UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Departamento de Matemáticas

ESTADÍSTICA
Facultad de Ciencias del Mar

PRÁCTICA 4
INTRODUCCIÓN A LA INFERENCIA ESTADÍSTICA. CURSO 2011-12

1. Se dispone de un procedimiento de análisis químico para identificar la presencia de un contaminante en agua. Se consideran los siguientes sucesos:

   - D: el procedimiento detecta la presencia del contaminante en el agua.
   - C : el agua está contaminada.

   Cuando este procedimiento se aplica a una muestra de agua se tienen las siguientes probabilidades:

$$\begin{array}{rclrcl} P\left(D\,|C\right) & = & p_V & P\left(\overline{D}\,|C\right) & = & 1-p_V \\ P\left(D\,|\overline{C}\right) & = & p_F & P\left(\overline{D}\,|\overline{C}\right) & = & 1-p_F \end{array}$$

   En estas condiciones, se toman n muestras de agua de una playa, se procesan con el análisis químico descrito y se cuenta el número X de muestras en que el análisis dio positivo (detectó la presencia del contaminante). Dado que en realidad no se sabe si la playa está contaminada o no, ni se conoce la frecuencia de los episodios de contaminación, se adopta la siguiente regla de decisión:

   a) Fijar un valor k.

   b) Si $X > k$ se decide que la playa está contaminada (y se ordena su cierre).

   c) Si $X \leq k$ se decide que la playa no está contaminada y se mantiene abierta al baño.

   a) **Desarrollar una función en R que, fijado un tamaño muestral n, calcule para cada $k = 0, 1, 2, \ldots, n$ las siguientes probabilidades:**

      1) La probabilidad de mantener abierta al baño la playa cuando en realidad está contaminada:

$$\alpha_k = P\left(X \leq k\,|C\right)$$

      2) La probabilidad de cerrar la playa cuando en realidad no está contaminada:

$$\beta_k = P\left(X > k\,|\overline{C}\right)$$

   Esta función debe recibir como argumentos de entrada los valores de n, p_V y p_F y devolver como resultado un `data.frame` que contenga las variables k,α_k y β_k.

Solución:
Si se dispone de n muestras de agua independientes y X=*"Número de muestras en las que el análisis da positivo"*, entonces $X \approx B\left(n, p\right)$ siendo p la probabilidad de detectar el contaminante en una muestra arbitraria. Además:

- Si la playa está contaminada: $p = p_V = P(D|C)$ (probabilidad de un verdadero positivo)
- Si la playa no está contaminada: $p = p_F = P(D|\overline{C})$ (probabilidad de un falso positivo)

En estas condiciones, los dos errores que pretenden controlarse mediante la regla de decisión especificada pueden escribirse como:

- $\alpha_k = P(\text{Decidir playa No Contaminada}|C) = P(X \leq k|C) = \sum_{j=0}^{k} P(X = j|C) =$
  $= \sum_{j=0}^{k} \binom{n}{j} p_V^j (1-p_V)^{n-j}$=`pbinom(k,n,pV)`
- $\beta_k = P(\text{Decidir playa Contaminada}|\overline{C}) = P(X > k|\overline{C}) = 1 - P(X \leq k|\overline{C}) =$
  $= 1 - \sum_{j=0}^{k} P(X = j|\overline{C}) = 1 - \sum_{j=0}^{k} \binom{n}{j} p_F^j (1-p_F)^{n-j}$=`1-pbinom(k,n,pF)`

La siguiente función calcula las probabilidades pedidas:

```
> errores = function(n, pV, pF) {
    ak = pbinom(0:n, n, pV)
    bk = 1 - pbinom(0:n, n, pF)
    data.frame(k = 0:n, alfa = formatC(ak, digits = 8, format = "f"),
        beta = formatC(bk, digits = 8, format = "f"))
 }
```

b) Utilizar la función desarrollada en el apartado anterior para determinar los valores de k, α_k y β_k para $n = 5, 10, 40$ y 50 siendo $p_V = 0{,}7$ y $p_F = 0{,}4$. ¿Existe en estos casos alguna combinación de n y k para la que se consiga que $\alpha_k \leq 0{,}01$ y $\beta_k \leq 0{,}05$?

Solución:
- Para $n = 5$:

```
> errores(5, 0.7, 0.4)
```

```
  k       alfa       beta
1 0 0.00243000 0.92224000
2 1 0.03078000 0.66304000
3 2 0.16308000 0.31744000
4 3 0.47178000 0.08704000
5 4 0.83193000 0.01024000
6 5 1.00000000 0.00000000
```

Vemos, pues, que para estos valores de n, p_V y p_F no hay posibilidad alguna de conseguir simultáneamente las tasas α y β buscadas.

- Para $n = 10$:

```
> errores(10, 0.7, 0.4)
```

```
    k       alfa       beta
1   0 0.00000590 0.99395338
2   1 0.00014369 0.95364260
3   2 0.00159039 0.83271025
4   3 0.01059208 0.61771940
5   4 0.04734899 0.36689674
6   5 0.15026833 0.16623862
```

```
7   6 0.35038928 0.05476188
8   7 0.61721721 0.01229455
9   8 0.85069165 0.00167772
10  9 0.97175248 0.00010486
11 10 1.00000000 0.00000000
```

Tampoco en este caso se consigue el objetivo buscado.

- Para $n = 40$:

```
> errores(40, 0.7, 0.4)[20:25, ]
```

```
    k       alfa       beta
20 19 0.00241936 0.12976571
21 20 0.00625450 0.07435156
22 21 0.01477705 0.03916797
23 22 0.03195126 0.01891075
24 23 0.06331288 0.00834177
25 24 0.11514665 0.00335086
```

- Para $n = 50$

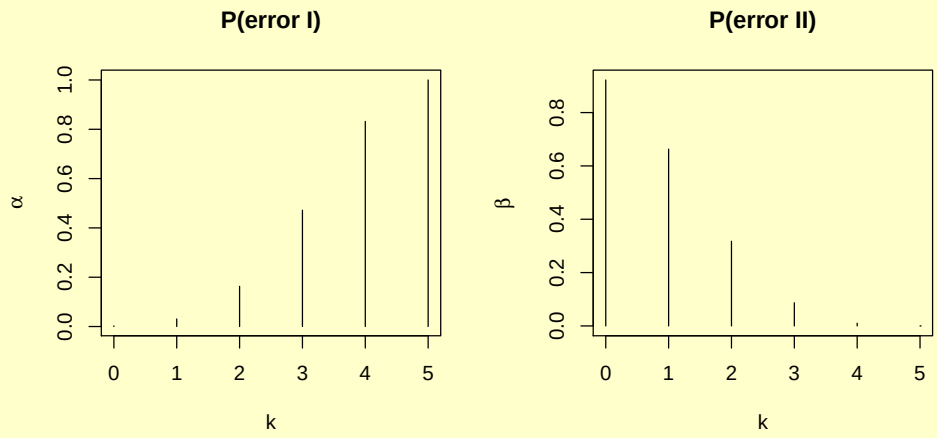
```
> errores(50, 0.7, 0.4)[24:29, ]
```

```
    k       alfa       beta
24 23 0.00034127 0.15616833
25 24 0.00093318 0.09780736
26 25 0.00236955 0.05734376
27 26 0.00559217 0.03140555
28 27 0.01227613 0.01603476
29 28 0.02508704 0.00761743
```

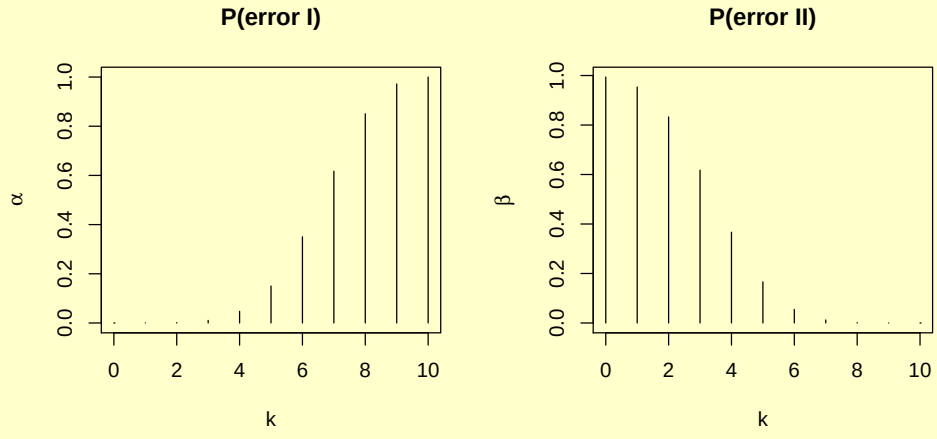
Vemos, pues, que para $n = 50$ y $k = 26$ sí se cumplen las condiciones $\alpha \leq 0{,}01$y $\beta \leq 0{,}05$. Posiblemente sea posible encontrar un valor de n algo menor para el que se consiga el mismo efecto.

Podemos ver los resultados anteriores gráficamente:

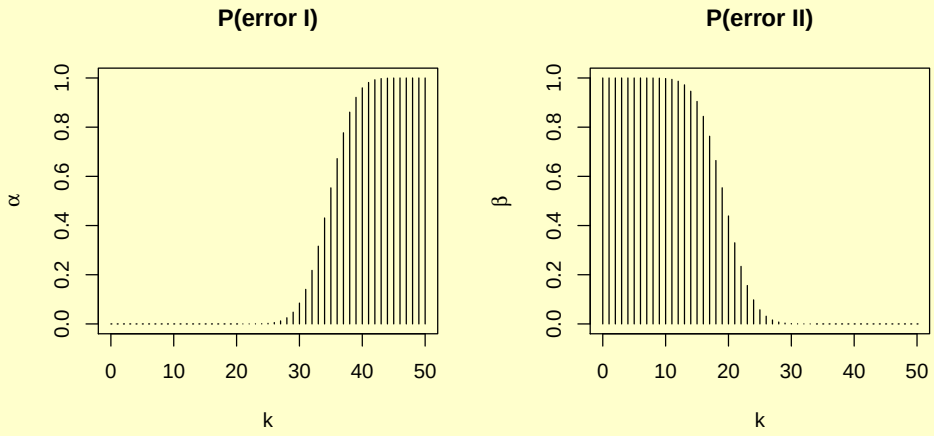
```
> pV = 0.7
> pF = 0.4
> n = 5
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> k = 0:n
> par(mfrow = c(1, 2))
> plot(k, pa, main = "P(error I)", ylab = expression(alpha), type = "h")
> plot(k, pb, main = "P(error II)", ylab = expression(beta), type = "h")
```




```
> n = 10
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> k = 0:n
> par(mfrow = c(1, 2))
> plot(k, pa, main = "P(error I)", ylab = expression(alpha), type = "h")
> plot(k, pb, main = "P(error II)", ylab = expression(beta), type = "h")
```




```
> n = 50
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> k = 0:n
> par(mfrow = c(1, 2))
> plot(k, pa, main = "P(error I)", ylab = expression(alpha), type = "h")
> plot(k, pb, main = "P(error II)", ylab = expression(beta), type = "h")
```



Podemos, pues, observar, que el comportamiento de ambos tipos de error es inverso: a medida que aumenta el valor de k, aumenta α y disminuye β. Cuando el valor de n es pequeño es posible que no haya ningún valor k para el que se consiga simultáneamente que α y β queden por debajo de ciertos umbrales; a medida que n aumenta, los valores de los errores correspondientes se van separando, hay más valores donde escoger, y empieza a ser posible encontrar valores de k para los que ambos errores estén acotados.

c) **Aproximación por el teorema central del límite:** Si X es el número de detecciones positivas en n análisis, cuando la playa está contaminada $X \approx B(n, p_V)$, y cuando la playa está limpia $X \approx B(n, p_F)$. Cuando n es un número grande, tenemos:

$$\begin{aligned} B(n,p_V) &\approx N\left(np_V, \sqrt{np_V(1-p_V)}\right) \\ B(n,p_F) &\approx N\left(np_F, \sqrt{np_F(1-p_F)}\right) \end{aligned}$$

Desarrollar una nueva función en R que, utilizando esta aproximación, permita obtener directamente los valores de n y k para α y β preespecificados. Comprobar con la distribución exacta (la binomial original) la bondad de esta aproximación.

Solución:
Tenemos:

$P(\text{Error I}) = P(\text{Decidir playa No Contaminada}\,|C) = P(X \le k\,|C) \cong P\left(Z \le \frac{k-np_V}{\sqrt{np_V(1-p_V)}}\right) = \alpha \Rightarrow$

$$\frac{k-np_V}{\sqrt{np_V(1-p_V)}} = \texttt{qnorm(alfa)}$$

$P(\text{Error II}) = P\left(\text{Decidir playa Contaminada}\,|\overline{C}\right) = P\left(X > k\,|\overline{C}\right) \cong P\left(Z > \frac{k-np_F}{\sqrt{np_F(1-p_F)}}\right) = \beta \Rightarrow$

$$\frac{k-np_F}{\sqrt{np_F(1-p_F)}} = \texttt{qnorm(1-beta)}$$

El sistema de ecuaciones resultante es sencillo de resolver:

$$\begin{aligned} k &= np_V + \texttt{qnorm(alfa)}\sqrt{np_V(1-p_V)} \\ k &= np_F + \texttt{qnorm(1-beta)}\sqrt{np_F(1-p_F)} \end{aligned}$$

$$n\,(p_F - p_V) = \texttt{qnorm(alfa)}\sqrt{np_V\,(1-p_V)} - \texttt{qnorm(1-beta)}\sqrt{np_F\,(1-p_F)}$$

$$\frac{n}{\sqrt{n}} = \frac{\texttt{qnorm(alfa)}\sqrt{p_V\,(1-p_V)} - \texttt{qnorm(1-beta)}\sqrt{p_F\,(1-p_F)}}{(p_F - p_V)}$$

$$n = \left(= \frac{\texttt{qnorm(alfa)}\sqrt{p_V\,(1-p_V)} - \texttt{qnorm(1-beta)}\sqrt{p_F\,(1-p_F)}}{(p_F - p_V)} \right)^2$$

En nuestro caso:

```
> alfa = 0.01
> beta = 0.05
> n = ((qnorm(alfa) * sqrt(pV * (1 - pV)) - qnorm(1 - beta) * sqrt(pF *
    (1 - pF)))/(pF - pV))^2
> n
```

```
[1] 38.93248
```

```
> k = n * pV + qnorm(alfa) * sqrt(n * pV * (1 - pV))
> k
```

```
[1] 20.60092
```

Dado que hemos utilizado la aproximación del teorema central del límite y el valor de n no ha resultado demasiado grande, podemos proceder a realizar el cálculo exacto de los errores I y II utilizando la probabilidad binomial para estos valores de n y k:

```
> n = 39
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> data.frame(k = 18:22, alfa = formatC(pa[19:23], digits = 8, format = "f"),
    beta = formatC(pb[19:23], digits = 8, format = "f"))
```

```
   k       alfa       beta
1 18 0.00159754 0.17132632
2 19 0.00433693 0.10205863
3 20 0.01072884 0.05588017
4 21 0.02422287 0.02802650
5 22 0.04998419 0.01283359
```

Como vemos, no se ha conseguido exactamente el objetivo buscado ($\alpha \leq 0{,}01$ y $\beta \leq 0{,}05$), aunque nos hemos aproximado bastante. Podemos afinar un poco más eligiendo n algo mayor:

```
> n = 40
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> data.frame(k = 18:22, alfa = formatC(pa[19:23], digits = 8, format = "f"),
    beta = formatC(pb[19:23], digits = 8, format = "f"))
```

```
   k       alfa       beta
1 18 0.00085399 0.20892878
2 19 0.00241936 0.12976571
3 20 0.00625450 0.07435156
4 21 0.01477705 0.03916797
5 22 0.03195126 0.01891075
```

```
> n = 41
> pa = pbinom(0:n, n, pV)
```

```
> pb = 1 - pbinom(0:n, n, pF)
> data.frame(k = 20:24, alfa = formatC(pa[21:25], digits = 8, format = "f"),
    beta = formatC(pb[21:25], digits = 8, format = "f"))

   k       alfa       beta
1 20 0.00356990 0.09651722
2 21 0.00881127 0.05324140
3 22 0.01992931 0.02701364
4 23 0.04135975 0.01256936
5 24 0.07886301 0.00534723
```

```
> n = 42
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> data.frame(k = 20:24, alfa = formatC(pa[21:25], digits = 8, format = "f"),
    beta = formatC(pb[21:25], digits = 8, format = "f"))

   k       alfa       beta
1 20 0.00199749 0.12248270
2 21 0.00514231 0.07055173
3 22 0.01214668 0.03750475
4 23 0.02635844 0.01834707
5 24 0.05261073 0.00823608
```

```
> n = 43
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> data.frame(k = 20:24, alfa = formatC(pa[21:25], digits = 8, format = "f"),
    beta = formatC(pb[21:25], digits = 8, format = "f"))

   k       alfa       beta
1 20 0.00109693 0.15222499
2 21 0.00294094 0.09132412
3 22 0.00724362 0.05072354
4 23 0.01641021 0.02601014
5 24 0.03423413 0.01228048
```

```
> n = 44
> pa = pbinom(0:n, n, pV)
> pb = 1 - pbinom(0:n, n, pF)
> data.frame(k = 20:24, alfa = formatC(pa[21:25], digits = 8, format = "f"),
    beta = formatC(pb[21:25], digits = 8, format = "f"))

   k       alfa       beta
1 20 0.00059183 0.18558807
2 21 0.00165013 0.11568447
3 22 0.00423174 0.06696377
4 23 0.00999360 0.03589550
5 24 0.02175738 0.01777234
```

Vemos de esta forma que para conseguir exactamente $\alpha \leq 0{,}01$ y $\beta < 0{,}05$ es preciso elegir $n = 44$ y $k = 23$.

d) Con los valores de n y k así obtenidos, simular la realización de n análisis con agua contaminada y n análisis con agua limpia. En cada caso decidir si con la regla de decisión elegida se toma la decisión correcta. Repetir este proceso 50.000 veces. ¿Cuál es la proporción de veces que se concluye que el agua está contaminada cuando está

limpia? ¿Cuál es la proporción de veces que se concluye que está limpia cuando está contaminada?.

Solución:
La regla de decisión consiste en tomar n muestras de agua y contar el número de detecciones positivas X; si $X > k$ decidimos que la playa está contaminada y si $X \leq k$ decidimos que está limpia.

Para simular la aplicación de esta regla de decisión en n muestras de agua limpia, basta tener en cuenta que en esas condiciones, en cada muestra existe una probabilidad p_F de detectar el contaminante (un falso positivo); por tanto $X \approx B(n, p_F)$. Podemos simular el valor de X mediante:

```
> n = 44
> pF = 0.4
> X = rbinom(1, n, pF)
> X

[1] 20
```

e implementar la regla de decisión mediante:

```
> k = 23
> decision = ifelse(X > k, "Playa Contaminada", "Playa Limpia")
> decision

[1] "Playa Limpia"
```

La siguiente función realiza la simulación y toma la decisión:

```
> simuLimpia = function(n, k, pF) {
     X = rbinom(1, n, pF)
     decision = ifelse(X > k, "Playa Contaminada", "Playa Limpia")
     return(decision)
 }
```

Replicamos 50.000 veces la toma de datos y la toma de la decisión:

```
> simulaciones = replicate(50000, simuLimpia(44, 23, 0.4))
```

y contamos el número de decisiones de cada clase que se han tomado, así como sus proporciones:

```
> table(simulaciones)

simulaciones
Playa Contaminada      Playa Limpia
             1805             48195

> prop.table(table(simulaciones))

simulaciones
Playa Contaminada      Playa Limpia
           0.0361            0.9639
```

Vemos de esta forma que con nuestra regla de decisión, la proporción de veces que hemos decidido que la playa está contaminada cuando realmente está limpia es 0.0361. Esta proporción es la que corresponde al error β que, tal como pretendíamos, es inferior a 0.05.

Un razonamiento similar nos conduce a construir la siguiente función que nos permite contar el número de veces que decidiríamos que está limpia una playa que realmente se encuentra contaminada:

```
> simuContaminada = function(n, k, pV) {
    X = rbinom(1, n, pV)
    decision = ifelse(X > k, "Playa Contaminada", "Playa Limpia")
    return(decision)
 }
```

Replicamos 50.000 veces la toma de datos y la toma de la decisión:

```
> simulaciones = replicate(50000, simuContaminada(44, 23, 0.7))
```

y contamos el número de decisiones de cada clase que se han tomado, así como sus proporciones:

```
> table(simulaciones)

simulaciones
Playa Contaminada      Playa Limpia
            49520               480

> prop.table(table(simulaciones))

simulaciones
Playa Contaminada      Playa Limpia
           0.9904             0.0096
```

Vemos de esta forma que con nuestra regla de decisión, la proporción de veces que hemos decidido que la playa está limpia cuando realmente está contaminada es 0.0096. Esta proporción es la que corresponde al error α que, tal como pretendíamos, es inferior a 0.01.

2. Se dispone ahora de un nuevo método de análisis de agua que en lugar de dar simplemente positivo o negativo, permite determinar la concentración exacta de contaminante en cada muestra. De acuerdo con la normativa sanitaria, la playa está contaminada si la concentración media μ de contaminante supera los 20 pg/cc. Como no se puede monitorizar el agua de toda la playa, se toman n muestras de agua repartidas a lo largo del litoral y se adopta como regla de decisión la siguiente:

   a) Se fija un valor μ_0.

   b) Se calcula la concentración media $\overline{X}$ de contaminante en las n muestras. Si $X \leq \mu_0$ se decide que la playa no está contaminada y se mantiene abierta al baño.

   c) Si $\overline{X} > \mu_0$ se decide que la playa está contaminada y se ordena su cierre.

   a) Suponiendo que la distribución de la concentración del contaminante es normal con desviación típica conocida $\sigma = 2$, **desarrollar una función en R que dados los valores de n y μ_0 calcule:**

      1) La probabilidad de mantener abierta al baño la playa cuando en realidad está contaminada:

$$\alpha = P\left(\overline{X} \leq \mu_0 \,|Cont\ (\mu > 20)\right)$$

      2) La probabilidad de cerrar la playa cuando en realidad no está contaminada:

$$\beta\left(\mu\right) = P\left(\overline{X} > \mu_0 \,|No\,Cont\ (\mu \leq 20)\right)$$

(Nótese que cuando la playa no está contaminada, entonces la concentración de contaminante es un valor desconocido $\mu < 20$ pg/cc; por tanto β es una función de dicho valor; constrúyase en tal caso β para 15 posibles valores de μ entre 16 y 20 (en R : `mu=seq(16,20,length=15)`).

3) Representa gráficamente estas funciones.

Solución:

Sea μ la concentración media (desconocida) de contaminante en la playa. En realidad, tanto α como β dependen de cuál sea el valor de μ. Concretamente, dado que $\sigma = 2$:

$$\alpha(\mu) = P\left(\overline{X} \leq \mu_0 \,|X \approx N(\mu, 2)\,,\ \text{siendo } \mu > 20\right)$$

$$\beta(\mu) = P\left(\overline{X} > \mu_0 \,|X \approx N(\mu, 2)\,,\ \text{siendo } \mu \leq 20\right)$$

Teniendo en cuenta que si la concentración sigue una distribución normal, entonces $\overline{X} \approx N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$, resulta:

$$\alpha(\mu) = P\left(\frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \leq \frac{\mu_0-\mu}{\sigma/\sqrt{n}}\middle|\, \mu \leq 20\right) = P\left(Z \leq \frac{\mu_0-\mu}{\sigma/\sqrt{n}}\middle|\, \mu \leq 20\right) = \text{pnorm}\left(\frac{\mu_0-\mu}{\sigma/\sqrt{\text{n}}}\right)$$

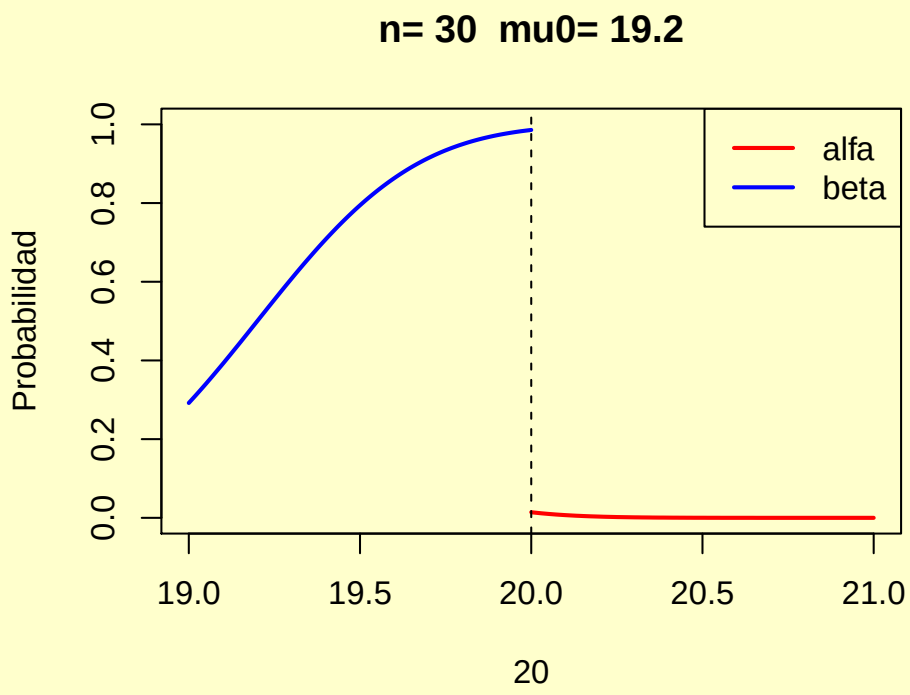
$$\beta(\mu) = P\left(\frac{\overline{X}-\mu}{\sigma/\sqrt{n}} > \frac{\mu_0-\mu}{\sigma/\sqrt{n}}\middle|\, \mu > 20\right) = P\left(Z > \frac{\mu_0-\mu}{\sigma/\sqrt{n}}\middle|\, \mu > 20\right) = 1-\text{pnorm}\left(\frac{\mu_0-\mu}{\sigma/\sqrt{\text{n}}}\right)$$

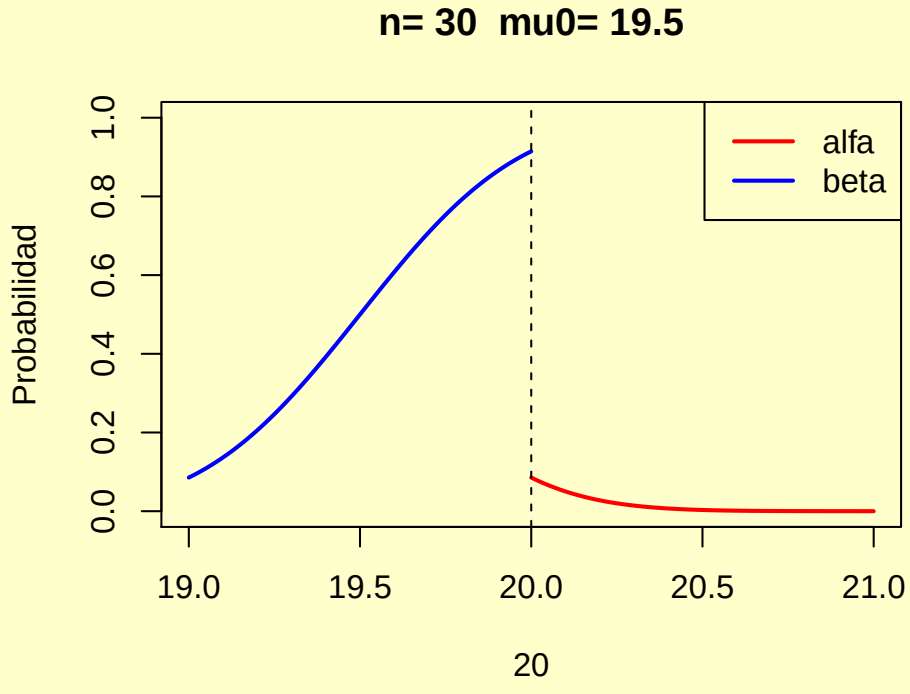
(donde α se calcularía sólo para $\mu \leq 20$ y β para $\mu > 20$). Podemos implementar α y β como funciones de μ_0, n y μ en R del siguiente modo

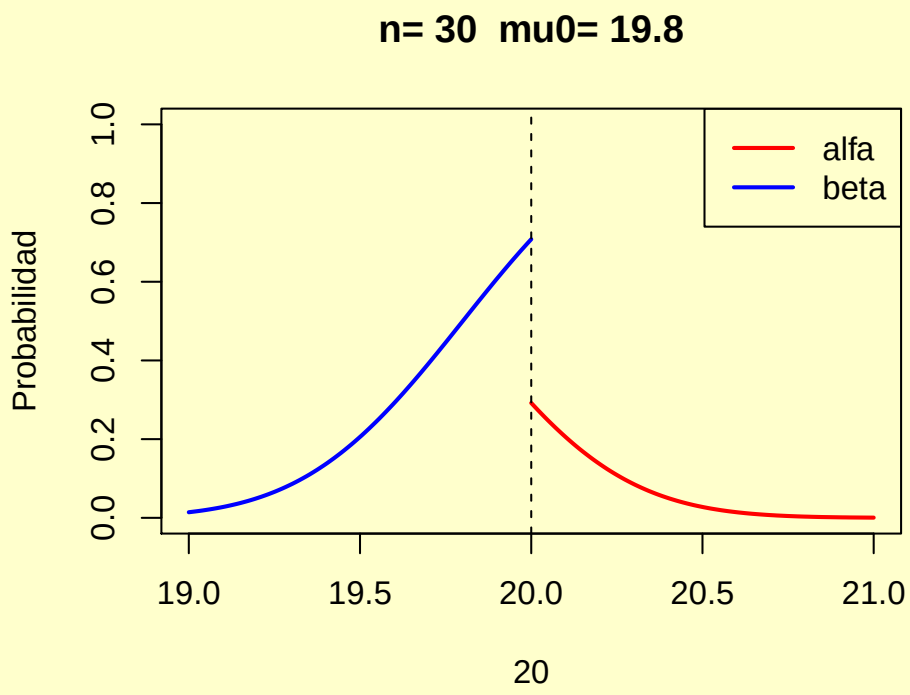
```
alfa = function(n, mu0, mu) {
    pnorm((mu0 - mu)/(2/sqrt(n)))
}
beta = function(n, mu0, mu) {
    1 - pnorm((mu0 - mu)/(2/sqrt(n)))
}
```

El siguiente código permite representarlas gráficamente para valores de μ entre 19 y 21 (téngase en cuenta que α sólo se define si $\mu > 20$, y β si $\mu \leq 20$). Hemos elegido a modo de ejemplo los valores $n = 30$ y $\mu_0 = 19{,}2$.

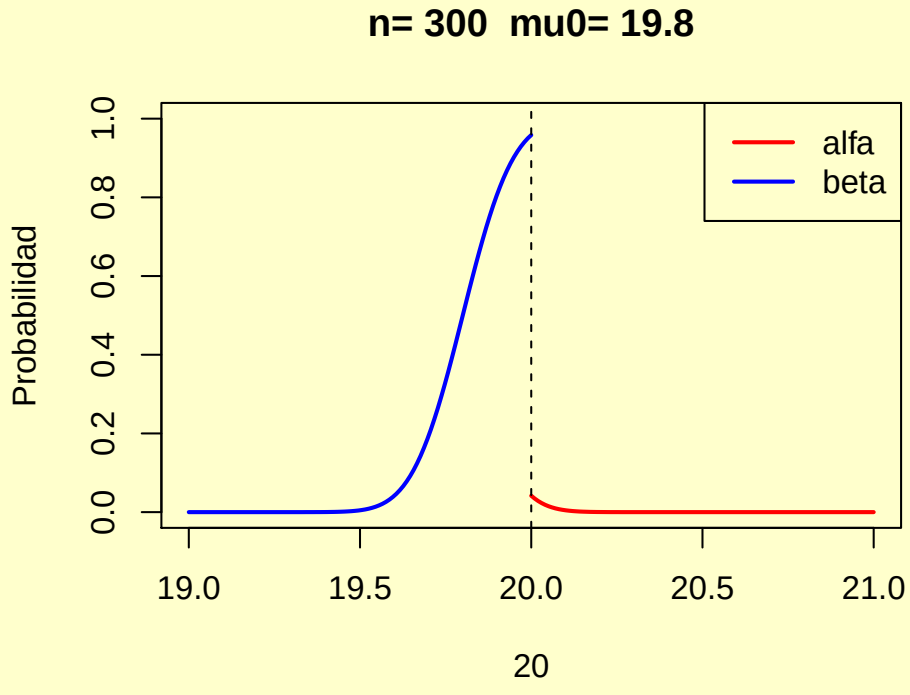
```
n = 30
mu0 = 19.2
mu.menorQue20 = seq(19, 20, length = 100)
mu.mayorQue20 = seq(20, 21, length = 100)
alfaValues = NULL
for (mu in mu.mayorQue20) alfaValues = rbind(alfaValues, alfa(n,
    mu0, mu))
betaValues = NULL
for (mu in mu.menorQue20) betaValues = rbind(betaValues, beta(n,
    mu0, mu))
plot(mu.mayorQue20, alfaValues, type = "l", col = "red", lwd = 2,
    xlim = c(19, 21), ylim = c(0, 1), xlab = mu, ylab = "Probabilidad")
lines(mu.menorQue20, betaValues, col = "blue", lwd = 2, ylim = c(0,
    1), xlab = mu, ylab = "Probabilidad")
abline(v = 20, lty = 2)
legend("topright", c("alfa", "beta"), col = c("red", "blue"),
    lwd = 2)
title(paste("n=", n, " mu0=", mu0))
```


n= 30 mu0= 19.2
Probabilidad
0.0
0.2
0.4
0.6
0.8
1.0
alfa
beta
19.0
19.5
20.0
20.5
21.0
20

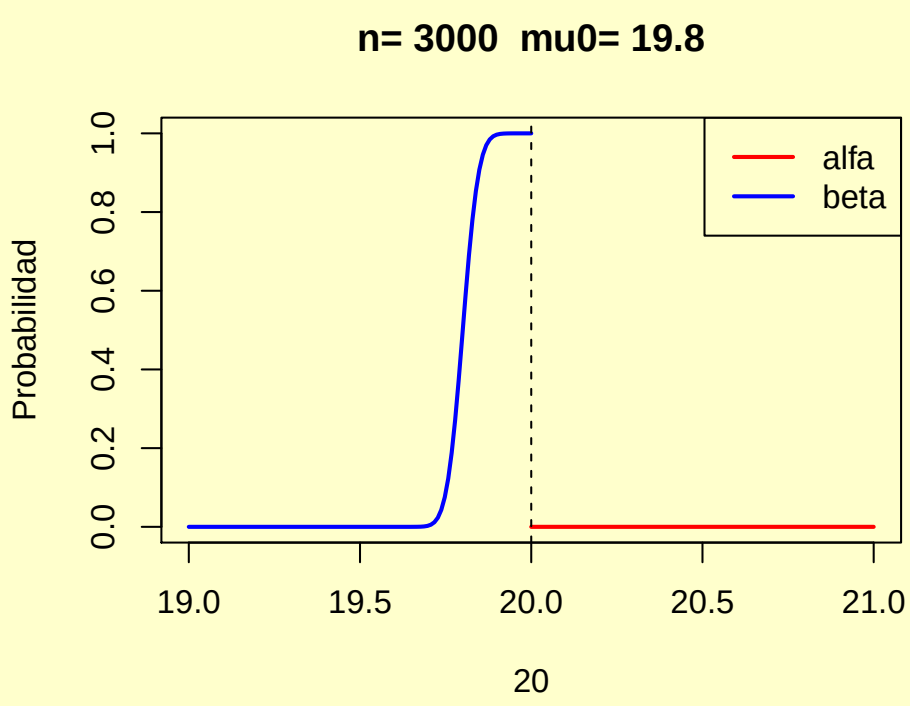

n= 30 mu0= 19.5
Probabilidad
0.0
0.2
0.4
0.6
0.8
1.0
alfa
beta
19.0
19.5
20.0
20.5
21.0
20



Vemos así que para un n fijo, el aumentar el valor de μ_0 tiene el efecto de incrementar α y disminuir β. Si para el mismo valor de $\mu_0 = 19{,}8$ incrementamos el tamaño de la muestra, hasta $n = 300$, el resultado es el siguiente:



Y para $n = 3000$:



Vemos así que al incrementar el tamaño de la muestra, los valores de α disminuyen para todos los $\mu > 20$. Al mismo tiempo la función β se va haciendo más vertical, lo que significa que β alcanza valores más pequeños más cerca de $\mu = 20$ por la izquierda (esto es, es más difícil decidir que $\mu > 20$ cuando μ es poco menor que 20).

Los gráficos anteriores nos muestran que el valor más grande de α se obtiene siempre para $\mu = 20$. Por tanto, la probabilidad:

$$P\left(\overline{X} \leq \mu_0 \,|\mu = 20\right)$$

constituye una cota superior del error α, esto es, $\alpha \leq P\left(\overline{X} \leq \mu_0 \,|\mu = 20\right)$. Entonces:

$$\alpha \leq P\left(\overline{X} \leq \mu_0 \,|\mu = 20\right) = P\left(Z \leq \frac{\mu_0 - 20}{2/\sqrt{n}}\right) = \text{pnorm}\left(\frac{\mu_0 - 20}{2/\sqrt{n}}\right)$$

Asimismo si la playa no está contaminada, la concentración real del contaminante será algún valor $\mu_1 < 20$ (obviamente desconocido). En este caso, la probabilidad de cerrar la playa sería:

$$\beta = P\left(\overline{X} > \mu_0 \,|\mu = \mu_1 < 20\right) = P\left(Z > \frac{\mu_0 - \mu_1}{2/\sqrt{n}}\right) = 1 - \text{pnorm}\left(\frac{\mu_0 - \mu_1}{2/\sqrt{\text{n}}}\right)$$

Como el valor de μ_1 no se conoce, lo que haremos en la práctica es calcular β para diversos valores de μ_1, lo que nos da una idea de como es el error β en función de μ_1 (si bien tras haber visto los gráficos anteriores ya tenemos una idea bastante clara). La siguiente función en R calcula, a partir de n y μ_0, una cota superior de α y diversos valores de β correspondientes a una colección de 15 posibles valores de μ_1, elegidos entre 16 y 20:

```
> errores=function(n,mu0){
        alfa=pnorm((mu0-20)/(2/sqrt(n)))
        mu1=seq(16,20,length=15)
    beta=1-pnorm((mu0-mu1)/(2/sqrt(n)))
    potencia=1-beta
    funcionBeta=data.frame(mu1=mu1,
```

```
                        beta=formatC(beta,format="f",digits=8),
                        potencia=formatC(potencia,format="f",digits=8))
        return(list(alfa=alfa,funcionBeta=funcionBeta))
 }
```

```
> errores(30, 19.5)

$alfa
[1] 0.08545176

$funcionBeta
        mu1       beta   potencia
1  16.00000 0.00000000 1.00000000
2  16.28571 0.00000000 1.00000000
3  16.57143 0.00000000 1.00000000
4  16.85714 0.00000000 1.00000000
5  17.14286 0.00000000 1.00000000
6  17.42857 0.00000001 0.99999999
7  17.71429 0.00000050 0.99999950
8  18.00000 0.00001996 0.99998004
9  18.28571 0.00044137 0.99955863
10 18.57143 0.00549530 0.99450470
11 18.85714 0.03915842 0.96084158
12 19.14286 0.16401837 0.83598163
13 19.42857 0.42245568 0.57754432
14 19.71429 0.72134630 0.27865370
15 20.00000 0.91454824 0.08545176
```

β prefijados, para un valor μ_1 también preespecificado. En efecto, tal como hemos visto, en el peor de los casos $\alpha = \text{pnorm}\left(\frac{\mu_0-20}{2/\sqrt{n}}\right)$, por lo que:

$$\frac{\mu_0 - 20}{2/\sqrt{n}} = \text{qnorm}\,(\alpha)$$

Asimismo:

$$1 - \beta = P\left(Z \leq \frac{\mu_0 - \mu_1}{2/\sqrt{n}}\right)$$

de donde:

$$\frac{\mu_0 - \mu_1}{2/\sqrt{n}} = \text{qnorm}\,(1-\beta)$$

De esta forma para determinar los valores de μ_0 y n para un valor de μ_1 determinado hemos de resolver el sistema:

$$\begin{cases} \frac{\mu_0-20}{2/\sqrt{n}} & = \text{qnorm}\,(\alpha) \\ \frac{\mu_0-\mu_1}{2/\sqrt{n}} & = \text{qnorm}\,(1-\beta) \end{cases}$$

Despejamos $\mu_0$en ambas ecuaciones:

$$\begin{cases} \mu_0 & = \frac{2}{\sqrt{n}}\text{qnorm}\,(\alpha) + 20 \\ \mu_0 & = \frac{2}{\sqrt{n}}\text{qnorm}\,(1-\beta) + \mu_1 \end{cases}$$

e igualamos y despejamos n:

$$\begin{aligned} \frac{2}{\sqrt{n}}\text{qnorm}\,(\alpha) + 20 &= \frac{2}{\sqrt{n}}\text{qnorm}\,(1-\beta) + \mu_1 \\ \frac{2}{\sqrt{n}}(\text{qnorm}\,(\alpha) - \text{qnorm}\,(1-\beta)) &= \mu_1 - 20 \\ n &= \left(\frac{2(\text{qnorm}\,(\alpha) - \text{qnorm}\,(1-\beta))}{\mu_1 - 20}\right)^2 \end{aligned}$$

El valor de μ_0 se calcula en cualquiera de las ecuaciones anteriores una vez que se haya calculado n. Podemos implementar fácilmente estos cálculos en R : (utilizamos la función `ceiling()` para redondear por exceso, ya que n debe ser un número entero)

```
> muestreo = function(alfa, beta, mu1) {
     n = ceiling((2 * (qnorm(alfa) - qnorm(1 - beta))/(mu1 - 20))^2)
     mu0 = 2 * qnorm(alfa)/sqrt(n) + 20
     return(c(n = n, mu0 = mu0))
 }
```

Por ejemplo, para $\alpha = 0{,}05$, $\beta = 0{,}10$ con $\mu_1 = 21$ esta función produce el siguiente resultado:

```
> muestreo(alfa = 0.05, beta = 0.1, mu1 = 19)

       n      mu0
35.00000 19.44394
```

lo que significa que si se desea garantizar un error α (decidir que está limpia cuando está contaminada) máximo de 0.05 y un error β (decidir que está contaminada cuando está limpia siendo la concentración real $\mu = 19$) de 0.10, se deben tomar $n = 35$ muestras de agua de la playa y decidir que está contaminada si la concentración media en esas muestras supera el valor $\mu_0 = 19{,}44394$.

Nótese que el error β es 0.10 si la concentración es 19 *pg/cc*. Si la concentración fuera mayor, pero sin llegar al valor 20 (de acuerdo con la norma sanitaria la playa no podría declararse contaminada) el error β aumentaría, es decir, es más fácil decretar que la playa está contaminada. En algún sentido, esto significa que *no nos importa demasiado* equivocarnos diciendo que la playa está contaminada si la concentración ya supera el valor 19.

b) En particular, y como aplicación práctica, utilizar este *script* para calcular los valores de n y μ_0 que permitan garantizar unas tasas de error $\alpha = 0{,}001$ y $\beta = 0{,}01$ (cuando $\mu = 18$).

Solución:

```
> muestreo(alfa = 0.001, beta = 0.01, mu1 = 18)

       n      mu0
30.00000 18.87161
```