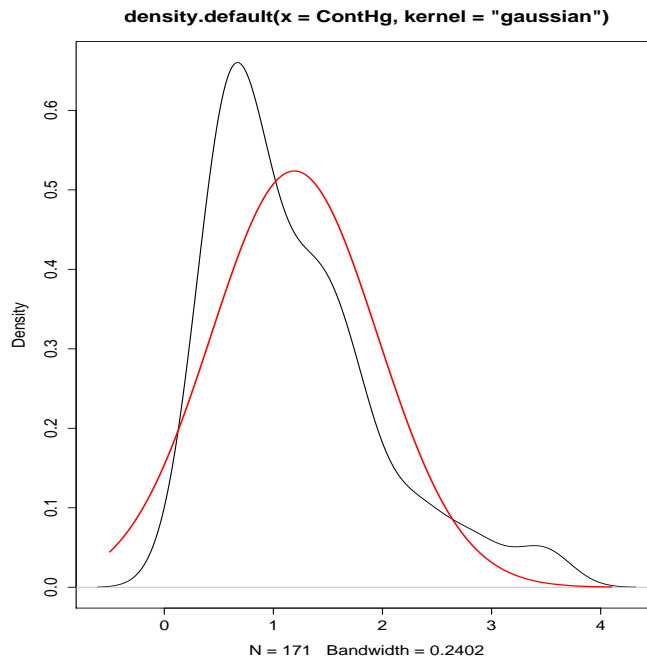


- (a) Representa un estimador de la función de densidad de la v.a. X = cantidad de contaminación por mercurio (en p.p.m.) en los peces capturados en los ríos norteamericanos Lumber y Wacamaw (ver fichero `Datos-mercurio.txt`). Comparar esta densidad estimada con la densidad normal de igual media y desviación típica (representada en la misma gráfica). En vista de las dos funciones ¿dirías que la función de densidad de X es aproximadamente normal?
- (b) Obtener un intervalo de confianza de nivel 0.95 para la media de X .
- (c) ¿Se puede considerar fiable este intervalo a pesar de la posible no-normalidad de X ?

(a)

```
> X = read.table('Datos-mercurio.txt')
> ContHg = X$V5
> plot(density(ContHg, kernel='gaussian'))
> t = seq(-0.5, 4.1, 0.01)
> m = mean(ContHg)
> desv = sd(ContHg)
> lines(t, dnorm(t, mean=m, sd=desv), type='l', lwd=2, col="red")
```



(b)

```
> t.test(ContHg)
```

One Sample t-test

```
data: ContHg
t = 20.4608, df = 170, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
1.076776 1.306733
sample estimates:
mean of x
```

1.191754

```
> tTestHg$conf.int
```

```
[1] 1.076776 1.306733
```

```
attr(,"conf.level")
```

```
[1] 0.95
```

(c) Sí, por el TCL. En realidad, si X no es normal un IC aproximado para $\mu = \mathbb{E}X$ sería

$$\left(\bar{x} \mp z_{\alpha/2} \frac{s}{\sqrt{n}} \right), \quad (1)$$

que podemos calcular con R:

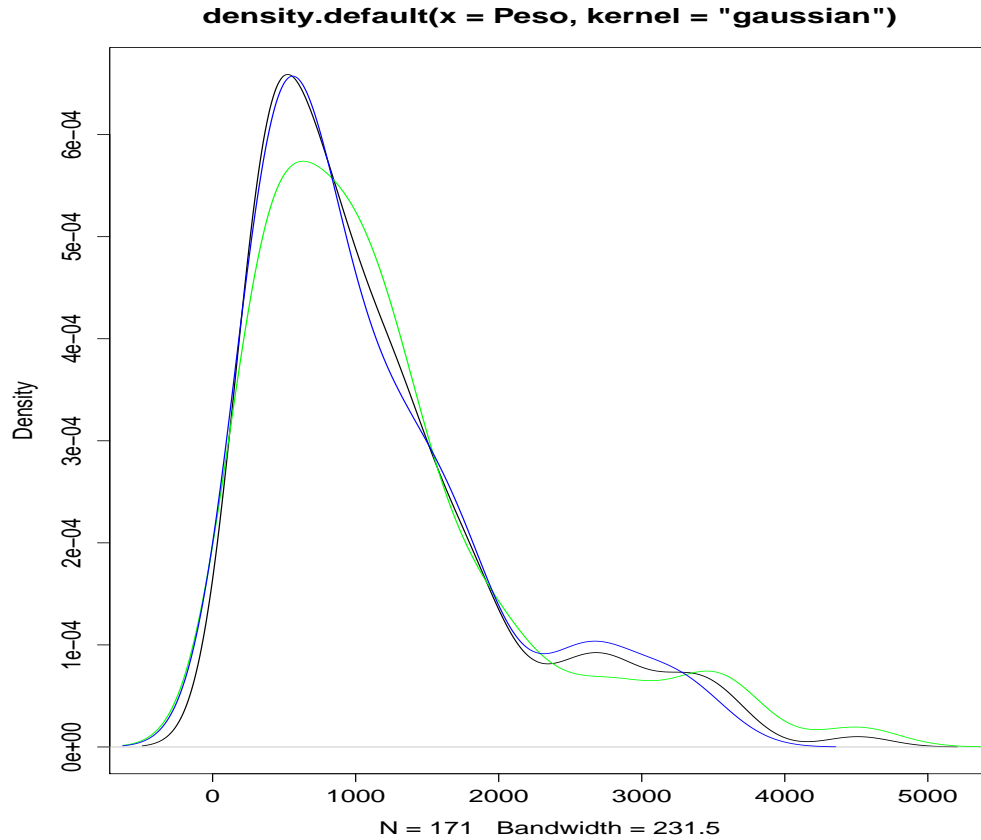
```
> source('norm.interval.R')
```

```
> norm.interval(ContHg)
```

```
[1] 1.077595 1.305914
```

Usando nuevamente los datos del fichero `Datos-mercurio.txt`, obtener intervalos de confianza para la media y la varianza de la variable “peso de los peces”. Usar en ambos casos los niveles 0.99 y 0.95 y comparar los intervalos obtenidos.

```
> Peso = X$V4
> plot(density(Peso,kernel='gaussian'))
> lines(density(PesoLumber,kernel='gaussian'),type='l',col="green")
> lines(density(PesoWacamaw,kernel='gaussian'),type='l',col="blue")
```



Claramente la distribución del peso no es gaussiana, pues el muestreo no se ha realizado en una población homogénea (dos ríos y varias estaciones de seguimiento, entre otros factores). Por ello utilizaremos el intervalo de confianza para la media dado en (1).

Intervalos para el peso esperado:

```
> source('norm.interval.R')
> norm.interval(Peso)
[1] 1016.686 1279.139
> norm.interval(Peso,nivel.conf=0.99)
[1] 975.4513 1320.3733
```

Respecto a los intervalos para la varianza, no podemos usar el intervalo

$$\left(\frac{(n-1)s^2}{\chi_{n-1;\alpha/2}^2}, \frac{(n-1)s^2}{\chi_{n-1;1-\alpha/2}^2} \right), \quad (2)$$

porque la distribución de los datos no es normal. Podemos usar el TCL para construir un intervalo de confianza $1-\alpha$ aproximado. Sea $Y = (X-\mu)^2$. Observemos que $\mathbb{E}(Y) = \sigma^2 = \mathbb{V}(X)$ y $\mathbb{V}(Y) = \mu_4 - \sigma^4$,

siendo $\mu_4 = \mathbb{E}(X - \mu)^4$ (suponiendo que el momento de orden 4 de X es finito). Entonces

$$\sqrt{n} \frac{\bar{Y} - \sigma^2}{\sqrt{\mu_4 - \sigma^4}} \xrightarrow[n \rightarrow \infty]{d} N(0, 1).$$

Por el teorema de Slutsky y el teorema de la aplicación continua, tenemos que

$$\sqrt{n} \frac{s^2 - \sigma^2}{\sqrt{\hat{\mu}_4 - s^4}} \xrightarrow[n \rightarrow \infty]{d} N(0, 1)$$

y un intervalo de confianza aproximado para σ^2 es

$$IC_{1-\alpha}(\sigma^2) \simeq \left(s^2 \mp z_{\alpha/2} \sqrt{\frac{\hat{\mu}_4 - s^4}{n}} \right),$$

donde $\hat{\mu}_4 = \sum_{i=1}^n (x_i - \mu)^4 / n$ y $s^4 = (s^2)^2$.

Reescribimos `var.interval.R` para incorporar este intervalo aproximado para la varianza:

```
var.interval = function(datos, nivel.conf = 0.95, Gauss.distr = 1) {

  v = var(datos)
  n = length(datos)

  if (Gauss.distr==1){
    gl = n - 1
    chiinf = qchisq((1 - nivel.conf)/2, gl)
    chisup = qchisq((1 - nivel.conf)/2, gl, lower.tail = FALSE)
    c(gl * v/chisup, gl * v/chiinf)
  } else {
    z = qnorm((1 - nivel.conf)/2, lower.tail = FALSE)
    mom4 = mean((datos-mean(datos))^4)
    c(v - z * sqrt((mom4-v^2)/n), v + z * sqrt((mom4-v^2)/n))
  }
}

> var(Peso)
[1] 766555.9
> var.interval(Peso,Gauss.distr=0)
[1] 552653.0 980458.7
> var.interval(Peso)
[1] 626525.8 959677.0
```