

```
T2<- 2*(-minimo$value+minimoH02$value)
T2
```

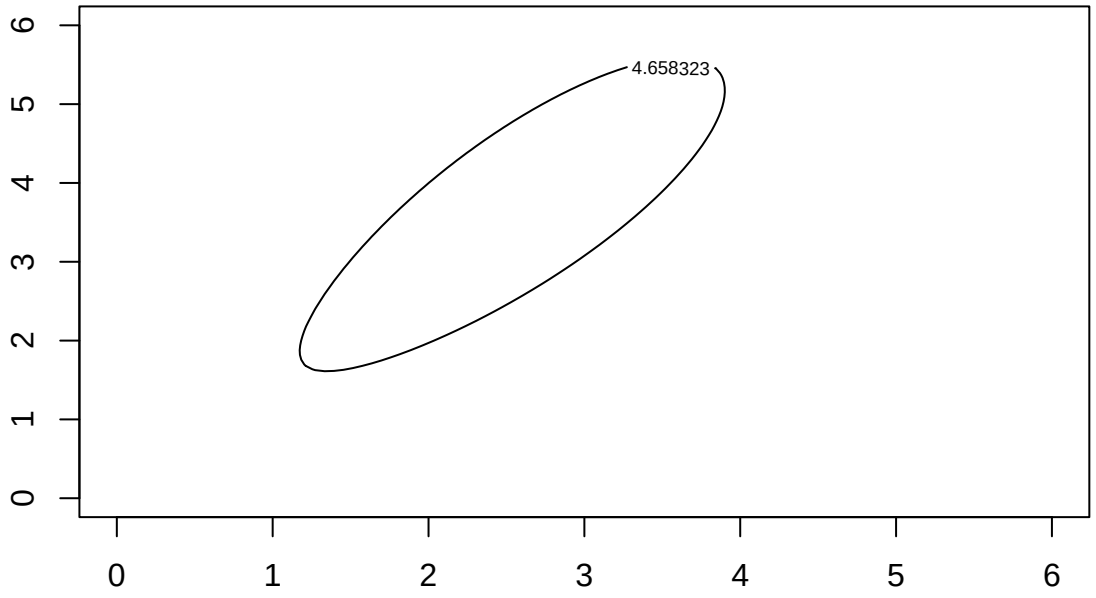
```
## [1] 5.269179
```

```
pvalorT2<-1-pchisq(T2,1)#cuando a=b tenemos un solo parametro
pvalorT2
```

```
## [1] 0.02170625
```

d. Obtener la region de confianza simultane para (a,b) con confianza aproximada 0.95 basada en el test de la razón de verosimilitudes.

```
d2<--minimo$value-qchisq(0.95,2)/2
aSeq<-seq(0,6,length.out=100) # Definir una secuencia con sentido para los parametros
bSeq<-seq(0,6,length.out=100)
o1<-outer(aSeq,bSeq,function(a,b)-30*lbeta(a,b)+A*(a-1)+B*(b-1)) # o definir lv
contour(aSeq,bSeq,o1,levels=c(d2))
```

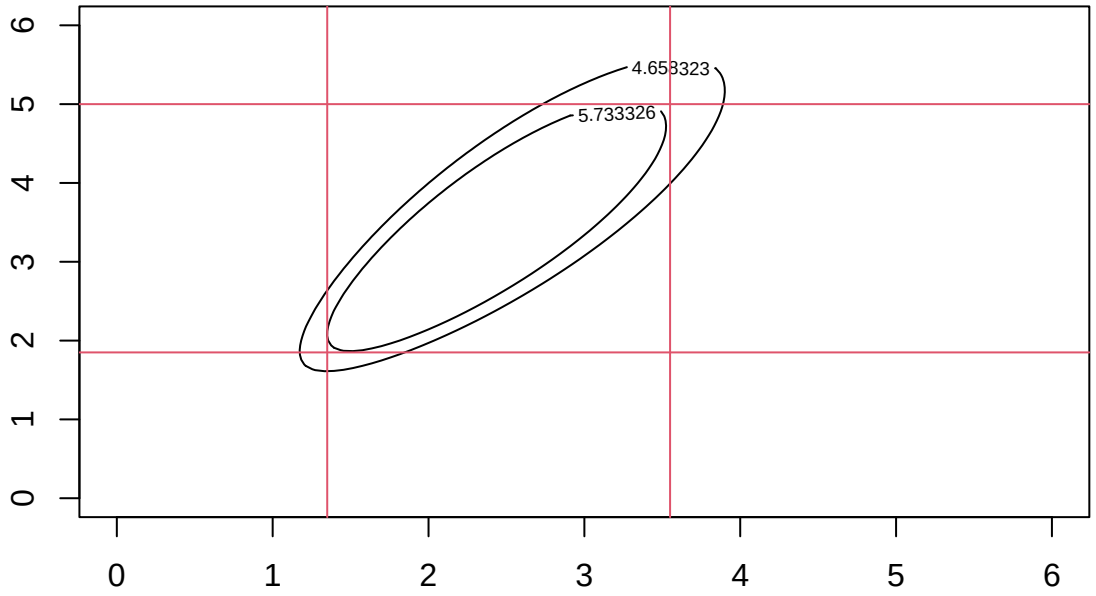



```
#Comprobamos que esta EMV
emv
```

```
## [1] 2.25584 3.18622
```

e. Obtener intervalos de confianza para a y b con confianza aproximada 0.95. para a y b

```
d1<--minimo$value-qchisq(0.95,1)/2
aSeq<-seq(0,6,length.out=100)# Definir una secuencia con sentido para los parametros
bSeq<-seq(0,6,length.out=100)
#definimos la cantidad f
o2<-outer(aSeq,bSeq,function(a,b)-30*lbeta(a,b)+A*(a-1)+B*(b-1)) # o definir lv
contour(aSeq,bSeq,o1,levels=c(d2))
contour(aSeq,bSeq,o2,levels=c(d1),add=TRUE)

# IC para a
abline(v=1.35,col=2)#extremo inferior
abline(v=3.55,col=2)#extremo superior
# IC para a
abline(h=1.85,col=2)#extremo inferior
abline(h=5,col=2)#extremo superior
```




```
# Se pueden calcular con los intervalos para Wald
```

f. Obtener el p-valor basado en el estadístico de Wald para contrastar la hipótesis $H_0 : a = 2$ vs $H_1 : a \neq 2$

```
#Se trata del valor de a con el que ha generado la muestra. No debería rechazarse
W1<-(emv[1]-2)^2/emvVar[1,1]
W1
```

```
## [1] 0.2152854
```

```
pvalorW1<-1-pchisq(W1,1)
pvalorW1
```

```
## [1] 0.6426559
```

g. Obtener el p-valor basado en el estadístico de Wald para contrastar la hipótesis $H_0 : a = b$ vs $H_1 : a \neq b$

```
g<-function(theta){
  a<-theta[1]
  b<-theta[2]
  return(a-b)
}
G<-cbind(1,-1)
uve<-G%*%emvVar%*%t(G)
W2<-((g(emv)-0)^2)/uve
W2
```

```
##          [,1]
## [1,] 4.049267
```

```
pvalorW2<-1-pchisq(W2,1)
pvalorW2
```

```
##            [,1]
## [1,] 0.04419053
```

3.3 Ejercicio Propuesto 1: Modelo normal

Sean X_1,...,X_{100} v.a.i.i.d. distribución $N(\mu, \sigma)$. Los datos observados de los que se dispone son: $\sum_{i=i}^{n} X_i = 134.94$ y $\sum_{i=1}^{n} X_i^2 = 547.43$.

a. Obtener el p-valor del test de Wald para contrastar $H_0 : \mu = \sigma$ contra $H_1 : \mu \neq \sigma$.
b. Obtener el p-valor del test de RV para contrastar la hipótesis nula $H_0 : \mu = \sigma$ contra $H_1 : \mu \neq \sigma$.
c. Obtener la región de confianza simultánea para (μ, σ), con confianza aproximada 0.95.
d. Obtener intervalos de confianza de Wald, con confianza aproximada 0.95, para ambos parámetros μ y σ.
e. Obtener intervalos de confianza basados en la RV, con confianza 0.95, para ambos parámetros μ y σ.

4 Práctica 4: Algoritmos de maximización de la verosmilitud

4.1 Ejercicio 1: Algoritmo NR y EM aplicados a un modelo de mixtura de dos normales conocidas

Sean $Y_1, \ldots, Y_{200}$ v.a.i.i.d. de un modelo de mixtura de dos normales conocidas $N(0,1)$ y $N(1,0.8)$ con densidad $g(y,p) = p*dnorm(y) + (1-p)*dnorm(y,1,0.8)$, es decir, g es la función de densidad de los datos observados (incompletos). A partir de a muestra dada, obtenga el EMV para p:

```
# Definir la muestra y de tamaño n=200
set.seed(1234)
x<-rnorm(200) # m.a.s. de tamaño 200 de una N(0,1)
y<-rnorm(200,1,0.8) # m.a.s. de tamaño 200 de una N(1,0.8)
r<-rbinom(200,1,0.3) # m.a.s. de tamaño 200 de una B(0.3)
y[r>0]<-x[r>0] # Reemplazar en y con los valores de x,
# en aquella posiciones donde r sea 1

# Aunque hemos simulado con p=0.3, trabajamos como desconocido
```

a. Mediante la función optim.

```
# Menos logverosimilitud de los datos observados (incompletos)
mlv<-function(p,z) { #Depende de p y z, en optim especificaremos los valores de z
    verosimilitud<-p*dnorm(z,0,1)+(1-p)*dnorm(z,1,0.8)
    return(-sum(log(verosimilitud)))
}
vi<-c(0.4)  #valor inicial para el calculo del emv
minimo<-optim(vi,mlv,z=y,method="Brent",lower=0.001,upper=0.999,hessian=T)#se especifica z
EMV_optim<-minimo$par
EMV_optim
```

```
## [1] 0.2854986
```

b. Mediante el algoritmo NR.

```
# Primera derivada de la logverosimilitud de los datos observados (incompletos)
lprima<-function(p){
    num<-dnorm(y)-dnorm(y,1,0.8)
    dnom<-p*dnorm(y)+(1-p)*dnorm(y,1,0.8)
    return(sum(num/dnom))
}

# Segunda derivada de la logverosimilitud de los datos observados (incompletos)
H<-function(p){
    num<--(dnorm(y)-dnorm(y,1,0.8))^2
```

```
    dnom<-(p*dnorm(y)+(1-p)*dnorm(y,1,0.8))^2
    return(sum(num/dnom))
}

# Fórmulas de actualización: theta^(k+1)=theta^(k)-H(theta^(k),X)^-1*lprima(theta^(k),X)
iter<-10 # Basta con iter<-6
p0<-0.2
p<-c(p0,rep(0,iter))
for(k in 2:(iter+1)){
        p[k]<-p[k-1]-(lprima(p[k-1])/H(p[k-1]))
}
EMV_NR<-p
EMV_NR
```

```
##  [1] 0.2000000 0.2706448 0.2851369 0.2854984 0.2854986 0.2854986 0.2854986
##  [8] 0.2854986 0.2854986 0.2854986 0.2854986
```

c. Mediante el algoritmo EM con valor inicial 0.2 y 10 iteraciones.

```
# Fórmulas de actualización:
#theta^(k+1)=1/n*sum((p^(k)*dnorm(y))/(p^(k)*dnorm(y)+(1-p^(k))*dnorm(y,1,0.8)))

iter<-10 # Aumentar 10, 25, 52
p0<-0.2
p<-c(p0,rep(0,iter))
for(k in 2:(iter+1)){
    num<-p[k-1]*dnorm(y)
    deno<-p[k-1]*dnorm(y)+(1-p[k-1])*dnorm(y,1,0.8)
        p[k]<-mean(num/deno)
}
EMV_EM<-p
EMV_EM
```

```
##  [1] 0.2000000 0.2200883 0.2358240 0.2479619 0.2572308 0.2642614 0.2695700
##  [8] 0.2735658 0.2765669 0.2788173 0.2805030
```

4.2 Ejercicio 2: Tabla de contingencia con datos faltantes

Se tienen 100 individuos que se clasifican en una de las 6 celdas de una tabla de contingencia 2×3, en la que se sabe que las filas y columnas son independientes, pero para la que se ha perdido la información de 35 individuos correspondientes a las celdas (1,3) y (2,2). Los datos observados fueron: La tabla observada es:

```
| 10 | 5  |  ? |
| 20 | ?  | 30 |
```

Se cuentan con $X_1, \dots X_{100}$ v.a.i.i.d. de una distribución P_θ tal que $Y_z = \sum_{i=1}^{100} \mathbb{I}(X_i = z), z = 1, \dots, 6$ con probabilidad $p_1, \dots, p_6$, y tal que $\sum_{z=1}^{6} p_z = 1$, de donde se sigue la siguiente tabla de probabilidad:

```
| p_1 | p_2  | p_3       | r_1   |
| p_4 | p_5  | p_6       | 1-r_1 |
| c_1 | c_2  | 1-c_1-c_2 |       |
```

Además se sabe que no se dispone de datos para Y_3 e Y_5 y que las filas y columnas son independientes, luego $(p_1, \dots, p_6) = (r_1c_1, r_1c_2, \dots, (1-r_1)(1-c_1-c_2))$, y por tanto $\theta = (r_1, c_1, c_2)$.

a. Obtener directamente el EMV para (r_1, c_1, c_2), maximizando la log-verosimilitud para los datos observados (incompletos).

```
n<-100
y<-c(10,5,2,3,35) # (y1,y2,y4,y6,100-(y1+y2+y4+y6))
w<-c(15,30,5) # (r1,c1,c2)
mlv<-function(tita) {
    r1<-tita[1];c1<-tita[2];c2<-tita[3]
    return(-sum(w*log(tita))-50*log(1-r1)-30*log(1-c1-c2)-35*log(r1*(1-c1-c2)+(1-r1)*c2))
    # sum(y*log(tita)) simplifica
}

vi<-c(1/3,1/3,1/6) # válidos
minimo<-optim(vi,mlv,hessian=T)
EMV_optim<-minimo$par
EMV_optim
```

```
## [1] 0.3748990 0.2999751 0.1751209
```

b. Utilizar el algoritmo EM para aproximar el EMV para (r_1, c_1, c_2), con valores iniciales $(r_1^0, c_1^0, c_2^0) = (1/3, 1/3, 1/6)$ y 20 iteraciones.

```
# Hemos calculado la probabilidad:
#P(Y3/Y3+Y5=35)=P(Y3=k,Y5=35-k)/P(Y3+Y5=35) =
# = (35!/k!*(35-k)!)*(P3/ P3+P5)^k) * (1-p3/ P3+P5)^(35-k)

n<-100
y1<-10;y2<-5;y4<-20;y6<-30    #datos observados

r1<-1/3; c1<-1/3; c2<-1/6; r2<-1-r1;c3<-1-c1-c2    #valores iniciales de los parámetros

iter<-20
for(k in 1:iter){

#paso E

    # Podemos hacerlo como en clase
    # Paso E:
    y31Star<-35*(r1*(1-c1-c2))/((r1*(1-c1-c2))+(1-r1)*c2)
    y51Star<-35-y31Star



    # O trabajar en función de p que es más simple
    #paso E
        #p3<-r1*c3; p5<-r2*c2
        #y31Star<-35*p3/(p3+p5); y51Star<-35-y31Star


#paso M
```

```
    r1<-(15+y31Star)/n
    c1<-0.3
    c2<-(40-y31Star)/n


    # O trabajar en función de p que es más simple
    #paso M
    #r1<-(y1+y2+y31Star)/n; r2<-1-r1
        #c1<-(y1+y4)/n; c2<-(y2+y51Star)/n; c3<-1-c1-c2

      cat("\n", "Iter",k,": r1=",r1,",c1=",c1,", c2=",c2,sep="")
}
```

```
##
## Iter1: r1=0.36,c1=0.3, c2=0.19
## Iter2: r1=0.3605505,c1=0.3, c2=0.1894495
## Iter3: r1=0.3610844,c1=0.3, c2=0.1889156
## Iter4: r1=0.361602,c1=0.3, c2=0.188398
## Iter5: r1=0.3621036,c1=0.3, c2=0.1878964
## Iter6: r1=0.3625895,c1=0.3, c2=0.1874105
## Iter7: r1=0.3630601,c1=0.3, c2=0.1869399
## Iter8: r1=0.3635155,c1=0.3, c2=0.1864845
## Iter9: r1=0.3639562,c1=0.3, c2=0.1860438
## Iter10: r1=0.3643824,c1=0.3, c2=0.1856176
## Iter11: r1=0.3647944,c1=0.3, c2=0.1852056
## Iter12: r1=0.3651926,c1=0.3, c2=0.1848074
## Iter13: r1=0.3655773,c1=0.3, c2=0.1844227
## Iter14: r1=0.3659487,c1=0.3, c2=0.1840513
## Iter15: r1=0.3663073,c1=0.3, c2=0.1836927
## Iter16: r1=0.3666534,c1=0.3, c2=0.1833466
## Iter17: r1=0.3669872,c1=0.3, c2=0.1830128
## Iter18: r1=0.3673091,c1=0.3, c2=0.1826909
## Iter19: r1=0.3676194,c1=0.3, c2=0.1823806
## Iter20: r1=0.3679184,c1=0.3, c2=0.1820816
```

4.3 Ejercicio 3: Ejemplo de micropropagación de raices

En un estudio de micropropagación de raices, se anotó el número de raices de 40 variedades de manzana. Los datos observados fueron:

Nº raices	0	1	2	3	4	5	6	7	8	9
Frecuencia	19	2	2	4	3	1	4	3	0	2

Se ha establecido como modelo para describir este fenémeno una mixtura de dos distribuciones f_1 y f_2 con proporciones p y $1-p$ respectivamente. En concreto, f_1 toma el valor cero con probabilidad 1 ($f_1(0)=1$), y f_2 es la función de masa de probabilidad de una distribución de Poisson(λ). Obtener el EMV para los parámetros:

a. A partir de los datos observados de forma aproximada (ver Práctica 3).

```
y<-c(rep(0,19),rep(1,2),rep(2,2),rep(3,4),rep(4,3),rep(5,1),rep(6,4),rep(7,3),rep(9,2))


mlv<-function(theta){
  p<-theta[1]
  lambda<-theta[2]
  return(-(19*log(p+(1-p)*exp(-lambda))+21*log(1-p)-21*lambda+sum(y)*log(lambda)))
  #NOTA: sum(y), empieza en 0
}

vi<-c(0.5,1)
minimo<-optim(vi,mlv,hessian=TRUE)
EMV_optim<-minimo$par
EMV_optim
```

```
## [1] 0.4698821 4.6207927
```

b. A partir de los datos completos, utilizando el algoritmo EM con valores iniciales $(p_0, \lambda_0) = (0.5, 1)$ y 8 iteraciones.

```
n<-40
iter<-8
p0<-0.5
l0<-1
p<-c(p0,rep(0,iter))
l<-c(l0,rep(0,iter))

for(i in 2:(iter+1)){
  p[i]<-(19/n)*(p[i-1]/(p[i-1]+(1-p[i-1])*exp(-l[i-1]))) # Para el resto de
  # frecuencias b_i^* es cero
  l[i]<-mean(y)*(1/(1-p[i]))

      cat("\n", "Iter",i,": p=",p[i],",l=",l[i],sep="")
}
```

```
##
## Iter2: p=0.3472528,l=3.753367
## Iter3: p=0.4549552,l=4.495044
## Iter4: p=0.4687308,l=4.611598
## Iter5: p=0.4697103,l=4.620116
## Iter6: p=0.4697751,l=4.62068
## Iter7: p=0.4697793,l=4.620717
## Iter8: p=0.4697796,l=4.62072
## Iter9: p=0.4697796,l=4.62072
```

```
EMV_EM<-c(p[iter+1],l[iter+1])
EMV_EM
```

```
## [1] 0.4697796 4.6207200
```

4.4 Ejercicio 4: Muestreo por lotes

Muestras de suelo, seleccionadas aleatoriamente, se combinan en lotes y se examina si en los lotes se detecta o no la presencia de una toxina. Por tanto, las muestras de suelo individuales no se examinan. De 100 lotes, de 10 muestras de suelo cada uno, se detectó la toxina en 12 lotes. Calcular el EMV para p = probabilidad de que la toxina esté presente en una muestra de suelo individual

a. A partir de los datos observados, tanto analíticamente como utilizando la función optim (ver Práctica 2).

```
#EMV obtenido de forma analítica es EMV_ana<-0.0127 pHat=1-(1-thetaHat)^0.1

# EMV con optim
sY<-12
mlv<-function(theta){
  return(-sY*log(theta)-(100-sY)*log(1-theta))
}
vi<-0.1
minimo<-optim(0.1,mlv,method="Brent",lower=0.01,upper=0.99,hessian=TRUE)
EMV_optim<-minimo$par
EMV_optim
```

```
## [1] 0.12
```

b. A partir de los datos completos, utilizando el algoritmo EM con valor inicial 0.1 y 8 iteraciones.

```
iter<-8
p0<-0.1
p<-c(p0,rep(0,iter))

for(i in 2:(iter+1)){
  p[i]<-3*p[i-1]/(25*(1-(1-p[i-1])^10))
}
EMV_EM<-p
EMV_EM
```

```
## [1] 0.10000000 0.01842408 0.01302880 0.01272047 0.01270302 0.01270204 0.01270198
## [8] 0.01270198 0.01270198
```

4.5 Ejercicio 5: Modelo exponencial negativa

Se sabe que la duración de unos componentes eléctricos sigue una distribución exponencial negativa con función de densidad $f(x, \lambda) = \lambda e^{-\lambda x}, x > 0$. Con el propósito de estudiar la duración media de los componentes se ponen a funcionar 100 de ellos y se observa que la duración de todos los componentes fue mayor de 3 y menor de 10 unidades de tiempo. Obtener un EMV para la duración media de los componentes $(1/\lambda)$.

a. A partir de los datos observados, tanto analíticamente como de forma aproximada con optim.

```r
# El EMV calculado de forma analitica es
emv<-log(10/3)/7
1/emv
```

```
## [1] 5.814085
```

```r
#EMV obtenido de forma analítica es EMV_ana<-5.814

mlv<-function(lambda) {
    return(-100*log(exp(-3*lambda)-exp(-10*lambda)))
}

vi<-5
minimo<-optim(vi,mlv,hessian=T)
EMV<-minimo$par
EMV_optim<-1/EMV
EMV_optim # Menos preciso
```

```
## [1] 5.818182
```

b. A partir de los datos completos, utilizando el algoritmo EM con un valor inicial de 5 y 10 iteraciones.

```r
iter<-10
l0<-5
l<-c(l0,rep(l0,iter))
for(i in 2:(iter+1)){
  num<-exp(-3*l[i-1])*(3+1/l[i-1])-exp(-10*l[i-1])*(10+1/l[i-1])
  deno<-exp(-3*l[i-1])-exp(-10*l[i-1])
  B<-num/deno
  l[i]<-1/B
}
EMV_EM<-1/l
EMV_EM
```

```
##  [1] 0.200000 3.200000 5.315369 5.753114 5.807157 5.813304 5.813997 5.814075
##  [9] 5.814084 5.814085 5.814085
```

4.6 Ejercicio Propuesto 1: Modelo exponencial negativa censurado

Se ponen en funcionamiento 100 y se anota su tiempo de fallo hasta un instante de tiempo $t = 1$ minuto. Se sabe que el tiempo de fallo de estos aparatos se modela según un modelo exponencial negativo con función de densidad $f(x, \lambda) = \lambda e^{-\lambda x}, x > 0$. Concretamente, se anotaron los siguiente tiempos de fallo:

```r
tFallo<-c(0.59683969, 0.34819508, 0.63876960, 0.03998816, 0.54655953, 0.70678049,
          0.43481007, 0.75293272, 0.36484336, 0.30681922)
```

Obtener un EMV para la duración media de los componentes $(1/\lambda)$:

a. A partir de los datos observados, tanto analíticamente como de forma aproximada.

b. A partir de los datos completos, utilizando el algoritmo EM con valor inicial 2 y 8 iteraciones.

4.7 Ejercicio Propuesto 2: Avanzado (Cambio variable)

Se ponen en funcionamiento dos aparatos cuyos tiempos de fallo se modelan según una exponencial negativa de parámetro λ. Sin embargo, únicamente se dispone de la información del primer aparato que falla. Es decir se tiene que X_1, X_2 i.i.d $\sim exp(\lambda)$ son los datos completos e $Y = \min(X_1, X_2)$ los datos observados (incompletos). Calcular el EMV para λ:

a. A partir de los datos observados, tanto analíticamente como de forma aproximada.

b. A partir de los datos completos, utilizando el algoritmo EM. En el paso E, se requiere utilizar el teorema de cambio de variable multidimensional. Solución $\widehat{\lambda} = 0.25$.

5 Práctica 5: Bootstrap

5.1 Ejercicio 1: Simualción *versus* bootstrap

Sea $X_1, \dots, X_{43}$ i.i.d. tal que $X_i \sim N(3,1), i = 1, \dots, 43$. Considere $Z = \overline{X}^3 = (\frac{\sum_{i=1}^{n} X_i}{n})^3$, y obtenga la varianza muestral de Z:

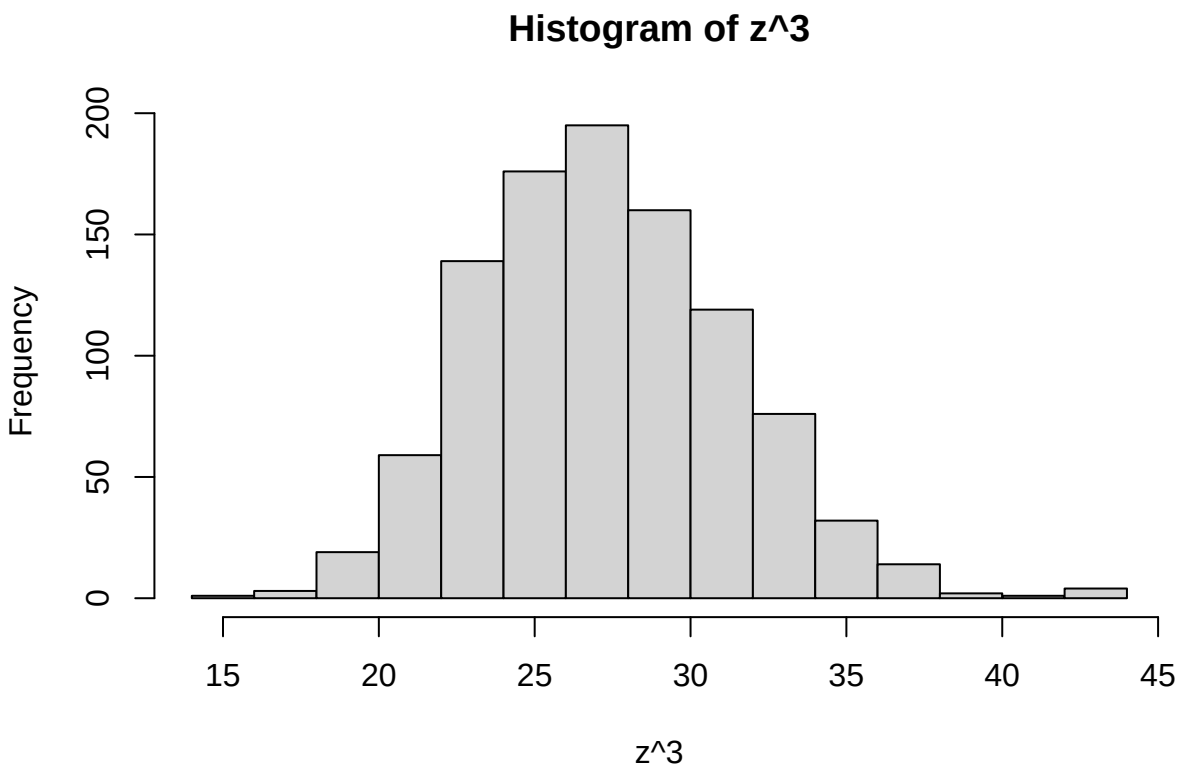
a. A partir de los conocimientos estudiados en IE1 e IEII ¿Es posible dar una expresión cerrada para calcular $Var(Z)$ de forma exacta? ¿Es posible calcular $Var(Z)$ de forma aproximada?

b. Por simulación $Var(Z)$.

```
# La simualcion permite aproximar cualquier caracteristica de la distribucion
n<-43
veces<-1000
y<-matrix(rnorm(n*veces,3,1),n,veces)
z<-apply(y,2,mean)
var(z^3)
```

```
## [1] 16.53763
```

```
par(mfrow=c(1,1))
hist(z^3)
```



c. Si únicamente se dispone de la muestra $X_1, \ldots, X_{43}$ y no se tiene información sobre su distribución.

```
# Bootstrap no parametrico (original)
set.seed(345)
x<-rnorm(n,3,1)
B<-1000

bootP<-matrix(0,B,length(x))
zStar<-c()
for(b in 1:B){
  bootP[b,]<-sample(x,length(x),replace=TRUE)
  zStar<-c(zStar,mean(bootP[b,])^3)
}
mean((zStar-mean(zStar))^2)
```

```
## [1] 15.84146
```

```
((B-1)/B)*var(zStar)# R trabaja con la quasi varianza, la teoria esta para la varianza
```

```
## [1] 15.84146
```

d. Si se dispone de la muestra $X_1, \ldots, X_{43}$ y además se sabe que su distribución es normal.

```r
# Bootstrap  parametrico (original)
#x<-rnorm(n,3,1)
B<-1000
muHat<-mean(x)
varHat<-var(x)

bootNP<-matrix(0,B,length(x))
zStarNP<-c()
for(b in 1:B){
  bootNP[b,]<-rnorm(n,muHat,sqrt(varHat))
  zStarNP<-c(zStarNP,mean(bootNP[b,])^3)
}
mean((zStarNP-mean(zStarNP))^2)
```

```
## [1] 17.63455
```

```r
((B-1)/B)*var(zStarNP)
```

```
## [1] 17.63455
```

e.- Considere $W = (Me(X_1, \dots, X_n))^3$, obtener el estimador bootstrap para $Sesgo(W)$. ¿Cómo es posible corroborar que el valor que obtenemos es correcto?

```r
# Bootstrap no parametrico (original)
set.seed(9321258)
x<-rnorm(n,3,1)
B<-1000

bootNP<-matrix(0,B,length(x))
wStar<-c()
for(b in 1:B){
  bootNP[b,]<-sample(x,length(x),replace=TRUE)
  wStar<-c(wStar,median(bootNP[b,])^3)
}
mean(wStar)-median(x)^3
```

```
## [1] -0.2830232
```

```r
mean(apply(y,2,median)^3)-3^3
```

```
## [1] 0.6326623
```

5.2 Ejercicio 2: Bootstrap. Modelo exponencial

Considerar una muestra de tamaño $n = 35$ de una distribución exponencial negativa de parámetro $\lambda = 0.5$ con función de densidad $f(x, \lambda) = \lambda \exp^{-\lambda x}, x > 0$.

a. Obtener el estimador bootstrap el sesgo y de la varianza del EMV de λ mediante bootstrap no paramétrico

```r
# Verdadero valor del parametro
lambda<-0.5
n<-35
x<-rexp(n,lambda)# datos
emv<-1/mean(x)#estimador

# Bootstrap no parametrico
B<-1000
mBootNP<-matrix(0,B,n)
emvStar<-rep(0,B)       #las B versiones bootstrap del estimador
for(b in 1:B){
    mBootNP[b,]<-sample(x,n,replace=T)
    emvStar[b]<-1/mean(mBootNP[b,])
}

# Estimador bootstrap de la varianza del EMV
mean((emvStar-mean(emvStar))^2)
```

```
## [1] 0.005845289
```

```r
vboot<-((B-1)/B)*var(emvStar)
vboot
```

```
## [1] 0.005845289
```

```r
# Estimador bootstrap del sesgo del EMV
sesgoboot<-mean(emvStar)-emv
sesgoboot
```

```
## [1] 0.0104515
```

b. Obtener el estimador bootstrap el sesgo y de la varianza del EMV mediante bootstrap paramétrico

```r
emv<-1/mean(x)#estimador

# Bootstrap no parametrico
B<-1000
mBootP<-matrix(0,B,n)
emvStarP<-rep(0,B)      #las B versiones bootstrap del estimador
for(b in 1:B){
    mBootP[b,]<-rexp(n,emv)
    emvStarP[b]<-1/mean(mBootP[b,])
}

# Estimador bootstrap de la varianza del EMV
mean((emvStarP-mean(emvStarP))^2)
```

```
## [1] 0.003932946
```

```r
vbootP<-((B-1)/B)*var(emvStarP)
vbootP
```

```
## [1] 0.003932946
```

```r
# Estimador bootstrap del sesgo del EMV
sesgobootP<-mean(emvStarP)-emv
sesgobootP
```

```
## [1] 0.01121677
```

c. Calcular mediante simulación, el sesgo y la varianza del estimador propuesto basado en una muestra de tamaño $n = 35$.

```r
# Estos cálculos se pueden hacer exactos utilizando la distribución gamma
# Exponencial(lambda~gama(1,lambda)

# Se obtiene: sesgo(emv)=lambda/(n-1) =0.0147  (para n=35, y lambda=0.5),
# desigualdad Jensen
# var(emv)=(lambda^2)*n^2/((n-1)^2)*(n-2)=0.0080    (para n=35, y lambda=0.5)


lambda<-0.5
n<-35
x<-rexp(n,lambda)# datos
emv<-1/mean(x)#estimador


lambda<-0.5
veces<-1000
y<-matrix(rexp(veces*n,lambda),veces,n) #generar 1000 muestras de tamaño n=35
# de la exponencial negativa 0.5.
lSim<-apply(y,1,mean)
lSim<-1/lSim
varestimador<-((veces-1)/veces)*var(lSim)
sesgoestimador<-mean(lSim)-lambda


varestimador # (lambda^2)*n^2/(((n-1)^2)*(n-2))=0.008
```

```
## [1] 0.008057114
```

```r
sesgoestimador # lambda/(n-1) =0.0147
```

```
## [1] 0.01386419
```

```r
# Varianza
(lambda^2)*n^2/(((n-1)^2)*(n-2))
```

```
## [1] 0.008027944
```

```r
# Sesgo
lambda/(n-1)
```

```
## [1] 0.01470588
```

5.3 Ejercicio 3: IC y CH Boostrap. Modelo normal

Considerar una muestra de tamaño $n = 30$ de una distribución normal estándar.

a. Calcular IC (percentil) bootstrap paramétrico de nivel $\alpha = 0.05$ para la media poblacional. Compara con el IC estandar que utilizarías en este caso.

```r
set.seed(12345)
x<-rnorm(30)
n<-30
B<-2000

mu<-mean(x)
ds<-sd(x)
Z<-matrix(rnorm(n*B,mu,ds),B,n)

muStar<-apply(Z,1,mean)


ext.inferior1<-quantile(muStar,0.025)
ext.superior1<-quantile(muStar,0.975)
ext.inferior1
```

```
##       2.5%
## -0.2600999
```

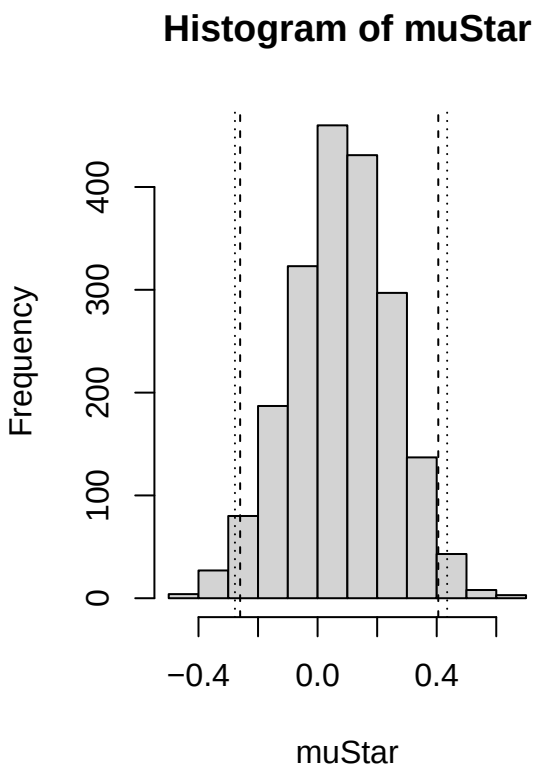
```r
ext.superior1
```

```
##    97.5%
## 0.405254
```

```r
#IC (t-Student) para la media es:
mu+c(-1,1)*qt(0.975,n-1)*ds/sqrt(n-1)
```

```
## [1] -0.2775214  0.4351354
```

```r
# Distribucion (histograma)
par(mfrow=c(1,2))
hist(muStar)
abline(v=c(ext.inferior1,ext.superior1),lty=2)#dashed
abline(v=c(mu+c(-1,1)*qt(0.975,n-1)*ds/sqrt(n-1)),lty=3)#dotted
```



b. Calcular IC (percentil) bootstrap no paramétrico de nivel $\alpha = 0.05$ para la media poblacional. Compara con el IC estandar que utilizarías en este caso.

```
Z<-matrix(sample(x,n*B,replace=TRUE),B,n)
T<-apply(Z,1,mean)

ext.inferior2<-quantile(T,0.025)
ext.superior2<-quantile(T,0.975)
ext.inferior2
```

```
##       2.5%
## -0.2418123
```

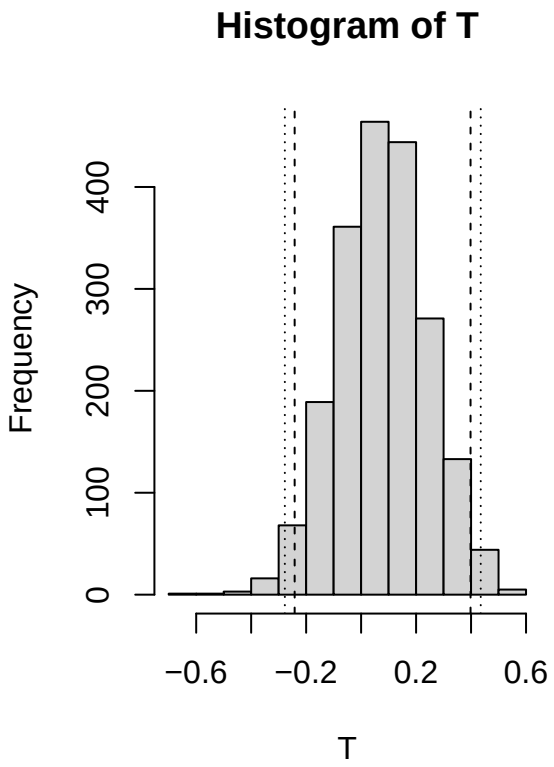
```
ext.superior2
```

```
##     97.5%
## 0.3981169
```

```
#IC (t-Student) para la media es:
mu+c(-1,1)*qt(0.975,n-1)*ds/sqrt(n-1)
```

```
## [1] -0.2775214  0.4351354
```

```r
# Distribucion (histograma)
par(mfrow=c(1,2))
hist(T)
abline(v=c(ext.inferior2,ext.superior2),lty=2)
abline(v=c(mu+c(-1,1)*qt(0.975,n-1)*ds/sqrt(n-1)),lty=3)
```



c. Considere $\varphi = \mu^3 + \mu^5$. Calcular IC (percentil) bootstrap no paramétrico de nivel $\alpha = 0.05$ para φ. Compara con el IC estandar que utilizarías en este caso. ¿Por qué hay diferencias? ¿Cuál es mejor?

```r
Z<-c()#marix(0,B,length(x))
for(j in 1:B){
  Z<-c(Z,
    (mean(sample(x,length(x),replace=TRUE)))^3+(mean(sample(x,length(x),replace=TRUE)))^5)
}

ext.inferior2<-quantile(Z,0.025)
ext.superior2<-quantile(Z,0.975)
ext.inferior2
```

```
##        2.5%
## -0.01458542
```

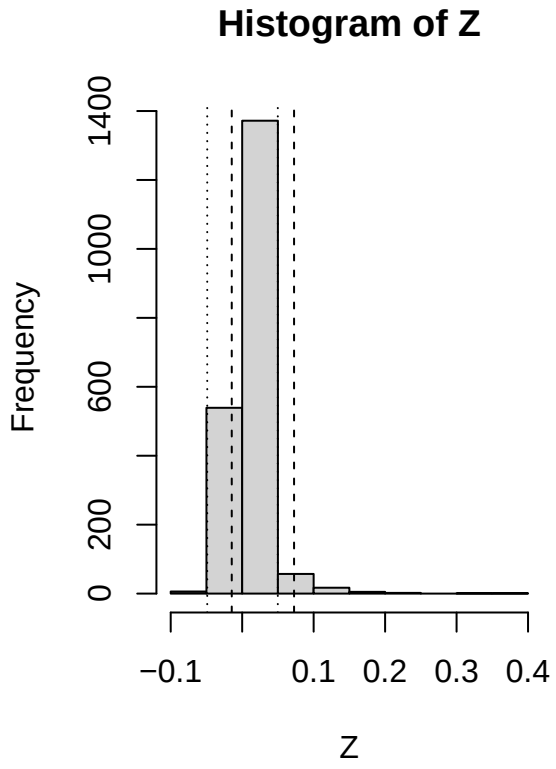
```r
ext.superior2
```

```
##      97.5%
## 0.07262164
```

```
#IC (normal) para la media es:
mean(x)^3+mean(x)^5+c(-1,1)*qnorm(0.975)*sd(Z)
```

```
## [1] -0.04888045  0.04986540
```

```
# Distribucion (histograma)
par(mfrow=c(1,2))
hist(Z)
abline(v=c(ext.inferior2,ext.superior2),lty=2)
abline(v=c(mean(x)^3+mean(x)^5+c(-1,1)*qnorm(0.975)*sd(Z)),lty=3)
```



d. A partir de los datos dados para el modelo normal. Obtener el valor crítico y el p-valor bootstrap para contrastar $H_0 : \mu = 0$ *vs* $H_1 : \mu \neq 0$ (σ desconocida). Consider $\alpha = 0.05$.

```
# El estadístico test es T=mean(x)*sqrt(n)/sd(x). Región crítica (|T|>C).
# Usando estimador de la quasivarianza. Si uso el de la varianza (como en teoria)
# sd(.)*(n-1)/n porque R trabaja siempre con la quasivarianza
# El valor crítico exacto para nivel 0.05 es: qt(0.975,29)=2.03
# El pvalor exacto es:  2*pt(-tobs,29), siendo tobs el valor observado de |T|

# Valor observado del estadiatico
tobs<-abs(mean(x)*sqrt(n)/sd(x))
tobs
```

```
## [1] 0.4600639
```

```
# Valor critico
qt(0.975,n-1)
```

```
## [1] 2.04523
```

```
# Pvalor
2*pt(-tobs,n-1)
```

```
## [1] 0.6489013
```

```
# d1) Bootstrap parametrico
muB<-rep(0,B)
sdB<-rep(0,B)
tobsB<-rep(0,B)
for(b in 1:B){
    muestra<-rnorm(n,mu,ds)
    muB[b]<-mean(muestra)
    sdB[b]<-sd(muestra)
    tobsB[b]<-abs(muB[b]*sqrt(n)/sdB[b])
}
vcB<-quantile(tobsB,probs=0.95) # Podeis comprobarlo subiendo
#n y B en un bootstrap paramétrico que este valor es 1.96
vcB
```

```
##      95%
## 2.222528
```

```
pvalorB1<-mean(tobsB>=tobs)
pvalorB1 # ¿Se deberia rechazar?
```

```
## [1] 0.6795
```

```
# d2) Bootstrap no parametrico
tobs<-abs(mu*sqrt(n)/ds)
muB<-rep(0,B)
sdB<-rep(0,B)
tobsB<-rep(0,B)
for(b in 1:B){
    muestra<-sample(x,length(x),replace=TRUE)
    muB[b]<-mean(muestra)
    sdB[b]<-sd(muestra)
    tobsB[b]<-abs(muB[b]*sqrt(n)/sdB[b])
}
vcB<-quantile(tobsB,0.95)
vcB
```

```
##      95%
## 2.286952
```

```
pvalorB2<-mean(tobsB>=tobs)
pvalorB2 # ¿Deberia estar de acuerdo con el bootstrap parametrico?
```

```
## [1] 0.671
```

5.4 Ejercicio Propuesto 1: Ejercicio de entrega del procesador (mixtura de normales)

En un procesador se está ejecutando un proceso de simulación, en el que un procedimiento estadístico es repetido hasta completar un millón de repeticiones. En alguna de estas repeticiones el proceso falla, aunque se ha programado para que el proceso no se detenga y descarte dicha repetición sustituyéndola por otra.

Tras observar un total de $n = 272$ tiempos de fallo durante dos semanas, la experiencia del investigador le hace pensar que los fallos provienen de dos causas diferentes. Concretamente, afirma que los tiempos de espera hasta que se produce fallo (Y) pueden modelarse como una mixtura de dos normales, es decir que cada $Y_i, i = 1, \dots, 272$ proviene bien de una distribución $N(\mu, \sigma)$ con probabilidad p, o bien de una distribución $N(\nu, \tau)$ con probabilidad $1 - p$.

A partir de los datos de la muestra de tiempos tiempo de fallo facilitada (tiempoFallo.RData) y los valores iniciales $(p_0, \mu_0, \sigma_0, \nu_0, \tau_0) = (0.3, 50, 4, 82, 6)$:

a. Obtenr IC Wald con confianza aproximada del 95% para los cinco parámetros a partir de la función optim.

b. Obtener IC 95% bootstrap (percentil) para los cinco parámetros, comparando los resultados con los obtenidos en el apartado a).

5.5 Ejercicio Propuesto 2: Adaptación Ejercicio 1 y 2 Práctica 3

Obtener el valor crítico y el p-valor bootstrap para resolver los contrastes de hipótesis propuestos en los apartados 1d) y 2g) de la Práctica 3.

6 Práctica 6: Tests de bondad de ajuste

6.1 Ejercicio 1: Adaptación Ejercicio 15

Para los datos: 1.0,2.3,4.2,7.1,10.4, utilizar el procedimiento más adecuado para contrastar la hipótesis nula:

a. Exponencial, $F_0(x) = 1-\exp(-\lambda x), x > 0$. Tanto a partir de los valores críticos de las tablas (en papel); como a partir de los valores críticos simulados para distintos n y α y concluir a partir de ellos.

```
# Datos
x<-c(1.0,2.3,4.2,7.1,10.4)

# H0 compuesta
#lillie.test unicamente para normalidad.

# Para obtener el estadistico podemos. Tambien nos da el pvalor, pero..
emv<-1/mean(x)
ks.test(x,pexp,emv) # Solo puedo usar estadistico. P-valor con la tabla simulada (Exp)
```

```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.18127, p-value = 0.9864
## alternative hypothesis: two-sided
```

```
ks.test(x/mean(x),pexp) # Mismo valor
```

```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  x/mean(x)
## D = 0.18127, p-value = 0.9864
## alternative hypothesis: two-sided
```

```
#Opcion 1: Obtenido el estadistico podemos ver taba Lilliefors
#para exponencial para obtener p-valor

#Opcion 2: Obtenido el estadistico podemos comparar con
#valores criticos simulados para Lilliefors (Exp)
# Tabla de Lilliefors para exponencial

# Modificada para n y alpha
enes<-c(4:12)
alphas<-c(0.95,0.9,0.75,0.5,0.25,0.10,0.075,0.05,0.001)
DnExp<-matrix(0,length(enes),length(alphas))
rownames(DnExp)<-as.character(enes)
```

```
colnames(DnExp)<-as.character(alphas)

fila<-0
for(k in enes){
    fila<-fila+1
    columna<-0
    for(j in alphas){
        columna<-columna+1
        matrixExp<-matrix(0,1000,k)
        Dns<-c()
        for(i in 1:1000){
            matrixExp[i,]<-rexp(k,emv) # Si en lugar de emv=0.2 fuese 5, que ocurre?
            xBar<-mean(matrixExp[i,])
            z<-matrixExp[i,]/xBar
            Dns<-c(Dns,ks.test(z,pexp)$statistic)
            #Dns<-c(Dns,ks.test(x,pexp,1/xBar)$statistic) # O tambien asi
        }
        DnExp[fila,columna]<-quantile(Dns,probs=1-j)
    }
}
DnExp
```

```
##         0.95       0.9      0.75       0.5      0.25       0.1     0.075
## 4  0.2069371 0.2288042 0.2597365 0.3171555 0.3843199 0.4455572 0.4634562
## 5  0.1910572 0.2058338 0.2360017 0.2838203 0.3456504 0.3965692 0.4293864
## 6  0.1691007 0.1855399 0.2163860 0.2620009 0.3170538 0.3719240 0.3911735
## 7  0.1640370 0.1750201 0.2077705 0.2436922 0.2946776 0.3431632 0.3614422
## 8  0.1488749 0.1636815 0.1948650 0.2305992 0.2769633 0.3239138 0.3422685
## 9  0.1450943 0.1550700 0.1866051 0.2223101 0.2689982 0.3095579 0.3258786
## 10 0.1377136 0.1509978 0.1743076 0.2085429 0.2504047 0.2976493 0.3087145
## 11 0.1281301 0.1435580 0.1673556 0.1976609 0.2480038 0.2838694 0.2881446
## 12 0.1273735 0.1378888 0.1604868 0.1912535 0.2296707 0.2717339 0.2773393
##         0.05     0.001
## 4  0.4829874 0.6596613
## 5  0.4315956 0.5373104
## 6  0.4073629 0.5595469
## 7  0.3869862 0.4826246
## 8  0.3596821 0.5498354
## 9  0.3474883 0.4698641
## 10 0.3179602 0.4410383
## 11 0.3158808 0.4202747
## 12 0.2958870 0.4314258
```

b. Calcular el p-valor por simulación para el test dado en a).

```
xBar<-mean(x)
tobs<-ks.test(x,pexp,1/xBar)$statistic
DnsH0<-c()
simExp<-matrix(0,1000,length(x))
# Lo repito como si no hubieramos hecho a), pero bastaria con considerar Dns
for(i in 1:1000){
  simExp[i,]<-rexp(length(x),1/xBar)
    DnsH0<-c(DnsH0,ks.test(simExp[i,],pexp,1/simExp[i,])$statistic)
```

```
}
pvalExp<-sum(DnsH0>tobs)/1000
pvalExp
```

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

c. Normal. Tanto desde funciones específicas de R, como a partir de los valores críticos de las tablas (en papel); y también desde de los valores críticos simulados para distintos n y α y concluir a partir de ellos.

```
# H0 compuesta
m<-mean(x)
s<-sd(x)

# Datos tipificados
z<-(x-m)/s

# Podemos hacerlo varias formas
# 1.- Usando el estadistico y p-valor de lillie.test de la libreria normtest
library(nortest)
lillie.test(x) # El p-valor sirve. Mismo valor
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  x
## D = 0.18356, p-value = 0.8379
```

```
lillie.test(z) # El p-valor sirve
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  z
## D = 0.18356, p-value = 0.8379
```

```
# 2.- Usando el estadistico de ks.test (para el estadistico)
#y las tablas Lilliefors (para pvalor)
ks.test(x,pnorm,m,s) # El p-valor no sirve. Mismo valor
```

```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.18356, p-value = 0.9844
## alternative hypothesis: two-sided
```

```
ks.test(z,pnorm) # El p-valor no sirve
```

```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  z
## D = 0.18356, p-value = 0.9844
## alternative hypothesis: two-sided
```

```
# 3.- Usando el estadistico de ks.test y las tablas Lilliefors simuladas (Norm)
enes<-c(4:12)
alphas<-c(0.95,0.9,0.85,0.8,0.75,0.5,0.25,0.1,0.05,0.01)
DnNorm<-matrix(0,length(enes),length(alphas))
rownames(DnNorm)<-as.character(enes)
colnames(DnNorm)<-as.character(alphas)
set.seed(1234)
fila<-0
for(k in enes){
    fila<-fila+1
    columna<-0
    for(j in alphas){
        columna<-columna+1
        matrixNorm<-matrix(0,1000,k)
        Dns<-c()
        for(i in 1:1000){
            matrixNorm[i,]<-rnorm(k,m,s)
            xBar<-mean(matrixNorm[i,])
            st<-sd(matrixNorm[i,])
            z<-(matrixNorm[i,]-xBar)/st
            Dns<-c(Dns,ks.test(z,pnorm)$statistic) # Se podria haber usado lillie.test
        }
        DnNorm[fila,columna]<-quantile(Dns,probs=1-j)
    }
}
DnNorm
```

```
##         0.95       0.9      0.85       0.8      0.75       0.5      0.25
## 4  0.1669930 0.1866025 0.1955884 0.2115697 0.2192155 0.2579405 0.2943471
## 5  0.1641690 0.1791922 0.1835936 0.1936115 0.2025611 0.2311249 0.2756100
## 6  0.1496122 0.1662484 0.1708479 0.1769048 0.1854476 0.2200505 0.2579886
## 7  0.1410891 0.1522087 0.1624851 0.1666594 0.1746485 0.2083721 0.2413215
## 8  0.1354395 0.1451497 0.1516915 0.1599003 0.1610306 0.1946597 0.2302603
## 9  0.1274311 0.1395766 0.1425364 0.1521915 0.1562692 0.1849371 0.2176267
## 10 0.1182865 0.1318627 0.1398881 0.1440289 0.1485597 0.1780265 0.2054360
## 11 0.1168919 0.1253377 0.1335612 0.1395344 0.1446921 0.1665040 0.1989754
## 12 0.1104904 0.1215888 0.1257444 0.1321678 0.1373488 0.1644175 0.1926626
##          0.1      0.05      0.01
## 4  0.3389818 0.3719604 0.4138511
## 5  0.3176857 0.3411115 0.4011310
## 6  0.3019926 0.3198396 0.3727551
## 7  0.2797531 0.2956719 0.3439593
## 8  0.2671091 0.2799000 0.3325922
## 9  0.2485654 0.2717517 0.3127179
## 10 0.2424919 0.2546090 0.2897199
## 11 0.2283164 0.2465327 0.2875371
## 12 0.2230690 0.2505217 0.2749604
```

```
################################
#        NOTA                  #
################################
# La libreia nortest permite calcular otros test
# cvm.test() # Cramer von Mises , n>7
# ad.test()  # Anderson Darling , n>7
# sftest()   # Shapiro Francia, se rechaza para valores pequenos
# shapiro.test() # No esta en la libreria nortest
```

d. Calcular por simulación la potencia para la alternativa Exp(7) para el test dado en c) de nivel $\alpha = 0.05$.

```
vc<-0.344 # A partir de la tabla simulada en c). Tambien 0.344 de la tabla Lilliefors (Norm)
DnsH1<-c()
simExp1<-matrix(0,1000,length(x))
for(i in 1:1000){
  simExp1[i,]<-rexp(length(x),7)
    DnsH1<-c(DnsH1,lillie.test(simExp1[i,])$statistic)
}
potencia<-sum(DnsH1>vc)/1000
potencia
```

```
## [1] 0.122
```

```
# ¿Como puede conseguirse aumentar potencia? 2 maneras.
#Disminuir alpha (no recomendado) o aumentar n


################################
#        NOTA                  #
################################
# Los valores criticos, pvalores y potencias pueden simularse
#para cualquier distribucion que conozcamos el estadistico
```

6.2 Ejercicio 2: Adaptación Ejercicio 16

Diez estudiantes se someten a un test. Los resultados (sobre 100) del test son 95,80,40,52,60,80,82,58,65,50.

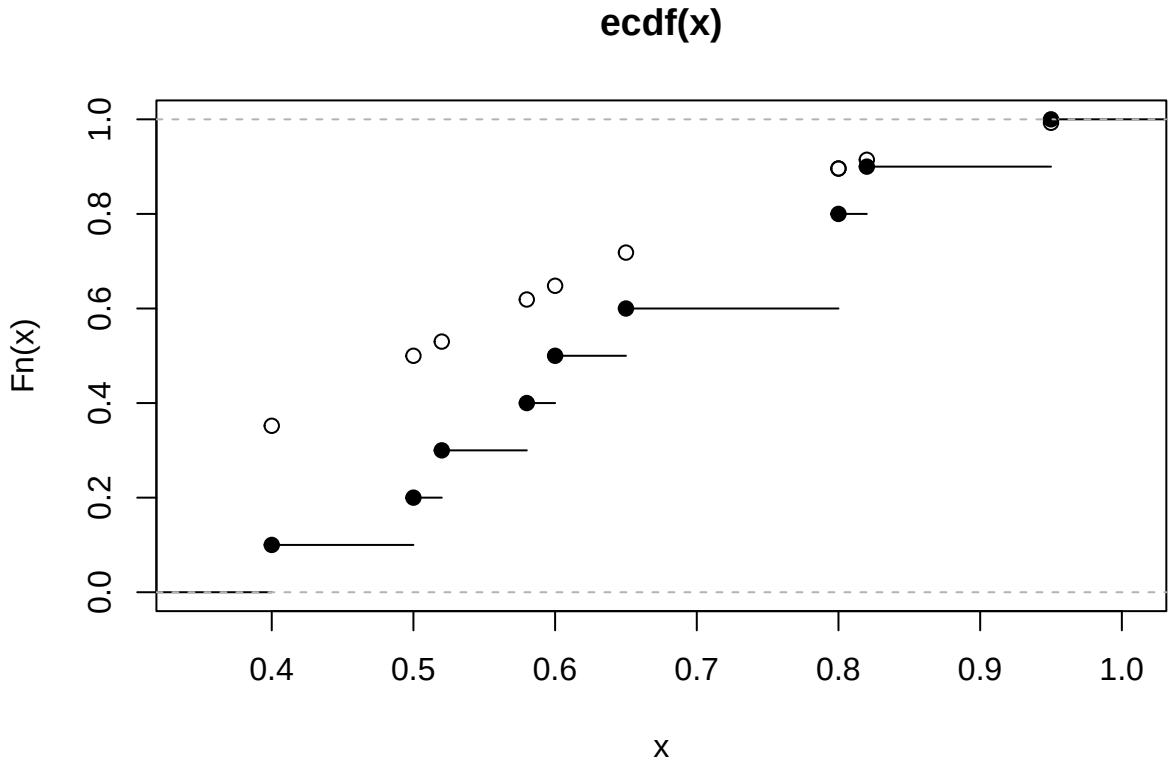
a. Contrastar la hipótesis nula $H_0 : F_0(x) = \begin{cases} 0 & si \quad x < 0 \\ x^2(3-2x) & si \quad 0 \leq x \leq 1 \\ 1 & si \quad x > 1 \end{cases}$, considere $\alpha = 0.05$.

```
# Necesitamos los datos sobre 1 punto
x<-c(95,80,40,52,60,80,82,58,65,50)/100
n<-length(x)

F0<-function(z){
  return(z^2*(3-2*z))
}
```

```r
#ordenamos lo datos y calculo F0 para datos ordendos
xo<-sort(x)
F0(xo)
```

```
##  [1] 0.352000 0.500000 0.529984 0.618976 0.648000 0.718250 0.896000 0.896000
##  [9] 0.914464 0.992750
```

```r
plot(ecdf(x)) # Fn
points(xo,F0(xo)) # F0
```




```r
# ¿Se rechazara H0?

# Estadistico K-S. Calculado directamente con la definicion
max((1:10)/n-F0(xo))
```

```
## [1] 0.00725
```

```r
max(F0(xo)-(0:9)/n)
```

```
## [1] 0.4
```

```
Dn<-max(max((1:10)/n-F0(xo)),max(F0(xo)-(0:9)/n))
Dn
```

```
## [1] 0.4
```

```
# Calculamos el p-valor usando las tablas
# Entre 0.05 y 0.1

# Estadistico K-S y pvalor calculado con la funcion ks.test
ks.test(x,F0)
```

```
## Warning in ks.test.default(x, F0): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.4, p-value = 0.08152
## alternative hypothesis: two-sided
```

```
# Tambien podemos usar solo el estadistico y
#buscar el p-valor en las tablas K-S menos preciso

# Podemos calcular ese pvalor por simulacion. Mandar como ejercicio de entrega
```

b.Obtener la potencia del test de K-S del apartado a) por simulación frente a la alternativa Beta(2,4).

```
# Valor critico del test de K-S tablas o `ks.test() o por simuladcion si lo tuvieramos
vc<-0.409 # De las tablas de la distribucion exacta de K-S

DnH1<-c()
for(i in 1:1000){
    sampleBeta<-rbeta(10,2,4)
    DnH1<-c(DnH1,ks.test(sampleBeta,F0)$statistic)
}
sum(DnH1>vc)/1000
```

```
## [1] 0.602
```

c. Calcular mediante simulación el tamagno muestral necesario para que la potencia sea al menos de 0.95.

```
# 1.- Lo hacemos por simulacion, tomamos los valores criticos de las tablas
nc<-c(10,20,30,40,50,60,70,80,90,100)
vcs<-c(0.409,0.294,0.242,0.210,1.36/sqrt(nc[5:10]))

potencia<-c()
for(k in 1:length(nc)){
  DnH1<-c()
  for(i in 1:1000){
    sampleBeta<-rbeta(nc[k],2,4)
```

```
    DnH1<-c(DnH1,ks.test(sampleBeta,F0)$statistic)
  }
  potencia[k]<-sum(DnH1>vcs[k])/1000
}
potencia
```

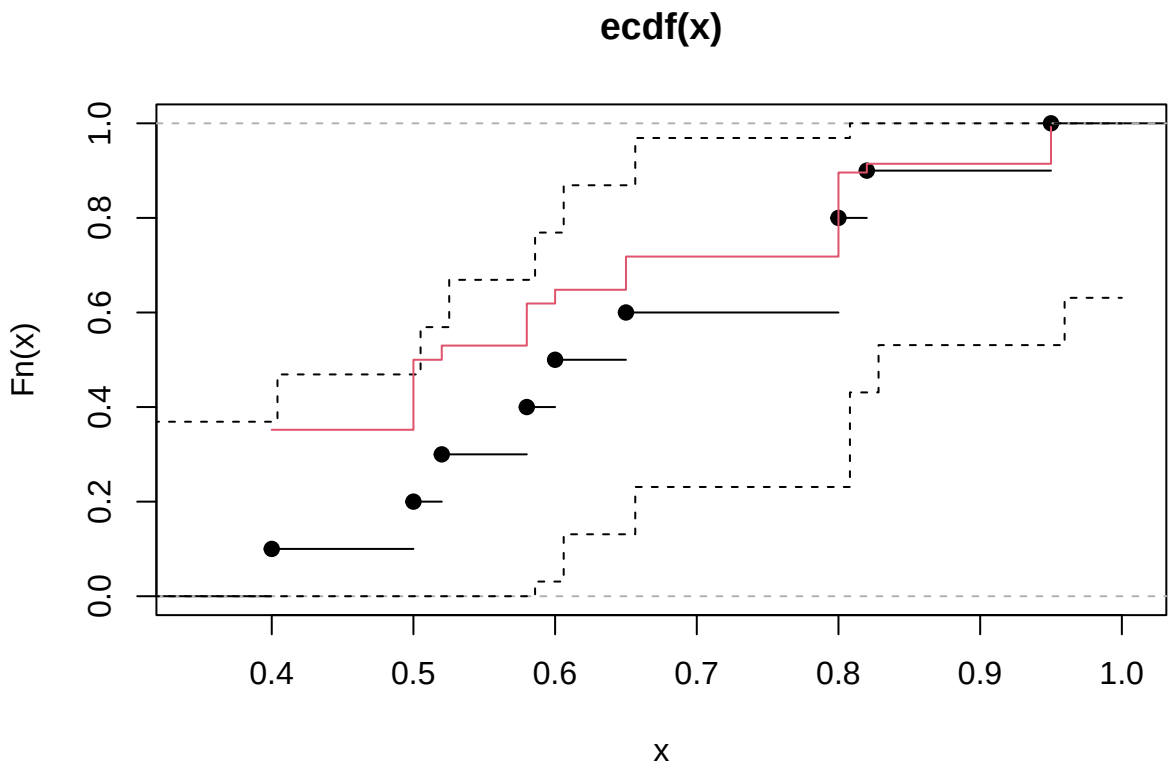
```
##  [1] 0.611 0.894 0.990 0.997 1.000 1.000 1.000 1.000 1.000 1.000
```

```
# Entre n=20 y n=30, puedo afinar un poco mas entre estos
#dos valores usando los valores criticos de la tabla
nc<-21:29
vcs<-c(0.287,0.281,0.275,0.269,0.264,0.259,0.254,0.250,0.246)

potencia<-c()
for(k in 1:length(nc)){
  DnH1<-c()
  for(i in 1:1000){
    sampleBeta<-rbeta(nc[k],2,4)
    DnH1<-c(DnH1,ks.test(sampleBeta,F0)$statistic)
  }
  potencia[k]<-sum(DnH1>vcs[k])/1000
}
potencia # n=25
```

```
## [1] 0.922 0.923 0.938 0.946 0.961 0.964 0.967 0.974 0.975
```

d. Obtener la banda de confianza simultanea para la función de distribución desconocida con confianza 0.9 ¿Contiene a la función de distribución bajo la H_0 dada en el apartado a)?

```
Fn<-ecdf(x)
plot(Fn)
ceros<-rep(0,100)
unos<-rep(1,100)
z<-seq(0,1,length=100)
L<-pmax(ceros,Fn(z)-0.369) # De donde sale este valor
U<-pmin(unos,Fn(z)+0.369)
lines(z,L,type="s",lty=2)
lines(z,U,type="s",lty=2)
lines(xo,F0(xo),type="s",col=2)
```



6.3 Ejercicio 3: Adaptación Ejercicio 13

Se lanzan 5 monedas en 100 ocasiones y se anota el número de caras en cada lanzamiento. Los resultados fueron:

Nº caras	0	1	2	3	4	5
Frecuencias	1	10	20	36	23	10

a. Estudiar si los datos soportan o no la hipótesis de que todas las monedas son equilibradas.

```
# H0: b(5,0.5) simple

n<-100
frecO<-c(1,10,20,36,23,10)
frecE<-c(dbinom(0:5,5,0.5)*n) # 5 o mas caras
frecE # <=5
```

```
## [1]  3.125 15.625 31.250 31.250 15.625  3.125
```

```
# Agrupamos la primera y la ultima clase
n<-100
frecO<-c(11,20,36,33)
frecE<-c(sum(dbinom(0:1,5,0.5))*n,dbinom(2:3,5,0.5)*n,sum(dbinom(4:5,5,0.5))*n)
frecE
```

```
## [1] 18.75 31.25 31.25 18.75
```

```
# Como en la teoria
tObs<-sum((frecO-frecE)^2/frecE)
tObs
```

```
## [1] 18.80533
```

```
pval<-1-pchisq(tObs,3)
pval
```

```
## [1] 0.0002999421
```

```
# Directamente
chisq.test(x=frecO,p=frecE/n)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  frecO
## X-squared = 18.805, df = 3, p-value = 0.0002999
```

b. Calcular la potencia en la alternativa Bin(5,0.55). ¿Cuánto vale n para que la potencia sea de 0.7?

```
p0<-frecE/n
p1<-c(sum(dbinom(0:1,5,0.55)),dbinom(2:3,5,0.55),sum(dbinom(4:5,5,0.55)))
D<-sum((p0-p1)^2/p0)
delta<-n*D
1-pchisq(qchisq(0.95,3),3,delta)
```

```
## [1] 0.4270591
```

```
# Calculamos n, usando las tablas
deltaNuevo<-8.792
nNuevo<-deltaNuevo/D
ceiling(nNuevo)
```

```
## [1] 182
```

c. Estudiar si los datos soportan o no la hipótesis de que todas las monedas tienen la misma probabilidad de cara.

```
# H0: b(5,p) compuesta

# EMV para p en base a los datos agrupados
n<-100
frecO<-c(11,20,36,33)

mlv<-function(p){
  return(-sum(frecO*log(c(sum(dbinom(0:1,5,p)),dbinom(2:3,5,p),sum(dbinom(4:5,5,p))))))
```

```
}
minimo<-optim(0.5,mlv,method="Brent",lower=0.01,upper=0.99)
emv<-minimo$par


p0<-c(sum(dbinom(0:1,5,emv)),dbinom(2:3,5,emv),sum(dbinom(4:5,5,emv)))

# Calculamos el estadistico y el pvalor
tobs<-chisq.test(x=frec0,p=p0)$statistic # Sirve este pvalor...
1-pchisq(tobs,6-1-1-1-1)
```

```
## X-squared
##  0.611787
```

6.4 Ejercicio 4: Adaptación Ejercicio 12

En la generación de 100 observaciones de una distribución Binomial Negativa [BN(2,2/3)], se obtuvieron los resultados siguientes:

Valores	0	1	2	3	4 o mas
Frecuencias	40	24	16	14	6

a. Contrastar el ajuste a una distribución de Poisson de media 1 (ignoramos binomial negativa).

```
n<-100
frec0<-c(40,24,16,14,6)
frecE<-c(dpois(0:3,1)*n,(1-ppois(3,1))*n)
frecE # <=5
```

```
## [1] 36.787944 36.787944 18.393972  6.131324  1.898816
```

```
chisq.test(x=frec0,p=frecE/n)
```

```
## Warning in chisq.test(x = frec0, p = frecE/n): Chi-squared approximation may be
## incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  frec0
## X-squared = 23.994, df = 4, p-value = 8.011e-05
```

```
# Agrupamos la ultima clase n<=5
```

b. Repetir el ajuste, agrupando clases hasta que el nº esperado de observaciones en cada clase sea mayor que 5.

```r
n<-100
frecO<-c(40,24,16,20)
frecE<-c(dpois(0:2,1)*n,(1-ppois(2,1))*n)
frecE
```

```
## [1] 36.78794 36.78794 18.39397  8.03014
```

```r
chisq.test(x=frecO,p=frecE/n)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  frecO
## X-squared = 22.88, df = 3, p-value = 4.278e-05
```

```r
# Agrupamos la ultima clase n<=10
```

c. Repetir el ajuste, agrupando clases hasta que el nº esperado de observaciones en cada clase sea mayor que 10.

```r
n<-100
frecO<-c(40,24,36)
frecE<-c(dpois(0:1,1)*n,(1-ppois(1,1))*n)
frecE
```

```
## [1] 36.78794 36.78794 26.42411
```

```r
chisq.test(x=frecO,p=frecE/n)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  frecO
## X-squared = 8.1959, df = 2, p-value = 0.01661
```

d. Estudiar el efecto que tiene sobre la potencia, en la alternativa BN(2,2/3), considerar unas clases u otras. Tomar el nivel 0.05 para el test.

```r
p0_5clases<-c(dpois(0:3,1),(1-ppois(3,1)))
p0_4clases<-c(dpois(0:2,1),(1-ppois(2,1)))
p0_3clases<-c(dpois(0:1,1),(1-ppois(1,1)))

p1_5clases<-c(dnbinom(0:3,2,2/3),(1-pnbinom(3,2,2/3)))
p1_4clases<-c(dnbinom(0:2,2,2/3),(1-pnbinom(2,2,2/3)))
p1_3clases<-c(dnbinom(0:1,2,2/3),(1-pnbinom(1,2,2/3)))

D_5clases<-sum(((p0_5clases-p1_5clases)^2)/p0_5clases)
D_4clases<-sum(((p0_4clases-p1_4clases)^2)/p0_4clases)
D_3clases<-sum(((p0_3clases-p1_3clases)^2)/p0_3clases)
```

```
delta_5clases<-n*D_5clases
delta_4clases<-n*D_4clases
delta_3clases<-n*D_3clases

vc_5clases<-qchisq(0.95,4)
vc_4clases<-qchisq(0.95,3)
vc_3clases<-qchisq(0.95,2)

1-pchisq(vc_5clases,4,delta_5clases)
```

```
## [1] 0.5635264
```

```
1-pchisq(vc_4clases,3,delta_4clases)
```

```
## [1] 0.4296486
```

```
1-pchisq(vc_3clases,2,delta_3clases)
```

```
## [1] 0.3211214
```

e. ¿Cuál debe ser el tamaño muestral para que la potencia sea al menos 0.9, utilizando 5 clases?

```
deltaNuevo_5clases<-15.405 # Tabla chi2 descentrada
nNuevo_5clases<-deltaNuevo_5clases/D_5clases
ceiling(nNuevo_5clases)
```

```
## [1] 210
```

6.5 Ejercicio Propuesto 1: Adaptación Ejercicio 17

Los diámetros, en cm, de nueve arandelas metálicas, fabricadas por una máquina fueron: 0.962, 1.066, 0.900, 0.846,0.807, 0.797, 0.814, 0.710, 0.676.

a. ¿Estos datos corresponden a una distribución normal? Calcular el pvalor por simulación.

b. Calcular por simulación la potencia para la alternativa N(0.8,0.1).

c. Obtener el tamagno muestral para que la potencia calculada en b) sea de 0.95

6.6 Ejercicio Propuesto 2: Adapatación Ejercicio 6

Ejercicio 6 de la hoja de ejercicios de TBA.

6.7 Ejercicio Propuesto 3: Cramer von Mises

Obtener por simlación la distribución bajo H_0 del estadístico de Cramer von Mises.

7 Práctica 7: Test basados en rangos 7

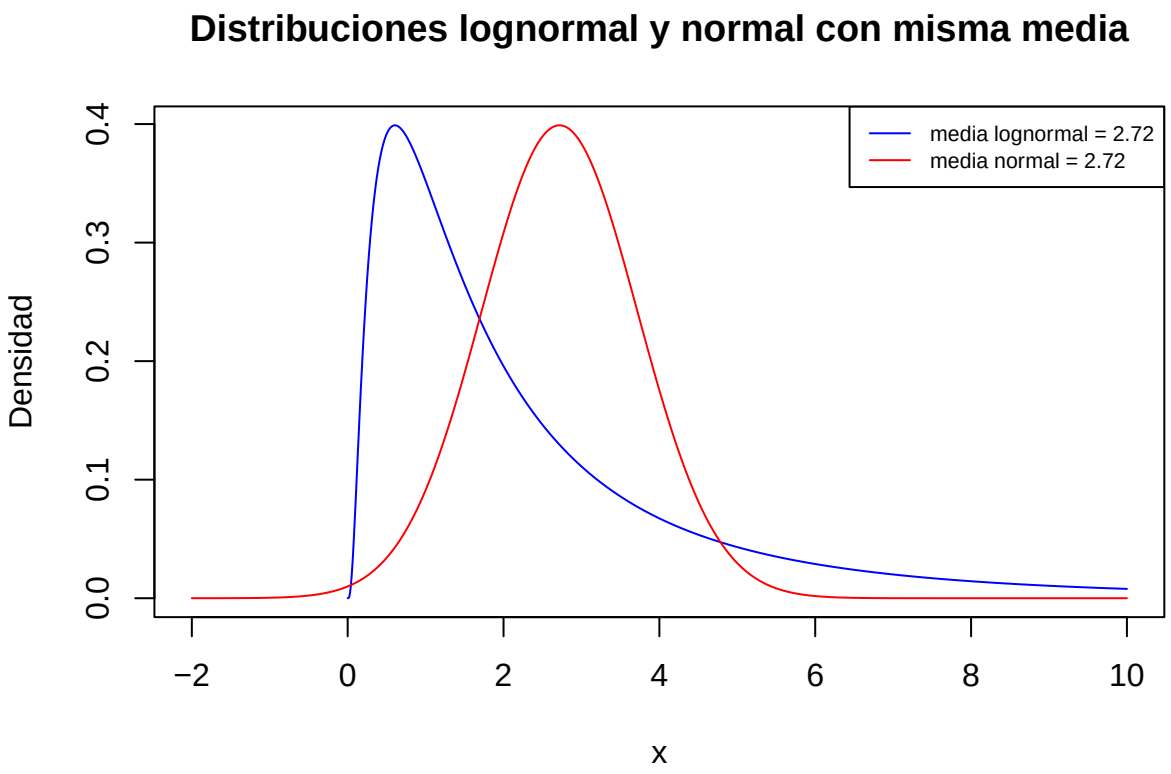
7.1 Ejercicio 1: Test de Wilcoxon (Mann Whitney) *vs* t-test

El t-test es un test estadístico paramétrico que permite contrastar la hipótesis nula de que las medias de dos poblaciones son iguales, frente a la hipótesis alternativa de que no lo son. Para que sus resultados sean válidos es necesario que se cumplan los supuestos de independencia, normalidad e igualdad de varianzas.

a. Evaluación t-test con distribuciones no normales para distintos tamagnos muestrales.

Distribución lognormal *normal*:

```r
x <- seq(0, 10, length = 1000)
y <- dlnorm(x = x, meanlog =  0.5, sdlog = 1)
plot(x, y, type = "l", lty = 1, xlab = "x", col = "blue", ylab = "Densidad",
     main = "Distribuciones lognormal y normal con misma media",
     xlim = c(-2, 10))
x_2 <- seq(-2, 10, length = 1000)
y_2 <- dnorm(x = x_2, mean = 2.718282, sd = 1)
lines(x_2, y_2, col = "red")
legend("topright",
       legend = c("media lognormal = 2.72", "media normal = 2.72"),
       col = c("blue", "red"), lty = 1, cex = 0.7)
```



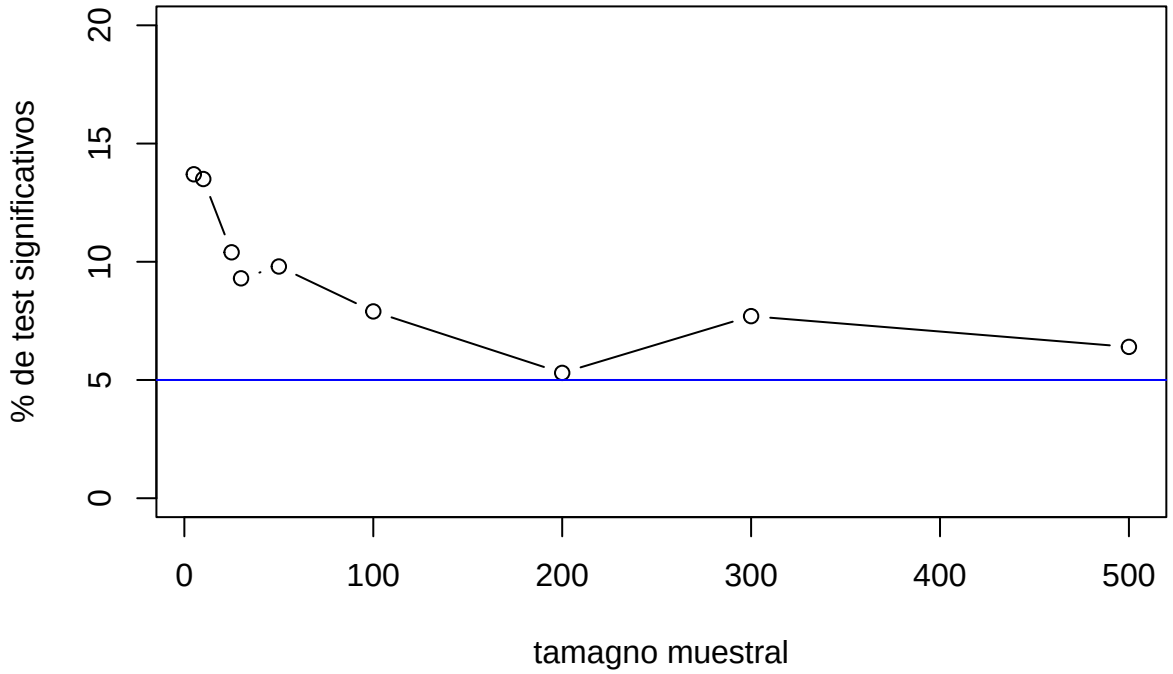
Para los tamagnos muestrales 5, 10, 20, 30, 50, 100, 200, 300, 500 se extraen 1000 observaciones de cada distribución y se identifica el porcentaje tests significativos (donde se rechaza H_0) para un nivel $\alpha = 0.05$.

```
testSignificativos<-c()
tamMuestral<-c(5,10,25,30,50,100,200,300,500)
for(i in tamMuestral){
  pValues<-c()
  for(j in 1:1000){
    muestraA<-rlnorm(n=i,meanlog=0.5,sdlog=1)
    muestraB<-rnorm(n=i,mean=2.718282,sd=1)
    pValues[j]<-t.test(muestraA,muestraB,var.equal=FALSE)$p.value
  }
  testSignificativos<-c(testSignificativos,mean(pValues<0.05)*100)
}
names(testSignificativos)<-c(5,10,23,30,50,100,200,300,500)
testSignificativos
```

```
##    5   10   23   30   50  100  200  300  500
## 13.7 13.5 10.4  9.3  9.8  7.9  5.3  7.7  6.4
```

La falta de normalidad que presenta una de las poblaciones, hace que para tamaños muestrales por debajo de 100 el procertanje de tests significativos está por encima de lo esperado (5%).

```
plot(x = tamMuestral, y = testSignificativos, type = "b", ylim = c(0,20),
     ylab = "% de test significativos", xlab = "tamagno muestral")
abline(h = 5, col = "blue")
```



Estos inconvenientes juistifican la necesidad de métodos no paramétricos. El test no paramétrico homólogo al t-test es el test de Wilcoxon-Mann-Whytney que hemos estudiado. Las condiciones de aplicabilidad de este test son que los datos tienen que ser independientes, ordinales (o poderse ordenar), el tamagno muestral no es necesario que sea grande, y tampoco que las muestras procedan de poblaciones normales, y que la variabilidad de los grupos sea similar (homocedasticidad).

Ante esta situación, ¿por qué no usamos siempre tests no paramétricos? En general los tests no paramétricos son menos potentes. En concreto, el test de Mann–Whitney–Wilcoxon es menos potente que el t-test (tienen menos probabilidad de rechazar la H_0 cuando realmente es falsa) ya que se centra en los rangos e ignora valores extremos. En el caso de los t-test, al trabajar con medias, si los tienen en cuenta. Sin embargo, esto hace a su vez que el test de Mann–Whitney–Wilcoxon sea una prueba más robusta que los t-test.

Por ejemplo (extremo), a continuación se tiene una muestra de datos correspondiente a una medida en dos grupos, cada uno con tres sujetos, entre paréntesis se escriben los rangos):

```
| Control     | Tratamiento  |
| 3.4 (1)     | 1233  (4)    |
| 3.7 (3)     | 1235  (6)    |
| 3.5 (2)     | 1234  (5)    |
```

A la vista de la tabla, parece evidente que el tratamiento aumenta drásticamente el valor que se está midiendo. Sin embargo, la prueba de Mann-Whitney pregunta si los rangos se distribuyeron al azar entre los grupos de control y tratados, cuál es la probabilidad de obtener los tres rangos más bajos en un grupo y los tres rangos más altos en el otro grupo. La prueba no paramétrica solo observa el rango, ignorando el hecho de que los valores tratados no son solo más altos, sino mucho más altos.

```
wilcox.test(c(3.4,3.7,3.5),c(1233,1235,1234))
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  c(3.4, 3.7, 3.5) and c(1233, 1235, 1234)
## W = 0, p-value = 0.1
## alternative hypothesis: true location shift is not equal to 0
```

Estos resultados no son significativamente diferentes ($\alpha = 0.05$). Este ejemplo muestra que con n=m=3, la prueba de Mann-Whitney nunca puede obtener un p-valor inferior a 0.05. En otras palabras, que para este ejemplo la prueba de Mann-Whitney tiene una potencia estadística nula.

Pero esto no es siempre así, con muestras grandes, la prueba de Mann-Whitney tiene casi tanta potencia estadística como la prueba t, en concreto, se ha demostrado que la pérdida de potencia es del 5%.

La figura de abajo está tomada de Zimmerman (1987) (https://www.jstor.org/stable/20151691?seq=3), donde se comparó por simulación los errores de tipo I y la potencia del t-test y del test de Wilcoxon-Mann-Whitney. Cuando las varianzas son iguales y los tamaños de las muestras son iguales (panel superior izquierda de la Figura 1), la función de potencia de la prueba t supera ligeramente a la de la prueba de Mann-Whitney, como era de esperar. En este caso, cuando se cumplen las suposiciones paramétricas, se sabe que la eficiencia relativa asintótica de la prueba de Mann-Whitney es de 0.955. Cuando los tamaños de las muestras son distintos y la muestra más pequeña tiene la menor varianza (panel intermedio inferior de la Figura 1), la prueba de Mann-Whitney es más potente que la prueba t en todo el rango de diferencias entre las medias.

Una de las aplicaciones más frecuentes del test de Mann–Whitney–Wilcoxon es su uso como alternativa al t-test cuando las muestras no proceden de poblaciones con distribución normal (asimetría o colas) o porque tienen un tamaño demasiado reducido para poder afirmarlo. Si las distribuciones de las poblaciones subyacentes se diferencian únicamente en localización, entonces el test de Mann–Whitney–Wilcoxon compara medianas,

En la práctica, el escenario en el que la única diferencia entre poblaciones es la localización es poco realista. Si las distribuciones tienen colas (asimetría) y las medias o medianas son distintas, es muy probable que las varianzas también lo sean. De hecho, la distribución normal es la única distribución estándar en la que la media y la varianza son independientes.

Es necesario evaluar estas características para poder determinar si el test de Mann–Whitney–Wilcoxon es suficientemente robusto para el estudio en cuestión.

Supóngase que se dispone de las siguientes muestras y de que se desea conocer si existe diferencia significativa entre las poblaciones de origen.

```
grupo_a <- c(5, 5, 5, 5, 5, 5, 7, 8, 9, 10)
grupo_b <- c(1, 2, 3 ,4, 5, 5, 5, 5, 5, 5)
par(mfrow = c(1,2))
hist(grupo_a, col = "blue", main = "")
hist(grupo_b, col = "red", main = "")
```

TABLE 1—Probability of Type I Errors, *t* Test, and *U* Test.

	U test				*t* test			
	1	2	3	*M*	1	2	3	*M*
Equal variances								
$N_1 = N_2 = 10$	.045	.038	.040	.041	.049	.046	.045	.047
$N_1 = 16, N_2 = 4$	.049	.049	.049	.049	.049	.048	.045	.048
$N_1 = 4, N_2 = 16$	.053	.045	.046	.048	.056	.042	.049	.049
Unequal variances								
$\sigma_1 > \sigma_2$								
$N_1 = N_2 = 10$	.074	.078	.075	.075	.067	.066	.061	.065
$N_1 = 16, N_2 = 4$	.005	.008	.005	.006	0	.001	.001	.001
$N_1 = 4, N_2 = 16$	.142	.135	.127	.134	.355	.351	.327	.344

NOTE: Entries are computer-generated probabilities that test statistics (*t* and *U*) exceed critical values associated with .05 significance level.

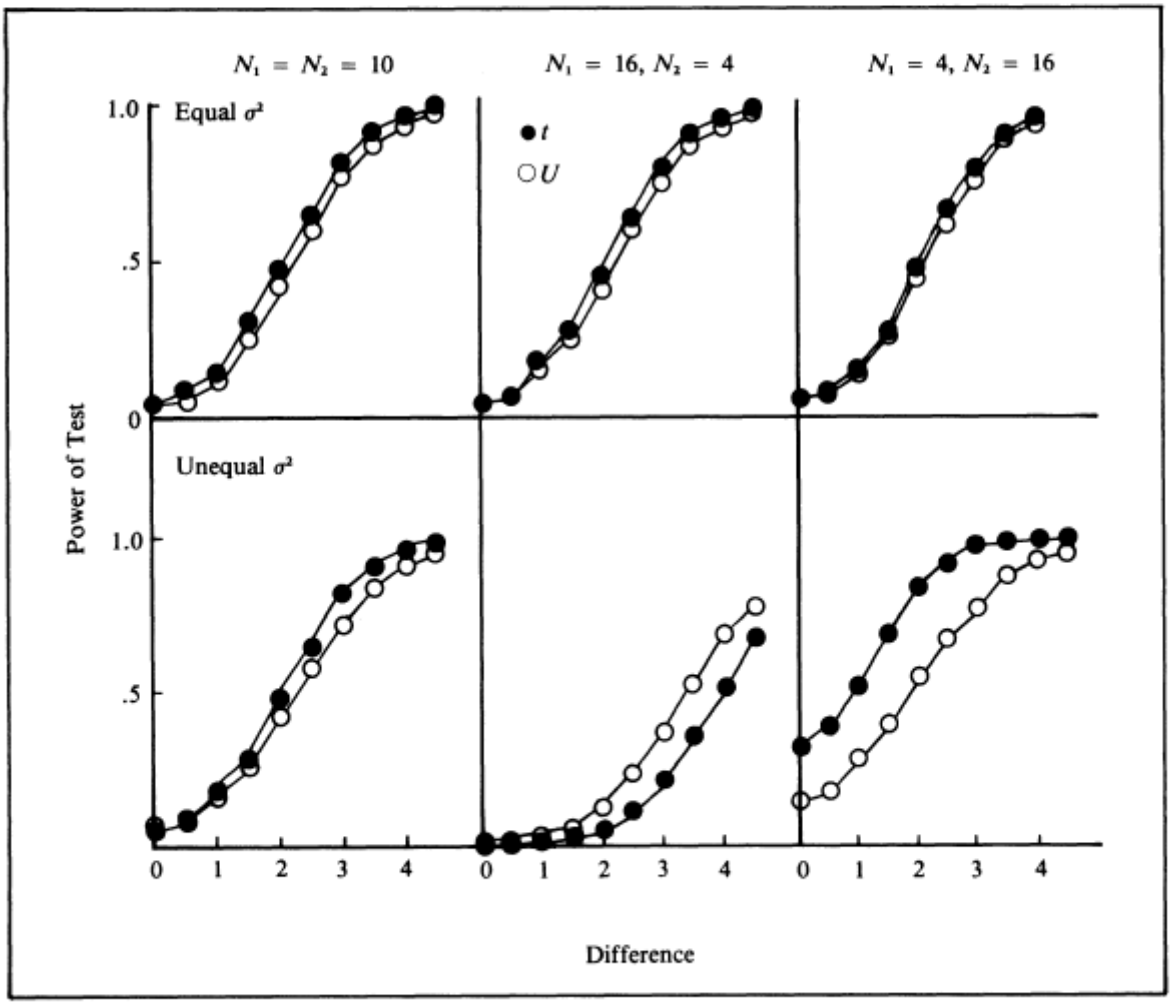
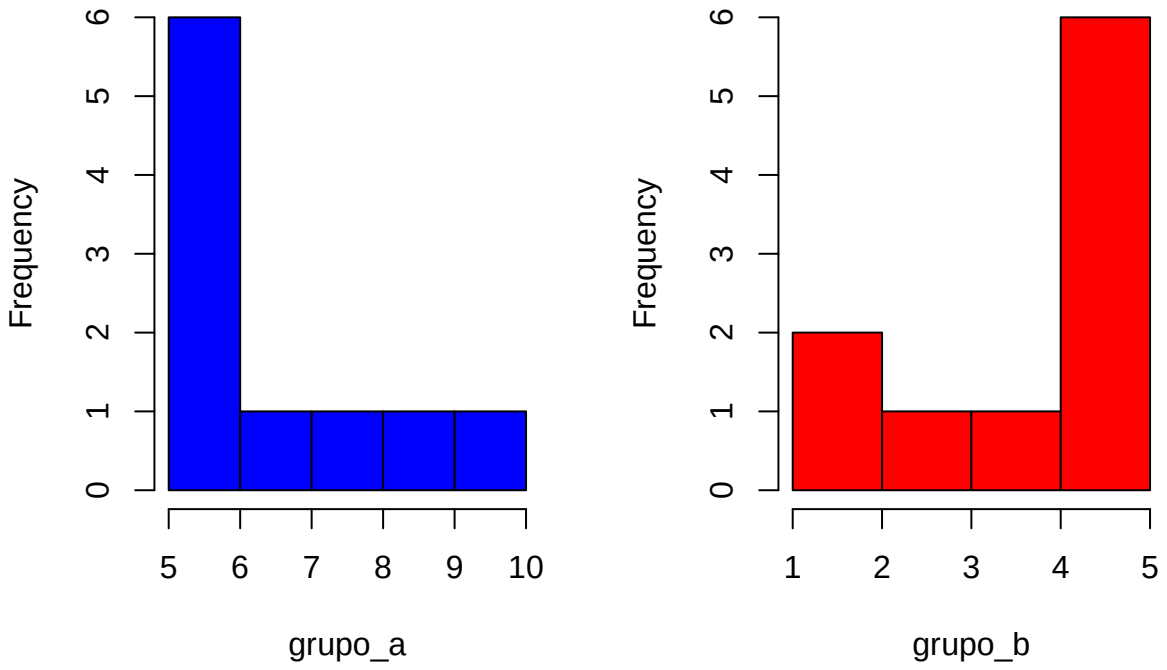


Figure 1—Power of Student *t* test and Mann-Whitney *U* test. Functions show computer-generated probabilities that test statistics (*t* and *U*) exceed critical values associated with .05 significance level. Differences between means are expressed in units of one-half a standard error of the difference.

Figura 7.1: P-valor y petencia t-test y Mann-Whitney



Aquí el tamaño muestral es pequeño y ambos grupos muestran asimetr?a, por lo que el t-test queda descartado. Una posible alternativa es emplear el test de Mann–Whitney–Wilcoxon:

```
wilcox.test(grupo_a, grupo_b, paired = FALSE)
```

```
## Warning in wilcox.test.default(grupo_a, grupo_b, paired = FALSE): cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  grupo_a and grupo_b
## W = 82, p-value = 0.007196
## alternative hypothesis: true location shift is not equal to 0
```

El p-valor obtenido indica que existen diferencias en la distribución de ambas poblaciones (como lo usaremos siempre nosotros). Pese a que si se calculan las medianas éstas coincide:

```
median(grupo_a)
```

```
## [1] 5
```

```
median(grupo_b)
```

```
## [1] 5
```

¿Es esto contradictorio? No, lo que está ocurriendo es que como las dos poblaciones tienen asimetrías en direcciones opuestas, es decir, sus diferencias van más allá de la localización (mediana), el test de Mann–Whitney–Wilcoxon no puede emplearse para comparar medianas.

7.2 Ejercicio 2: Simulación para W_S y W_{XY}

a. Obtener por simulación los pvalores de la tabla dada para la distribución de W_{XY}.

```
# Simulacion basada en 100 repeticiones
rep<-100  # Podeis probar con 1000, problema tiempo!
# Definimos el array que almacena los pvalores de la tabla
tablaWXY<-array(0,dim=c(10,10,51),dimnames = list(paste0("k1_",1:10),
                            paste0("k2_",1:10),
                            paste0("a_",0:50)))

for(k1 in 1:10){
    for(k2 in k1:10){
        pval<-c()
        for(a in 0:50){
            logPval<-c()
            for(veces in 1:rep){
                all<-1:(k1+k2)
                control<-sample(all,k1,replace=FALSE)
                trat<-setdiff(all,control)
                Ws<-sum(trat)
                Wxy<-Ws-k2*(k2+1)/2
                logPval<-c(logPval,ifelse(Wxy<=a,1,0))
            }
            # Obtenemos el pvalor por simulacion
            pval<-c(pval,sum(logPval)/rep)
        }
        tablaWXY[k1,k2,]<-pval
    }

}

# Veamos algunos ejemplos y comparemos los resultados son las tablas dadas
tablaWXY[3,3,]
```

```
##  a_0  a_1  a_2  a_3  a_4  a_5  a_6  a_7  a_8  a_9 a_10 a_11 a_12 a_13 a_14 a_15
## 0.05 0.10 0.18 0.39 0.47 0.70 0.79 0.89 0.97 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## a_16 a_17 a_18 a_19 a_20 a_21 a_22 a_23 a_24 a_25 a_26 a_27 a_28 a_29 a_30 a_31
## 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## a_32 a_33 a_34 a_35 a_36 a_37 a_38 a_39 a_40 a_41 a_42 a_43 a_44 a_45 a_46 a_47
## 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## a_48 a_49 a_50
## 1.00 1.00 1.00
```

```
tablaWXY[4,4,]
```

```
##  a_0  a_1  a_2  a_3  a_4  a_5  a_6  a_7  a_8  a_9 a_10 a_11 a_12 a_13 a_14 a_15
```

```
## 0.00 0.02 0.07 0.10 0.11 0.23 0.36 0.39 0.55 0.74 0.76 0.83 0.91 0.93 0.95 0.96
## a_16 a_17 a_18 a_19 a_20 a_21 a_22 a_23 a_24 a_25 a_26 a_27 a_28 a_29 a_30 a_31
## 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## a_32 a_33 a_34 a_35 a_36 a_37 a_38 a_39 a_40 a_41 a_42 a_43 a_44 a_45 a_46 a_47
## 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## a_48 a_49 a_50
## 1.00 1.00 1.00
```

```
tablaWXY[5,5,]
```

```
##  a_0  a_1  a_2  a_3  a_4  a_5  a_6  a_7  a_8  a_9 a_10 a_11 a_12 a_13 a_14 a_15
## 0.01 0.02 0.01 0.03 0.06 0.12 0.07 0.13 0.21 0.33 0.42 0.50 0.46 0.58 0.67 0.68
## a_16 a_17 a_18 a_19 a_20 a_21 a_22 a_23 a_24 a_25 a_26 a_27 a_28 a_29 a_30 a_31
## 0.66 0.87 0.85 0.88 0.95 1.00 0.99 0.97 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## a_32 a_33 a_34 a_35 a_36 a_37 a_38 a_39 a_40 a_41 a_42 a_43 a_44 a_45 a_46 a_47
## 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## a_48 a_49 a_50
## 1.00 1.00 1.00
```

```
tablaWXY[10,10,] # 0.059 con 1000 repeticiones
```

```
##  a_0  a_1  a_2  a_3  a_4  a_5  a_6  a_7  a_8  a_9 a_10 a_11 a_12 a_13 a_14 a_15
## 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.01
## a_16 a_17 a_18 a_19 a_20 a_21 a_22 a_23 a_24 a_25 a_26 a_27 a_28 a_29 a_30 a_31
## 0.01 0.00 0.03 0.03 0.01 0.01 0.02 0.03 0.03 0.04 0.04 0.03 0.04 0.08 0.09 0.08
## a_32 a_33 a_34 a_35 a_36 a_37 a_38 a_39 a_40 a_41 a_42 a_43 a_44 a_45 a_46 a_47
## 0.06 0.10 0.13 0.14 0.10 0.16 0.25 0.18 0.27 0.25 0.37 0.27 0.31 0.31 0.41 0.41
## a_48 a_49 a_50
## 0.50 0.49 0.50
```

```
pwilcox(28,10,10,"l") # En la tabla es 0.0526
```

```
## Warning in pwilcox(28, 10, 10, "l"): NAs introducidos por coerción
```

```
## [1] 0.05256122
```

b. Para el caso paricular de un experimento con 5 individuos, de los cuales 3 fueron asignados a tratamiento. Obtener por simulación la distribución de W_s y W_{XY} bajo H_0. ¿Est?n igualmente distribuidas baho H_0? ¿En torno a qué valor se localiza cada una de estas distribuciones? ¿Cuál es mínimo valor que toman los estadísticos bajo H_0?

```
k1<-2
k2<-3

rep<-1000

Ws_dist<-c()
Wxy_dist<-c()
for(veces in 1:rep){
    all<-1:(k1+k2)
    control<-sample(all,k1,replace=FALSE)
```

```
    trat<-setdiff(all,control)
    Ws_dist<-c(Ws_dist,sum(trat))
    Wxy_dist<-c(Wxy_dist,sum(trat)-k2*(k2+1)/2)
}

# Con la notacion que hemos visto en clase
m<-k1
n<-k2
N<-m+n

# Comprobamos algunos aspectos respecto la distribucion de Ws bajo H0
par(mfrow=c(1,2))
barplot(table(Ws_dist))
summary(Ws_dist)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   6.000   8.000   9.000   8.966  10.000  12.000
```

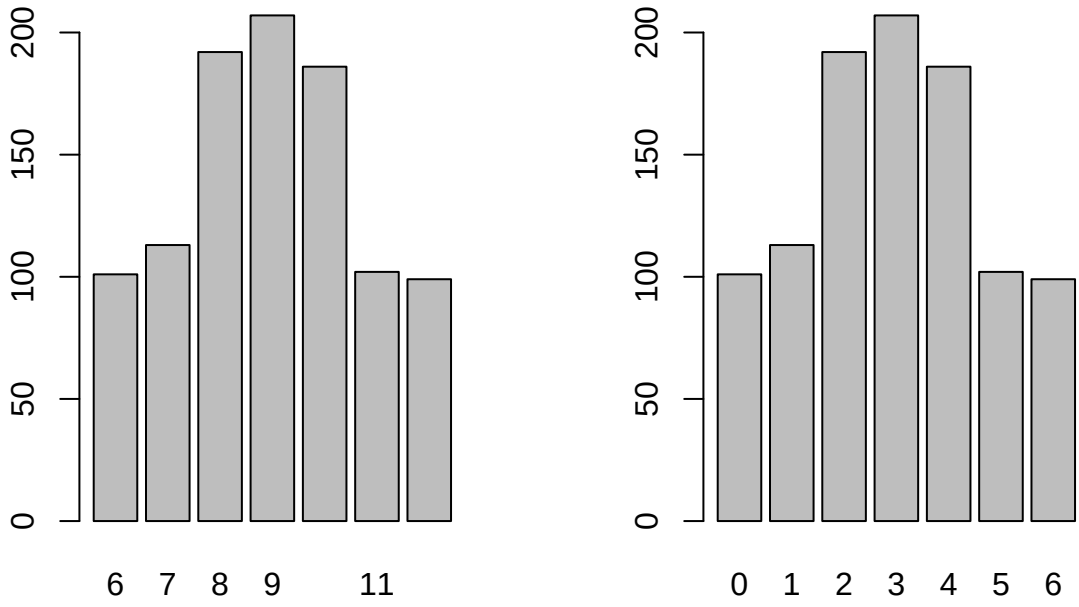
```
# La distribucion de Ws bajo H0 es simetrica respecto de n*(N+1)/2
n*(N+1)/2
```

```
## [1] 9
```

```
# El minimo valor de Ws es n*(n+1)/2
n*(n+1)/2
```

```
## [1] 6
```

```
# Comprobamos algunos aspectos respecto la distribucion de WXY bajo H0
barplot(table(Wxy_dist))
```




```
summary(Wxy_dist)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.966   4.000   6.000
```

```
# La distribucion de WXY bajo H0 es simetrica respecto de m*n/2
m*n/2
```

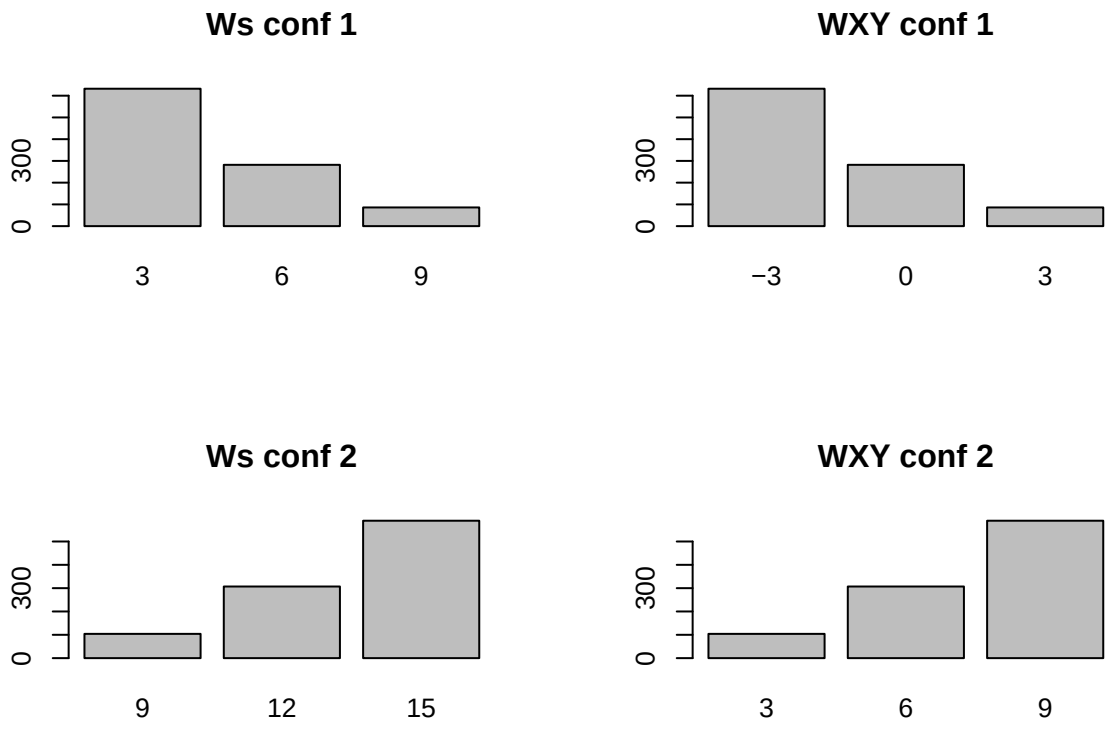
```
## [1] 3
```

```
# El minimo valor de WXY 0
```

c. Para este mismo experimento, y en el caso de que hubiera coincidencias las distrdistribuciones de W_s y W_{XY} est?n igualmente distribuidas bajo H_0?

```
k1<-2
k2<-3

rep<-1000

Ws_dist1<-c() # Distribucion Ws para configuracion de coincidencias 1
Wxy_dist1<-c() # Distribucion WXY para configuracion de coincidencias 1
Ws_dist2<-c() # Distribucion Ws para configuracion de coincidencias 2
Wxy_dist2<-c() # Distribucion WXY para configuracion de coincidencias 2
```

```
for(veces in 1:rep){
    all<-1:(k1+k2)
    control<-sample(all,k1,replace=FALSE)
    # Configuracion de coincidencias 1 (el minimo de los que queden)
    conf1<-min(setdiff(all,control))
    trat1<-rep(conf1,k2)
    # Configuracion de coincidencias 2 (el maximo de los que queden)
    conf2<-max(setdiff(all,control))
    trat2<-rep(conf2,k2)
    Ws_dist1<-c(Ws_dist1,sum(trat1))
    Wxy_dist1<-c(Wxy_dist1,sum(trat1)-k2*(k2+1)/2)
    Ws_dist2<-c(Ws_dist2,sum(trat2))
    Wxy_dist2<-c(Wxy_dist2,sum(trat2)-k2*(k2+1)/2)
}

par(mfrow=c(2,2))
barplot(table(Ws_dist1),main="Ws conf 1")
barplot(table(Wxy_dist1),main="WXY conf 1")
barplot(table(Ws_dist2),main="Ws conf 2")
barplot(table(Wxy_dist2),main="WXY conf 2")
```



7.3 Ejercicio 3: Adaptación del Ejercicio 4

Se quiere comparar el crecimiento de la población en zonas rurales y en zonas urbanas. Para ello se utiliza el porcentaje de cambio en la población entre 2010 y 2020. Se eligen al azar 7 zonas urbanas y 9 zonas rurales y se calculan los porcentajes de crecimiento en cada zona:

Zonas rurales: 1.1, -21.7, -16.3, -11.3, -10.4, -7.0, -2.0, 1.9, 6.2 Zonas urbanas: -2.4, 9.9, 14.2, 18.4, 20.1, 23.1, 70.4

a. Contrastar si es mayor la tasa de crecimiento en las zonas urbanas.

```
# Datos
x<-c(1.1,-21.7,-16.3,-11.3,-10.4,-7.0,-2.0,1.9,6.2) # Rural
y<-c(-2.4,9.9,14.2,18.4,20.1,23.1,70.4) # Urbano
m<-length(x)
n<-length(y)
N<-n+m

# Manera 1. Utilizando pwilcox

# Calculamos los rangos
z<-c(x,y)
r<-rank(z)
r
```

```
##  [1]  8  1  2  3  4  5  7  9 10  6 11 12 13 14 15 16
```

```
# Calculo de Ws
WSu=sum(r[(m+1):N])
WSu
```

```
## [1] 87
```

```
# Calculo de WXY
WXYu<-WSu-n*(n+1)/2
WXYu
```

```
## [1] 59
```

```
# Calculo del pvalor
1-pwilcox(WXYu-1,n,m)
```

```
## [1] 0.001048951
```

```
# Manera 2. Utilizando wilcox.test (cuando hay datos)
wilcox.test(y,x,"g") # Help: the one-sided alternative "greater"
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  y and x
## W = 59, p-value = 0.001049
## alternative hypothesis: true location shift is greater than 0
```

```r
#is that x is shifted to the right of y
wilcox.test(x,y,"l") # Se obtiene lo mismo
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  x and y
## W = 4, p-value = 0.001049
## alternative hypothesis: true location shift is less than 0
```

b. Estimar el efecto del tratamiento.

```r
# Manera 1. Calcular a mano las mn diferencias y
# estimar el efecto con la mediana

d<-c()
for(i in 1:m){
    for(j in 1:n){
        d<-c(d,y[j]-x[i])
    }
}
median(d)
```

```
## [1] 22.1
```

```r
# Manera 2. Utilizando los argumentos de wilcox.test

wilcox.test(y,x,"g",conf.int=TRUE,conf.level=0.95)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  y and x
## W = 59, p-value = 0.001049
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  13.9  Inf
## sample estimates:
## difference in location
##                   22.1
```

c. Aproximar la potencia en $\Delta = 2$ si $F \sim N(\mu, \hat{}2)$ con $\sigma^2 = 4$ y $\alpha = 0.05$.

```r
muAlpha<-qnorm(0.95)

# La diferencia de v.a. normales es N(0,2*sqrt(2)) cl normales
fStar<-function(sigma){
  return(1/(sigma*sqrt(2*pi)))
}

Delta<-2
```

```r
# Potencia aproximada asint

pnorm(sqrt((12*n*m)/(N+1))*fStar(2*sqrt(2))*Delta-muAlpha)
```

```
## [1] 0.5934122
```

d. Suponiendo n=m calcular el tamagno muestral necesario para que la potencia en $\Delta = 2$ sea 0.9.

```r
# Manera 1. Calcular a mano las mn diferencias y estimar el efecto
# con la mediana

n2<-(qnorm(0.95)-qnorm(0.1))^2/(6*Delta^2*fStar(2*sqrt(2))^2)
n2
```

```
## [1] 17.93608
```

```r
ceiling(n2) # para tratamiento y control
```

```
## [1] 18
```

7.4 Ejercicio 4: Adaptación Ejercicio 1

La duración en horas de una serie de bombillas fue: 518, 174, 613, 2010, 2139, 156, 450, 536. Tras un nuevo proceso de fabricación, la duración de 25 bombillas fue:

899, 326, 2118, 839, 820, 1423, 1687, 1010, 3011, 1739, 1185, 1320, 646, 505, 4236, 4481, 1433, 1806, 400, 421, 335, 1164, 1713, 1356, 390.

a. Contrastar si la duración de la nueva serie supera significativamente a la de la primera.

```r
# Datos
x<-c(518,174,614,2010,2139,156,450,536)
y<-c(899,326,2118,839,820,1423,1687,1010,3011,1739,1185,1320,
     646,505,4236,4481,1433,1806,400,421,335,1164,1713,1356,390)

m<-length(x)
n<-length(y)
N<-n+m

# Manera 1. Utilizando pwilcox

# Calculamos los rangos
z<-c(x,y)
r<-rank(z)
r
```

```
##  [1] 10  2 12 28 30  1  8 11 16  3 29 15 14 22 24 17 31 26 19 20 13  9 32 33 23
## [26] 27  6  7  4 18 25 21  5
```

```r
# Calculo de Ws
WS=sum(r[(m+1):N])
WS
```

```
## [1] 459
```

```r
# Calculo de WXY
WXY<-WS-n*(n+1)/2
WXY
```

```
## [1] 134
```

```r
# Calculo del pvalor
1-pwilcox(WXY-1,n,m)
```

```
## [1] 0.08121473
```

```r
# Manera 2. Utilizando wilcox.test (cuando hay datos)

wilcox.test(y,x,alternative="g")
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  y and x
## W = 134, p-value = 0.08121
## alternative hypothesis: true location shift is greater than 0
```

```r
# O analogamente
wilcox.test(x,y,alternative="l")
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  x and y
## W = 66, p-value = 0.08121
## alternative hypothesis: true location shift is less than 0
```

b. Estimar la diferencia en la duración de las bombillas.

```r
# Manera 1. Calcular a mano las mn diferencias
# y estimar el efecto con la mediana

d<-c()
for(i in 1:m){
    for(j in 1:n){
        d<-c(d,y[j]-x[i])
    }
}
median(d)
```

```
## [1] 491
```

```
# Manera 2. Utilizando los argumentos de wilcox.test

wilcox.test(y,x,"g",conf.int=TRUE,conf.level=0.95)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  y and x
## W = 134, p-value = 0.08121
## alternative hypothesis: true location shift is greater than 0
## 95 percent confidence interval:
##  -60 Inf
## sample estimates:
## difference in location
##                    491
```

c. Obtener IC al menos 95% para la diferencia de la duración (Δ). ¿Cuál es la confianza exacta de dicho intervalo?

```
# Calculo de i,j
wilcox.test(y,x,conf.int=TRUE,conf.level=0.95)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  y and x
## W = 134, p-value = 0.1624
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -128 1169
## sample estimates:
## difference in location
##                    491
```

```
i<-qwilcox(0.025,m,n)
i
```

```
## [1] 54
```

```
# Conocido i, tengo j
j<-m*n-i+1
j
```

```
## [1] 147
```

```
# Comprobaciones
pwilcox(i,n,m) # No vale
```

```
## [1] 0.02738272
```

```
pwilcox(i-1,n,m)
```

```
## [1] 0.02468382
```

```
# IC al menos 1-alpha
d[i]
```

```
## [1] 225
```

```
d[j]
```

```
## [1] 1008
```

```
# Confianza exacta
1-2*pwilcox(i-1,n,m)
```

```
## [1] 0.9506324
```

7.5 Ejercicio 5: Adapatación Ejercicio 8

Se tienen 6 enfermos que padecen cierta enfermedad. El grado de dolor que padecen se codifica en: A=no tiene dolor, B=dolor soportable, C=dolor muy intenso. Para estudiar si cierto tratamiento mitiga el dolor de estos enfermos se eligen 3 de ellos al azar que se dejan como control, y se aplica el tratamiento a los otros 3, siendo los resultados:

Control: A, C, C Tratamiento: B, B, C.

Estudiar si el tratamiento produce efecto.

La muestra conjunta ordenada C C C B B A semi-rangos 2 2 2 4.5 4.5 6

$W_S^* = 11$

Distribución exacta (S*1, S*2, S*3) (2,2,2) (2,2,4.5) (2,2,6) (2,4.5,4.5) (2,4.5,6) (4.5,4.5,6) probabilidades 1/20 6/20 3/20 3/20 6/20 1/20 W_S^* 6 8.5 10 11 12.5 15

```
$P(W_S^* >=11)= 0.5$
```

NO SE RECHAZA LA HIPÓTESIS

```
# No tenemos datos, no podemos usar wilcox.test

# ¿Es correcto?
1-pwilcox(10,3,3)
```

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

7.6 Ejercicio Propuesto 1: Adaptación Ejercicio 5

En un examen se utilizan dos tipos de pruebas A y B, que se asignan al azar a los alumnos. A medida que los alumnos entregan el examen, el profesor anota el tipo de prueba y se obtiene:

```
B B B A A B A B B B B A B A A B A A A
```

¿Es razonable pensar que se requiere más tiempo para realizar una prueba que la otra?

Bibliografía

Alvo, M. (2022). *Statistical inference and machine learning for big data*. Springer Nature.

Casella, G. and Berger, R. L. (2021). *Statistical inference*. Cengage Learning.

D'Agostino, R. (2017). *Goodness-of-fit-techniques*. Routledge.

Gibbons, J. D. and Chakraborti, S. (2014). *Nonparametric statistical inference: revised and expanded*. CRC press.

Lehmann, E. L. and Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media.

Lehmann, E. L. et al. (1975). Statistical methods based on ranks. *Nonparametrics. San Francisco, CA, Holden-Day*.

Manteiga, M. T. G. (2012). *Estadística aplicada: Una visión instrumental*. Ediciones Díaz de Santos.

Millar, R. B. (2011). *Maximum likelihood estimation and inference: with examples in R, SAS and ADMB*. John Wiley & Sons.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Romano, J. P. and Lehmann, E. (2005). Testing statistical hypotheses.

Wasserman, L. (2004). *All of statistics: a concise course in statistical inference*, volume 26. Springer.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2023a). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.37, https://pkgs.rstudio.com/bookdown/.

Xie, Y. (2023b). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.45.