1. Enunciado

En una empresa el tiempo que se tarda en fabricar una unidad de cierto producto se comporta como una variable aleatoria con distribución exponencial $\text{Exp}(\lambda)$. Cuando el sistema de producción funciona correctamente se verifica que $\lambda = \lambda_0 = 1$; en caso contrario, $\lambda < \lambda_0$. Para analizar el funcionamiento del proceso de producción se consideran los tiempos que se tarda en fabricar $n = 16$ unidades, elegidas al azar.

1. Contrasta las hipótesis $H_0\colon \lambda = \lambda_0$, $H_1\colon \lambda < \lambda_0$ a nivel de significación $\alpha = 0{,}05$.

2. Calcula el P-valor del contraste para la siguiente muestra:

   ```
   c(0.9, 1.3, 0.2, 0.2, 0.5, 3, 1.3, 0.6, 1.1, 0.2, 1.5, 0.9,
   1.3, 4.5, 1.2, 1.1)
   ```

3. Calcula el mínimo tamaño de muestra para que la potencia sea 0,8 para $\lambda = \frac{3}{4}$.

4. ¿Qué cambiaría si el contraste fuera $H_0\colon \lambda \geq \lambda_0$, $H_1\colon \lambda < \lambda_0$.

2. Resolución

```
> anchura <- 55 # del renglón
> options(width=anchura)
> landa0 <- 1
> n <- 16
> alfa <- 0.05
> set.seed(1) # para generar la muestra del enunciado
> X <- 0.1+round(rexp(n,landa0),1) # 0.1+ para evitar el cero
> cat(deparse(X,anchura),sep="\n") # para poder copiar y pegar

c(0.9, 1.3, 0.2, 0.2, 0.5, 3, 1.3, 0.6, 1.1, 0.2, 1.5, 0.9,
1.3, 4.5, 1.2, 1.1)
```

2.1. Contrasta las hipótesis

Teniendo en cuenta las hipótesis parece razonable rechazar H_0 cuando la estimación de λ sea menor de lo esperado bajo H_0. El estimador máximo-verosímil de λ es $\hat{\lambda} = \frac{n}{\sum X_i}$ y por tanto la región crítica se puede plantear como

$$
\begin{aligned}
\text{R.C.} &= \left\{(x_1,\ldots,x_n) \,\middle|\, \hat{\lambda} < c\right\} = \left\{(x_1,\ldots,x_n) \,\middle|\, \frac{n}{\sum x_i} < c\right\} \\
&= \left\{(x_1,\ldots,x_n) \,\middle|\, \sum x_i > \frac{n}{c}\right\} = \left\{(x_1,\ldots,x_n) \,\middle|\, \sum x_i > k\right\}
\end{aligned}
$$

$$
\begin{aligned}
0{,}05 = \alpha &= P\left(\text{R.C.} \mid H_0\right) = P\left(\sum X_i > k \;\middle|\; H_0\right) = P\left[\gamma(n, \lambda_0) > k\right] \\
&\Longrightarrow k = 23{,}097
\end{aligned}
$$

Haciendo el cálculo con R:

```
> (k <- qgamma(1-alfa, n, landa0))
[1] 23.09713
```

Dada la muestra del enunciado

```
> X
 [1] 0.9 1.3 0.2 0.2 0.5 3.0 1.3 0.6 1.1 0.2 1.5 0.9
[13] 1.3 4.5 1.2 1.1
```

queda comprobar si el valor del estadístico $\sum x_i$ cae dentro de la R.C.:

```
> print(sum(X)) > k # ¿rechazamos H0?
[1] 19.8
[1] FALSE
```

2.2. Calcula el P-valor

El P-valor

$$
P\left[\sum X_i > \sum x_i \;\middle|\; H_0\right] = P\left[\gamma(n, \lambda_0) > \sum x_i\right]
$$

vale

```
> (pval <- 1 - pgamma (sum(X), n, landa0))
[1] 0.1671028
> pval < alfa # ¿rechazamos H0?
[1] FALSE
```

2.3. Calcula tamaño de muestra para potencia

Bajo la hipótesis de que $\lambda = \frac{3}{4}\lambda_0$ se tiene que $\sum X_i \equiv \gamma\left(n, \frac{3}{4}\lambda_0\right)$, luego

$$
\begin{aligned}
\text{potencia} &= P\left[\text{R.C.} \;\middle|\; \lambda = \frac{3}{4}\lambda_0\right] = P\left[\sum X_i > k_n \;\middle|\; \lambda = \frac{3}{4}\lambda_0\right] \\
&= P\left[\gamma\left(n, \frac{3}{4}\lambda_0\right) > k_n\right]
\end{aligned}
$$

Se puede hacer una búsqueda del n necesario para obtener la potencia deseada (0,8). Se puede comprobar que con el tamaño muestral[1] del enunciado, $n = 16$, no es suficiente,

[1] Nótese que k depende de n, es decir, $k = k_n$, luego hay que redefinir k.

```
> potMin <- 0.8
> landa1 <- 3/4 * landa0
> k   <- function (m) qgamma(1-alfa, m, landa0)
> pot <- function (m) 1 - pgamma(k(m), m, landa1)
> n
```

```
[1] 16
```

```
> k(n)
```

```
[1] 23.09713
```

```
> pot(n)
```

```
[1] 0.342771
```

así que buscaremos a partir de ahí. Varias posibilidades:

- Búsqueda exhaustiva. Podemos buscar
  - hasta un valor fijo suficientemente grande, escogido a ojo:

    ```
    > enes <- n:100
    > (probs <- setNames (pot(enes), enes))
    ```

    ```
           16        17        18        19        20 
    0.3427710 0.3553391 0.3676914 0.3798345 0.3917742 
           21        22        23        24        25 
    0.4035154 0.4150626 0.4264195 0.4375896 0.4485762 
           26        27        28        29        30 
    0.4593821 0.4700101 0.4804626 0.4907421 0.5008508 
           31        32        33        34        35 
    0.5107910 0.5205647 0.5301740 0.5396210 0.5489075 
           36        37        38        39        40 
    0.5580355 0.5670070 0.5758238 0.5844879 0.5930009 
           41        42        43        44        45 
    0.6013649 0.6095817 0.6176530 0.6255808 0.6333668 
           46        47        48        49        50 
    0.6410128 0.6485208 0.6558924 0.6631295 0.6702339 
           51        52        53        54        55 
    0.6772074 0.6840518 0.6907689 0.6973605 0.7038283 
           56        57        58        59        60 
    0.7101741 0.7163996 0.7225067 0.7284971 0.7343725 
           61        62        63        64        65 
    0.7401347 0.7457853 0.7513261 0.7567588 0.7620850 
           66        67        68        69        70 
    0.7673065 0.7724248 0.7774418 0.7823589 0.7871779 
           71        72        73        74        75 
    0.7919003 0.7965277 0.8010617 0.8055039 0.8098558 
    ```

```
       76        77        78        79        80
0.8141190 0.8182950 0.8223853 0.8263914 0.8303147
       81        82        83        84        85
0.8341567 0.8379189 0.8416026 0.8452093 0.8487404
       86        87        88        89        90
0.8521972 0.8555811 0.8588935 0.8621355 0.8653086
       91        92        93        94        95
0.8684141 0.8714531 0.8744269 0.8773369 0.8801841
       96        97        98        99       100
0.8829698 0.8856951 0.8883613 0.8909694 0.8935206
```

```
> probs [probs > potMin] [1]
```

```
       73
0.8010617
```

- hasta obtener la potencia necesaria, y parar entonces:

```
> m <- n
> while (pot(m) < potMin) m <- m+1
> c (m, pot(m))
```

```
[1] 73.0000000  0.8010617
```

- Mediante un algoritmo numérico. La función `uniroot` busca una raíz de su argumento en cierto intervalo.

```
> fun <- function (m) pot(m) - potMin
> (sol <- uniroot (fun, c(n,1000))$root)
```

```
[1] 72.76403
```

```
> c(floor(sol), pot(floor(sol)))
```

```
[1] 72.0000000  0.7965277
```

```
> c(ceiling(sol), pot(ceiling(sol)))
```

```
[1] 73.0000000  0.8010617
```

2.4. ¿Qué cambiaría con $H_0\colon \lambda \geq \lambda_0$?

Sea $\lambda > \lambda_0$. Entonces

$$\begin{aligned}
P\,(\text{R.C.} \mid \lambda) &= P\left(\sum X_i > k \,\middle|\, \lambda\right) = P\left(\gamma(n,\lambda) > k\right) = P\left(\gamma(n,1) > k\lambda\right) \\
&< P\left(\gamma(n,1) > k\lambda_0\right) = P\left(\gamma(n,\lambda_0) > k\right) = P\left(\sum X_i > k \,\middle|\, \lambda_0\right) \\
&= P\,(\text{R.C.} \mid \lambda_0) = \alpha \\
&\Longrightarrow \forall \lambda > \lambda_0, \quad P\,(\text{R.C.} \mid \lambda) \leq \alpha
\end{aligned}$$

Por tanto, la región crítica sería la misma que para $H_0\colon \lambda = \lambda_0$ y el contraste se resolvería de manera exactamente igual.