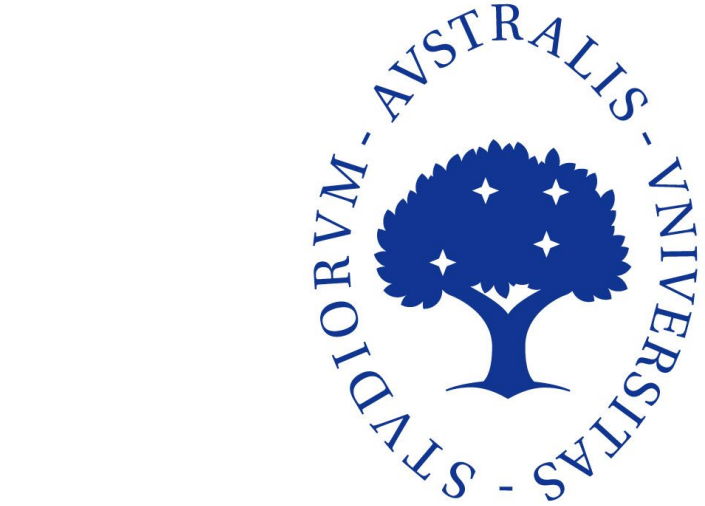



UNIVERSIDAD
AUSTRAL

**Maestría en Ciencia de Datos**

**Materia: Estadística y Probabilidad**

**Grupo 3**

**Alumnos: Maslaton Mariano**

**Maslaton Carlos**

# Ejercicio N°1 - Estadística Descriptiva

Para este análisis, se seleccionó el "Adult Income Dataset", disponible en el UCI Machine Learning Repository. Este conjunto de datos también puede obtenerse ejecutando el siguiente código:

In [1]:
```
options(warn = -1)
```

In [2]:
```
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"
download.file(url, destfile = "adult.data")
```

In [3]:
```
column_names <- c("age", "workclass", "fnlwgt", "education", "education_num",
                  "marital_status", "occupation", "relationship", "race",
                  "sex", "capital_gain", "capital_loss", "hours_per_week",
                  "native_country", "income")
data <- read.csv("adult.data", header = FALSE, sep = ",", col.names = column_names,
                 na.strings = " ?")
```

**a) Describir para este caso la población, la muestra tomada, el experimento que estará usando, las variables bajo análisis y cualquier otra característica relevante para el procedimiento. Si la base es simulada, exponga los motivos por los cuales el procedimiento es viable.**

**Descripción cualitativa:** Se trata de un dataset que contiene información demográfica y socioeconómica de individuos adultos extraída del censo de EE.UU. de 1994.

El dataset posee la siguiente cantidad de filas y variables:

In [4]:
```
tamano_dataset <- nrow(data)
cantidad_variables <- ncol(data)

cat("Cantidad de filas:", tamano_dataset, "\n")
cat("Cantidad de variables:", cantidad_variables)
```

```
Cantidad de filas: 32561
Cantidad de variables: 15
```

**Descripción estadística del dataset:**

In [5]:
```
summary(data)
```

```
     age         workclass             fnlwgt         education
Min.   :17.00   Length:32561       Min.   :  12285   Length:32561
1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character
Median :37.00   Mode  :character   Median : 178356   Mode  :character
Mean   :38.58                      Mean   : 189778
3rd Qu.:48.00                      3rd Qu.: 237051
Max.   :90.00                      Max.   :1484705
education_num   marital_status      occupation        relationship
Min.   : 1.00   Length:32561       Length:32561       Length:32561
1st Qu.: 9.00   Class :character   Class :character   Class :character
Median :10.00   Mode  :character   Mode  :character   Mode  :character
Mean   :10.08
3rd Qu.:12.00
Max.   :16.00
    race               sex             capital_gain    capital_loss
Length:32561       Length:32561       Min.   :    0   Min.   :   0.0
Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
Mode  :character   Mode  :character   Median :    0   Median :   0.0
                                      Mean   : 1078   Mean   :  87.3
                                      3rd Qu.:    0   3rd Qu.:   0.0
                                      Max.   :99999   Max.   :4356.0
hours_per_week  native_country        income
Min.   : 1.00   Length:32561       Length:32561
1st Qu.:40.00   Class :character   Class :character
Median :40.00   Mode  :character   Mode  :character
Mean   :40.44
3rd Qu.:45.00
Max.   :99.00
```

**Extracción de la muestra:** se seleccionó una muestra aleatoria de $n$ = 2000 registros del "Adult Income Dataset".

Esta muestra busca ser una representación adecuada para realizar análisis estadísticos descriptivos de las características demográficas y socioeconómicas de los individuos incluidos.

In [6]:
```r
selected_columns <- c("age", "education", "occupation", "hours_per_week", "income")
adult_data_selected <- data[selected_columns]
```

In [7]:
```r
set.seed(47)
n <- 2000
sample_indices <- sample(1:nrow(adult_data_selected), size = n, replace = FALSE)
sample_data <- adult_data_selected[sample_indices, ]
```

**Variables bajo análisis:**

- **age**: Edad del individuo.
  - **Tipo**: Numérica.
  - **Descripción**: Representa la edad de cada individuo en la muestra. Esta variable permite analizar la distribución etaria y sus posibles correlaciones con otras variables.

- **education**: Nivel educativo.
  - **Tipo**: Categórica.
  - **Descripción**: Indica el nivel educativo alcanzado por los individuos, con categorías como "Bachelors", "HS-grad", "Masters", etc. Esta variable es esencial para estudiar la relación entre educación e ingresos.
- **occupation**: Ocupación.
  - **Tipo**: Categórica.
  - **Descripción**: Describe el tipo de trabajo que desempeña cada individuo, con categorías como "Tech-support", "Craft-repair", "Other-service", etc. Analizar esta variable puede revelar tendencias ocupacionales y su relación con el ingreso.
- **hours_per_week**: Horas trabajadas por semana.
  - **Tipo**: Numérica.
  - **Descripción**: Indica el número de horas que cada individuo trabaja por semana. Esta variable es crucial para estudiar los patrones laborales y su impacto en los ingresos.
- **income**: Ingreso anual.
  - **Tipo**: Categórica.
  - **Descripción**: Categorizada en ">50K" (más de USD 50,000) y "<=50K" (menos de USD 50,000). Esta variable es el objetivo principal del análisis, permitiendo estudiar cómo diferentes factores demográficos y laborales influyen en los niveles de ingreso.

**b) Generar un set de estadística descriptiva sobre la misma que le permita resumir la información obtenida, explicando para cada una de ellas su significado.**

In [8]:
```
age_mean <- mean(sample_data$age, na.rm = TRUE)
age_median <- median(sample_data$age, na.rm = TRUE)
age_sd <- sd(sample_data$age, na.rm = TRUE)
age_min <- min(sample_data$age, na.rm = TRUE)
age_max <- max(sample_data$age, na.rm = TRUE)
age_iqr <- IQR(sample_data$age, na.rm = TRUE)

cat("Estadísticas Descriptivas para la Edad (age):\n")
cat("Media:", age_mean, "\n")
cat("Mediana:", age_median, "\n")
cat("Desviación Estándar:", age_sd, "\n")
cat("Mínimo:", age_min, "\n")
cat("Máximo:", age_max, "\n")
cat("Rango Intercuartílico (IQR):", age_iqr, "\n\n")
```

```
Estadísticas Descriptivas para la Edad (age):
Media: 38.2565
Mediana: 37
Desviación Estándar: 13.25313
Mínimo: 17
Máximo: 90
Rango Intercuartílico (IQR): 19
```

In [9]:
```r
hours_mean <- mean(sample_data$hours_per_week, na.rm = TRUE)
hours_median <- median(sample_data$hours_per_week, na.rm = TRUE)
hours_sd <- sd(sample_data$hours_per_week, na.rm = TRUE)
hours_min <- min(sample_data$hours_per_week, na.rm = TRUE)
hours_max <- max(sample_data$hours_per_week, na.rm = TRUE)
hours_iqr <- IQR(sample_data$hours_per_week, na.rm = TRUE)

cat("Estadísticas Descriptivas para las Horas Trabajadas por Semana (hours_per_week
cat("Media:", hours_mean, "\n")
cat("Mediana:", hours_median, "\n")
cat("Desviación Estándar:", hours_sd, "\n")
cat("Mínimo:", hours_min, "\n")
cat("Máximo:", hours_max, "\n")
cat("Rango Intercuartílico (IQR):", hours_iqr, "\n\n")
```

```
Estadísticas Descriptivas para las Horas Trabajadas por Semana (hours_per_week):
Media: 40.4245
Mediana: 40
Desviación Estándar: 11.80101
Mínimo: 1
Máximo: 99
Rango Intercuartílico (IQR): 5
```

In [10]:
```r
# Distribución de frecuencias para la variable 'education'
education_freq <- table(sample_data$education)
education_prop <- prop.table(education_freq)

cat("Distribución de Frecuencias para el Nivel Educativo (education):\n")
print(education_freq)
cat("Distribución de Frecuencias Relativas (Proporciones):\n")
print(education_prop)
cat("\n")
```

```
Distribución de Frecuencias para el Nivel Educativo (education):

        10th          11th          12th       1st-4th       5th-6th
          54            67            29             9            20
     7th-8th           9th    Assoc-acdm     Assoc-voc     Bachelors
          46            26            71            73           338
   Doctorate       HS-grad       Masters     Preschool   Prof-school
          18           646           129             1            37
Some-college
         436
Distribución de Frecuencias Relativas (Proporciones):

        10th          11th          12th       1st-4th       5th-6th
      0.0270        0.0335        0.0145        0.0045        0.0100
     7th-8th           9th    Assoc-acdm     Assoc-voc     Bachelors
      0.0230        0.0130        0.0355        0.0365        0.1690
   Doctorate       HS-grad       Masters     Preschool   Prof-school
      0.0090        0.3230        0.0645        0.0005        0.0185
Some-college
      0.2180
```

In [11]:
```r
# Distribución de frecuencias para la variable 'occupation'
occupation_freq <- table(sample_data$occupation)
occupation_prop <- prop.table(occupation_freq)

cat("Distribución de Frecuencias para la Ocupación (occupation):\n")
print(occupation_freq)
cat("Distribución de Frecuencias Relativas (Proporciones):\n")
print(occupation_prop)
cat("\n")
```

```
Distribución de Frecuencias para la Ocupación (occupation):

     Adm-clerical       Armed-Forces       Craft-repair    Exec-managerial
              233                  1                260                274
  Farming-fishing  Handlers-cleaners  Machine-op-inspct      Other-service
               61                 74                119                206
  Priv-house-serv     Prof-specialty    Protective-serv              Sales
               10                265                 41                209
     Tech-support   Transport-moving
               63                 87
Distribución de Frecuencias Relativas (Proporciones):

     Adm-clerical       Armed-Forces       Craft-repair    Exec-managerial
     0.1224382554       0.0005254861       0.1366263794       0.1439831844
  Farming-fishing  Handlers-cleaners  Machine-op-inspct      Other-service
     0.0320546506       0.0388859695       0.0625328429       0.1082501314
  Priv-house-serv     Prof-specialty    Protective-serv              Sales
     0.0052548607       0.1392538098       0.0215449291       0.1098265896
     Tech-support   Transport-moving
     0.0331056227       0.0457172885
```

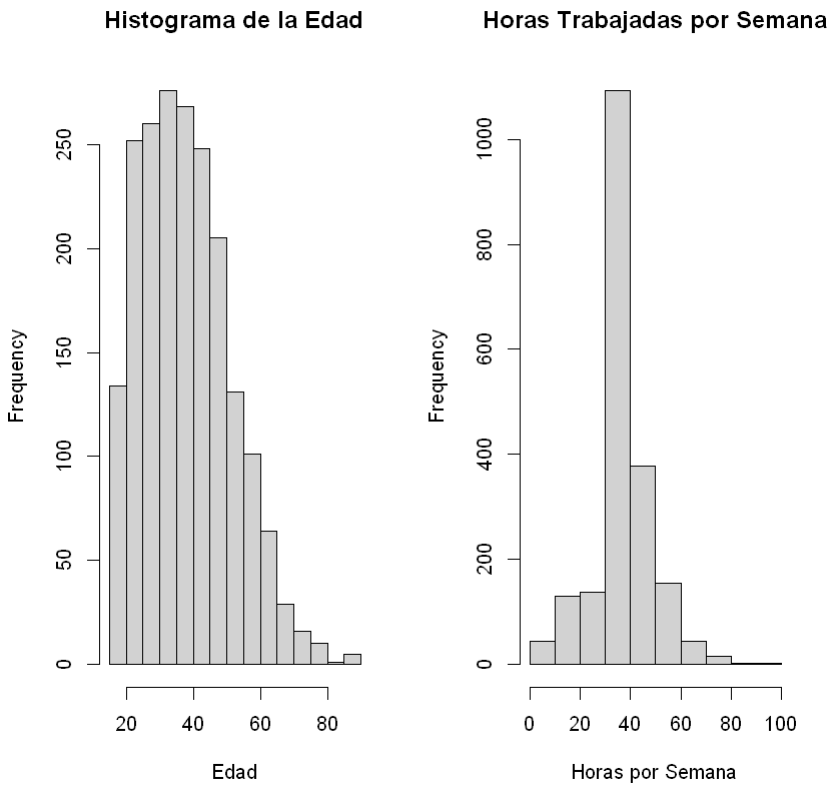
In [12]:
```r
# Distribución de frecuencias para la variable 'income'
income_freq <- table(sample_data$income)
```

```r
income_prop <- prop.table(income_freq)

cat("Distribución de Frecuencias para el Ingreso Anual (income):\n")
print(income_freq)
cat("Distribución de Frecuencias Relativas (Proporciones):\n")
print(income_prop)
cat("\n")
```

```
Distribución de Frecuencias para el Ingreso Anual (income):

 <=50K   >50K
  1503    497
Distribución de Frecuencias Relativas (Proporciones):

 <=50K   >50K
0.7515 0.2485
```

**c) Generar un histograma para una de las variables cuantitativas, utilizando la forma que considere más correcta para agrupar las categorías. Elija la variable que mayor simetría consiga en el histograma resultante.**

In [13]:
```r
par(mfrow=c(1,2))

hist(sample_data$age, main = "Histograma de la Edad", xlab = "Edad", breaks = 25)
hist(sample_data$hours_per_week, main = "Horas Trabajadas por Semana",
     xlab = "Horas por Semana", breaks = 8)

par(mfrow=c(1,1))
```



# Ejercicio N°2 - Probabilidad

## Ejercicio 1

**1.1. Si lanzamos una moneda, ¿Cuál es la probabilidad esperada de obtener una cara?. Si lanzamos una moneda 10 veces, ¿Cuál es la cantidad esperada de caras?**

Sea $X$ es una variable aleatoria que sigue una distribución de Bernoulli, donde la probabilidad de obtener "cara" es 0.5. Al evento "cara" se le asigna el valor 1, y al evento "cruz" se le asigna el valor 0. La esperanza $E[X]$ se obtiene al multiplicar cada resultado posible $X$ por su probabilidad correspondiente $P(X)$ y luego, sumar estos productos [1]:

$$E(X_i) = 1 \cdot p(\text{cara}) + 0 \cdot p(\text{cruz}) = 0.5$$

La cantidad esperada de caras al lanzar una moneda 10 veces es:

$$\text{número de caras} = 10 \cdot E(X_{\text{cara}}) = 10 \cdot 0.5 = 5$$

**1.2. Lanzar una moneda 10 veces y contar el número de caras. Repetirlo 8 veces y almacenar el número de caras para cada una.**

In [14]:
```
simular_lanzamientos <- function(n) {
  sum(sample(c("cara", "cruz"), size = n, replace = TRUE) == "cara")
}

resultados_8 <- replicate(8, simular_lanzamientos(10))
media_8 <- mean(resultados_8)
desviacion_std_8 <- sd(resultados_8)

print(paste("Número de caras:",sum(resultados_8)))
```

```
[1] "Número de caras: 36"
```

**1.3. Lanzar una moneda 10 veces, contar el número de caras, almacenar el resultado y repetirlo 1000 veces.**

In [15]:
```
resultados_1k <- replicate(1000, simular_lanzamientos(10))
media_1k <- mean(resultados_1k)
desviacion_std_1k <- sd(resultados_1k)

print(paste("Número de caras:",sum(resultados_1k)))
```

```
[1] "Número de caras: 5006"
```

**1.4. ¿Cómo difieren los resultados del experimento en (2) de los resultados en el experimento (3)? Justificar**

A medida que aumenta el número de repeticiones del experimento, la media de los resultados de estas repeticiones tiende a aproximarse más consistentemente al valor

esperado de la distribución. Esto implica que el promedio de los resultados obtenidos en múltiples repeticiones se acercará a la media de 5 caras.

In [16]:
```
print(paste("Media para 8 repeticiones:", media_8,
            "y Desviación Estándar:", desviacion_std_8))
print(paste("Media para 1000 repeticiones:", media_1k,
            "y Desviación Estándar:", desviacion_std_1k))
```

```
[1] "Media para 8 repeticiones: 4.5 y Desviación Estándar: 1.19522860933439"
[1] "Media para 1000 repeticiones: 5.006 y Desviación Estándar: 1.61696585572441"
```

## Ejercicio 2

**Una persona te propone jugar un juego con dados, el cual te solicita tirar 2 dados.**

**• Si sale un 7, te pagarán 3 pesos.**

**• Si sale un 11, te pagarán 5 pesos.**

**• Si sale cualquier otra combinación, deberás pagar 0.70 pesos.**

### 2.1. ¿Cuál es la probabilidad de sacar un siete?

Es posible hallar la probabilidad de obtener un siete al realizar el producto cartesiano de los espacios muestrales de los dados para construir un nuevo espacio muestral, dispuesto en forma de matriz, en donde cada elemento de la matriz es igual a la suma de los índices de cada posición de la fila y columna:

In [17]:
```
set1 <- 1:6
set2 <- 1:6

suma_matriz <- outer(set1, set2, FUN = "+")

print(suma_matriz)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    2    3    4    5    6    7
[2,]    3    4    5    6    7    8
[3,]    4    5    6    7    8    9
[4,]    5    6    7    8    9   10
[5,]    6    7    8    9   10   11
[6,]    7    8    9   10   11   12
```

De la disposición anterior, basta con sumar la cantidad veces en que el resultado fue 7 y dividir dicho valor por la cantidad total de elementos de la matriz:

In [18]:
```
cantidad_siete <- sum(suma_matriz == 7)
print(paste("P(7)=", cantidad_siete/(6*6)))
```

```
[1] "P(7)= 0.166666666666667"
```

### 2.2. ¿Cuál es la probabilidad de sacar un once?

Para la solución de este inciso, se repite el procedimiento previo y se crea una máscara en la que se suma la cantidad de elementos de la matriz que sean iguales a 11 y se lo divide por el total de elementos de la matriz:

In [19]:
```
cantidad_once <- sum(suma_matriz == 11)
print(paste("P(11)=", cantidad_once/(6*6)))
```

```
[1] "P(11)= 0.0555555555555556"
```

**2.3. ¿Cuál es la probabilidad de sacar un siete o un once?**

Dado que son eventos excluyentes, el valor de la probabilidad se la obtiene con la suma [2]:

In [20]:
```
print(paste("P(7U11)=", cantidad_siete/(6*6)+cantidad_once/(6*6)))
```

```
[1] "P(7U11)= 0.222222222222222"
```

**2.4. Simular tirar 2 dados mediante la función Roll1Dice(). Simular tirar 2 dados 100 veces y almacenar los resultados. Calcular los puntos anteriores (1, 2 y 3) a partir de los datos.**

Definición de la función solicitada en el ejercicio:

In [21]:
```
Roll1Dice <- function() {
  sample(1:6, 1)
}
```

In [22]:
```
set.seed(87)
veces <- 100
resultados <- replicate(veces, Roll1Dice() + Roll1Dice())
```

**Cálculo de P(7), P(11) y P(7U11):**

In [23]:
```
print(paste("P(7)=", sum(resultados == 7)/veces))
print(paste("P(11)=", sum(resultados == 11)/veces))
print(paste("P(7U11)=", sum(resultados == 7)/veces + sum(resultados == 11)/veces))
```

```
[1] "P(7)= 0.18"
[1] "P(11)= 0.07"
[1] "P(7U11)= 0.25"
```

**2.5. Suponga que jugó 10 veces y obtuvo una ganancia de $ 30. ¡Qué fácil parece ser el juego! ¡Debería seguir jugando! ¿Es correcta la suposición? Demostrar con una simulación.**

Esta afirmación puede ser respondida al calcular la esperanza matemática del juego.

$$E(X) = 3 \cdot P(7) + 5 \cdot P(11) - 0.7 \cdot (1 - P(7) - P(11))$$

Reemplazando los valores de probabilidad de ocurrencia, la ganancia esperada por juego es de:

$$E(X) = 3 \cdot \frac{6}{36} + 5 \cdot \frac{2}{36} - 0.7 \cdot \frac{28}{36} = 0.23333$$

Si se juega, por ejemplo 10000 veces, la ganancia esperada es de:

$$n \cdot E(X) = 10000 \cdot 0.23333 = 2333.3$$

La conclusión a la que se arriba, es que debería seguir jugando, ya que el valor esperado del juego es positivo. También es posible arribar a esta conclusión mediante una simulación:

In [24]:
```
veces <- 10000
resultados <- replicate(veces, Roll1Dice() + Roll1Dice())

print(paste("Ganancia:", sum(resultados == 7) * 3 + sum(resultados == 11) * 5 -
            sum((resultados != 7) & (resultados != 11)) * 0.7))
```

```
[1] "Ganancia: 2373.4"
```

**2.6. Ahora dicha persona te ofrece disminuir el monto a pagar a $ 0.68. ¿Deberías aceptarlo?**

La penalidad por no obtener un 7 o un 11 en la tirada de dados es menor, por lo que la ganancia esperada por jugada aumenta:

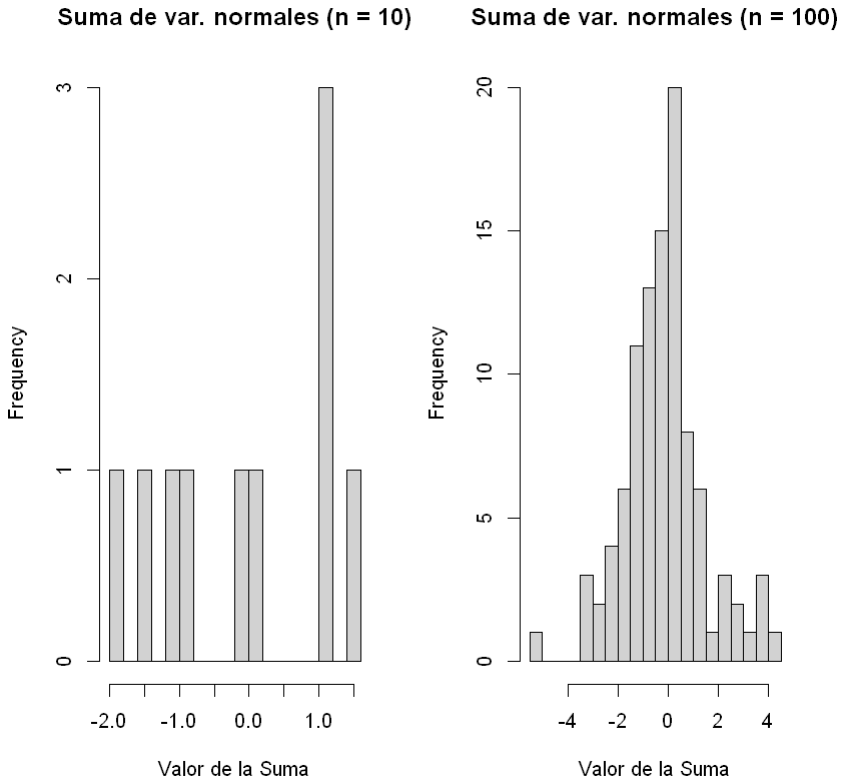
$$E(X) = 3 \cdot \frac{6}{36} + 5 \cdot \frac{2}{36} - 0.68 \cdot \frac{28}{36} = 0.24888$$

# Ejercicio N°3 - Variables Aleatorias

**3.1. Simular la suma de dos variables normales mediante la función rnorm(x,mu,sigma) en R con media igual a 0 y desvío igual a 1. Probar con el tamaño de muestra n = 10 y n = 100. ¿Qué observa si grafica ambos objetos?**

In [25]:
```
n <- c(10, 100)

for (valor in n) {
    nombre_variable <- paste0("suma_n", valor)
    assign(nombre_variable, NULL)

    for (i in 1:valor) {
        suma <- rnorm(1, 0, 1) + rnorm(1, 0, 1)
        assign(nombre_variable, c(get(nombre_variable), suma), envir = .GlobalEnv)
    }
}

par(mfrow = c(1, 2))  # Organizar los gráficos en una fila de dos columnas

hist(suma_n10, breaks = 20, main = "Suma de var. normales (n = 10)",
     xlab = "Valor de la Suma")
hist(suma_n100, breaks = 20, main = "Suma de var. normales (n = 100)",
     xlab = "Valor de la Suma")
```

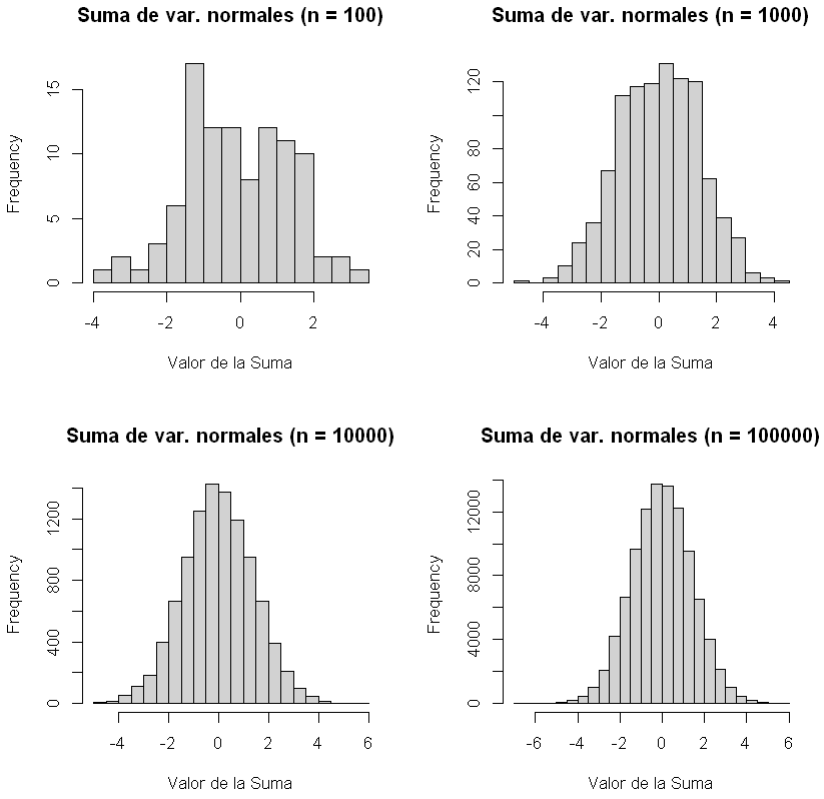
```
par(mfrow = c(1, 1))
```




**3.2. Simular la suma de dos variables normales mediante la función rnorm(x,mu,sigma) en R con media igual a 0 y desvío igual a 1. Probar con distintos valores de tamaño de muestra. Podría probar con n = 100, n = 1000, n = 10000, n = 100000. Gráficar para estos distintos valores de muestra. Comprobar que la media (valor esperado) es igual a la suma de sus valores esperados y la varianza es igual a la suma de varianzas individuales.**

In [26]:
```
options(scipen = 999)
n <- c(100, 1000, 10000, 100000)

for (valor in n) {
    nombre_matriz <- paste0("datos_n", valor)
    datos <- matrix(NA, nrow = valor, ncol = 2)

    for (i in 1:valor) {
        datos[i, 1] <- rnorm(1, 0, 1)
        datos[i, 2] <- rnorm(1, 0, 1)
    }
    assign(nombre_matriz, datos, envir = .GlobalEnv)
}

par(mfrow = c(2, 2))

hist(rowSums(datos_n100),
     breaks = 20, main = "Suma de var. normales (n = 100)", xlab = "Valor de la Sum
hist(rowSums(datos_n1000),
```

```
     breaks = 20, main = "Suma de var. normales (n = 1000)", xlab = "Valor de la Su
hist(rowSums(datos_n10000),
     breaks = 20, main = "Suma de var. normales (n = 10000)", xlab = "Valor de la S
hist(rowSums(datos_n100000),
     breaks = 20, main = "Suma de var. normales (n = 100000)", xlab = "Valor de la

par(mfrow = c(1, 1))
```



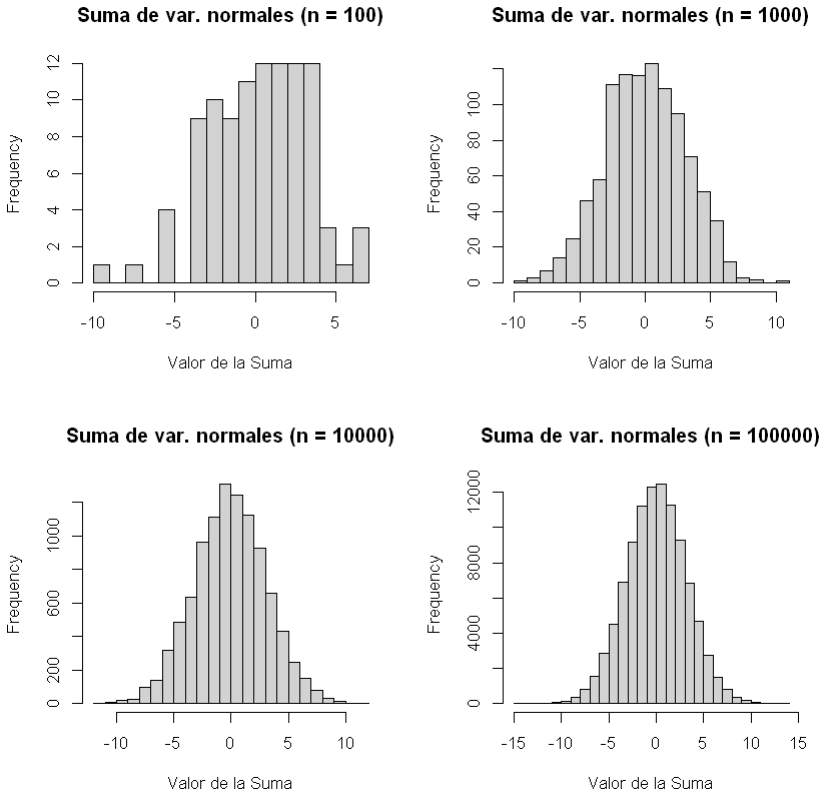

In [27]:
```
nombres_matrices <- c("datos_n100", "datos_n1000", "datos_n10000", "datos_n100000")

for (nombre_matriz in nombres_matrices) {
    print(paste("Para la matriz", nombre_matriz))
    print(paste("Media de la suma:", sum(apply(get(nombre_matriz), 2, mean))))
    print(paste("Suma de las medias:", mean(get(nombre_matriz)[, 1])
                + mean(get(nombre_matriz)[, 2])))
    print(paste("Varianza de la suma:", sum(apply(get(nombre_matriz), 2, var))))
    print(paste("Suma de las varianza:", var(get(nombre_matriz)[, 1])
                + var(get(nombre_matriz)[, 2])))
    cat("\n")
}
```

```
[1] "Para la matriz datos_n100"
[1] "Media de la suma: -0.0632288379607235"
[1] "Suma de las medias: -0.0632288379607235"
[1] "Varianza de la suma: 2.15539593443556"
[1] "Suma de las varianza: 2.15539593443556"

[1] "Para la matriz datos_n1000"
[1] "Media de la suma: 0.0117383382620096"
[1] "Suma de las medias: 0.0117383382620096"
[1] "Varianza de la suma: 1.91476000820771"
[1] "Suma de las varianza: 1.91476000820771"

[1] "Para la matriz datos_n10000"
[1] "Media de la suma: -0.00722844789776151"
[1] "Suma de las medias: -0.00722844789776151"
[1] "Varianza de la suma: 1.9918779220084"
[1] "Suma de las varianza: 1.9918779220084"

[1] "Para la matriz datos_n100000"
[1] "Media de la suma: -0.00590240107202146"
[1] "Suma de las medias: -0.00590240107202146"
[1] "Varianza de la suma: 2.00765217933784"
[1] "Suma de las varianza: 2.00765217933784"
```

**3.3. Simular la suma de diez variables normales mediante la función rnorm en R con media igual a 0 y desvío igual a 1. Probar con distintos valores de tamaño de muestra. Podría probar con n = 100, n = 1000, n = 10000, n = 100000. Gráficar para estos distintos valores de muestra. Comprobar que la media (valor esperado) es igual a la suma de sus valores esperados y la varianza es igual a la suma de varianzas individuales.**

In [28]:
```r
options(scipen = 999)
n <- c(100, 1000, 10000, 100000)

for (valor in n) {
    nombre_matriz <- paste0("datos_n", valor)
    datos <- matrix(NA, nrow = valor, ncol = 10)

    for (i in 1:valor) {
        for (j in 1:10) {
            datos[i, j] <- rnorm(1, 0, 1)
        }
    }
    assign(nombre_matriz, datos, envir = .GlobalEnv)
}

par(mfrow = c(2, 2))

hist(rowSums(datos_n100),
     breaks = 20, main = "Suma de var. normales (n = 100)", xlab = "Valor de la Sum
hist(rowSums(datos_n1000),
     breaks = 20, main = "Suma de var. normales (n = 1000)", xlab = "Valor de la Su
hist(rowSums(datos_n10000),
```

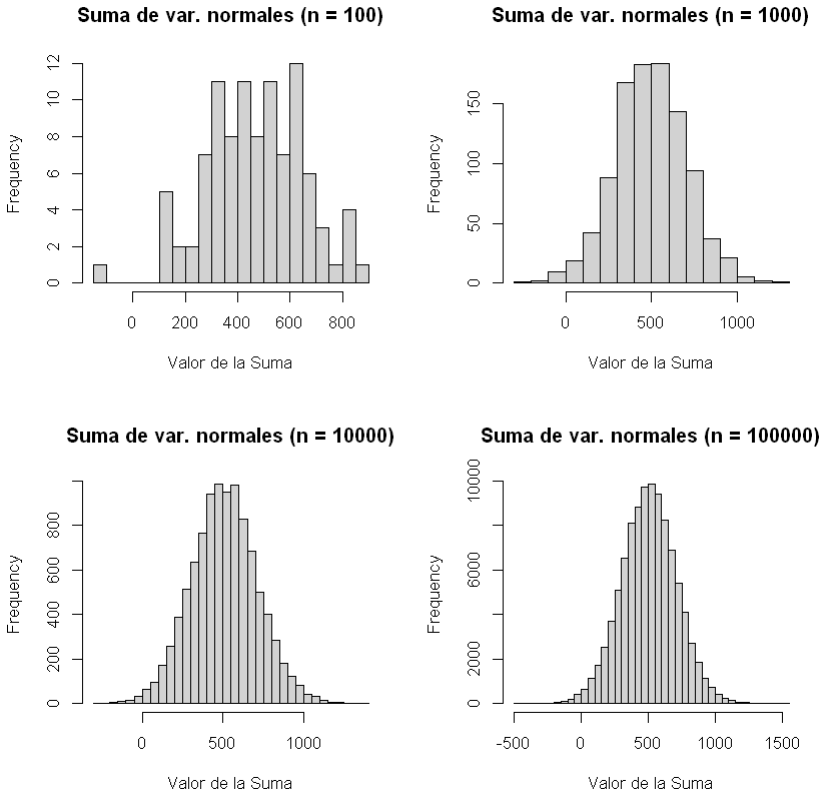
```
     breaks = 20, main = "Suma de var. normales (n = 10000)", xlab = "Valor de la S
hist(rowSums(datos_n100000),
     breaks = 20, main = "Suma de var. normales (n = 100000)", xlab = "Valor de la

par(mfrow = c(1, 1))
```




### 3.4. Extra: Probar lo anterior pero sumando normales con distintas medias y desvíos.

In [29]:
```
options(scipen = 999)
n <- c(100, 1000, 10000, 100000)

for (valor in n) {
    nombre_matriz <- paste0("datos_n", valor)
    datos <- matrix(NA, nrow = valor, ncol = 10)

    for (i in 1:valor) {
        for (j in 1:10) {
            datos[i, j] <- rnorm(1, sample(1:100, 1), sample(1:100, 1))
        }
    }
    assign(nombre_matriz, datos, envir = .GlobalEnv)
}

par(mfrow = c(2, 2))

hist(rowSums(datos_n100),
     breaks = 15, main = "Suma de var. normales (n = 100)", xlab = "Valor de la Sum
hist(rowSums(datos_n1000),
     breaks = 20, main = "Suma de var. normales (n = 1000)", xlab = "Valor de la Su
hist(rowSums(datos_n10000),
```

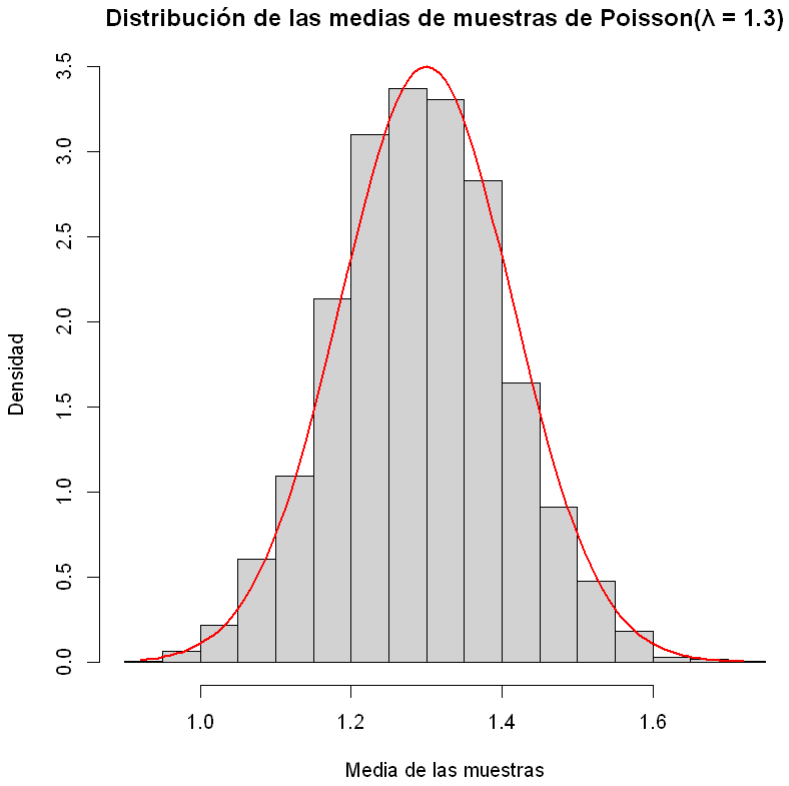
```
     breaks = 25, main = "Suma de var. normales (n = 10000)", xlab = "Valor de la S
hist(rowSums(datos_n100000),
     breaks = 30, main = "Suma de var. normales (n = 100000)", xlab = "Valor de la

par(mfrow = c(1, 1))
```




# Ejercicio N°4 - Teorema Central del Límite

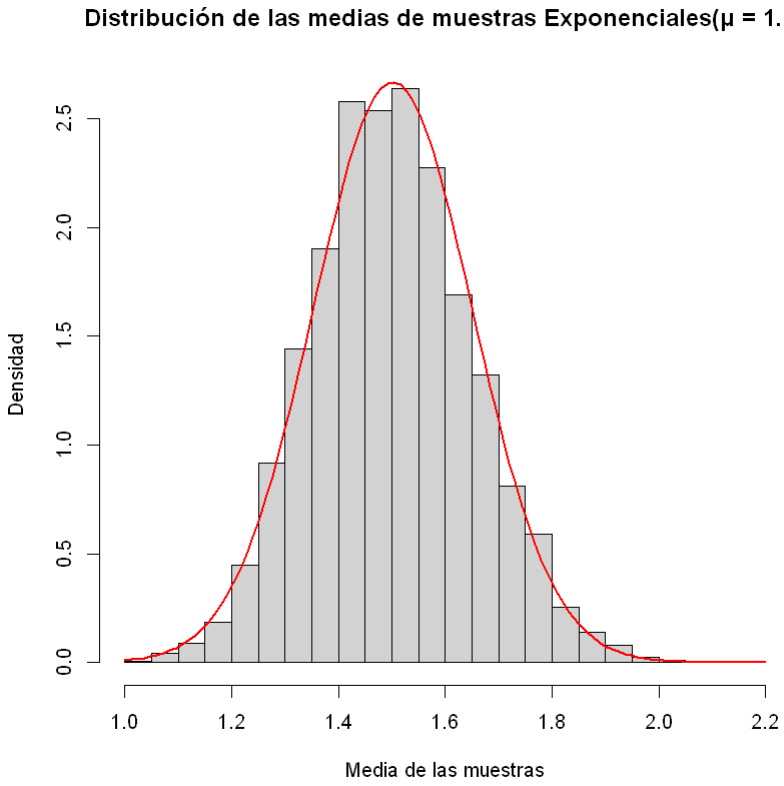
### 4.1. Poisson de parámetro λ = 1.3

In [30]:
```
lambda <- 1.3
n_muestras <- 5000
tamaño_muestra <- 100

medias_muestras <- replicate(n_muestras, mean(rpois(tamaño_muestra, lambda)))

hist(medias_muestras, breaks = 20, probability = TRUE,
     main = "Distribución de las medias de muestras de Poisson(λ = 1.3)",
     xlab = "Media de las muestras", ylab = "Densidad")

x <- seq(min(medias_muestras), max(medias_muestras), length = 100)
y <- dnorm(x, mean = lambda, sd = sqrt(lambda/tamaño_muestra))
lines(x, y, col = "red", lwd = 2)
```




### 4.2. Exponencial de parámetro μ = 1.5

In [31]:
```r
mu <- 1.5
lambda <- 1 / mu
n_muestras <- 5000
tamaño_muestra <- 100

set.seed(58)
medias_muestras <- replicate(n_muestras, mean(rexp(tamaño_muestra, lambda)))

hist(medias_muestras, breaks = 20, probability = TRUE,
     main = "Distribución de las medias de muestras Exponenciales(μ = 1.5)",
     xlab = "Media de las muestras", ylab = "Densidad", border = "black")

curve(dnorm(x, mean = mean(medias_muestras), sd = sd(medias_muestras)),
      add = TRUE, col = "red", lwd = 2)
```

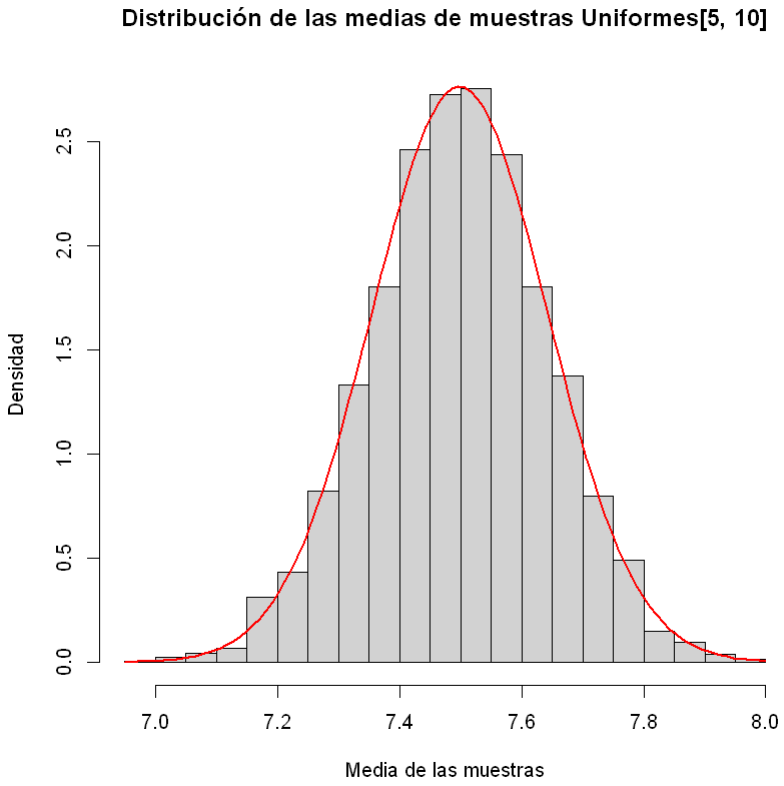



### 4.3. Uniforme en el intervalo [5,10]

In [32]:
```
min_val <- 5
max_val <- 10
n_muestras <- 5000
tamaño_muestra <- 100
```

In [33]:
```
set.seed(32)
medias_muestras <- replicate(n_muestras,
                             mean(runif(tamaño_muestra, min = min_val, max = max_va
```

In [34]:
```
hist(medias_muestras, breaks = 20, probability = TRUE,
     main = "Distribución de las medias de muestras Uniformes[5, 10]",
     xlab = "Media de las muestras", ylab = "Densidad", border = "black")

curve(dnorm(x, mean = mean(medias_muestras), sd = sd(medias_muestras)),
      add = TRUE, col = "red", lwd = 2)
```

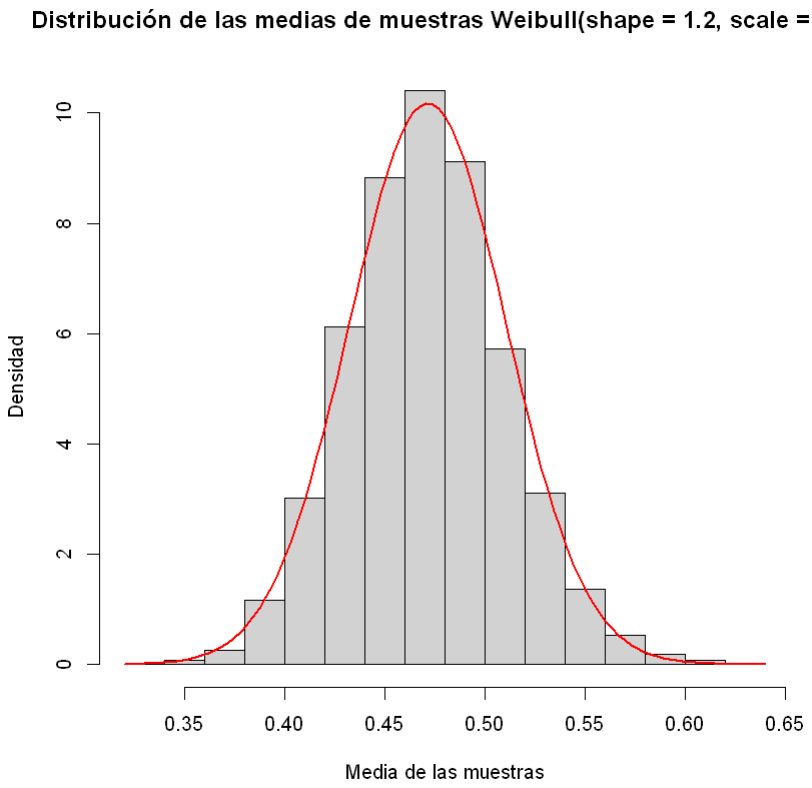



### 4.4. Weibull de parámetros shape = 1.2 y scale = 0.5

In [35]:
```
shape <- 1.2
scale <- 0.5
n_muestras <- 5000
tamaño_muestra <- 100
```

In [36]:
```
set.seed(69)
medias_muestras <- replicate(n_muestras,
                             mean(rweibull(tamaño_muestra, shape, scale)))
```

In [37]:
```
# Graficar el histograma de las medias de las muestras
hist(medias_muestras, breaks = 20, probability = TRUE,
     main = "Distribución de las medias de muestras Weibull(shape = 1.2, scale = 0.
     xlab = "Media de las muestras", ylab = "Densidad", border = "black")

# Superponer la densidad normal teórica
curve(dnorm(x, mean = mean(medias_muestras), sd = sd(medias_muestras)),
      add = TRUE, col = "red", lwd = 2)
```




# Ejercicio N°5

**Proceda a cargar el siguiente dataset de un repositorio en github,**

In [38]:
```
# install.packages("repmis")
```

In [39]:
```
url = "https://github.com/hllinas/DatosPublicos/blob/main/Estudiantes.Rdata?raw=fal
repmis::source_data(url)
datos <- Estudiantes
```

```
Downloading data from: https://github.com/hllinas/DatosPublicos/blob/main/Estudiante
s.Rdata?raw=false


SHA-1 hash of the downloaded data file is:
6bf9d5a19779293538bd61d55d0662bdaf8100a1
```

'Estudiantes'

**Realizar los siguientes ejercicios. Interprete todas sus respuestas.**

**a) Considerar solamente las observaciones que van desde la 2 hasta la 35 y definir el data frame "datos2a35". Verificar su tamaño, variables y estructura.**

In [40]:
```
datos2a35 <- datos[2:35, ]
```

Tamaño de datos2a35:

In [41]:
```
dim(datos2a35) # tamaño
```

34 · 46

Variables de datos2a35:

In [42]:
```
names(datos2a35)
```

'Observacion' · 'ID' · 'Sexo' · 'SexoNum' · 'Edad' · 'Fuma' · 'Estatura' · 'Colegio' · 'Estrato' · 'Financiacion' · 'Acumulado' · 'P1' · 'P2' · 'P3' · 'Final' · 'Definitiva' · 'Gastos' · 'Ingreso' · 'Gas' · 'Clases' · 'Ley' · 'PandemiaCat' · 'PandemiaNum' · 'Likert1' · 'Likert2' · 'Likert3' · 'Likert4' · 'Likert5' · 'AGPEQ1' · 'AGPEQ2' · 'AGPEQ3' · 'SATS1' · 'SATS2' · 'SATS3' · 'SATS4' · 'IDARE1.1' · 'IDARE1.2' · 'IDARE1.3' · 'IDARE1.4' · 'IDARE1.5' · 'IDARE2.6' · 'IDARE2.7' · 'IDARE2.8' · 'IDARE2.9' · 'IDARE2.10' · 'Puntaje'

Estructura de datos2a35:

In [43]:
```
str(datos2a35)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':       34 obs. of  46 variables:
 $ Observacion : num  2 3 4 5 6 7 8 9 10 11 ...
 $ ID          : chr  "SB11201910004475" "SB11201910011427" "SB11201910041975" "SB11
201910013623" ...
 $ Sexo        : chr  "Masculino" "Masculino" "Masculino" "Femenino" ...
 $ SexoNum     : num  1 1 1 0 0 0 0 0 1 0 ...
 $ Edad        : chr  "21.07" "20.92" "18.41" "16.64" ...
 $ Fuma        : chr  "Si" "Si" "Si" "Si" ...
 $ Estatura    : chr  "Baja" "Alta" "Alta" "Alta" ...
 $ Colegio     : chr  "Privado" "Privado" "Privado" "Privado" ...
 $ Estrato     : num  2 2 2 1 2 1 1 2 1 1 ...
 $ Financiacion: chr  "Beca" "Beca" "Beca" "Beca" ...
 $ Acumulado   : chr  "3.96" "3.85" "3.69" "4.01" ...
 $ P1          : chr  "2.3" "3.4" "2.5" "3.1" ...
 $ P2          : chr  "4.9" "3.6" "4.2" "3.5" ...
 $ P3          : chr  "3.7" "2.0" "5.0" "5.0" ...
 $ Final       : chr  "3.3" "1.9" "2.5" "3.0" ...
 $ Definitiva  : chr  "3.55" "2.73" "3.55" "3.65" ...
 $ Gastos      : chr  "72.1" "85.2" "56.6" "64.6" ...
 $ Ingreso     : chr  "2.07" "2.84" "1.55" "2.32" ...
 $ Gas         : chr  "24.17" "22.27" "23.08" "27.26" ...
 $ Clases      : chr  "Presencial" "Virtual" "Virtual" "Virtual" ...
 $ Ley         : chr  "En desacuerdo" "En desacuerdo" "En desacuerdo" "En desacuerd
o" ...
 $ PandemiaCat : chr  "De acuerdo" "De acuerdo" "De acuerdo" "Ni de acuerdo, ni en d
esacuerdo" ...
 $ PandemiaNum : num  3 3 3 2 3 3 3 1 3 1 ...
 $ Likert1     : num  3 2 5 1 3 4 1 2 5 5 ...
 $ Likert2     : num  2 3 4 1 2 3 2 1 4 4 ...
 $ Likert3     : num  4 3 2 5 3 3 3 4 1 1 ...
 $ Likert4     : num  1 4 5 2 1 2 3 1 5 5 ...
 $ Likert5     : num  1 2 1 4 4 2 1 3 4 4 ...
 $ AGPEQ1      : chr  "Ni de acuerdo, ni en desacuerdo" "De acuerdo" "De acuerdo" "T
otalmente de acuerdo" ...
 $ AGPEQ2      : chr  "En desacuerdo" "En desacuerdo" "Ni de acuerdo, ni en desacuer
do" "Ni de acuerdo, ni en desacuerdo" ...
 $ AGPEQ3      : chr  "En desacuerdo" "De acuerdo" "Totalmente de acuerdo" "Totalmen
te en desacuerdo" ...
 $ SATS1       : chr  "En desacuerdo" "En desacuerdo" "Totalmente en desacuerdo" "In
deciso" ...
 $ SATS2       : chr  "De acuerdo" "Totalmente de acuerdo" "Totalmente de acuerdo"
"De acuerdo" ...
 $ SATS3       : chr  "Indeciso" "Indeciso" "En desacuerdo" "En desacuerdo" ...
 $ SATS4       : chr  "Totalmente en desacuerdo" "De acuerdo" "De acuerdo" "Indecis
o" ...
 $ IDARE1.1    : chr  "Bastante" "Bastante" "Poco" "Poco" ...
 $ IDARE1.2    : chr  "Poco" "Poco" "No en lo absoluto" "Poco" ...
 $ IDARE1.3    : chr  "Mucho" "Bastante" "Bastante" "Bastante" ...
 $ IDARE1.4    : chr  "No en lo absoluto" "Bastante" "Bastante" "Bastante" ...
 $ IDARE1.5    : chr  "Bastante" "Poco" "No en lo absoluto" "Poco" ...
 $ IDARE2.6    : chr  "Frecuentemente" "Algunas veces" "Casi siempre" "Algunas vece
s" ...
 $ IDARE2.7    : chr  "Algunas veces" "Algunas veces" "Frecuentemente" "Algunas vece
s" ...
 $ IDARE2.8    : chr  "Algunas veces" "Algunas veces" "Frecuentemente" "Frecuentemen
te" ...
```

```
 $ IDARE2.9    : chr  "Frecuentemente" "Frecuentemente" "Casi nunca" "Casi nunca"
...
 $ IDARE2.10   : chr  "Frecuentemente" "Algunas veces" "Algunas veces" "Casi nunca"
...
 $ Puntaje     : num  78 77 70 68 65 54 50 36 35 35 ...
```

**Todas los puntos siguientes resolverlo con el dataset "datos2a35",**

**b) Definir el objeto "Sexo" (género de los estudiantes). Conviértalo en factor y diga cuáles son sus respectivos niveles.**

In [44]:
```
Sexo <- as.factor(datos2a35$Sexo)
```

Los niveles de la variable Sexo son los siguientes:

In [45]:
```
levels(Sexo)
```

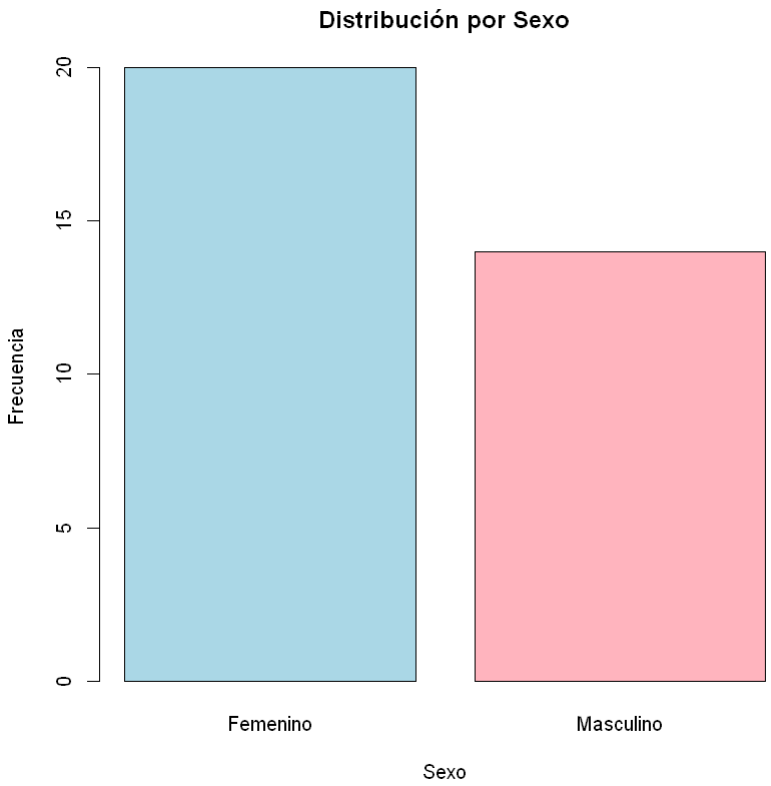
'Femenino' · 'Masculino'

**c) Construir una tabla de frecuencias para la variable Sexo y el diagrama de barras correspondiente.**

In [46]:
```
cat("Tabla de Frecuencias para la variable Sexo\n\n")
tabla_frecuencias <- table(Sexo)
print(tabla_frecuencias)
```

```
Tabla de Frecuencias para la variable Sexo

Sexo
 Femenino Masculino
       20        14
```

In [47]:
```
barplot(tabla_frecuencias, main = "Distribución por Sexo", xlab = "Sexo",
        ylab = "Frecuencia", col = c("lightblue", "lightpink"))
```




**d) Determinar la proporción de mujeres.**

La proporción de mujeres en la muestra se obtiene dividiendo el número de ocurrencias de 'Femenino' entre el total de elementos de la variable 'Sexo':

In [48]:
```
p_ <- sum(datos2a35$Sexo == 'Femenino') / length(Sexo)
round(p_, 4)
```

0.5882

**e) Mediante el método de la región crítica: Al nivel del 5%, determine si el porcentaje poblacional de mujeres es menor o igual que el 30%. Escribir un resumen del enunciado del problema, verificar los supuestos, concluya, diga cuál es la fórmula, el valor de prueba, el valor crítico, la región crítica e interprete.**

**Resumen:** se busca determinar si el porcentaje poblacional de mujeres es menor o igual al 30% en una muestra de estudiantes. Para esto, se plantean las siguientes hipótesis:

$$H_0 : p \leq 0.30$$

$$H_1 : p > 0.30$$

**Supuestos:** la distribución de la proporción muestral se la pueda aproximar como una curva normal. Para verificar esto, es necesario que el tamaño de la muestra multiplicado por la proporción bajo hipótesis debe ser mayor a 10 [3]:

$$np = 34 \times 0.30 = 10.2$$

Lo mismo debe esperarse al multiplicar el tamaño de la muestra por el recíproco de la proporción bajo hipótesis:

$$n(1-p) = 34 \times 0.70 = 23.8$$

**Fórmula:** en el cálculo se emplea el estadístico de prueba de proporciones para una muestra:

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

**Valor Crítico y Valor de Prueba:**

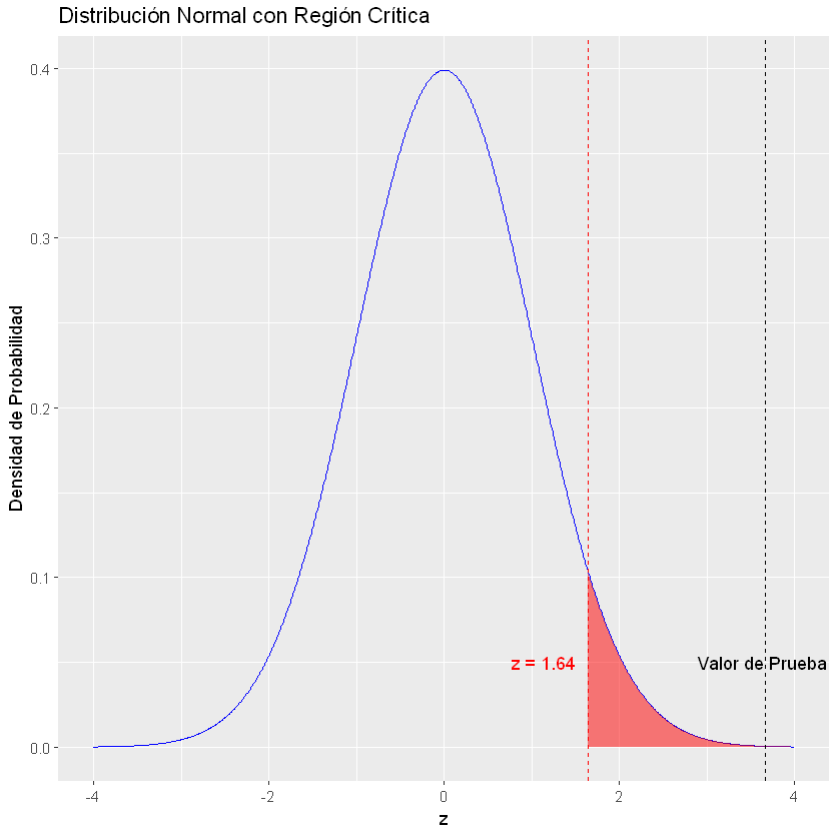
In [49]:
```
p <- 0.30
n <- nrow(datos2a35)
alpha <- 0.05
```

In [50]:
```
Zc <- qnorm(1 - alpha)
Zemp <- (p_ - p) / sqrt(p * (1 - p) / n)

print(paste("Valor Crítico (Zc):", round(Zc,2)))
print(paste("Valor de Prueba (Zemp):", round(Zemp, 2)))
```

```
[1] "Valor Crítico (Zc): 1.64"
[1] "Valor de Prueba (Zemp): 3.67"
```

**Región crítica:**

In [51]:
```
library(ggplot2)

x <- seq(-4, 4, length = 1000)
y <- dnorm(x)
data <- data.frame(x = x, y = y)

ggplot(data, aes(x = x, y = y)) +
  geom_line(color = "blue") +
  geom_area(data = subset(data, x >= Zc), aes(y = y), fill = "red", alpha = 0.5) +
  geom_vline(xintercept = Zc, color = "red", linetype = "dashed") +
  geom_vline(xintercept = Zemp, color = "black", linetype = "dashed") +
  labs(title = "Distribución Normal con Región Crítica", x = "z",
       y = "Densidad de Probabilidad") +
  annotate("text", x = Zc - 0.5, y = 0.05,
           label = paste("z =", round(Zc, 2)), color = "red") +
  annotate("text", x = Zemp + 0.3, y = 0.05,
           label = paste("Valor de Prueba =", round(Zemp, 2)), color = "black")
```




Dado que el valor de prueba (Zemp) es mayor al valor crítico (Zc), hay suficiente evidencia para rechazar la hipótesis nula.

**f) Mediante el método del P-valor: Determine si el porcentaje poblacional de mujeres es menor o igual que el 30%. Halle el P-valor, interprete y compare su decisión con el inciso (e).**

Para responder este inciso, se procede a calcular la probabilidad acumulada a derecha de valor de prueba Zemp y se compara esa área con el nivel de significancia:

In [52]:
```
p_value <- pnorm(Zemp, lower.tail = FALSE)

print(paste("P-value:", round(p_value,6)))
print(paste("Alpha:", alpha))
```

```
[1] "P-value: 0.000122"
[1] "Alpha: 0.05"
```

Como el p-valor es más pequeño que el nivel de significancia del 5%, hay suficiente evidencia para rechazar la hipótesis nula. Por lo que se arriba a la misma conclusión que en el inciso (e).

**g) Realizar la misma prueba del inciso (h) con la función prop.test y compare los resultados obtenidos.**

In [53]:
```
resultado_prop_test <- prop.test(sum(datos2a35$Sexo == 'Femenino'), n, p,
                                 alternative = "greater", correct=FALSE)
```

```
resultado_prop_test
```

```
	1-sample proportions test without continuity correction

data:  sum(datos2a35$Sexo == "Femenino") out of n, null probability p
X-squared = 13.451, df = 1, p-value = 0.0001224
alternative hypothesis: true p is greater than 0.3
95 percent confidence interval:
 0.4479564 1.0000000
sample estimates:
        p
0.5882353
```

Para efectuar la comparación entre el p-value obtenido en el inciso anterior y el obtenido al emplear la función prop.test, se procede a calcular la diferencia entre ambos valores, en donde puede observarse que el resultado tiene un valor numéricamente insignificante:

In [54]:
```
resultado_prop_test$p.value - p_value
```

0.00000000000000000001897353801849 63

**h) Construir un intervalo del 95% de confianza para la proporción poblacional de mujeres y compare los resultados obtenidos en los incisos anteriores.**

Para la estimación del intervalo de confianza, se emplearán los procedimientos descritos por Levine en [4] (p. 250)

In [55]:
```
Zc <- qnorm(1 - alpha)
SE <- sqrt(p_ * (1 - p_) / n)
margen_error <- Zc * SE
```

In [56]:
```
p_ + c(-1, 1) * margen_error
```

0.449403832191521 · 0.727066756043773

Al efectuar el cálculo del intervalo de confianza para obtener una estimación general de la proporción, se obtuvo como resultado:

$$CI = (0.4494038, 0.7270667)$$

Sin embargo, el resultado obtenido por la prueba de hipótesis con la función prop.test fue:

$$CI = (0.4479564, 1.0000000)$$

Esta discrepancia en el extremo derecho del intervalo se debe a que se realizó una prueba de hipótesis a derecha, que resulta en un intervalo más amplio y asimétrico, en tanto que el intervalo de confianza representa el rango con distribución normal donde se espera que esté la proporción verdadera.

**i) Construya intervalos de Confianza del 95% para la media mediante bootstrap con 10000 repeticiones.**

In [57]:
```
library(boot)
```

In [58]:
```
proporcion_femenino <- function(data, indices) {
  return(mean(data[indices]))
}

datos2a35$esFemenino <- datos2a35$Sexo == "Femenino"
bootstrap <- boot(data = datos2a35$esFemenino,
                  statistic = proporcion_femenino, R = 10000)
intervalo_confianza <- boot.ci(bootstrap, type = "perc", conf = 0.95)

print(intervalo_confianza)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = bootstrap, conf = 0.95, type = "perc")

Intervals :
Level     Percentile
95%   ( 0.4118,  0.7647 )
Calculations and Intervals on Original Scale
```

# Ejercicio N°6

In [59]:
```
# install.packages("naivebayes")
# install.packages("caret")
library(naivebayes)
library(caret)

set.seed(43)
data(infert, package="datasets")
```

```
naivebayes 1.0.0 loaded

For more information please visit:

https://majkamichal.github.io/naivebayes/

Cargando paquete requerido: lattice


Adjuntando el paquete: 'lattice'


The following object is masked from 'package:boot':

    melanoma
```

**a) Convertir las variables predictoras que sean categóricas en factores.**

In [60]:
```r
infert <- within(infert, {
  education <- as.factor(education)
  induced <- as.factor(induced)
  case <- as.factor(case)
  spontaneous <- as.factor(spontaneous)
  stratum <- as.factor(stratum)
  pooled.stratum <- as.factor(pooled.stratum)
})
```

**b) Entrenar el algoritmo de Naive Bayes con el 70% de los datos y el resto utilizarlos para testear.**

In [61]:
```r
trainIndex <- createDataPartition(infert$case, p = 0.7, list = FALSE)
trainData <- infert[trainIndex, ]
testData <- infert[-trainIndex, ]

common_columns <- intersect(colnames(trainData), colnames(testData))
trainData <- trainData[, common_columns]
testData <- testData[, common_columns]
```

In [62]:
```r
nb_model <- naive_bayes(case ~ ., data = trainData, laplace = 1)
```

**c) Calcular la matriz de confusión sobre las predicciones realizadas sobre los datos de testing.**

In [63]:
```r
predictions <- predict(nb_model, testData)
matriz_confusion <- confusionMatrix(predictions, testData$case)
matriz_confusion
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 28 19
         1 21  5

               Accuracy : 0.4521
                 95% CI : (0.3352, 0.573)
    No Information Rate : 0.6712
    P-Value [Acc > NIR] : 1.0000

                  Kappa : -0.2157

 Mcnemar's Test P-Value : 0.8744

            Sensitivity : 0.5714
            Specificity : 0.2083
         Pos Pred Value : 0.5957
         Neg Pred Value : 0.1923
             Prevalence : 0.6712
         Detection Rate : 0.3836
   Detection Prevalence : 0.6438
      Balanced Accuracy : 0.3899

       'Positive' Class : 0
```

**d) Calcular las métricas de accuracy, especificidad y sensibilidad.**

In [64]:
```r
accuracy <- matriz_confusion$overall['Accuracy']
sensitivity <- matriz_confusion$byClass['Sensitivity']
specificity <- matriz_confusion$byClass['Specificity']

print(paste("Accuracy:", accuracy))
print(paste("Sensibilidad:", sensitivity))
print(paste("Especificidad:", specificity))
```

```
[1] "Accuracy: 0.452054794520548"
[1] "Sensibilidad: 0.571428571428571"
[1] "Especificidad: 0.208333333333333"
```

# Referencias


[1] Levine, Krehbiel y Berenson (2006). Distribución de probabilidad de una variable aleatoria discreta. Estadística para Administración Cuarta Edición (pp. 155-156). Pearson Educación

[2] Lincoln L. Chao (1993). Teoría elemental de la probabilidad. Estadística para las ciencias administrativas Tercera Edición (p. 81). McGraw Hill

[3] “Elementary Statistics“, PennState, https://online.stat.psu.edu/stat200/book/export/html/144

[4] Levine, Krehbiel y Berenson (2006). Estimación del Intervalo de Confianza para una Proporción. Estadística para Administración Cuarta Edición (pp. 250-252). Pearson Educación.