1. Stopping generating new simulation data

Write a program to generate standard normal random variables until you have generated n of them, where $n \geq 100$ is such that $S/\sqrt{n} < 0{,}01$, where S is the sample standard deviation of the n data values. Note that this is the "Method for Determining When to Stop Generating New Data". Also, answer the following questions:

Debido a que se va a generar variables aleatorias **normales estándar** X_i, es decir, $X_i \sim \mathcal{N}(0,1)$ y se debe parar de generar cuando $\frac{S}{\sqrt{n}} < 0{,}01$.

```r
x <- numeric()
s <- 1
n <- 0

while(n < 100 || s / sqrt(n) >= 0.01) {
  x <- c(x, rnorm(1))
  n <- length(x)
  s <- sd(x)
}

cat("Cantidad de Xi generados", n, "\n")

## Cantidad de Xi generados 10246

cat("Desviación estándar", round(s, 5), "\n")

## Desviación estándar 1.01221

cat("Error estándar S/sqrt(n)", round(s/sqrt(n), 5), "\n")

## Error estándar S/sqrt(n) 0.01
```

En el código se declara una variable s inicializada en 1 para que luego sea nuevamente computada a la desviación estándar de x que es un vector que va a almacenar todas las variables normales aleatorias generadas. Según el enunciado tenemos una condición de que al menos deben haber 100 variables aleatorias generadas ($n \geq 100$) y que el error estándar sea menor a 0.01 ($S/\sqrt{n} < 0{,}01$). En el ciclo `while`, por lo tanto, tiene sentido que el ciclo continue si alguna de las afirmaciones anteriores son falsas, por eso queda la condición de esa manera. Se hace uso de la función `rnorm(1)` para que genere una variable aleatoria normal con los valores por defecto (promedio 0 y desviación 1).

1.1. How many normals do you think will be generated? Give an analytic estimate.

1.1.1. Respuesta

Aunque ya se tengan los resultados de la simulación, se puede hacer un estimado analítico con la condición $S/\sqrt{n} < 0{,}01$ donde se puede despejar n para saber cuántas variables se necesitan para parar el criterio.

Se tiene en primer lugar la inecuación:

$$\frac{S}{\sqrt{n}} < 0{,}01$$

Se eleva ambas partes con menos 1:

$$\left(\frac{S}{\sqrt{n}}\right)^{-1} > 0{,}01^{-1}$$
$$\frac{\sqrt{n}}{S} > 100$$

Multiplicando ambas partes por S:

$$\cancel{S} \times \frac{\sqrt{n}}{\cancel{S}} > 100 \times S$$
$$\sqrt{n} > 100 \times S$$

Ahora se cancela la raíz elevando ambas partes al cuadrado:

$$(\cancel{\sqrt{n}})^{\cancel{2}} > (100 \times S)^2$$

Quedando entonces:

$$n > 10000 \times S^2$$

Para hacer el ejercicio se utilizó en R la función `rnorm(1, mean = 0, sd = 1)`, de esta manera genera únicamente 1 valor con media 0 y desviación 1, por lo tanto, podemos hacer $S = 1$ para estimar cuántos n necesitamos para que se cumpla condición y deje de generar variables aleatorias, por lo tanto:

$$n > 10000 \times (1)^2$$
$$n > 10000$$

Este valor se acerca bastante al que se imprime en la simulación.

1.2. How many normals did you generate?

1.2.1. Respuesta

Se han generado `n=` 10246 normales.

1.3. What is the sample mean of all the normals generated?

1.3.1. Respuesta

Utilizando el comando `mean()` al vector `x` se obtiene 0,0097696.

1.4. What is the sample variance?

1.4.1. Respuesta

La varianza muestral se calcula como:

$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

Esto se puede sacar con la función `var()` de R, por lo tanto, aplicando eso al vector `x` se obtiene 1,0245682.

1.5. Comment on the results of (1.3) and (1.4). Were they surprising?

1.5.1. Respuesta

El resultado de 1.3 es la media de `x` que es 0,0097696 y el de 1.4 es la varianza muestral de `x` dando 1,0245682. Estos valores muy sorprendentes no fueron... Se espera que al simular muchas veces una distribución normal, esta nos dé los valores de la media y la desviación estándar al cuadrado. Pero sí es interesante que dado este algoritmo para detenerse después de cruzar un $S/\sqrt{n}$ nos dé valores muy cercanos a una distribución normal con $\mu = 0$ y $\sigma = 1$.

2. Gaining confidence with confidence intervals

We know that the $\mathcal{U}(-1,1)$ r.v. has mean 0. Use a sample of size 1000 to estimate the mean and give a 95 % confidence interval (CI). Does the CI contain 0? Repeat the above a large number of times (≥ 100). What percentage of time does the CI contain 0? Write your code so that it produces output similar to the following:

```
Number of trials: 10

Sample mean  lower bound  upper bound  contains mean?

    -0.0733      -0.1888       0.0422               1
    -0.0267      -0.1335       0.0801               1
    -0.0063      -0.1143       0.1017               1
    -0.0820      -0.1869       0.0230               1
    -0.0354      -0.1478       0.0771               1
    -0.0751      -0.1863       0.0362               1
    -0.0742      -0.1923       0.0440               1
     0.0071      -0.1011       0.1153               1
     0.0772      -0.0322       0.1867               1
    -0.0243      -0.1370       0.0885               1

100 percent of CI's contained the mean
```

2.1. Respuesta

Se sabe que para tener un intervalo de confianza de 95 %, el límite inferior y superior serán, respectivamente:

$$\left(\bar{X} - 1{,}96\frac{S}{\sqrt{n}}, \bar{X} + 1{,}96\frac{S}{\sqrt{n}}\right)$$

El siguiente código permite sacar el intervalo de confianza del 95 % y también el estimado de la media.

```r
n <- 1000
x <- runif(n, -1, 1)
xmean <- mean(x)
S <- sd(x)
L <- xmean - 1.96 * S/sqrt(n)
U <- xmean + 1.96 * S/sqrt(n)
cat("El estimado es", xmean, "\n")

## El estimado es -0.01769999

cat("95% está entre (", L, ", ", U, ") \n", sep="")

## 95% está entre (-0.05425794, 0.01885796)
```

Aquí se estimó la media de la distribución uniforme, dando un resultado de $-0{,}0177$, muy cercano a 0. El intervalo de confianza está entre $-0{,}0542579$ y $0{,}018858$, en este caso sí contiene a 0.[1] Ahora, se va a realizar el proceso unas 200 veces para conocer el porcentaje de intervalos de confianza que contienen 0.

```
trials <- 200
n <- 1000
true_mean <- 0

results <- data.frame(
  sample_mean = numeric(trials),
  lower_bound = numeric(trials),
  upper_bound = numeric(trials),
  contains_mean = integer(trials)
)

for (i in 1:trials){
  x <- runif(n, -1, 1)
  xmean <- mean(x)
  S <- sd(x)
  L <- xmean - 1.96 * S/sqrt(n)
  U <- xmean + 1.96 * S/sqrt(n)
  contains <- as.integer(L <= true_mean & true_mean <= U)

  results[i, ] <- c(xmean, L, U, contains)
}

cat("Número de intentos:", trials, "\n")

## Número de intentos: 200

print(results, digits= 4)

##     sample_mean lower_bound upper_bound contains_mean
## 1     0.0199645   -0.015891   0.0558202             1
## 2     0.0342190   -0.001083   0.0695213             1
## 3    -0.0043886   -0.039886   0.0311094             1
## 4     0.0094988   -0.025761   0.0447589             1
## 5     0.0079659   -0.028089   0.0440209             1
## 6     0.0020347   -0.033697   0.0377664             1
## 7     0.0106194   -0.024952   0.0461910             1
## 8    -0.0088935   -0.044907   0.0271201             1
```

[1] Aquí hay que hacer una aclaración, los autores utilizaron knitr, una herramienta para la generación de reportes dinámicos en R, entonces cada vez que se compile este archivo `.Rnw` va a salir una salida diferente y puede ser que ya el intervalo no contenga 0.

```
## 9    -0.0024477   -0.037986   0.0330911             1
## 10    0.0138358   -0.021823   0.0494952             1
## 11   -0.0277445   -0.063327   0.0078380             1
## 12   -0.0003710   -0.036058   0.0353156             1
## 13    0.0118711   -0.023813   0.0475557             1
## 14   -0.0122962   -0.049061   0.0244683             1
## 15   -0.0042528   -0.039474   0.0309685             1
## 16    0.0096765   -0.026285   0.0456379             1
## 17   -0.0124555   -0.049611   0.0246998             1
## 18    0.0146936   -0.021171   0.0505578             1
## 19   -0.0084436   -0.044355   0.0274673             1
## 20   -0.0013152   -0.037463   0.0348330             1
## 21   -0.0240895   -0.059246   0.0110674             1
## 22   -0.0173336   -0.052878   0.0182112             1
## 23    0.0023738   -0.033865   0.0386129             1
## 24    0.0180604   -0.018396   0.0545169             1
## 25    0.0217302   -0.014696   0.0581559             1
## 26   -0.0104135   -0.046763   0.0259365             1
## 27    0.0175230   -0.018296   0.0533423             1
## 28    0.0025968   -0.031902   0.0370960             1
## 29    0.0018093   -0.034079   0.0376971             1
## 30   -0.0015889   -0.038490   0.0353118             1
## 31    0.0069606   -0.028604   0.0425256             1
## 32   -0.0177219   -0.052905   0.0174614             1
## 33   -0.0216579   -0.058233   0.0149174             1
## 34   -0.0022139   -0.038668   0.0342401             1
## 35    0.0071967   -0.028553   0.0429462             1
## 36    0.0441168    0.007908   0.0803254             0
## 37   -0.0013089   -0.037477   0.0348595             1
## 38    0.0181460   -0.018594   0.0548856             1
## 39    0.0197039   -0.015769   0.0551767             1
## 40    0.0200501   -0.016112   0.0562124             1
## 41    0.0065212   -0.029902   0.0429443             1
## 42    0.0006384   -0.035674   0.0369507             1
## 43    0.0307422   -0.005276   0.0667605             1
## 44   -0.0259476   -0.061334   0.0094385             1
## 45    0.0200290   -0.015977   0.0560354             1
## 46    0.0054573   -0.031101   0.0420154             1
## 47    0.0195048   -0.016563   0.0555729             1
## 48    0.0430542    0.007518   0.0785905             0
## 49   -0.0254912   -0.061474   0.0104918             1
## 50    0.0095610   -0.026593   0.0457148             1
## 51    0.0008706   -0.034968   0.0367095             1
## 52   -0.0006827   -0.036851   0.0354861             1
```

```
## 53   -0.0156003   -0.051367   0.0201660             1
## 54   -0.0142682   -0.050449   0.0219124             1
## 55   -0.0350559   -0.070797   0.0006856             1
## 56    0.0142718   -0.021882   0.0504256             1
## 57    0.0060927   -0.030428   0.0426134             1
## 58    0.0051196   -0.031803   0.0420420             1
## 59   -0.0090637   -0.044250   0.0261228             1
## 60   -0.0081285   -0.043161   0.0269041             1
## 61   -0.0082764   -0.043996   0.0274434             1
## 62   -0.0029126   -0.038711   0.0328857             1
## 63   -0.0199913   -0.055236   0.0152532             1
## 64   -0.0194289   -0.055599   0.0167416             1
## 65   -0.0071183   -0.043204   0.0289677             1
## 66    0.0308946   -0.004684   0.0664733             1
## 67   -0.0212037   -0.056604   0.0141961             1
## 68   -0.0128739   -0.048394   0.0226463             1
## 69   -0.0212730   -0.057758   0.0152123             1
## 70   -0.0072707   -0.042279   0.0277377             1
## 71    0.0008460   -0.035890   0.0375822             1
## 72   -0.0017893   -0.038015   0.0344367             1
## 73    0.0170532   -0.018404   0.0525105             1
## 74    0.0219561   -0.013767   0.0576797             1
## 75    0.0030160   -0.032498   0.0385297             1
## 76   -0.0156734   -0.050970   0.0196235             1
## 77    0.0185552   -0.018147   0.0552578             1
## 78    0.0301246   -0.005138   0.0653873             1
## 79    0.0165100   -0.019569   0.0525891             1
## 80    0.0180046   -0.019124   0.0551331             1
## 81    0.0100945   -0.025206   0.0453956             1
## 82   -0.0004606   -0.035782   0.0348608             1
## 83    0.0086784   -0.026846   0.0442027             1
## 84   -0.0378245   -0.074012  -0.0016373             0
## 85    0.0189552   -0.016534   0.0544449             1
## 86    0.0059819   -0.030238   0.0422023             1
## 87    0.0155873   -0.020996   0.0521706             1
## 88   -0.0065733   -0.042031   0.0288844             1
## 89    0.0156252   -0.020263   0.0515132             1
## 90   -0.0061103   -0.041895   0.0296741             1
## 91    0.0155084   -0.019820   0.0508368             1
## 92   -0.0137805   -0.049408   0.0218472             1
## 93    0.0187632   -0.017983   0.0555096             1
## 94   -0.0014986   -0.037414   0.0344172             1
## 95    0.0057812   -0.029057   0.0406199             1
## 96    0.0277228   -0.008536   0.0639819             1
```

```
## 97   -0.0143180   -0.050118   0.0214824             1
## 98    0.0091817   -0.025388   0.0437510             1
## 99    0.0019480   -0.033936   0.0378318             1
## 100   0.0118412   -0.023150   0.0468326             1
## 101  -0.0176815   -0.053021   0.0176584             1
## 102  -0.0017249   -0.037785   0.0343349             1
## 103  -0.0455806   -0.081134  -0.0100272             0
## 104  -0.0128831   -0.048301   0.0225348             1
## 105  -0.0027030   -0.038849   0.0334433             1
## 106   0.0309335   -0.005098   0.0669647             1
## 107  -0.0073644   -0.042297   0.0275685             1
## 108  -0.0319742   -0.068071   0.0041228             1
## 109  -0.0050140   -0.041198   0.0311697             1
## 110  -0.0150870   -0.052320   0.0221456             1
## 111   0.0024803   -0.033646   0.0386064             1
## 112  -0.0052342   -0.040488   0.0300194             1
## 113  -0.0041399   -0.039699   0.0314193             1
## 114  -0.0153843   -0.050634   0.0198651             1
## 115  -0.0199169   -0.056134   0.0163000             1
## 116  -0.0204244   -0.056703   0.0158538             1
## 117   0.0210436   -0.013898   0.0559850             1
## 118   0.0102881   -0.025995   0.0465709             1
## 119  -0.0011791   -0.036481   0.0341230             1
## 120   0.0025604   -0.033245   0.0383657             1
## 121   0.0218732   -0.014263   0.0580096             1
## 122   0.0126629   -0.022507   0.0478327             1
## 123   0.0056923   -0.030237   0.0416221             1
## 124   0.0040422   -0.031132   0.0392163             1
## 125   0.0487916    0.012510   0.0850735             0
## 126  -0.0109882   -0.047204   0.0252278             1
## 127   0.0167426   -0.018899   0.0523844             1
## 128   0.0023855   -0.033473   0.0382441             1
## 129  -0.0077238   -0.043445   0.0279972             1
## 130   0.0100189   -0.024793   0.0448307             1
## 131  -0.0250632   -0.060677   0.0105503             1
## 132   0.0224976   -0.012585   0.0575806             1
## 133   0.0091443   -0.027104   0.0453925             1
## 134   0.0062196   -0.029628   0.0420673             1
## 135   0.0284327   -0.007355   0.0642204             1
## 136  -0.0062583   -0.041864   0.0293473             1
## 137   0.0012255   -0.034251   0.0367018             1
## 138  -0.0090470   -0.045086   0.0269915             1
## 139  -0.0117640   -0.047639   0.0241113             1
## 140  -0.0180202   -0.053485   0.0174446             1
```

```
## 141   0.0212207   -0.013526   0.0559675             1
## 142  -0.0163934   -0.052997   0.0202096             1
## 143   0.0045991   -0.030985   0.0401829             1
## 144  -0.0117983   -0.047318   0.0237213             1
## 145   0.0113114   -0.023540   0.0461627             1
## 146   0.0013240   -0.034303   0.0369512             1
## 147   0.0190568   -0.017104   0.0552176             1
## 148  -0.0116534   -0.047520   0.0242133             1
## 149   0.0111602   -0.025520   0.0478405             1
## 150   0.0053606   -0.030071   0.0407924             1
## 151   0.0057598   -0.029915   0.0414349             1
## 152   0.0090574   -0.026190   0.0443044             1
## 153  -0.0161896   -0.051702   0.0193230             1
## 154  -0.0083077   -0.043475   0.0268591             1
## 155   0.0034378   -0.031928   0.0388032             1
## 156   0.0052875   -0.030878   0.0414527             1
## 157   0.0060863   -0.029918   0.0420908             1
## 158  -0.0168365   -0.052907   0.0192336             1
## 159   0.0176617   -0.019588   0.0549119             1
## 160   0.0022533   -0.033863   0.0383696             1
## 161   0.0292167   -0.006906   0.0653392             1
## 162  -0.0385446   -0.073548  -0.0035413             0
## 163  -0.0351941   -0.070218  -0.0001704             0
## 164  -0.0018054   -0.038479   0.0348687             1
## 165   0.0008637   -0.034783   0.0365100             1
## 166   0.0077165   -0.027965   0.0433977             1
## 167   0.0082130   -0.027294   0.0437203             1
## 168  -0.0088570   -0.044455   0.0267411             1
## 169  -0.0342351   -0.069967   0.0014967             1
## 170   0.0116390   -0.024231   0.0475086             1
## 171  -0.0046088   -0.040359   0.0311413             1
## 172   0.0045677   -0.031589   0.0407248             1
## 173  -0.0019452   -0.037388   0.0334978             1
## 174   0.0310318   -0.006429   0.0684926             1
## 175  -0.0160528   -0.051798   0.0196924             1
## 176  -0.0087615   -0.044682   0.0271587             1
## 177  -0.0283470   -0.063532   0.0068382             1
## 178  -0.0157275   -0.052135   0.0206799             1
## 179  -0.0334389   -0.068840   0.0019624             1
## 180  -0.0200608   -0.056122   0.0160001             1
## 181   0.0085797   -0.027468   0.0446278             1
## 182  -0.0090172   -0.045356   0.0273212             1
## 183  -0.0049175   -0.040822   0.0309868             1
## 184   0.0343324   -0.001419   0.0700840             1
```

```
## 185   0.0077679   -0.027925   0.0434613            1
## 186   0.0317991   -0.003187   0.0667853            1
## 187  -0.0038990   -0.040010   0.0322121            1
## 188   0.0088476   -0.027746   0.0454410            1
## 189  -0.0055029   -0.042107   0.0311008            1
## 190  -0.0251823   -0.060097   0.0097326            1
## 191  -0.0037984   -0.039467   0.0318698            1
## 192   0.0070678   -0.029066   0.0432018            1
## 193  -0.0016435   -0.038212   0.0349253            1
## 194   0.0050599   -0.031313   0.0414328            1
## 195  -0.0186166   -0.054697   0.0174640            1
## 196  -0.0196905   -0.055552   0.0161714            1
## 197  -0.0378134   -0.073500  -0.0021267            0
## 198  -0.0049283   -0.040444   0.0305870            1
## 199  -0.0092888   -0.044479   0.0259016            1
## 200  -0.0079792   -0.043975   0.0280167            1
```

```
porcentaje <- mean(results$contains_mean) * 100

cat("\n", porcentaje, "% de los intervalos de",
"confianza contienen la media real\n")
```

```
##
##  96 % de los intervalos de confianza contienen la media real
```

En conclusión, el 96 % de los intervalos generados incluyen al 0.

3. Standard deviation of a proportion

Assume a manager is using the sample proportion $\hat{p}$ to estimate the proportion p of a new shipment of computer chips that are defective. He doesn't know p for this shipment, but in previous shipments it has been close to 0,01, that is 1 % of chips have been defective.

3.1. If the manager wants the standard deviation of $\hat{p}$ to be about 0,02, how large a sample should she take based on the assumption that the rate of defectives has not changed dramatically?

En clase se vieron los estimadores de probabilidad, donde se quiere estimar

$$p = (PX \in A)$$

Donde A es el subconjunto del espacio de estados de Ω de X. Es decir, para nuestro

problema, este subconjunto de espacios en la muestra que tomó la administradora del cargamento de chips defectuosos. Se puede definir la variable indicadora Z como:

$$Z = \begin{cases} 1, & X \in A \\ 0, & X \notin A \end{cases}$$

1 significa que sí está defectuoso y 0 que no lo está. Se puede escribir el estimador como:

$$p = E[Z]$$

En clase se realizó una demostración de como el valor esperado de Z se le puede asignar a p. Se define la varianza de Z como:

$$\begin{aligned} \text{Var}(Z) &= E[Z^2] - E[Z]^2 \\ &= (1^2 \times P(X \in A) + 0^2 \times P(X \notin A)) - (P(X \in A)) \\ &= p - p^2 \end{aligned}$$

Aplicando factorización:

$$\text{Var}(Z) = p(1-p)$$

Estimar p se puede realizar mediante el promedio muestral de Z, es decir, se puede estimar la proporción $\hat{p}$ con el promedio de los chips que son defectuosos:

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} Z_i$$

Se puede calcular entonces, la varianza del estimador:

$$\text{Var}(\hat{p}) = \frac{1}{n} \text{Var} \left(\sum_{i=1}^{n} Z_i \right)$$

Por independencia de los valores del cargamento de chips Z_i se puede meter la varianza de estos en la suma:

$$\begin{aligned} \text{Var}(\hat{p}) &= \frac{1}{n^2} \sum_{i=1}^{n} \text{Var}(Z_i) \\ &= \frac{1}{n^2} \sum_{i=1}^{n} p(1-p) = \frac{1}{n} p(1-p) \end{aligned}$$

Por lo tanto, una aproximación es:

$$\text{Var}(\hat{p}) \approx \frac{1}{n} \hat{p}(1-\hat{p})$$

Teniendo en cuenta que la desviación estándar es el cuadrado de la varianza, se puede despejar:

$$\sigma_{\hat{p}} \approx \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

El problema nos dice que la desviación estándar de $\hat{p}$ ($\sigma_{\hat{p}}$) debe ser sobre 0.02 y nos piden hallar cuántas muestras debe tomar la administradora, por lo tanto, en la parte izquierda de la ecuación reemplazamos por 0,02, se despeja para n y la proporción de defectuosos se toma como 0,01 debido a que en el problema se nos dice que no ha cambiado tanto.

$$\begin{aligned}
\sigma_{\hat{p}} &\approx \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \\
0{,}02 &\approx \sqrt{\frac{0{,}1(1-0{,}1)}{n}} \\
0{,}02 &\approx \sqrt{\frac{0{,}0099}{n}} \\
(0{,}02)^2 &\approx \left(\cancel{\sqrt{\frac{0{,}0099}{n}}}\right)^{\cancel{2}} \\
0{,}0004 &\approx \frac{0{,}0099}{n} \\
n &\approx \frac{0{,}0099}{0{,}0004} = 24{,}75
\end{aligned}$$

Por lo tanto, la administradora debe tomar un muestra de por lo menos 25 chips para tener una desviación estándar de al menos 0,02.

3.2. Now suppose something went wrong with the production run and the actual proportion of defectives in the shipment is 0.3, that is 30 % are defective. Now what would be the actual standard deviation of $\hat{p}$ for the sample size you choose in a)?

Teniendo 25 chips y cambiando el valor de la proporción de defectuosos, se obtiene:

$$\begin{aligned}
\sigma_{\hat{p}} &\approx \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \\
&\approx \sqrt{\frac{0{,}3(1-0{,}3)}{25}} \\
&\approx 0{,}091
\end{aligned}$$

Es decir, si la proporción de detección es del 30 %, la desviación estándar de $\hat{p}$ con una cantidad de chips de 25, será aproximadamente 0,091.

4. Bets

You pay 10,000 pesos to participate in a bet game, which consists in tossing two coins together. If two heads fall, you earn 15,000 pesos. If one head and one tail fall you earn 5,000 pesos. In any other case you earn nothing. Let X the random variable of your profit.

4.1. Analytically find the probability mass function p_X, the mean $E[X]$, and variance $\mathrm{Var}(X)$ of X.

Dado que X es la variable aleatoria de la ganancia, podemos listar los posibles valores que puede tomar de acuerdo a los lanzamientos:

Salida moneda	**Ganancia**	**Ganancia X**	**Probabilidad**
HH	15,000	+5,000	0.25
HT-TH	5,000	−5,000	0.5
TT	0	−10,000	0.25

Cuadro 1: Posibles salidas y ganancias al tirar dos monedas, (H) cara y (T) sello.

Del cuadro 1 se toma que los posibles valores para las ganancias son:

$$X \in \{-5000, -10000, 5000\}$$

La función de probabilida de masa:

$$p_X(x) = \begin{cases} 0{,}25, & x = 5000 \\ 0{,}5, & x = -5000 \\ 0{,}25, & x = -10000 \\ 0, & \text{En otro caso} \end{cases}$$

Para sacar el valor esperado se multiplica la probabilidad por cada valor de X:

$$E[X] = \sum_{I=1}^{3} x\dot{p}_X(x) = 5000(0{,}25) + (-5000)(0{,}5) + (-10000)(0{,}25)$$

$$E[X] = 1250 - 2500 - 2500 = -3750$$

Es decir, en promedio se pierde 3,750 pesos por jugada.
Para calcular la varianza se puede utilizar la fórmula:

$$\mathrm{Var}(X) = E[X^2] - (E[X])^2$$

Se computa $E[X^2]$:

$$\begin{aligned} E[X^2] &= 5000^2(0{,}25) + (-5000)^2(0{,}5) + (-10000)^2(0{,}25) \\ &= 25000000(0{,}25) + 25000000(0{,}5) + 100000000(0{,}25) \\ &= 6250000 + 12500000 + 25000000 = 43750000 \end{aligned}$$

Ahora se le resta el cuadrado de la media:

$$\text{Var}(X) = 43750000 - (-3750)^2 = 43750000 - 14062500 = 29687500$$

4.2. Write a code that simulates the r.v. X using the command sample. Generate an *iid* `sample` $\{X_i\}$ of size $n = 10^5$.

```
outcomes <- c(-10000, -5000, 5000)
probabilities <- c(0.25, 0.5, 0.25)

n <- 10^5

X <- sample(outcomes, size = n , replace = TRUE, prob = probabilities)

length(X)

## [1] 100000

summary(X)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -10000   -5000   -5000   -3734    5000    5000
```

En este fragmento de código se realiza la creación del vector `X` de tamaño 10^5 creado con la función `sample` que toma las salidas que se explicaron anteriormente, el tamaño, que se reemplace y la probabilidad.

4.3. Modify your code to calculate: (i) the estimated mean profit $\bar{X}_j$ for each sample subsequence: $\{X_1, X_2, ..., X_j\}$, $j = 2, 3, \ldots, n$, (ii) the 95 % CI's of each estimated mean profit $\bar{X}_j$.

```
mean_estimates <- numeric(n)
lower_CI <- numeric(n)
upper_CI <- numeric(n)
ie <- 1.96
```

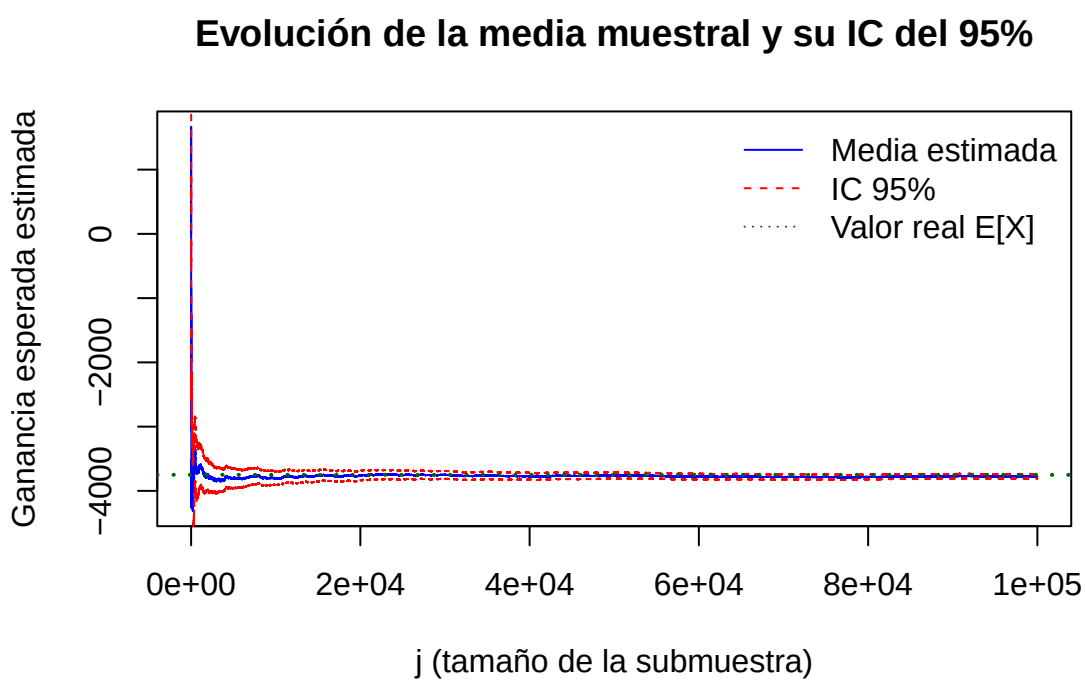
```r
for (j in 2:n) {
  x_sub <- X[1:j]
  mean_j <- mean(x_sub)
  sd_j <- sd(x_sub)
  se_j <- sd_j / sqrt(j)

  mean_estimates[j] <- mean_j
  lower_CI[j] <- mean_j - ie * se_j
  upper_CI[j] <- mean_j + ie * se_j
}
```

En esta parte del código se crean tres vectores `mean_estimates`, `lower_CI`, `upper_CI` de tamaño 10^5 y también una variable `ie`. El bucle en cada iteración calcula la media muestral de los valores $\bar{X}_j$, la desviación estándar `sd_j` y el error estándar, estos son necesarios para luego calcular el intervalo de confianza junto con la media.

4.4. Plot $\bar{X}_j$ and their 95 % CI's in terms of $j = 2, \ldots, 10^5$. Add an horizontal line corresponding to the actual value $E[X]$.

```r
# Your plotting code remains the same
plot(2:n, mean_estimates[2:n], type = "l", col = "blue", lwd = 1,
     xlab = "j (tamaño de la submuestra)",
     ylab = "Ganancia esperada estimada",
     main = "Evolución de la media muestral y su IC del 95%")
lines(2:n, lower_CI[2:n], col = "red", lty = 2)
lines(2:n, upper_CI[2:n], col = "red", lty = 2)
abline(h = -3750, col = "darkgreen", lty = 3, lwd = 2)
legend("topright", legend = c("Media estimada", "IC 95%", "Valor real E[X]"),
       col = c("blue", "red", "darkgreen"), lty = c(1, 2, 3), bty = "n")
```



Se puede ver que a medida que se aumenta la submuestra j, la media estimada $\bar{X}$ se acerca más al valor esperado $E[X]$ y también, el intervalo de confianza se va acortando más.

4.5. Repeat *c)* and *d)* to estimate the probabilities $p_X(x)$, their 95 % CI's, and their plots for each $j = 2, \ldots, 10^5$, adding the actual values.

```
# Posibles valores de X
outcomes <- c(-10000, -5000, 5000)
probs_true <- c(0.25, 0.5, 0.25)

# Inicializar matrices para almacenar las estimaciones y CIs
p_estimates <- matrix(0, nrow = n, ncol = 3)
p_lower <- matrix(0, nrow = n, ncol = 3)
p_upper <- matrix(0, nrow = n, ncol = 3)

z <- 1.96  # Nivel de confianza del 95%

for (j in 2:n) {
  x_sub <- X[1:j]
  for (i in 1:3) {
    val <- outcomes[i]
    p_hat <- mean(x_sub == val)
```

```
    se <- sqrt(p_hat * (1 - p_hat) / j)

    p_estimates[j, i] <- p_hat
    p_lower[j, i] <- max(0, p_hat - z * se)
    p_upper[j, i] <- min(1, p_hat + z * se)
  }
}
```

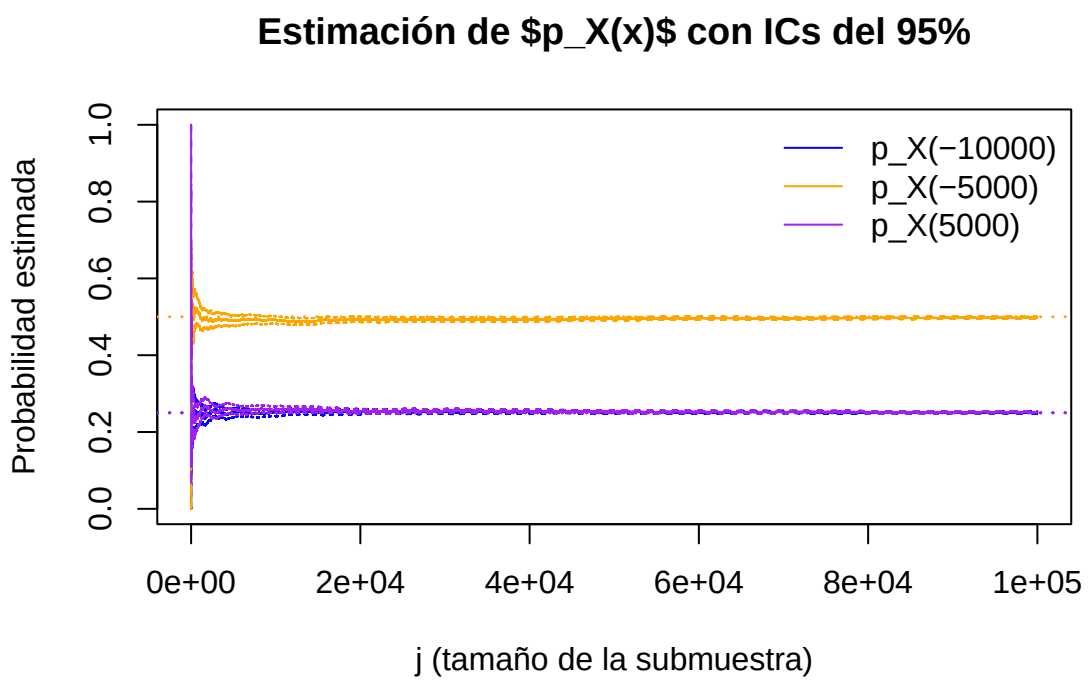
Gráfico de las probabilidades estimadas con sus ICs para cada posible valor de X:

```
colors <- c("blue", "orange", "purple")
labels <- c("p_X(-10000)", "p_X(-5000)", "p_X(5000)")
true_values <- c(0.25, 0.5, 0.25)

plot(2:n, p_estimates[2:n,1], type = "l", col = colors[1], ylim = c(0,1),
     xlab = "j (tamaño de la submuestra)", ylab = "Probabilidad estimada",
     main = "Estimación de $p_X(x)$ con ICs del 95%")

for (i in 1:3) {
  lines(2:n, p_estimates[2:n,i], col = colors[i], lwd = 1)
  lines(2:n, p_lower[2:n,i], col = colors[i], lty = 2)
  lines(2:n, p_upper[2:n,i], col = colors[i], lty = 2)
  abline(h = true_values[i], col = colors[i], lty = 3, lwd = 1.5)
}

legend("topright", legend = labels, col = colors, lty = 1:1, bty = "n")
```



Se observa cómo las probabilidades estimadas convergen a los valores reales a medida que aumenta el tamaño de muestra j, y cómo los intervalos de confianza se vuelven más estrechos.

5. Bootstrap

This exercise is based on the article Introduction to Bootstrapping in Statistics with an Example and the dataset body fat.csv that contains the body fat percentages of 92 adolescent girls. Generate a program that gives:

5.1. An histogram of the sample data.

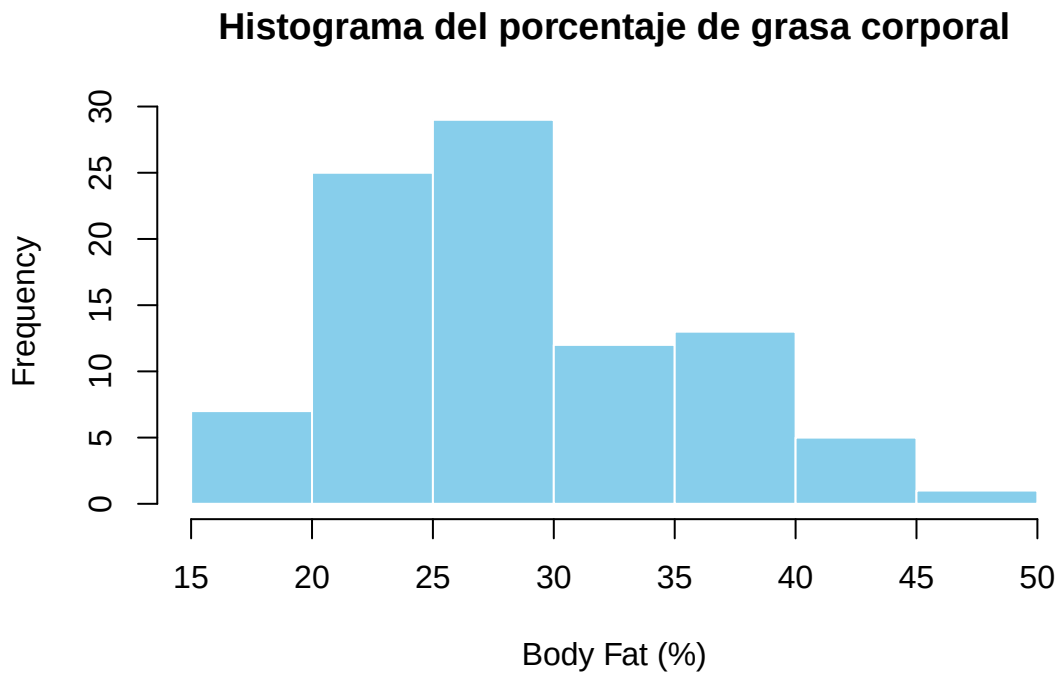
```
# Leer el archivo
fat_data <- read.csv("body_fat.csv", stringsAsFactors = FALSE)

# Verifica el nombre exacto de la columna
colnames(fat_data) <- trimws(colnames(fat_data))  # Eliminar espacios en el nombre
columna_objetivo <- colnames(fat_data)[1]         # Obtener el nombre real

# Convertir a numérico por si acaso
fat_data[[columna_objetivo]] <- as.numeric(trimws(fat_data[[columna_objetivo]]))

# Histograma
```

```r
hist(fat_data[[columna_objetivo]],
     main = "Histograma del porcentaje de grasa corporal",
     xlab = "Body Fat (%)",
     col = "skyblue",
     border = "white")
```



5.2. The 95 % confidence interval of the mean of the data from the traditional method (i.e., via the Central Limit Theorem).

```r
# Estadísticas necesarias
media <- mean(fat_data[[columna_objetivo]], na.rm = TRUE)
sd <- sd(fat_data[[columna_objetivo]], na.rm = TRUE)
n <- sum(!is.na(fat_data[[columna_objetivo]]))  # número de datos no NA

# Cálculo del error estándar
se <- sd / sqrt(n)

# Intervalo de confianza del 95%
alpha <- 0.05
z <- qnorm(1 - alpha/2)  # valor crítico Z para 95%
li <- media - z * se
ls <- media + z * se
```

```r
# Mostrar resultado
cat(sprintf("The 95%% confidence interval for the mean is: [%.2f, %.2f]", li, ls))

## The 95% confidence interval for the mean is: [27.14, 29.99]
```

Como se puede apreciar en el programa, hay un 95 % de confianza para que la media esté en el intervalo [27,1370357, 29,9933991].

5.3. A number of 500 bootstrapped samples from the original dataset, with 92 observations each.

```r
# Vector original limpio
original_data <- na.omit(fat_data[[columna_objetivo]])
n <- length(original_data)

# Generar 500 muestras bootstrap
bootstrap_samples <- replicate(500, sample(original_data, size = n,
                     replace = TRUE), simplify = FALSE)
```

Este código utiliza la función `replicate()` para repetir la operación 500 veces, la función `sample()` toma una muestra con reemplazo del tamaño de 92.

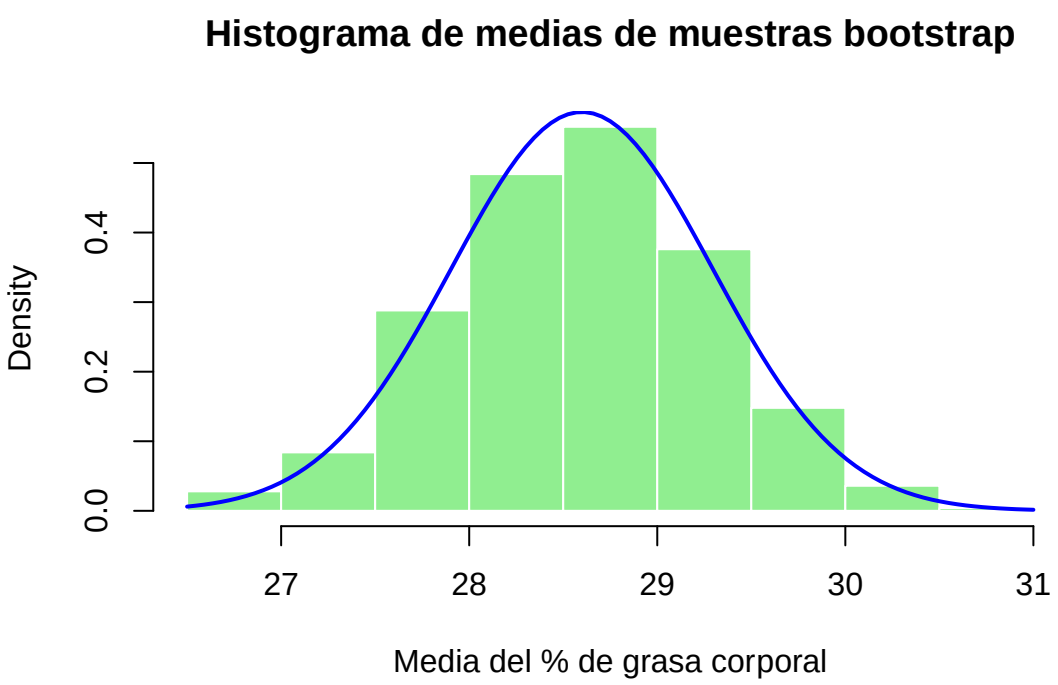
5.4. An histogram of the means of each bootstrapped sample.

```r
# Calcular la media de cada muestra bootstrap
bootstrap_means <- sapply(bootstrap_samples, mean)

# Histograma de las medias
hist(bootstrap_means,
     main = "Histograma de medias de muestras bootstrap",
     xlab = "Media del % de grasa corporal",
     col = "lightgreen",
     border = "white",
     probability = TRUE)  # Importante: escala el eje y para densidades

# Parámetros de la normal
mean_boot <- mean(bootstrap_means)
sd_boot <- sd(bootstrap_means)

# Añadir curva de distribución normal
curve(dnorm(x, mean = mean_boot, sd = sd_boot),
      col = "blue", lwd = 2, add = TRUE)
```



Se puede ver como mas medias de muestras del bootstrap se asemejan a una distribución normal.

5.5. A 95 % bootstrapped confidence interval of the mean.

```
# Calcular intervalo de confianza bootstrap del 95%
bootstrap_ci <- quantile(bootstrap_means, probs = c(0.025, 0.975))

bootstrap_ci
```

```
##     2.5%    97.5%
## 27.20592 29.90144
```

Entoncs, el 95 % del intervalo de confianza para la media es:

$$[27{,}21,\ 29{,}9]$$

5.6. A comparison of both confidence intervals

Con el método tradicional del teórema central del límite:

$$[27{,}14,\ 29{,}99]$$

Con el método *bootstrap*

[27,21, 29,9]

6. Reliability of a system

Suppose a 3-out-of-4 system where each component is functioning with probabilities $p = \{pi\}$:

6.1. Write the structure function of the system $\phi(x)$.

La estructura funciona si al menos 3 de los 4 componentes están estructurados. Dado que hay dos estados; funciona ($\phi = 1$) o lo contrario ($\phi = 0$). Se puede tener un estado de los componentes $x = (x_1, x_2, x_3, x_4)$, donde cada $x_i \in \{0, 1\}$ representa si el componente i está funcionando.

La estructura entonces es:

$$\phi(x_1, x_2, x_3, x_4) = \begin{cases} 1 & \text{si } x_1 + x_2 + x_3 + x_4 \geq 3 \\ 0 & \text{en otro caso} \end{cases}$$

6.2. Deduce analytically the reliability function $R(\mathbf{p})$. Evaluate it when $\{p_i\} = \{0{,}9, 0{,}5, 0{,}2, 0{,}1\}$

$$R(\mathbf{p}) = \sum_{\substack{S \subseteq \{1,2,3,4\} \\ |S|=3}} \left(\prod_{i \in S} p_i \prod_{j \notin S} (1 - p_j) \right) + \prod_{i=1}^{4} p_i$$

For $\mathbf{p} = \{0{,}9, 0{,}5, 0{,}2, 0{,}1\}$, we obtain:

$$\begin{aligned} R(\mathbf{p}) &= 0{,}9 \cdot 0{,}5 \cdot 0{,}2 \cdot 0{,}9 + 0{,}9 \cdot 0{,}5 \cdot 0{,}1 \cdot 0{,}8 + 0{,}9 \cdot 0{,}2 \cdot 0{,}1 \cdot 0{,}5 \\ &\quad + 0{,}5 \cdot 0{,}2 \cdot 0{,}1 \cdot 0{,}1 + 0{,}9 \cdot 0{,}5 \cdot 0{,}2 \cdot 0{,}1 \\ &= 0{,}081 + 0{,}036 + 0{,}009 + 0{,}001 + 0{,}009 = \boxed{0{,}136} \end{aligned}$$