Profesor: Rodrigo Del Rosso

Alumno: José Valdés

Materia: Estadística

EXAMEN ESTADÍSTICA 2023

MAESTRÍA EN EXPLOTACIÓN DE DATOS Y GESTIÓN DEL CONOCIMIENTO

TABLA DE CONTENIDO

TABLA DE ILUSTRACIONES

TABLA DE ANEXOS

Anexos	nombre	extensión	Url
Anexo No. 1	examenEstadistica2023	html	Archivo web
Anexo No. 2	Dataset ejercicio 1	csv	Archivo

INTRODUCCIÓN

El presente documento se desarrolla en el marco de la Maestría En Explotación De Datos Y Gestión Del Conocimiento de la Universidad Austral de Argentina, materia Estadística impartida durante el primer año de la cursada.

Con el desarrollo del presente documento se busca articular todos los conocimientos adquiridos durante el desarrollo del primer cuatrimestre del primer año del desarrollo de la maestría, se utiliza como herramienta base el software RStudio y la funcionalidad de Markdown.

EXAMEN ESTADÍSTICA AÑO 2023

A continuación, se presenta la secuencia de seis ejercicios de las distintas temáticas impartidas durante la cursada de la materia estadística, esta fue desarrollada durante el periodo 25/08/2022 hasta el 20/10/2022. El presente documento fue desarrollado durante el periodo 03/02/2023 hasta el 17/02/2023.

Para el desarrollo del archivo Markdown se realiza la validación de la instalación de los paquetes necesarios para ejecutar el script. A continuación, se presenta el código utilizado:

```
rm(list = ls())


# Bibliotecas a cargar
check_packages <- function(packages) {
  if (all(packages %in% rownames(installed.packages()))) {
    TRUE
  } else{
    cat(
      "Instalar los siguientes packages antes de ejecutar el presente script\n",
      packages[!(packages %in% rownames(installed.packages()))],
      "\n"
    )
  }
}
packages_needed <- c("repmis")


# Se llama a la funcion check_packages
check_packages(packages_needed)
```

```
## [1] TRUE
```

```
library(repmis)
```

Con la función check_packages se realiza la verificación de los paquetes necesarios para ejecutar cada ejerció, en caso de ser necesario algún paquete adicional, esta función se utiliza para solo referenciar sobre la variable packages_needed una lista de los nombres de los paquetes que se requieran. Para el desarrollo de los seis ejercicios sin entrar a utilizar paquetes que permitan tener una visualización de datos más estética, solo se requirió utilizar el paquete repmis.

Ejercicio N° 1 - Estadística Descriptiva

Seleccione una base de datos pública de su interés y seleccione una muestra de n observaciones aleatorias al azar. Alternativamente, simule una muestra aleatoria simple de al menos 1000 registros. Para la entrega de este examen deberá adjuntar la muestra seleccionada y el procedimiento que permitir obtener los registros muestrales. A partir de la misma se solicita lo siguiente:

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

Solución del ejercicio: (David M. Levine, 2023)

Base de datos públicas a utilizar en el ejercicio: (r-coder.com, 2023)

```
##Se comparte URL del archivo trabajado

URLEjer1<-
"https://d3c33hcgiwev3.cloudfront.net/1sf2mERLTmuH9phES65rBw_a8f877dcd6d84ca3b48e8390d40c74f1_Ordenes_productos_C1_M2.csv?Expires=1675641600&Signature=iVg7NljP9Bz3iZmt3LKXouVnrPxlN5TmuSB7ccYcV~cyUHVjqTX5zp-qVG8J7XSd6Td9ID5gzkC~eYn-PxAmVNgOj6QfdFNdcfy3WYQ~ISMgg6dfp2N~Y48dDpoglh5PuBeWCKA1dBRES3F3H2dHD-FntYvYic9L9-D84grVR-0_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A"
```

```
#url_archivo <- paste(URLEjer1,"Ordenes_productos_C1_M2.csv", sep = ";")


#Se carga el archivo de forma local, esto por no saber cómo cargado desde la
web.

file<-"C:/Users/Josvaldes/Documents/Maestria/Austral/1
ano/Estadistica/examen/Ordenes_productos_C1_M2.csv"

ordenesProductosRetiel <- read.csv(file, sep = ";", header = TRUE)
```

Se procede con la selección de una muestra aleatoria al azar, prueba con 1000 registros:

```
#se transforma la lista contenida en el objeto ordenesProductosRetiel en un data
frame


df<-data.frame(matrix(unlist(ordenesProductosRetiel), nrow =
length(ordenesProductosRetiel), byrow = TRUE))


#se selecciona una muestra aleatoria simple del data frame

muestra1000registros <- df[sample(nrow(df), size=1000, replace = TRUE, prob =
NULL)]


#Se cambia los nombres de las variables a trabajar

value <-
c("orden_id","order_item_id","producto_id","vendedor_id","fecha_envio_limite","p
recio","valor_flete","codigo_postal_vendedor","ciudad_vendedor","departamento_ve
ndedor","nombre_categoria_producto","longitud_nombre_producto","longitud_descrip
cion_producto","cantidad_fotos_producto","peso_g_producto","longitud_cm_producto
","altura_cm_producto","ancho_cm_producto")


row.names(muestra1000registros) <- value


#Se usa la función de transpuesta para regresar las filas y las columnas a la
estructura original y dejar el objeto como un data frame

df00<-as.data.frame(t(muestra1000registros))
```

Debido a que las conversiones ocasionaron que el tipo de datos cambiara todo a carácter se hace la corrección de las columnas al tipo de dato del dataset original:

```
df00$precio=as.numeric(df00$precio)

df00$valor_flete=as.numeric(df00$valor_flete)

df00$codigo_postal_vendedor=as.integer(df00$codigo_postal_vendedor)

df00$longitud_nombre_producto=as.integer(df00$longitud_nombre_producto)

df00$longitud_descripcion_producto=as.integer(df00$longitud_descripcion_producto
)

df00$cantidad_fotos_producto=as.integer(df00$cantidad_fotos_producto)

df00$peso_g_producto=as.integer(df00$peso_g_producto)

df00$longitud_cm_producto=as.integer(df00$longitud_cm_producto)

df00$altura_cm_producto=as.integer(df00$altura_cm_producto)

df00$ancho_cm_producto=as.integer(df00$ancho_cm_producto)

str(df00)
```

```
## 'data.frame':    1000 obs. of  18 variables:

##  $ orden_id                    : chr  "28050PK20B" "94053PO72A"
"107500PO59A" "72523PG54A" ...

##  $ order_item_id               : chr  "B" "A" "A" "A" ...

##  $ producto_id                 : chr  "PK20" "PO72" "PO59" "PG54" ...

##  $ vendedor_id                 : chr  "VE9159" "VE3276" "VE5389" "VE5229"
...

##  $ fecha_envio_limite          : chr  "17/08/2017 8:15" "11/02/2018 18:32"
"7/04/2018 18:12" "18/01/2018 2:39" ...

##  $ precio                      : num  433 68.4 271.9 61 51.5 ...

##  $ valor_flete                 : num  82.7 9.11 30.72 52.69 11.1 ...

##  $ codigo_postal_vendedor      : int  66001 52051 52435 81001 73001 50683
52356 81001 52435 8001 ...

##  $ ciudad_vendedor             : chr  "Pereira" "Arboleda" "Mallama"
"Arauca" ...
```

```
##  $ departamento_vendedor        : chr  "Risaralda" "Nari\xf1o" "Nari\xf1o"
"Arauca" ...

##  $ nombre_categoria_producto    : chr  "Deportes" "" "Productos ecoamigables"
"Salud" ...

##  $ longitud_nombre_producto     : int  25 NA 6 21 23 5 13 21 6 NA ...

##  $ longitud_descripcion_producto: int  5 NA 7 33 16 35 19 33 7 NA ...

##  $ cantidad_fotos_producto      : int  4 NA 27 0 35 26 29 0 27 NA ...

##  $ peso_g_producto              : int  5270 NA 2486 258 884 286 811 258 2486
NA ...

##  $ longitud_cm_producto         : int  9 NA 17 22 45 29 29 22 17 NA ...

##  $ altura_cm_producto           : int  27 NA 11 24 26 21 9 24 11 NA ...

##  $ ancho_cm_producto            : int  29 NA 14 25 18 18 16 25 14 NA ...
```

Sobre el objeto ordenesProductosRetiel se entrega el dataset utilizado para el ejercicio y sobre el objeto df00 se entrega la muestra aleatoria simple de 1000 observaciones del dataset.

• 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.

```
#La población utilizada es la que está en el dataset ordenesProductosRetiel las
características del objeto son las siguiente: 10134 obs. of 18 variables:
str(ordenesProductosRetiel)
```

```
## 'data.frame':    10134 obs. of  18 variables:

##  $ orden_id                     : chr  "107500PO59A" "37493PS22B"
"28050PK20B" "52187PA10A" ...

##  $ order_item_id                : chr  "A" "B" "B" "A" ...
```

```
##  $ producto_id                 : chr  "PO59" "PS22" "PK20" "PA10" ...

##  $ vendedor_id                 : chr  "VE5389" "VE1558" "VE9159" "VE3159"
...

##  $ fecha_envio_limite          : chr  "7/04/2018 18:12" "20/10/2017 9:07"
"17/08/2017 8:15" "23/09/2017 23:27" ...

##  $ precio                      : num  271.9 115.7 433 108.4 51.5 ...

##  $ valor_flete                 : num  30.72 4.68 82.7 35.39 11.1 ...

##  $ codigo_postal_vendedor      : int  52435 52203 66001 52435 73001 52356
52240 50683 52381 66001 ...

##  $ ciudad_vendedor             : chr  "Mallama" "Colon" "Pereira" "Mallama"
...

##  $ departamento_vendedor       : chr  "Nari\xf1o" "Nari\xf1o" "Risaralda"
"Nari\xf1o" ...

##  $ nombre_categoria_producto   : chr  "Productos ecoamigables"
"Carnicer\xeda" "Deportes" "Electrodom\xe9sticos" ...

##  $ longitud_nombre_producto    : int  6 10 25 10 23 13 8 5 7 1 ...

##  $ longitud_descripcion_producto: int  7 31 5 1 16 19 37 35 11 39 ...

##  $ cantidad_fotos_producto     : int  27 20 4 6 35 29 24 26 30 33 ...

##  $ peso_g_producto             : int  2486 256 5270 734 884 811 621 286 30
16 ...

##  $ longitud_cm_producto        : int  17 43 9 46 45 29 26 29 50 11 ...

##  $ altura_cm_producto          : int  11 2 27 48 26 9 41 21 26 42 ...

##  $ ancho_cm_producto           : int  14 21 29 22 18 16 29 18 17 12 ...
```

```
#La muestra utilizada es la que está en el objeto df00 las características del
objeto son las siguiente: 1000 obs. of 18 variables:

str(df00)
```

```
## 'data.frame':    1000 obs. of  18 variables:

##  $ orden_id                     : chr  "28050PK20B" "94053PO72A"
"107500PO59A" "72523PG54A" ...

##  $ order_item_id                : chr  "B" "A" "A" "A" ...

##  $ producto_id                  : chr  "PK20" "PO72" "PO59" "PG54" ...

##  $ vendedor_id                  : chr  "VE9159" "VE3276" "VE5389" "VE5229"
...

##  $ fecha_envio_limite           : chr  "17/08/2017 8:15" "11/02/2018 18:32"
"7/04/2018 18:12" "18/01/2018 2:39" ...

##  $ precio                       : num  433 68.4 271.9 61 51.5 ...

##  $ valor_flete                  : num  82.7 9.11 30.72 52.69 11.1 ...

##  $ codigo_postal_vendedor       : int  66001 52051 52435 81001 73001 50683
52356 81001 52435 8001 ...

##  $ ciudad_vendedor              : chr  "Pereira" "Arboleda" "Mallama"
"Arauca" ...

##  $ departamento_vendedor        : chr  "Risaralda" "Nari\xf1o" "Nari\xf1o"
"Arauca" ...

##  $ nombre_categoria_producto    : chr  "Deportes" "" "Productos ecoamigables"
"Salud" ...

##  $ longitud_nombre_producto     : int  25 NA 6 21 23 5 13 21 6 NA ...

##  $ longitud_descripcion_producto: int  5 NA 7 33 16 35 19 33 7 NA ...

##  $ cantidad_fotos_producto      : int  4 NA 27 0 35 26 29 0 27 NA ...

##  $ peso_g_producto              : int  5270 NA 2486 258 884 286 811 258 2486
NA ...

##  $ longitud_cm_producto         : int  9 NA 17 22 45 29 29 22 17 NA ...

##  $ altura_cm_producto           : int  27 NA 11 24 26 21 9 24 11 NA ...
```

```
##  $ ancho_cm_producto           : int  29 NA 14 25 18 18 16 25 14 NA ...
```

```
#El experimento que se usa es un muestreo aleatorio simple
#las variables bajo análisis se describen en las salidas de la funcion str()
```

Como se observan en las salidas del dataset utilizado relacionado al sector retail, se tiene una base de 10134 observaciones y 18 variables, de esta población se realiza un muestreo aleatorio simple con el cual se toma una muestra aleatoria de 1000 observaciones.

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

```
#Se valida el resumen de los datos del data frame (muestra)
summary(df00)
```

```
##    orden_id         order_item_id      producto_id        vendedor_id
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##  fecha_envio_limite     precio        valor_flete     codigo_postal_vendedor
##  Length:1000        Min.   : 13.48   Min.   : 0.320   Min.   : 8001
##  Class :character   1st Qu.: 51.57   1st Qu.: 8.002   1st Qu.:52240
##  Mode  :character   Median :110.96   Median :12.500   Median :52435
##                     Mean   :153.27   Mean   :23.092   Mean   :56272
##                     3rd Qu.:244.44   3rd Qu.:33.680   3rd Qu.:66001
##                     Max.   :432.99   Max.   :83.650   Max.   :81001
```

```
##

##  ciudad_vendedor    departamento_vendedor nombre_categoria_producto

##  Length:1000        Length:1000           Length:1000

##  Class :character   Class :character      Class :character

##  Mode  :character   Mode  :character      Mode  :character

##

##  longitud_nombre_producto longitud_descripcion_producto
cantidad_fotos_producto

##  Min.   : 1.00            Min.   : 1.00                 Min.   : 0.00

##  1st Qu.: 6.00            1st Qu.: 7.00                 1st Qu.: 6.00

##  Median :10.00            Median :19.00                 Median :26.00

##  Mean   :11.42            Mean   :21.47                 Mean   :21.92

##  3rd Qu.:21.00            3rd Qu.:35.00                 3rd Qu.:30.00

##  Max.   :25.00            Max.   :39.00                 Max.   :35.00

##  NA's   :399              NA's   :399                   NA's   :399

##  peso_g_producto longitud_cm_producto altura_cm_producto ancho_cm_producto

##  Min.   :  16    Min.   : 9.00        Min.   : 2.00      Min.   :12.00

##  1st Qu.: 256    1st Qu.:17.00        1st Qu.:11.00      1st Qu.:16.00

##  Median : 621    Median :29.00        Median :26.00      Median :18.00

##  Mean   :1004    Mean   :29.51        Mean   :25.07      Mean   :19.73

##  3rd Qu.: 884    3rd Qu.:45.00        3rd Qu.:41.00      3rd Qu.:25.00

##  Max.   :5270    Max.   :50.00        Max.   :48.00      Max.   :29.00
```

```
##  NA's   :399     NA's   :399          NA's   :399        NA's   :399
```

Para un análisis más detallado se hace la revisión de cada una de las variables, se excluyen las cualitativas:

Variable Precio

```
#Población
summary(ordenesProductosRetiel$precio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.01   34.16   83.12  119.79  165.51 1262.94
```

```
#muestra
summary(df00$precio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.48   51.57  110.96  153.27  244.44  432.99
```

Variable valor flete

```
#Población
summary(ordenesProductosRetiel$valor_flete)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    5.76   13.98   20.05   27.45  183.15
```

```
#muestra
summary(df00$valor_flete)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.320   8.002  12.500  23.092  33.680  83.650
```

longitud_nombre_producto

```
#Población
summary(ordenesProductosRetiel$longitud_nombre_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   10.00   20.00   20.07   30.00   40.00      23
```

```
#muestra
summary(df00$longitud_nombre_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    6.00   10.00   11.42   21.00   25.00     399
```

longitud_nombre_producto

```
#Población
summary(ordenesProductosRetiel$longitud_nombre_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   10.00   20.00   20.07   30.00   40.00      23
```

```
#muestra
summary(df00$longitud_nombre_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    6.00   10.00   11.42   21.00   25.00     399
```

longitud_descripcion_producto

```
#Población
summary(ordenesProductosRetiel$longitud_descripcion_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   10.00   20.00   19.84   30.00   40.00      23
```

```
#muestra
summary(df00$longitud_descripcion_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    7.00   19.00   21.47   35.00   39.00     399
```

cantidad_fotos_producto

```
#Población
summary(ordenesProductosRetiel$cantidad_fotos_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    10.0    20.0    20.1    30.0    40.0      23
```

```
#muestra
summary(df00$cantidad_fotos_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    6.00   26.00   21.92   30.00   35.00     399
```

altura_cm_producto

```
#Población
summary(ordenesProductosRetiel$altura_cm_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    9.00   17.00   17.91   26.00   64.00      23
```

```
#muestra
summary(df00$altura_cm_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2.00   11.00   26.00   25.07   41.00   48.00     399
```

ancho_cm_producto

```
#Población
summary(ordenesProductosRetiel$ancho_cm_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   15.00   23.00   22.91   30.00   72.00      23
```

```
#muestra
summary(df00$ancho_cm_producto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's

##   12.00   16.00   18.00   19.73   25.00   29.00     399
```

• 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.

Variable Precio

```
#Histograma población

hist(x = ordenesProductosRetiel$precio, main = "Histograma de precio -
Población",

     xlab = "Precio", ylab = "Frecuencia",

     col = "purple")
```

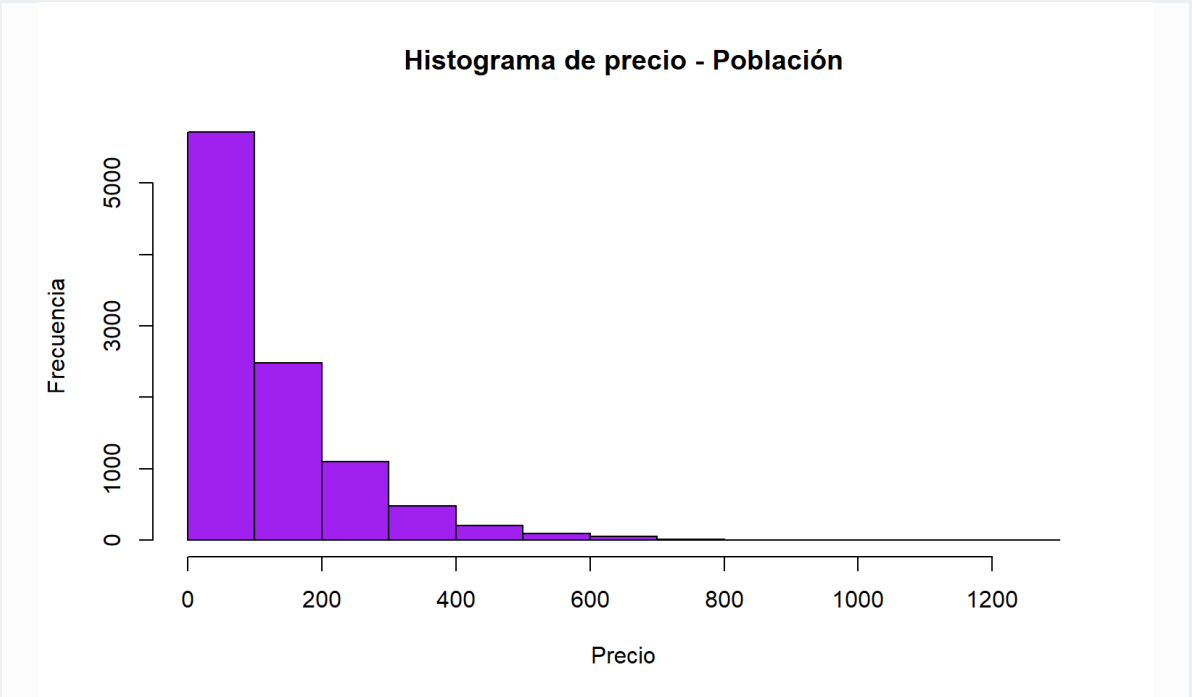

Ilustración 1: Histograma de precio - Población.

```
#Histograma muestra

hist(x = df00$precio, main = "Histograma de precio - Muestra",

     xlab = "Precio", ylab = "Frecuencia",
```

```
      col = "purple")
```

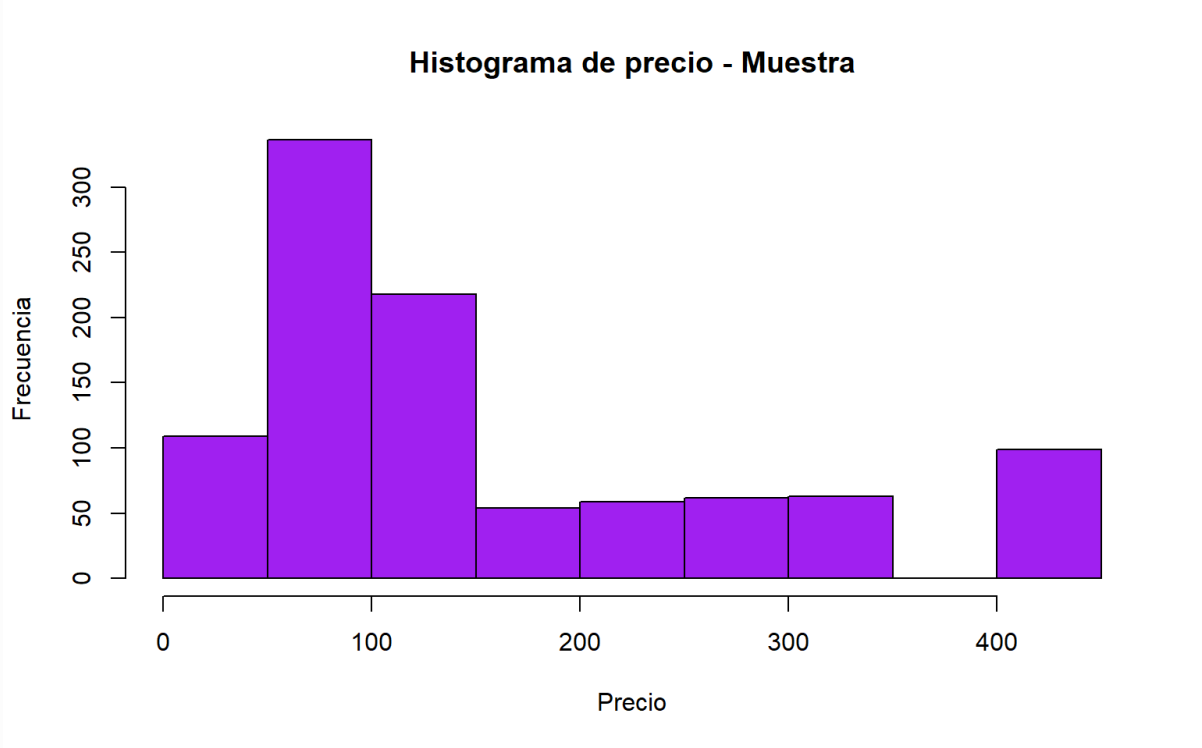

Ilustración 2: Histograma de precio - Muestra.

Variable valor flete

```
#Histograma población

hist(x = ordenesProductosRetiel$valor_flete, main = "Histograma de valor flete -
Población",

     xlab = "valor flete", ylab = "Frecuencia",

     col = "purple")
```

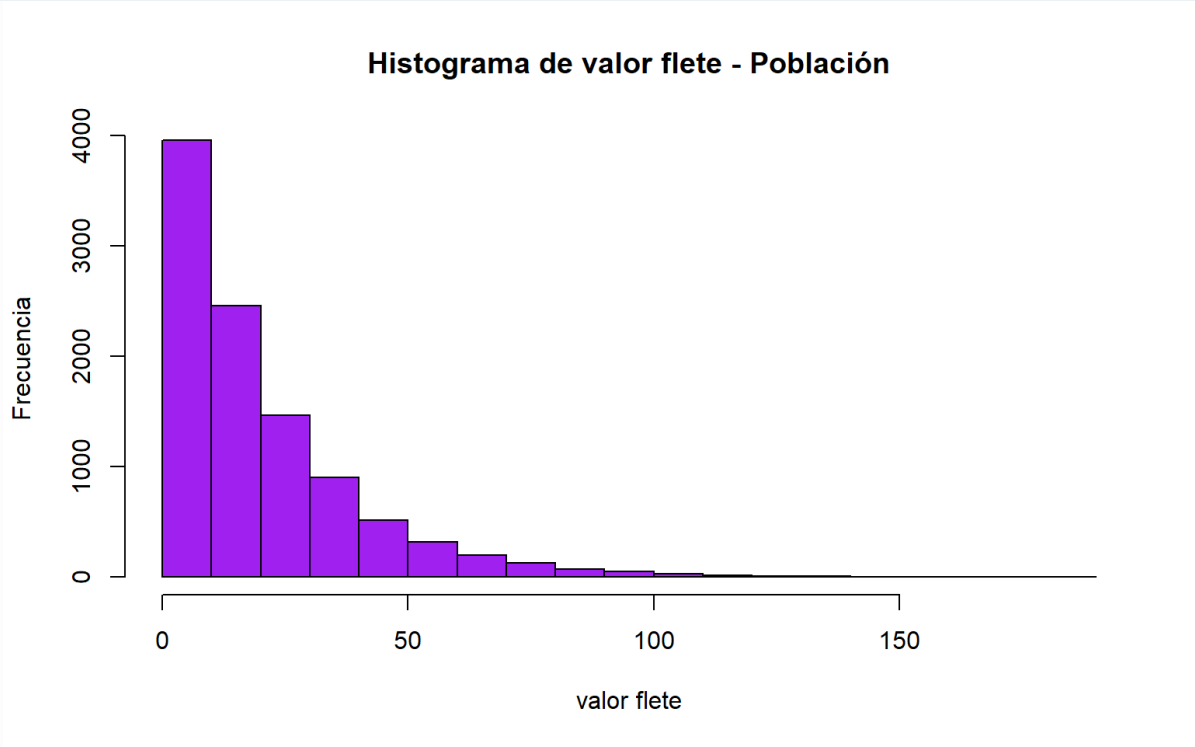

Ilustración 3: Histograma de valor flete - Población.

```
#Histograma muestra
hist(x = df00$valor_flete, main = "Histograma de valor flete - Muestra",
     xlab = "valor flete", ylab = "Frecuencia",
     col = "purple")
```

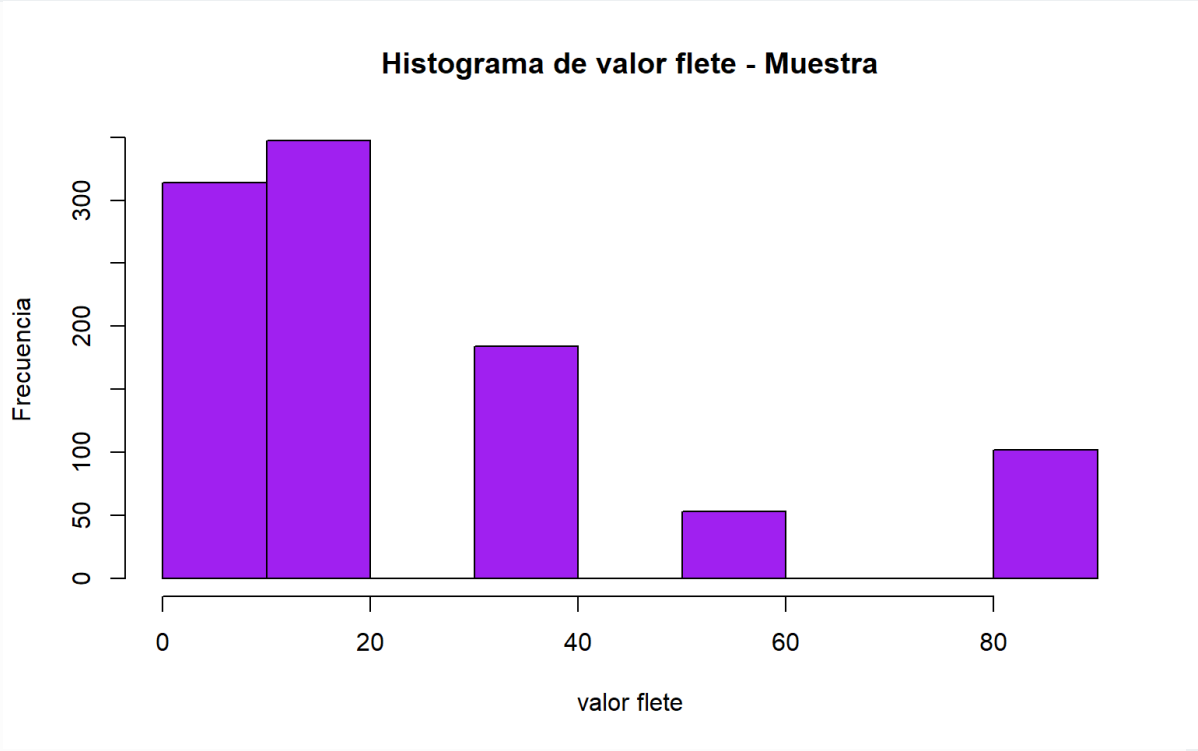

Ilustración 4: Histograma de valor flete - Muestra.

longitud_nombre_producto

```
#Histograma población

hist(x = ordenesProductosRetiel$longitud_nombre_producto, main = "Histograma de
longitud_nombre_producto - Población",

     xlab = "longitud_nombre_producto", ylab = "Frecuencia",

     col = "purple")
```

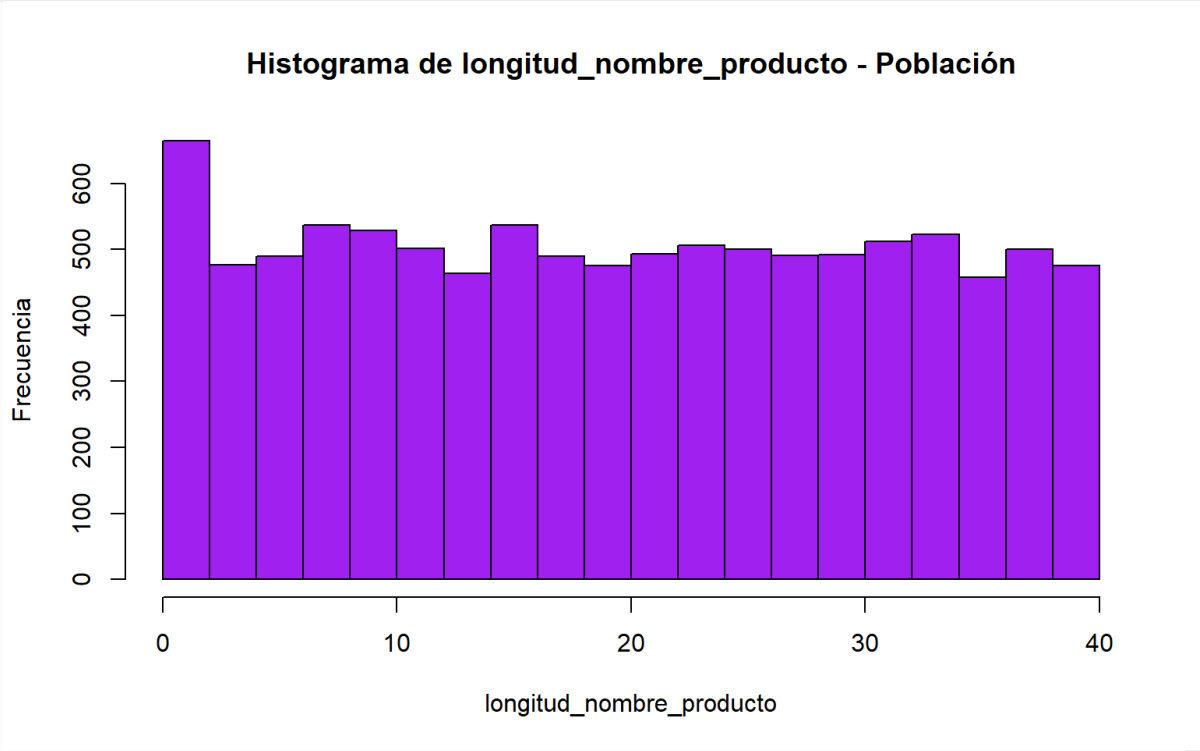

Ilustración 5: Histograma de longitud_nombre_producto - Población.

```
#Histograma muestra

hist(x = df00$longitud_nombre_producto, main = "Histograma de
longitud_nombre_producto - Muestra",

     xlab = "longitud_nombre_producto", ylab = "Frecuencia",

     col = "purple")
```

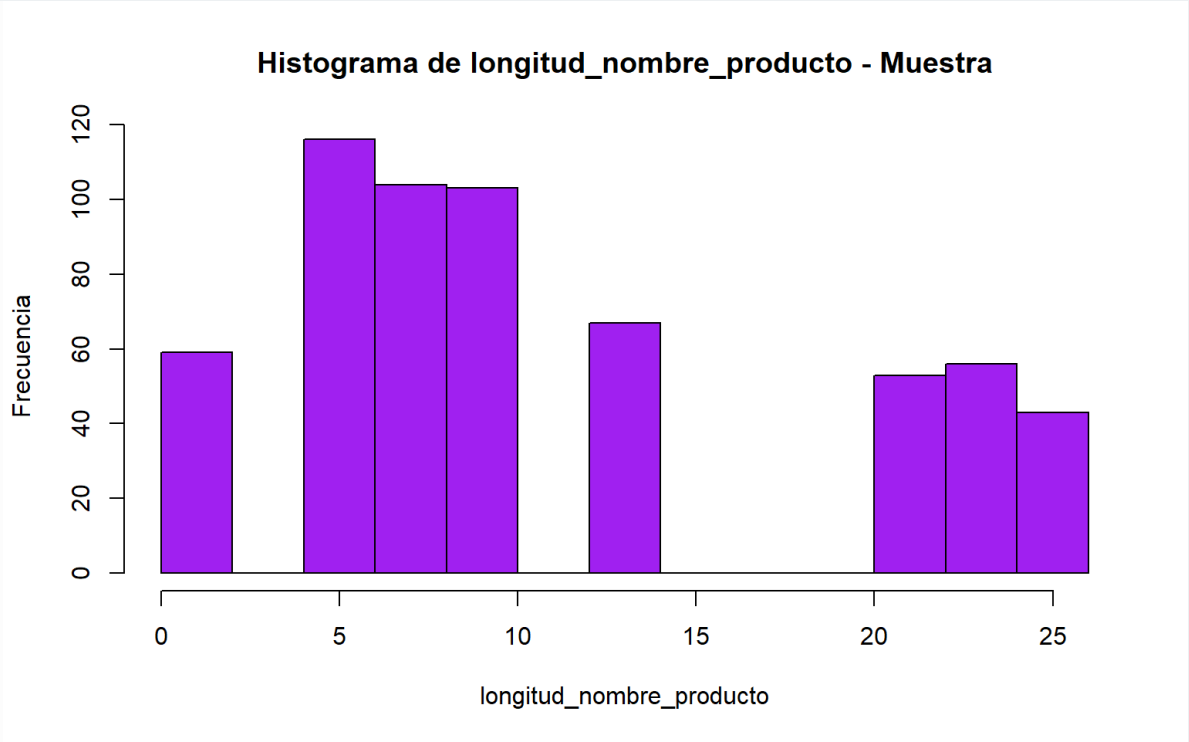

Ilustración 6: Histograma de longitud_nombre_producto - Muestra.

longitud_descripcion_producto

```
#Histograma población

hist(x = ordenesProductosRetiel$longitud_descripcion_producto, main =
"Histograma de longitud_descripcion_producto - Población",

     xlab = "longitud_descripcion_producto", ylab = "Frecuencia",

     col = "purple")
```

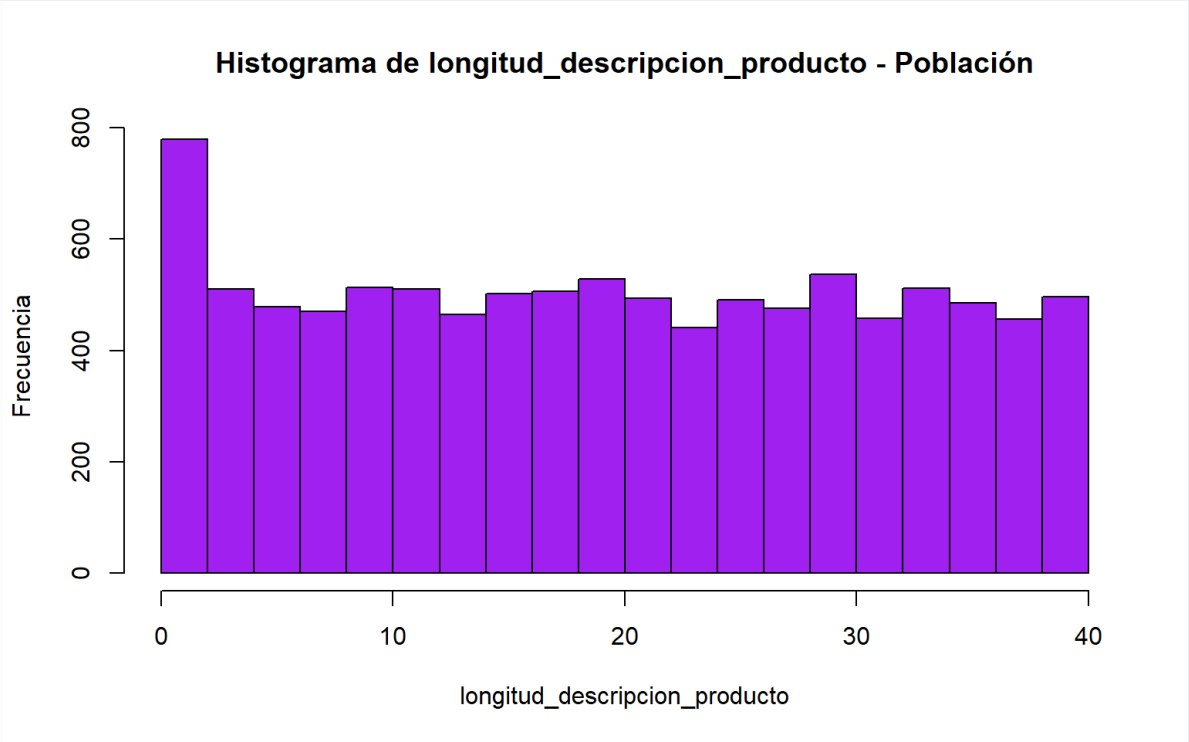

Ilustración 7: Histograma de longitud_descripcion_producto - Población.

```
#Histograma muestra

hist(x = df00$longitud_nombre_producto, main = "Histograma de
longitud_descripcion_producto - Muestra",

     xlab = "longitud_descripcion_producto", ylab = "Frecuencia",

     col = "purple")
```

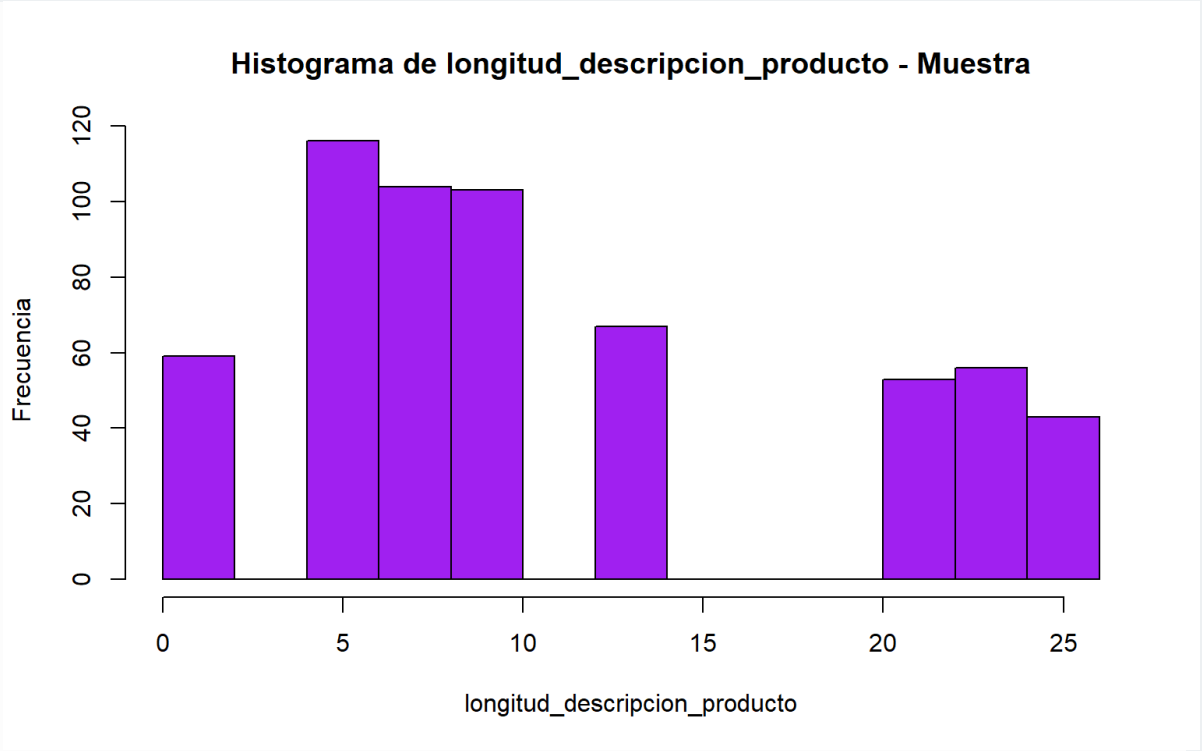

Ilustración 8: Histograma de longitud_descripcion_producto - Muestra.

cantidad_fotos_producto

```
#Histograma población

hist(x = ordenesProductosRetiel$cantidad_fotos_producto, main = "Histograma de
cantidad_fotos_producto - Población",

     xlab = "cantidad_fotos_producto", ylab = "Frecuencia",

     col = "purple")
```

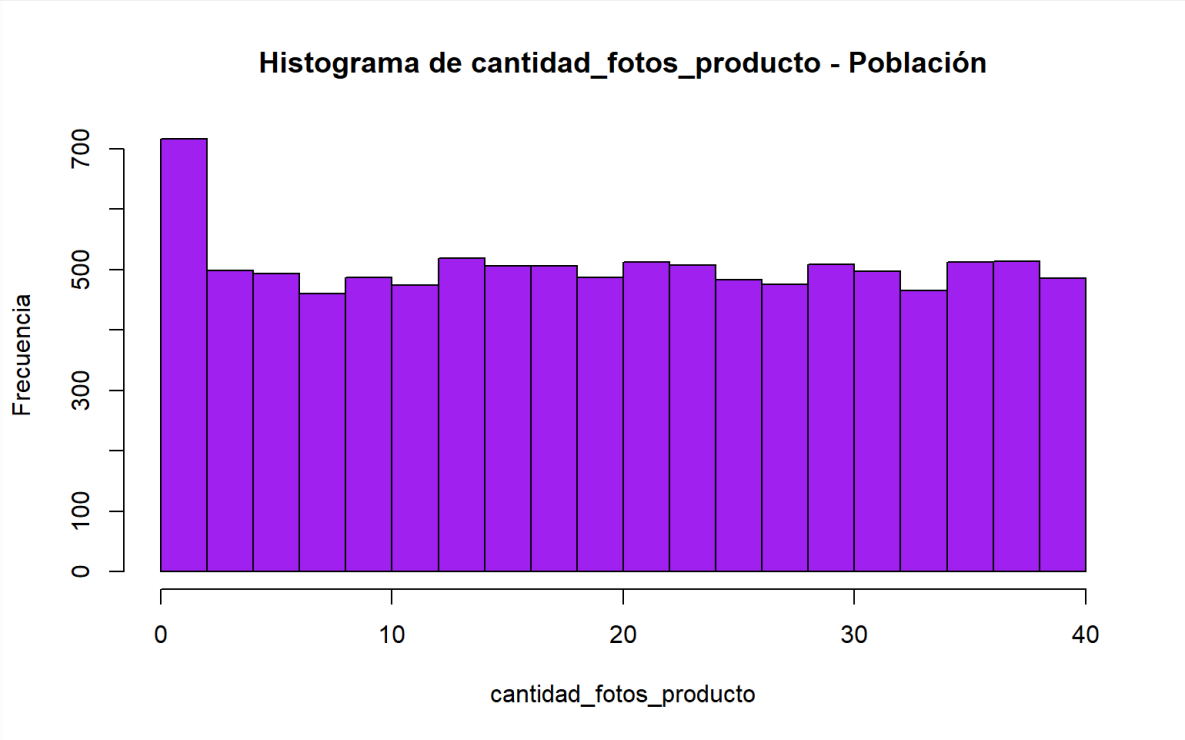

Ilustración 9: Histograma de cantidad_fotos_producto - Población.

```
#Histograma muestra

hist(x = df00$cantidad_fotos_producto, main = "Histograma de
cantidad_fotos_producto - Muestra",

     xlab = "cantidad_fotos_producto", ylab = "Frecuencia",

     col = "purple")
```

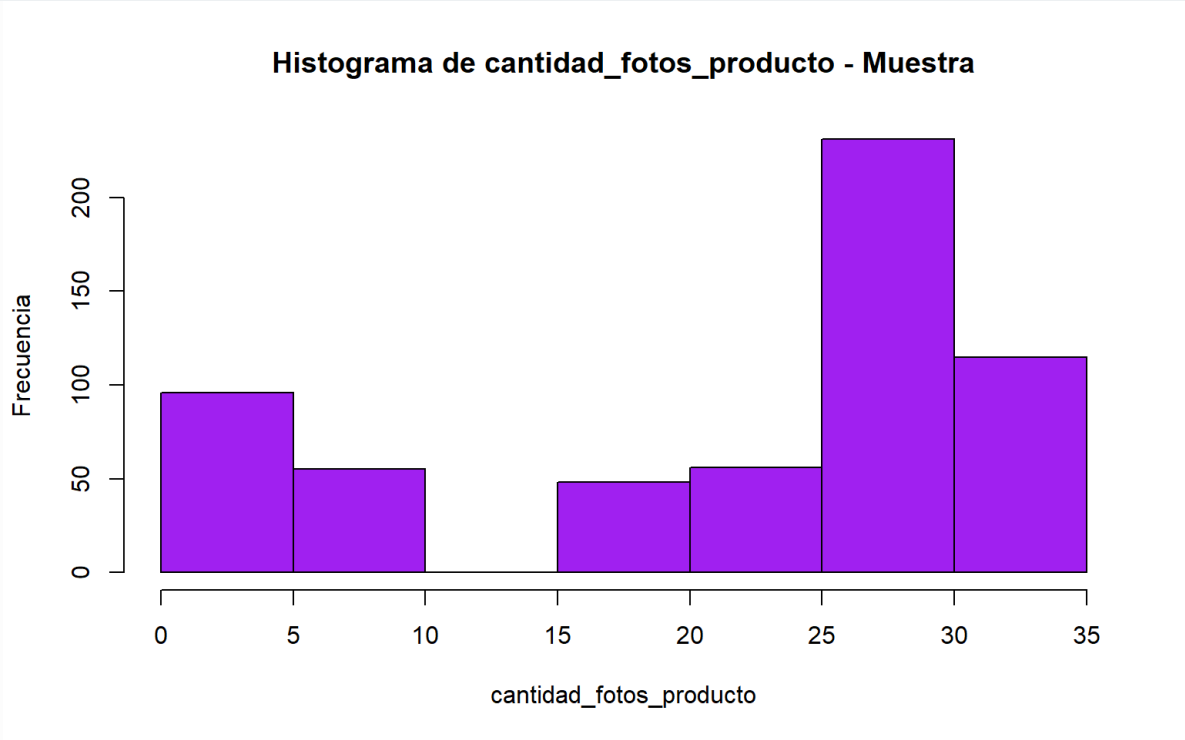

Ilustración 10: Histograma de cantidad_fotos_producto - Muestra.

altura_cm_producto

```
#Histograma población

hist(x = ordenesProductosRetiel$altura_cm_producto, main = "Histograma de
altura_cm_producto - Población",

     xlab = "altura_cm_producto", ylab = "Frecuencia",

     col = "purple")
```

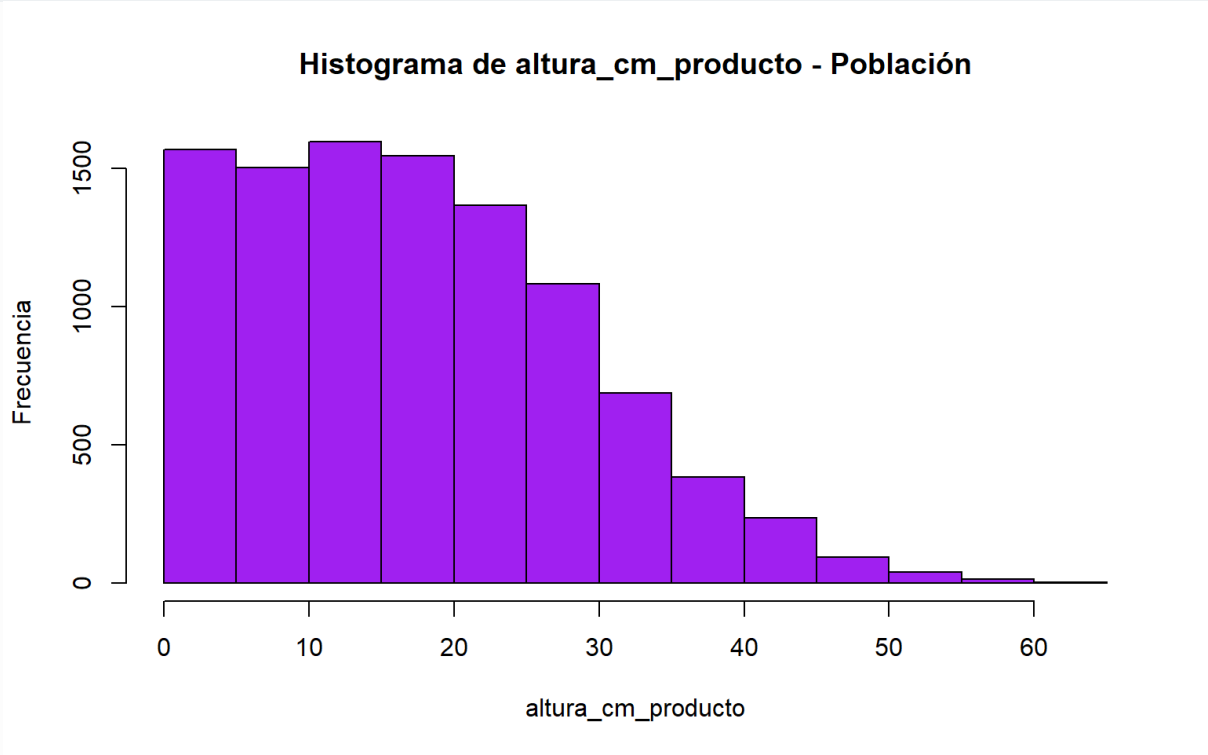

Ilustración 11: Histograma de altura_cm_producto - Población.

```
#Histograma muestra

hist(x = df00$altura_cm_producto, main = "Histograma de altura_cm_producto -
Muestra",

     xlab = "altura_cm_producto", ylab = "Frecuencia",

     col = "purple")
```

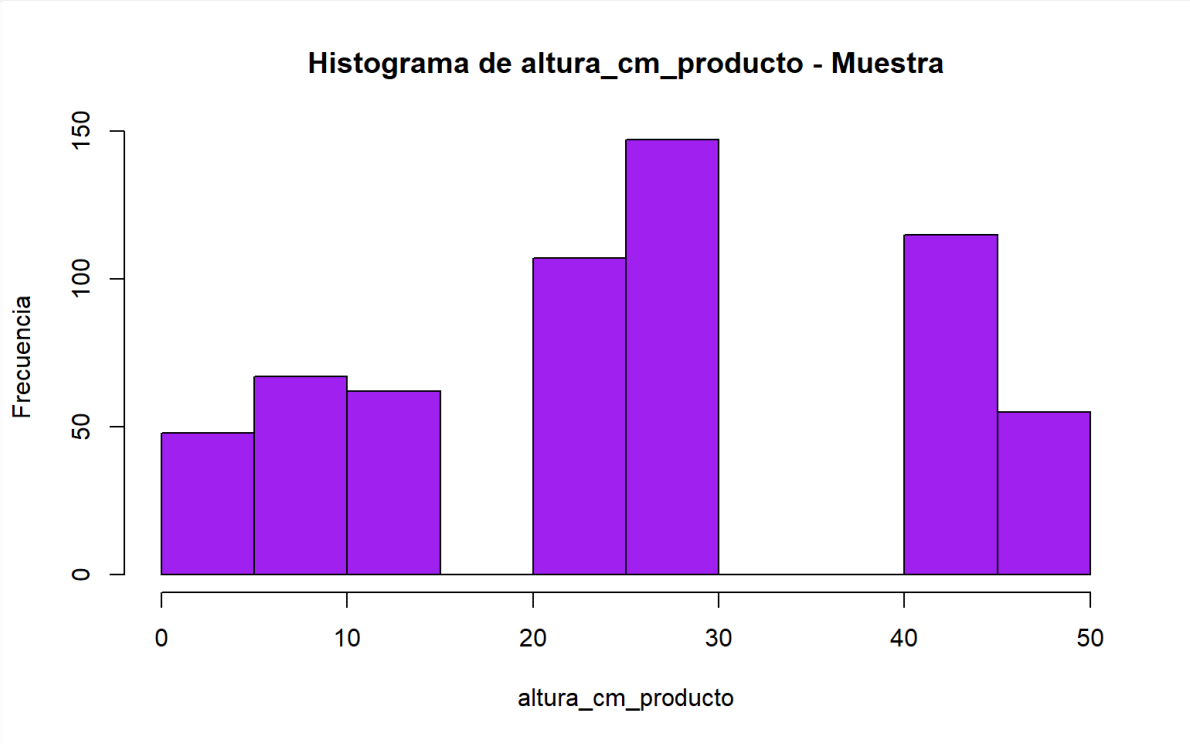

Ilustración 12: Histograma de altura_cm_producto - Muestra.

ancho_cm_producto

```
#Histograma población

hist(x = ordenesProductosRetiel$ancho_cm_producto, main = "Histograma de
ancho_cm_producto - Población",

     xlab = "ancho_cm_producto", ylab = "Frecuencia",

     col = "purple")
```

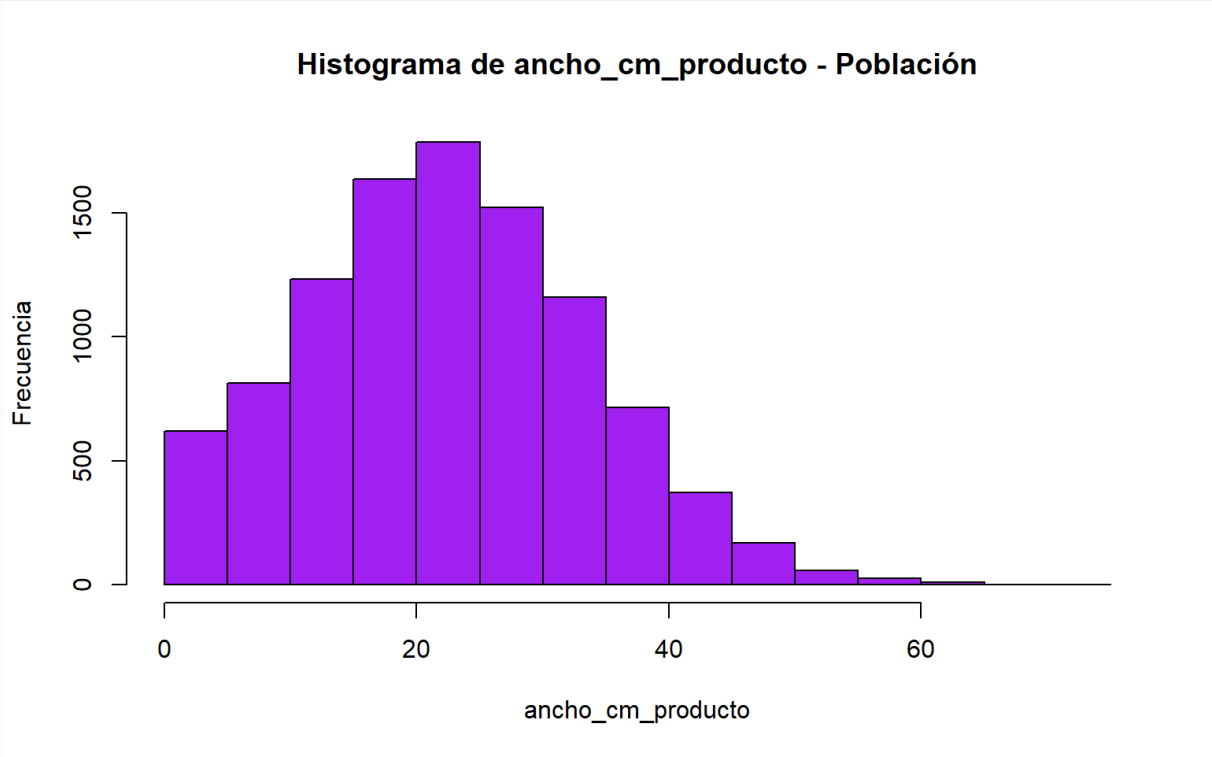

Ilustración 13: Histograma de ancho_cm_producto - Población.

```
#Histograma muestra

hist(x = df00$ancho_cm_producto, main = "Histograma de ancho_cm_producto -
Muestra",

     xlab = "ancho_cm_producto", ylab = "Frecuencia",

     col = "purple")
```

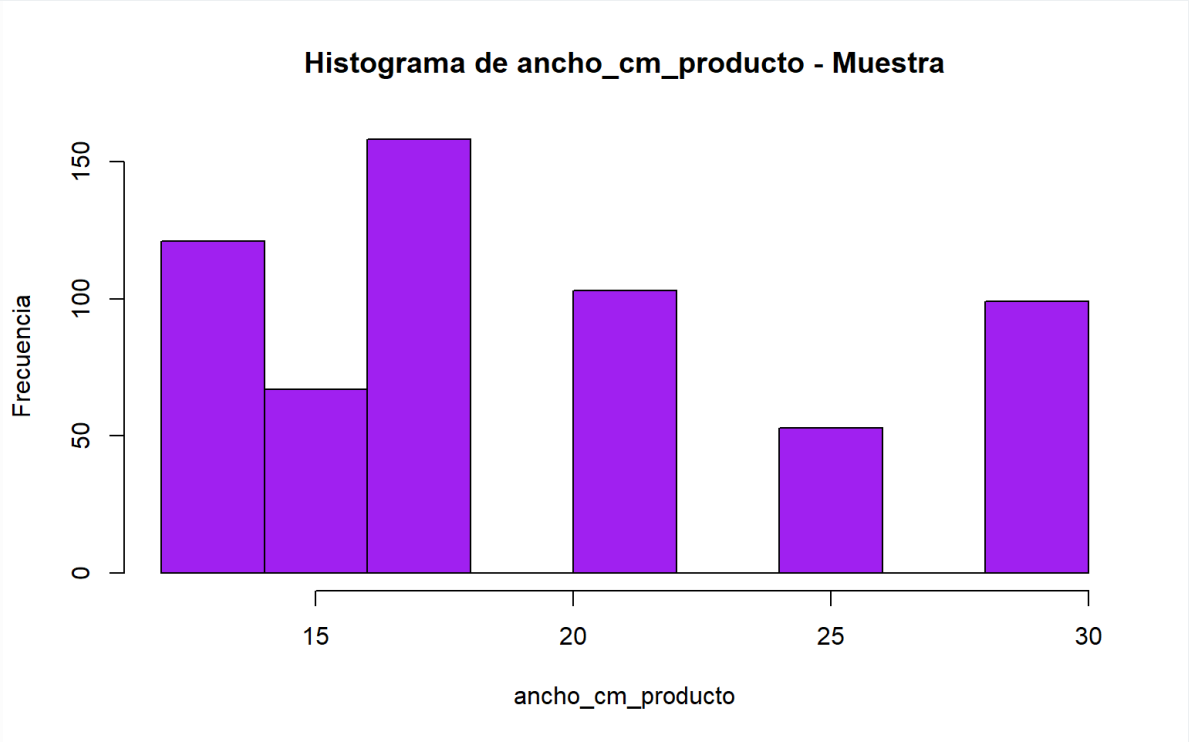

Ilustración 14: Histograma de ancho_cm_producto - Muestra.

Como se observa en las imágenes anteriores, las variables cuantitativas que presenta mayor simetría son:

- longitud_nombre_producto
- longitud_descripcion_producto
- cantidad_fotos_producto

Estas variables representan aspectos del producto que no son significativos, para observar características relevantes se toman gráficos en boxplot de las siguientes variables:

- precio
- valor flete
- altura_cm_producto ancho_cm_producto

```
# Boxplot precio, Valor Flete, altura_cm_producto y ancho_cm_producto - Población

boxplot(ordenesProductosRetiel$precio, ordenesProductosRetiel$valor_flete, ordenesProductosRetiel$altura_cm_producto, ordenesProductosRetiel$ancho_cm_producto)
```

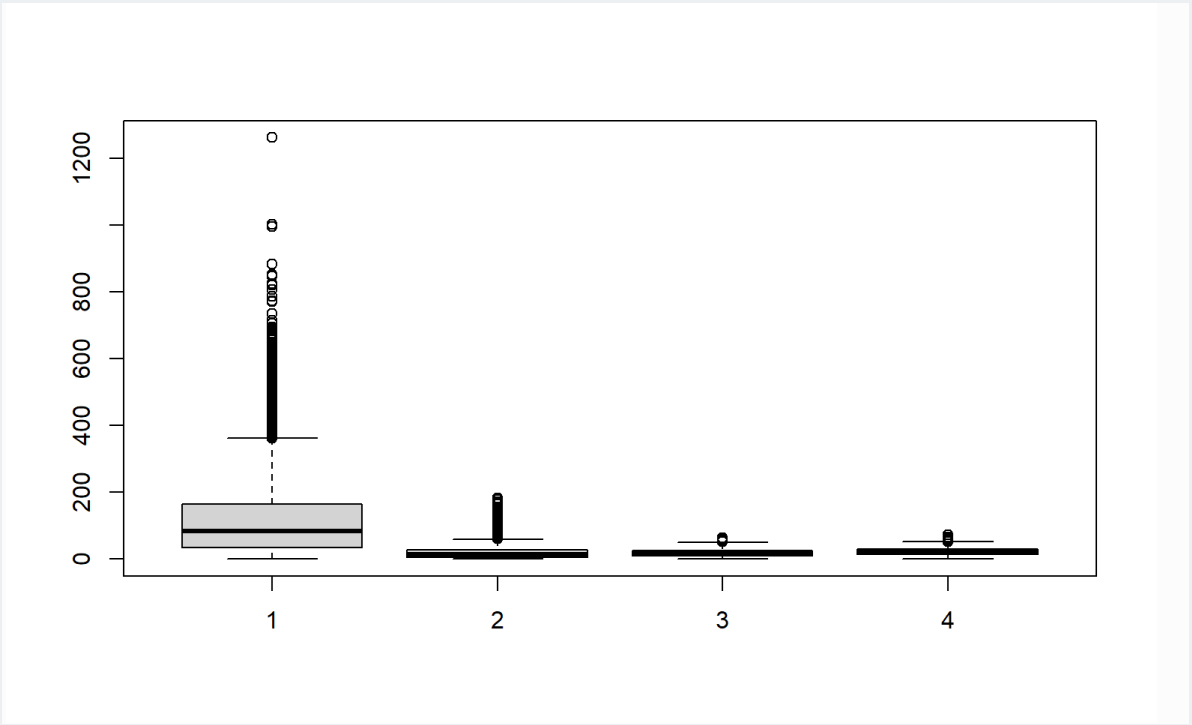

Ilustración 15: Boxplot precio, Valor Flete, altura_cm_producto y ancho_cm_producto - Población.

```
# Boxplot precio, Valor Flete, altura_cm_producto y ancho_cm_producto - muestra

boxplot(df00$precio, df00$valor_flete, df00$altura_cm_producto,
df00$ancho_cm_producto)
```

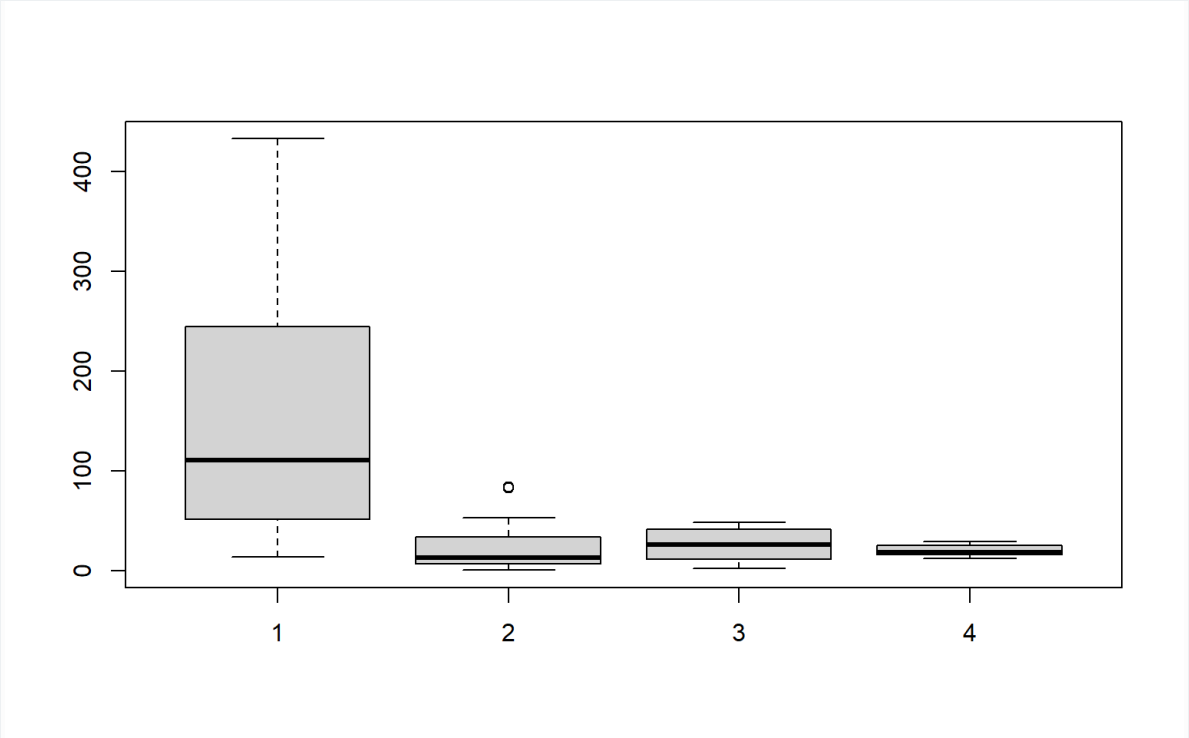

Ilustración 16: Boxplot precio, Valor Flete, altura_cm_producto y ancho_cm_producto - muestra.

Como se observa en las salidas, los boxplots de la población no se alcanzan a visualizar para determinar cuál variable es la más simétrica, pero al revisar las salidas de la muestra se observa que la tercera variable (altura_cm_producto) es la variable cuantitativa que representa mayor información del proyecto que tiende a la simetría. De esto podemos concluir que en el set de datos la altura de los productos tiende a ser simétrica.

Ejercicio N° 2 - Probabilidad

Ejercicio 1 1. Si lanzamos un dado equilibrado de 6 lados, ¿Cuál es la probabilidad esperada de obtener el número 5?. Si lanzamos el dado 10 veces, ¿Cuál es la cantidad esperada de veces que sale dicho número?

Solución del ejercicio: (CHAO, 1993)

R/ La probabilidad de cada una de las caras de un dado equilibrado corresponde a 1/6, esto significa que cada cara tiene la misma probabilidad de presentarse al momento de lanzar el dado, por tal razón, la probabilidad de obtener el numero 5 es del 1/6. Si se lanza el dado equilibrado 10 veces, la probabilidad esperada de obtener el número 5 sería la

multiplicación de la cantidad de veces que se lanza el dado por la probabilidad que se obtenga el número deseado: 10 * (1/6) = 1.67 Esto quiere decir que si lanzamos un dado equilibrado 10 veces se tiene la probabilidad de obtener el número 5, una o dos veces de los 10 lanzamientos realizados. Se aclara que el tema de probabilidad se utiliza para predecir resultados a largo plazo, por lo que si realizamos este experimento puede que no se logre la probabilidad estimada, esto por ser un evento con pocas repeticiones, estos eventos son difíciles de predecir y dependen de muchos factores que puedan afectar la prueba ejecutada.

Lanzar un dado 10 veces y contar el número de veces que sale 5. Repetirlo 15 veces y almacenar la cantidad de apariciones de dicho número para cada una.

R/ Como se comenta en el punto anterior, estos eventos se pueden calcular con el número de lanzamientos por la probabilidad de obtener cada cara: 10 * (1/6) = 1.67 Almacenar estos resultados y repetirlo 15 veces más se puede simular en la herramienta R de la siguiente forma:

```
# para obtener resultados reproducibles, se crea una semilla
set.seed(123)
num_lanzamientos <- 10
num_simulaciones <- 15
resultado <- numeric(num_simulaciones)
for (i in 1:num_simulaciones) {
  dados_Lanzados <- sample(1:6, num_lanzamientos, replace = TRUE)
  resultado[i] <- sum(dados_Lanzados == 5)
  cat("Simulación ", i, ": ", resultado[i], " veces el número 5.\n")
  cat("  Datos de los 10 lanzamientos", dados_Lanzados, "\n")
}
```

```
Simulación  1 :  1  veces el número 5.
  Datos de los 10 lanzamientos 3 6 3 2 2 6 3 5 4 6
Simulación  2 :  1  veces el número 5.
  Datos de los 10 lanzamientos 6 1 2 3 5 3 3 1 4 1
Simulación  3 :  1  veces el número 5.
  Datos de los 10 lanzamientos 1 5 3 2 2 1 6 3 4 6
Simulación  4 :  2  veces el número 5.
  Datos de los 10 lanzamientos 1 3 5 4 2 5 1 1 2 3
Simulación  5 :  4  veces el número 5.
  Datos de los 10 lanzamientos 4 5 5 3 6 1 2 5 5 4
Simulación  6 :  2  veces el número 5.
```

```
  Datos de los 10 lanzamientos 5 2 1 1 3 1 6 5 1 2
Simulación  7 :  0  veces el número 5.
  Datos de los 10 lanzamientos 4 4 6 6 3 6 6 1 6 2
Simulación  8 :  2  veces el número 5.
  Datos de los 10 lanzamientos 1 2 4 5 5 6 3 1 4 6
Simulación  9 :  0  veces el número 5.
  Datos de los 10 lanzamientos 1 6 1 3 6 4 1 6 6 3
Simulación  10 :  3  veces el número 5.
  Datos de los 10 lanzamientos 6 5 3 6 2 5 5 3 2 2
Simulación  11 :  0  veces el número 5.
  Datos de los 10 lanzamientos 2 4 2 2 6 4 4 6 1 6
Simulación  12 :  1  veces el número 5.
  Datos de los 10 lanzamientos 6 6 3 6 3 6 1 3 5 2
Simulación  13 :  2  veces el número 5.
  Datos de los 10 lanzamientos 6 6 3 2 5 6 5 3 4 4
Simulación  14 :  2  veces el número 5.
  Datos de los 10 lanzamientos 6 4 5 3 1 6 2 1 2 5
Simulación  15 :  0  veces el número 5.
  Datos de los 10 lanzamientos 3 4 4 1 6 6 4 1 6 3
```

Como se observa en la salida anterior, se tiene la simulación del lanzamiento de un dado equilibrado 10 veces, repetida durante 15 oportunidades, en la descripción de la salida se observa los resultados obtenidos y el conteo que se realiza dentro de la estructura cíclica para determinar la cantidad del número 5 obtenida. Del experimento realizado se puede contrastar que el número de veces que el número 5 aparece en la prueba es 21 veces, en R se obtendría de la siguiente forma:

```
sum(resultado)
```

```
## [1] 21
```

3. Lanzar el dado 100 veces, contar el número de apariciones de dicho número 5, almacenar el resultado y repetirlo 1000 veces.

R/ Con relación al ejercicio anterior se implementa el mismo código y se retira las salidas de cada simulación para obtener solo el resultado final:

```
# para obtener resultados reproducibles, se crea una semilla
set.seed(123)
num_lanzamientosEje23 <- 100
num_simulacionesEje23 <- 1000
resultadoEje23 <- numeric(num_simulacionesEje23)
```

```
for (i in 1:num_simulacionesEje23) {
  dados_LanzadosEje23 <- sample(1:6, num_lanzamientosEje23, replace = TRUE)
  resultadoEje23[i] <- sum(dados_LanzadosEje23 == 5)
}
sum(resultadoEje23)
```

```
## [1] 16774
```

Como se observa en la salida, la simulación muestra que se obtendrían 16774 veces el número 5, esto al lanzar un dado equilibrado 100 veces y repetir el experimento 1000 veces.

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

R/ Los resultados difieren de las cantidades de repeticiones que se realizan en el punto 2 (10 lanzamientos de un dado equilibrado, conteo de la cantidad de veces que aparece el número 5 y repetido 15 veces) y el punto 3 (100 lanzamientos, conteo del número 5 y repetidos 1000 veces). Estas diferencias afectan la precisión y la fiabilidad de los resultados, entre más lanzamientos y repeticiones haya más preciso y confiable será la prueba. En tal sentido, los resultados del punto 3 son más confiables y precisos por tener un mayor número de repeticiones.

Ejercicio 2

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

- Si sale un 7, te pagarán $ 60.
- Si sale un 6, te pagarán $ 30.
- Si sale cualquier otra combinación, deberás pagar $ 35

1. ¿Cuál es la probabilidad de sacar un siete?

R/La probabilidad de sacar un 7 es de 6/36 o 1/6, esto debido a que existen 6 formas de obtener un 7 a través de las siguientes combinaciones:

1. 1 y 6 = 7
2. 2 y 5 = 7

3. 3 y 4 = 7
4. 4 y 3 = 7
5. 5 y 2 = 7
6. 6 y 1 = 7

Como se observa para dos dados de seis lados, existen seis posibles combinaciones para obtener un 7 y por lo tanto la probabilidad de que ocurra este evento es 1/6.

2. ¿Cuál es la probabilidad de sacar un seis?

R/ De la misma forma se podría ver las posibilidades de obtener un número seis a través de las siguientes combinaciones:

1. 5 y 1 = 6
2. 1 y 5 = 6
3. 4 y 2 = 6
4. 2 y 4 = 6
5. 6 y 0 = 6
6. 0 y 6 = 6
7. 3 y 3 = 6

7/36 probabilidad de sacar un 6 con dos dados.

3. ¿Cuál es la probabilidad de sacar un siete o un seis?

R/ La probabilidad de obtener un 7 o 6 con los dos dados será la suma de las probabilidades obtenidas en los puntos anteriores: 6 (combinaciones de dos dados para obtener un 7) + 7 (combinaciones de dos dados para obtener un 6) = 13

13/36 es la probabilidad de sacar dos dados un 6 o un 7.

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.

R/ Para desarrollar el ejercicio en R se utiliza el siguiente código:

```
# para obtener resultados reproducibles, se crea una semilla
```

```r
set.seed(123)

# Se define la función Roll1Dice para simular un dado de seis caras

Roll1Dice <- function() {

  return(sample(1:6, 1, replace = TRUE))

}


funcionLazar2Dados <- function(nrow){

# Se inicializa una matriz de 100 filas y 2 columnas para almacenar los
resultados

results <- matrix(0, nrow, ncol = 2)


# Se realiza la simulación de lanzar 2 dados 100 veces

for (i in 1:nrow) {

  results[i, 1] <- Roll1Dice()

  results[i, 2] <- Roll1Dice()

}


# Se imprimen los resultados de lanzar dos dados 100 veces

return(print(results))

}


# Se llama la función para presentar los resultados del ejercicio 4:

ejer4 <- funcionLazar2Dados(100)
```

```
##       [,1] [,2]

##  [1,]    3    6

##  [2,]    3    2

##  [3,]    2    6

##  [4,]    3    5

##  [5,]    4    6
```

```
##  [6,]    6    1
##  [7,]    2    3
##  [8,]    5    3
##  [9,]    3    1
## [10,]    4    1
## [11,]    1    5
## [12,]    3    2
## [13,]    2    1
## [14,]    6    3
## [15,]    4    6
## [16,]    1    3
## [17,]    5    4
## [18,]    2    5
## [19,]    1    1
## [20,]    2    3
## [21,]    4    5
## [22,]    5    3
## [23,]    6    1
## [24,]    2    5
## [25,]    5    4
## [26,]    5    2
## [27,]    1    1
```

```
##  [28,]    3    1
##  [29,]    6    5
##  [30,]    1    2
##  [31,]    4    4
##  [32,]    6    6
##  [33,]    3    6
##  [34,]    6    1
##  [35,]    6    2
##  [36,]    1    2
##  [37,]    4    5
##  [38,]    5    6
##  [39,]    3    1
##  [40,]    4    6
##  [41,]    1    6
##  [42,]    1    3
##  [43,]    6    4
##  [44,]    1    6
##  [45,]    6    3
##  [46,]    6    5
##  [47,]    3    6
##  [48,]    2    5
##  [49,]    5    3
```

```
##  [50,]    2    2
##  [51,]    2    4
##  [52,]    2    2
##  [53,]    6    4
##  [54,]    4    6
##  [55,]    1    6
##  [56,]    6    6
##  [57,]    3    6
##  [58,]    3    6
##  [59,]    1    3
##  [60,]    5    2
##  [61,]    6    6
##  [62,]    3    2
##  [63,]    5    6
##  [64,]    5    3
##  [65,]    4    4
##  [66,]    6    4
##  [67,]    5    3
##  [68,]    1    6
##  [69,]    2    1
##  [70,]    2    5
##  [71,]    3    4
```

```
##  [72,]    4    1
##  [73,]    6    6
##  [74,]    4    1
##  [75,]    6    3
##  [76,]    4    3
##  [77,]    5    4
##  [78,]    4    4
##  [79,]    6    1
##  [80,]    2    3
##  [81,]    4    3
##  [82,]    1    6
##  [83,]    5    5
##  [84,]    2    3
##  [85,]    5    6
##  [86,]    6    1
##  [87,]    4    2
##  [88,]    4    5
##  [89,]    5    6
##  [90,]    5    5
##  [91,]    1    2
##  [92,]    1    2
##  [93,]    5    5
```

```
##  [94,]    1    2
##  [95,]    5    4
##  [96,]    2    6
##  [97,]    2    3
##  [98,]    1    1
##  [99,]    5    5
## [100,]    3    2
```

```
# Se utiliza la función para simular los datos del ejercicio 3


resultadoejer3<-0
for (i in 1:100) {
  # Se valida si la suma de cada una de las filas es igual a 6 o 7, y se guarda en la variable resultadoejer3
  resultadoejer3[i] <- sum(ejer4[i,])==6 | sum(ejer4[i,])==7


}
sum(resultadoejer3)
```

```
## [1] 22
```

Se observa que al validar la matriz construida de los resultados se obtiene de las 100 muestras 22 pruebas que cumplen la condición de sumar 6 o 7, esto en probabilidad representa 22% de las pruebas (22/100), lo obtenido en el ejercicio 3 mostraba que la probabilidad de obtener un 6 o 7 con dos dados era 36,11% (13/36). Se esperaría que si se aumenta el número de muestras se acerque a esta probabilidad.

```
# para obtener resultados reproducibles, se crea una semilla
set.seed(123)
```

```r
# Se utiliza la función para simular los datos del ejercicio 3 con n=10000
ejer3_10mil <- funcionLazar2Dados(10000)
```

```
##          [,1] [,2]
##     [1,]    3    6
##     [2,]    3    2
##     [3,]    2    6
##     [4,]    3    5
##     [5,]    4    6
##     [6,]    6    1
##     [7,]    2    3
##     [8,]    5    3
##     [x,]    .    .
##    [xx,]    .    .
##    [xx,]    .    .
##   [xxx,]    .    .
##   [xxx,]    .    .
##   [xxx,]    .    .
##   [xxxx,]    .    .
##   [xxxx,]    .    .
##   [xxxx,]    .    .
##  [9989,]    2    4
##  [9990,]    5    6
```

```
##  [9991,]    6    5
##  [9992,]    6    1
##  [9993,]    6    4
##  [9994,]    1    6
##  [9995,]    1    6
##  [9996,]    2    5
##  [9997,]    6    6
##  [9998,]    4    4
##  [9999,]    3    5
## [10000,]    5    4
```

```
resultadoejer3_10mil<-0
for (i in 1:10000) {
  # Se valida si la suma de cada una de las filas es igual a 6 o 7, y se guarda en la variable resultadoejer3_10mil
  resultadoejer3_10mil[i] <- sum(ejer3_10mil[i,])==6 | sum(ejer3_10mil[i,])==7

}
probabilidadEjer3_10mil <- sum(resultadoejer3_10mil)/10000
probabilidadEjer3_10mil
```

```
## [1] 0.3
```

Como se observa la probabilidad se encuentra al rededor del 30% mostrando que se acerca al valor estimado inicialmente.

Para el caso del ejercicio 2 se tiene lo siguiente:

```
resultadoejer3_10mil<-0

for (i in 1:10000) {

  # Se valida si la suma de cada una de las filas es igual a 6, y se guarda en
la variable resultadoejer3_10mil

  resultadoejer3_10mil[i] <- sum(ejer3_10mil[i,])==6


}

probabilidadEjer3_10mil <- sum(resultadoejer3_10mil)/10000

probabilidadEjer3_10mil
```

```
## [1] 0.1405
```

Con relación al ejercicio 2 (7/36) = 0,1944, se obtiene de la simulación 0,1405.

Para el caso del ejercicio 1 se tiene:

```
resultadoejer3_10mil<-0

for (i in 1:10000) {

  # Se valida si la suma de cada una de las filas es igual a 6, y se guarda en
la variable resultadoejer3_10mil

  resultadoejer3_10mil[i] <- sum(ejer3_10mil[i,])==7


}

probabilidadEjer3_10mil <- sum(resultadoejer3_10mil)/10000

probabilidadEjer3_10mil
```

```
## [1] 0.1595
```

Con relación al ejercicio 1 (1/36) = 0,1666, se obtiene de la simulación 0,1595.

5. Suponga que jugaron 10 veces y obtuvieron una ganancia de $ 300. ¡Qué fácil parece ser el juego! ¡Debería seguir jugando! ¿Es correcta la suposición? Demostrar con una simulación.

```r
# Simulación de jugar el juego de lanzar dos dados 10 veces
# para obtener resultados reproducibles, se crea una semilla
set.seed(123)
# Se utiliza la función para simular los datos con n=10
ejer5 <- funcionLazar2Dados(10)
```

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

```r
resultadoejer56<-0
resultadoejer57<-0
resultadoejer5dif<-0
for (i in 1:10) {
  # Se valida si la suma de cada una de las filas es igual a 6, y se guarda en la variable resultadoejer56
  resultadoejer56[i] <- sum(ejer5[i,])==6
  # Se valida si la suma de cada una de las filas es igual a 6, y se guarda en la variable resultadoejer57
```

```
  resultadoejer57[i] <- sum(ejer5[i,])==7

   # Se valida si la suma de cada una de las filas es diferente a 6 o 7, y se
guarda en la variable resultadoejer5dif

  resultadoejer5dif[i] <- sum(ejer5[i,])!=6 | sum(ejer5[i,])!=7


}

pagoNum6 <- sum(resultadoejer56)*60

pagoNum7 <- sum(resultadoejer57)*30

menosNumDif67 <- sum(resultadoejer5dif)*35


  # Se calcula la ganancia del juego

  totalJuego <- pagoNum6+pagoNum7-menosNumDif67


  totalJuego
```

```
## [1] -320
```

La suposición es incorrecta, la simulación muestra que con la semilla presentada los resultados generan una pérdida de $ 320. No sería prudente seguir jugando.

6. Ahora dicha persona te ofrece disminuir el monto a pagar a $ 20. ¿Deberías aceptarlo?

R/ Se procede a realizar una simulación para determinar la conveniencia:

```
# Simulación de jugar el juego de lanzar dos dados 10 veces

# para obtener resultados reproducibles, se crea una semilla

set.seed(123)

# Se utiliza la función para simular los datos con n=10

ejer5 <- funcionLazar2Dados(10)
```

```
##       [,1] [,2]

##  [1,]    3    6
```

```
##  [2,]    3    2

##  [3,]    2    6

##  [4,]    3    5

##  [5,]    4    6

##  [6,]    6    1

##  [7,]    2    3

##  [8,]    5    3

##  [9,]    3    1

## [10,]    4    1
```

```
resultadoejer56<-0

resultadoejer57<-0

resultadoejer5dif<-0

for (i in 1:10) {

  # Se valida si la suma de cada una de las filas es igual a 6, y se guarda en
la variable resultadoejer56

  resultadoejer56[i] <- sum(ejer5[i,])==6

  # Se valida si la suma de cada una de las filas es igual a 6, y se guarda en
la variable resultadoejer57

  resultadoejer57[i] <- sum(ejer5[i,])==7

   # Se valida si la suma de cada una de las filas es diferente a 6 o 7, y se
guarda en la variable resultadoejer5dif

  resultadoejer5dif[i] <- sum(ejer5[i,])!=6 | sum(ejer5[i,])!=7


}

pagoNum6 <- sum(resultadoejer56)*60

pagoNum7 <- sum(resultadoejer57)*30

menosNumDif67 <- sum(resultadoejer5dif)*20
```

```
  # Se calcula la ganancia del juego
  totalJuego <- pagoNum6+pagoNum7-menosNumDif67


  totalJuego
```

```
## [1] -170
```

Se observa que, aunque se disminuya el valor que hay q pagar con la perdida se seguiría perdiendo en el juego, en esta ocasión por $170.

Ejercicio N° 3 - Variables Aleatorias

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?

Solución del ejercicio: (Durand, 2008)

```
# Si se toma representativamente la variable normal como n=10 y n=100 se puede
realizar la suma de la siguiente forma


#n=10
n <- 10
x <- rnorm(n, mean = 0, sd = 1)
y <- rnorm(n, mean = 0, sd = 1)
z <- x + y


#n=100
n2 <- 100
x2 <- rnorm(n2, mean = 0, sd = 1)
y2 <- rnorm(n2, mean = 0, sd = 1)
z2 <- x2 + y2
```

Para graficar la simulación anterior se utiliza el siguiente código:

```
#Grafica para n=10
par(mfrow = c(2,2))
hist(x, main = "Histograma de X (n = 10)")
hist(y, main = "Histograma de Y (n = 10)")
hist(z, main = "Histograma de X + Y (n = 10)")
```

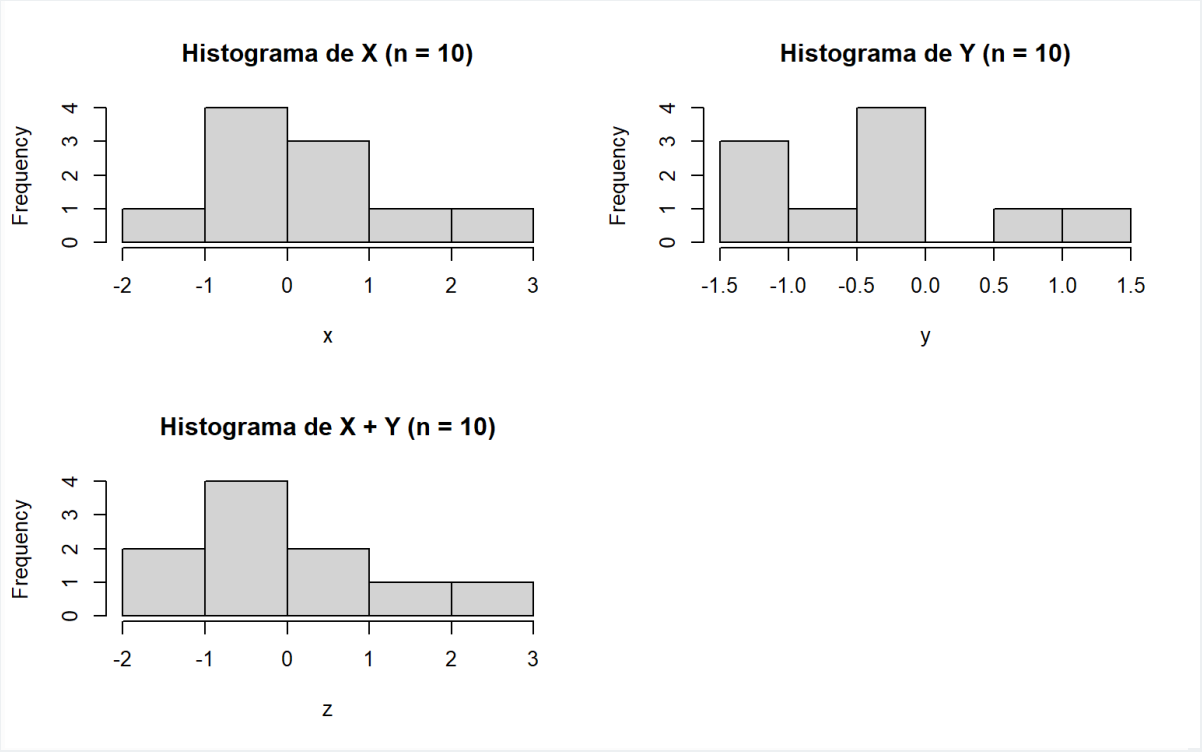

Ilustración 17: Histograma de X, Y y X+Y (n = 10).

```
#Grafica para n=100
par(mfrow = c(2,2))
hist(x2, main = "Histograma de X2 (n = 100)")
hist(y2, main = "Histograma de Y2 (n = 100)")
hist(z2, main = "Histograma de X2 + Y2 (n = 100)")
```

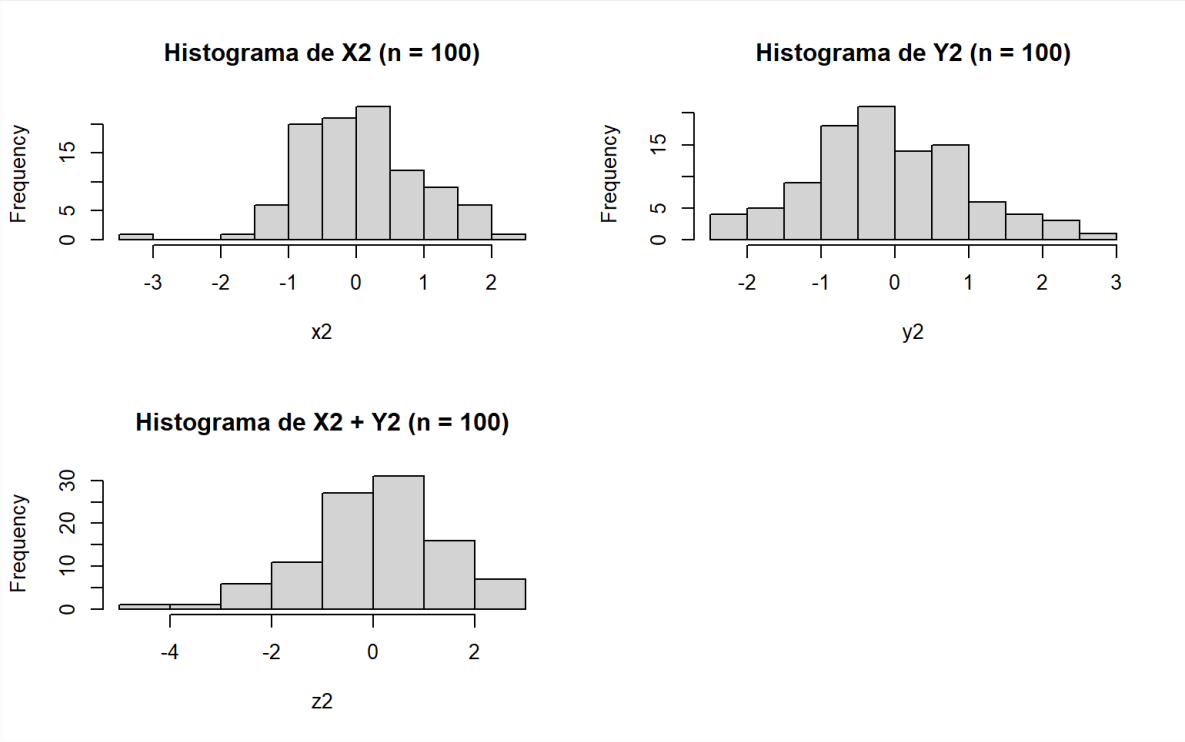

Ilustración 18: Histograma de X, Y y X+Y (n = 100).

¿Qué observa si grafica ambos objetos? R/ Se observa en las gráficas anteriores que las variables normales graficadas individualmente tienden a mostrar una curva normal, al sumarlas sigue esta misma tendencia. Al aumentar las muestras se observa que el parecido a una curva normal perfecta se va observando más definida, si se aumentara el valor de n se vería este aspecto. A continuación, se presenta la simulación con n=10000:

```
#n=10000
n3 <- 10000
x3 <- rnorm(n3, mean = 0, sd = 1)
y3 <- rnorm(n3, mean = 0, sd = 1)
z3 <- x3 + y3


#Grafica para n=10000
par(mfrow = c(2,2))
hist(x3, main = "Histograma de X3 (n = 10000)")
hist(y3, main = "Histograma de Y3 (n = 10000)")
hist(z3, main = "Histograma de X3 + Y3 (n = 10000)")
```

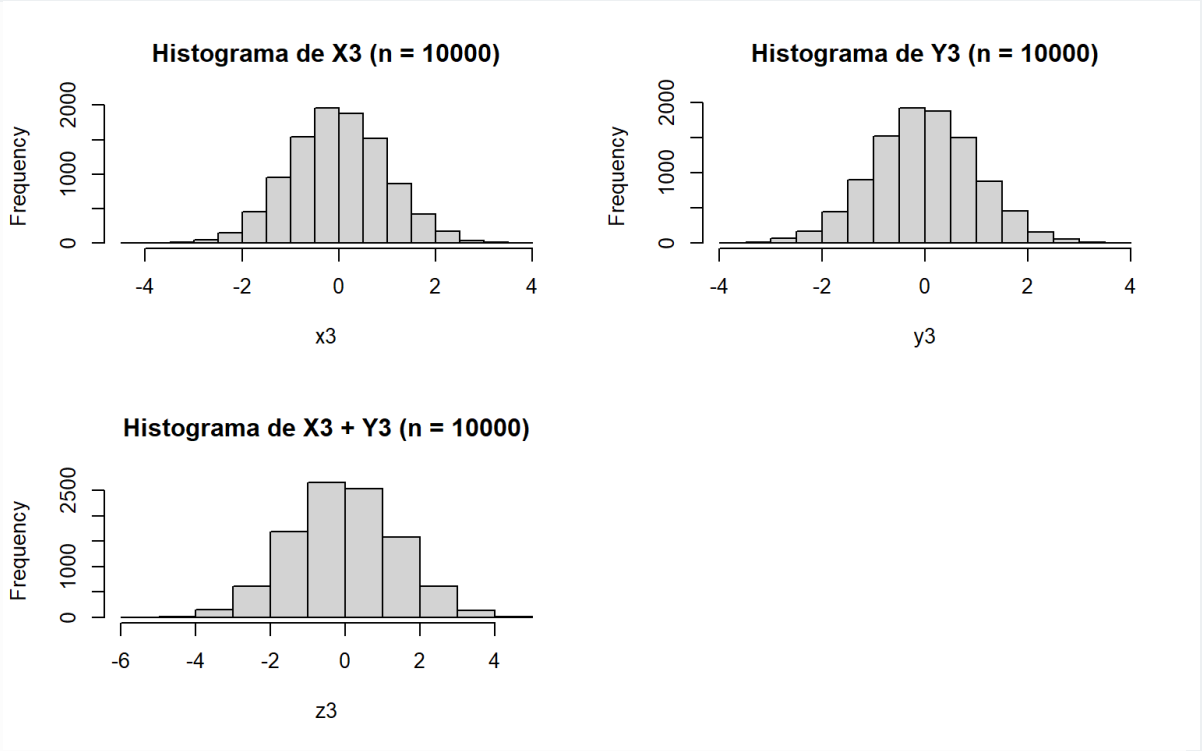

Ilustración 19: Histograma de X, Y y X+Y (n = 10000).

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. Graficar 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.

```
#Se presenta Código de la simulación a través de un bucle
nEjer32 <- c(100, 1000, 10000, 100000)
xEjer32 <- list()
yEjer32 <- list()
zEjer32 <- list()
for (i in 1:length(nEjer32)) {
  xEjer32[[i]] <- rnorm(nEjer32[i], mean = 0, sd = 1)
  yEjer32[[i]] <- rnorm(nEjer32[i], mean = 0, sd = 1)
  zEjer32[[i]] <- xEjer32[[i]] + yEjer32[[i]]
}
```

```r
#grafica
par(mfrow = c(3,3))
for (i in 1:length(nEjer32)) {
  hist(xEjer32[[i]], main = paste("Histograma de X (n =", nEjer32[i], ")"))
  hist(yEjer32[[i]], main = paste("Histograma de Y (n =", nEjer32[i], ")"))
  hist(zEjer32[[i]], main = paste("Histograma de X + Y (n =", nEjer32[i], ")"))
}
```

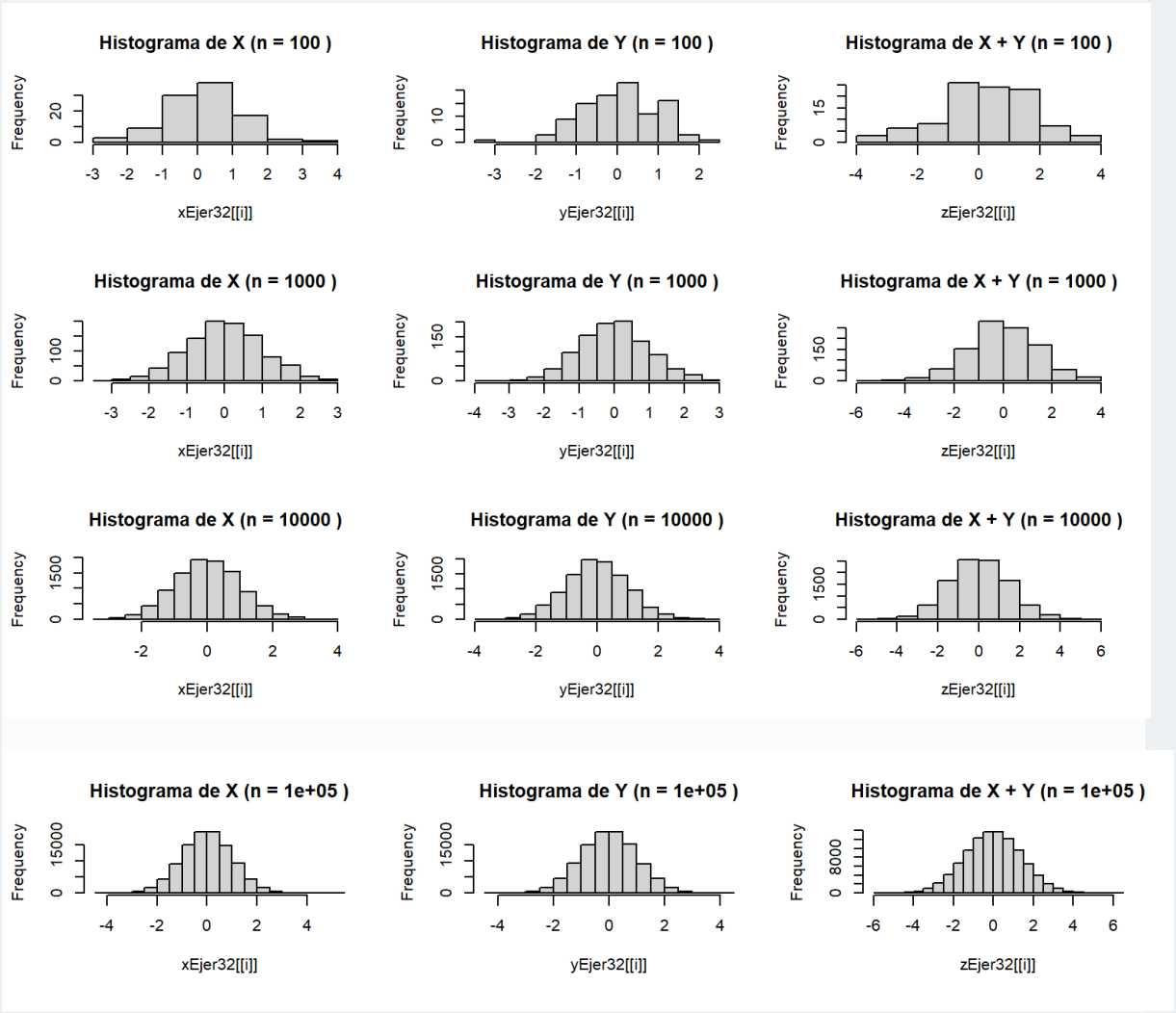

Ilustración 20: Histograma de X, Y y X+Y (n = 100, 1000, 10000, 100000).

Como se observó en el ejercicio anterior, al graficar cada variable normal se observa que tiende a representar una gráfica normal, al sumarla genera otra grafica normal, para este caso se entiende que la media seria (0+0=0) y su desviación estándar (raíz cuadra de 1 + raíz

cuadrada de 1 = raíz cuadrada de 2). Además, se puede comprobar que la media y la desviación corresponden a la suma de las sumas individuales de cada muestra de la siguiente forma:

```
for (i in 1:length(nEjer32)) {
  print(paste("Muestra con tamaño", nEjer32[i]))
  print(paste("Media X =", mean(xEjer32[[i]])))
  print(paste("Media Y =", mean(yEjer32[[i]])))
  print(paste("Media X + Y =", mean(zEjer32[[i]])))
  print(paste("Varianza X =", var(xEjer32[[i]])))
  print(paste("Varianza Y =", var(yEjer32[[i]])))
  print(paste("Varianza X + Y =", var(zEjer32[[i]])))
  print("")
}
```

```
## [1] "Muestra con tamaño 100"

## [1] "Media X = 0.14542589128843"

## [1] "Media Y = 0.0537392635221665"

## [1] "Media X + Y = 0.199165154810597"

## [1] "Varianza X = 1.08952317211717"

## [1] "Varianza Y = 0.892768885881121"

## [1] "Varianza X + Y = 2.2905888605101"

## [1] ""

## [1] "Muestra con tamaño 1000"

## [1] "Media X = 0.00194631974077503"

## [1] "Media Y = -0.00737847007053988"

## [1] "Media X + Y = -0.00543215032976486"
```

```
## [1] "Varianza X = 0.962745805726622"

## [1] "Varianza Y = 0.957444205069351"

## [1] "Varianza X + Y = 1.89140028201185"

## [1] ""

## [1] "Muestra con tamaño 10000"

## [1] "Media X = 0.0125326731394645"

## [1] "Media Y = 0.001861226873392"

## [1] "Media X + Y = 0.0143939000128565"

## [1] "Varianza X = 0.998344501557535"

## [1] "Varianza Y = 1.01344493642016"

## [1] "Varianza X + Y = 2.02615392989628"

## [1] ""

## [1] "Muestra con tamaño 1e+05"

## [1] "Media X = -0.000825942757150163"

## [1] "Media Y = -0.00485100729110009"

## [1] "Media X + Y = -0.00567695004825026"

## [1] "Varianza X = 1.00455269196256"

## [1] "Varianza Y = 0.998918754481666"

## [1] "Varianza X + Y = 2.00296545478529"

## [1] ""
```

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. Graficar 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.

```r
# Function to generate n random normal variables and sum them
sum_normals <- function(n){
  return(rowSums(matrix(rnorm(n * 10, mean = 0, sd = 1), ncol = 10)))
}

# Plot the distribution for different sample sizes
par(mfrow = c(2, 2))
hist(sum_normals(100), main = "n = 100")
hist(sum_normals(1000), main = "n = 1000")
hist(sum_normals(10000), main = "n = 10000")
hist(sum_normals(100000), main = "n = 100000")
```

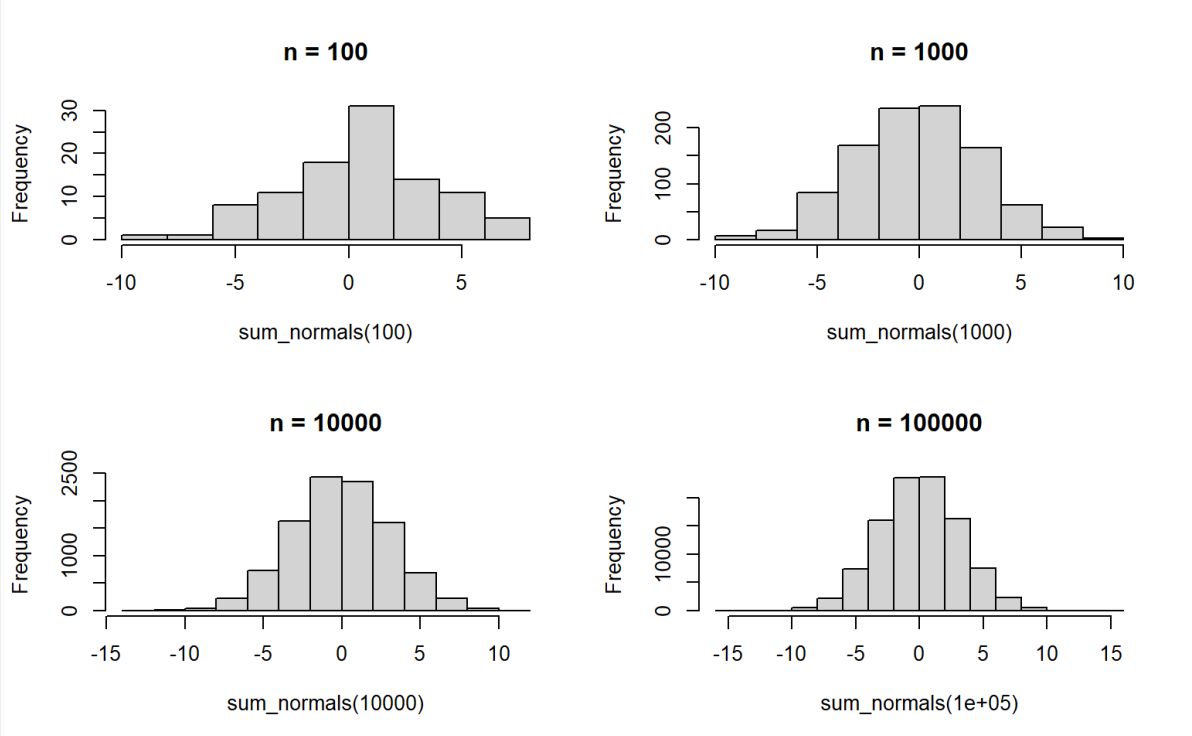

Ilustración 21: Plot the distribution for different sample sizes.

Para verificar 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, se puede desarrollar de la siguiente forma:

```
# Verify that the mean is equal to the sum of individual means
n <- 100000
mean(sum_normals(n))
```

```
## [1] 0.002698117
```

```
# Output: -0.0108


# Verify that the variance is equal to the sum of individual variances
var(sum_normals(n))
```

```
## [1] 9.933176
```

```
# Output: 10.0297
```

Como se observa en la salida, la suma de medias igual 0 resultará en una media igual a cero (0+0+0+…+0=0) y su varianza (1 + 1 + … + 1= 10).

4. Extra: Probar lo anterior, pero sumando normales con distintas medias y desvíos.

```
# Function to generate n random normal variables with different means and
standard deviations
sum_diff_normals <- function(n){
  means <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
  stds <- c(0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5)
  normals <- matrix(NA, n, 10)


  for (i in 1:10){
```

```
    normals[, i] <- rnorm(n, mean = means[i], sd = stds[i])
  }

  return(rowSums(normals))
}

# Plot the distribution for a given sample size
n <- 100000
hist(sum_diff_normals(n), main = "Sum of different normal variables")
```

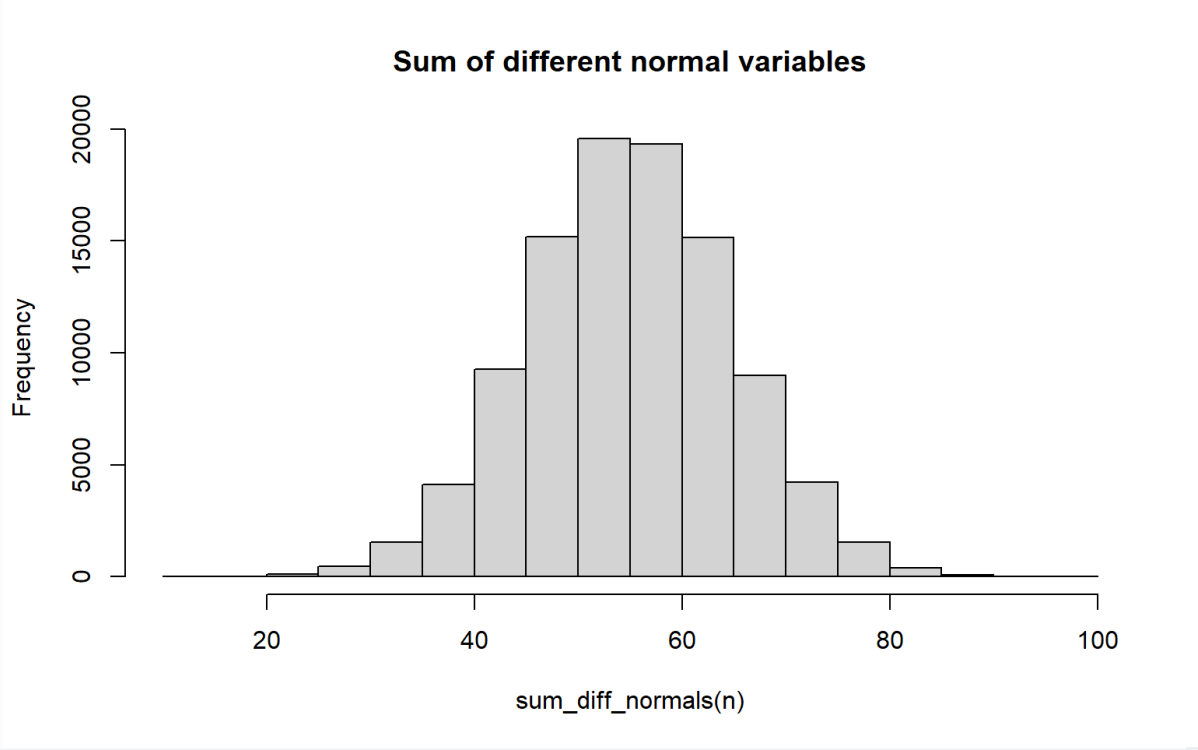

Ilustración 22: Sum of different normal variables.

Como se ha visto en los ejercicios anteriores la suma de medias y varianzas de forma individual generan un resultado que sigue siendo normal con media y varianza igual a las sumas individuales.

```
# Verify that the mean is equal to the sum of individual means
mean(sum_diff_normals(n))
```

```
## [1] 54.9839
```

```
# Verify that the variance is equal to the sum of individual variances
var(sum_diff_normals(n))
```

```
## [1] 96.76338
```

Ejercicio N° 4 - Teorema Central del Límite

En clase hemos visto que la media de variables Normales es una Normal. Ahora bien, ¿Ocurrirá lo mismo si las variables que se promedian no son normales? Se plantea el siguiente ejercicio para que intenten resolver, de forma tal que descubran al Teorema Central del Límite. Repetir el proceso visto en clase mediante R cuando la variable aleatoria original se distribuye de la forma siguiente, 1. Poisson de parámetro λ = 1.5 2. Exponencial de parámetro μ = 2.5 3. Uniforme en el intervalo [10,30] 4. Weibull de parámetros shape = 2 y scale = 1. Realizar un Gráfico de Histograma y plotear la densidad de una variable normal para cada caso.

Solución del ejercicio: (CHAO, 1993)

```
# Function to generate n random variables with a Poisson distribution with
parameter lambda = 1.5
mean_poisson <- function(n){
  return(mean(rpois(n, lambda = 1.5)))
}


# Plot the distribution of the mean of Poisson variables
n <- 100000
hist(mean_poisson(n), main = "Mean of Poisson variables", xlab = "Value", ylab =
"Frequency")
curve(dnorm(x, mean = 1.5, sd = sqrt(1.5/n)), col = "red", add = TRUE)
```

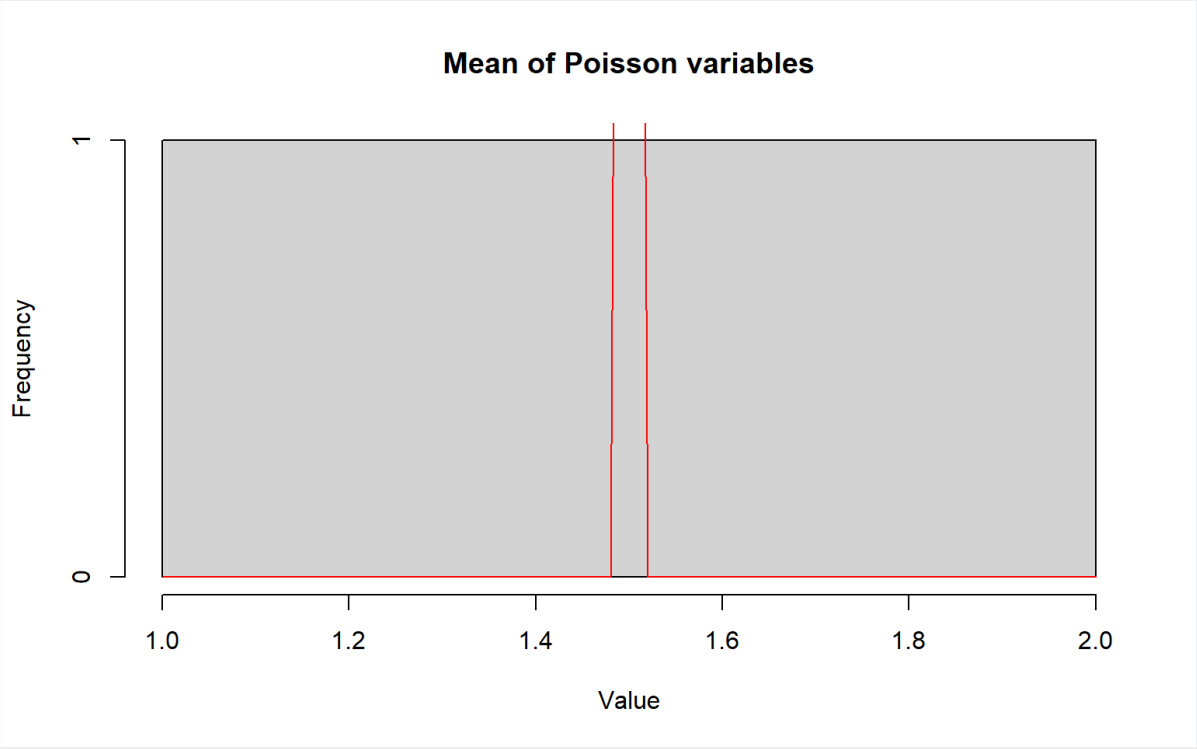

Ilustración 23: Mean of Poisson variables.

```r
# Function to generate n random variables with an Exponential distribution with
parameter mu = 2.5

mean_exponential <- function(n){

  return(mean(rexp(n, rate = 1/2.5)))

}


# Plot the distribution of the mean of Exponential variables

n <- 100000

hist(mean_exponential(n), main = "Mean of Exponential variables", xlab =
"Value", ylab = "Frequency")

curve(dnorm(x, mean = 2.5, sd = sqrt(2.5/n)), col = "red", add = TRUE)
```

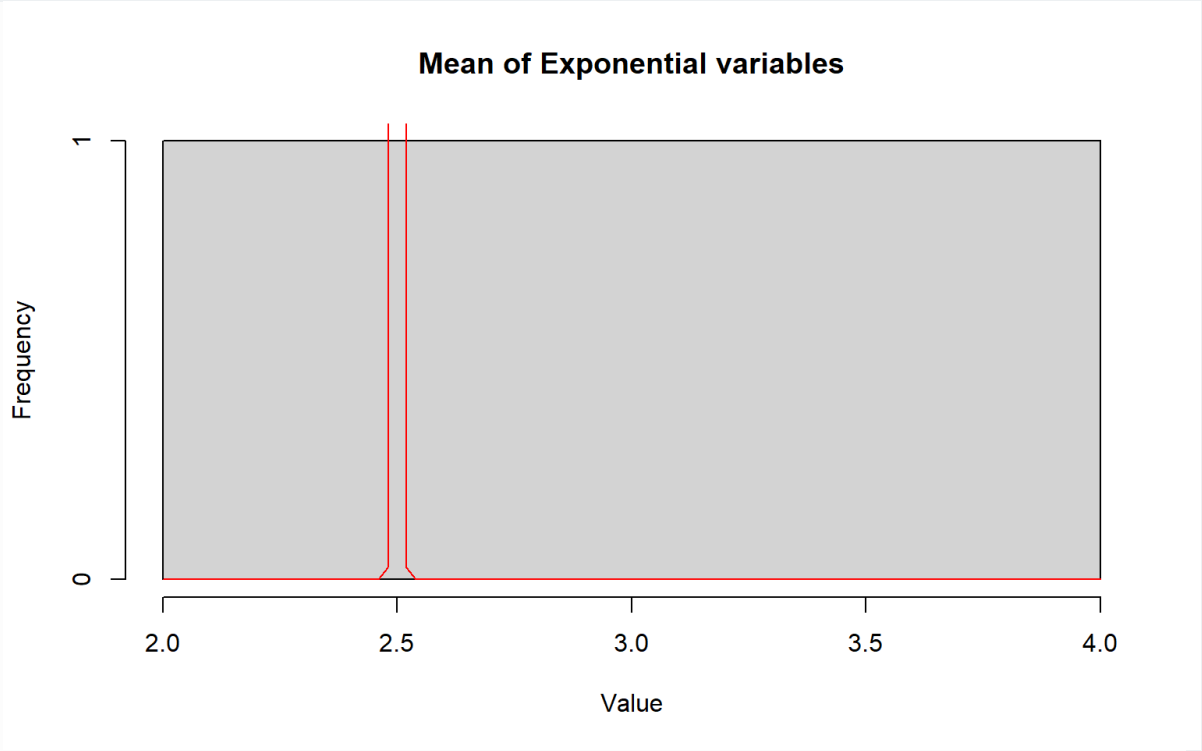

Ilustración 24: Plot the distribution of the mean of Exponential variables.

```r
# Function to generate n random variables with a Uniform distribution between 10 and 30

mean_uniform <- function(n){

  return(mean(runif(n, min = 10, max = 30)))

}


# Plot the distribution of the mean of Uniform variables

n <- 100000

hist(mean_uniform(n), main = "Mean of Uniform variables", xlab = "Value", ylab = "Frequency")

curve(dnorm(x, mean = 20, sd = sqrt(60/12/n)), col = "red", add = TRUE)
```

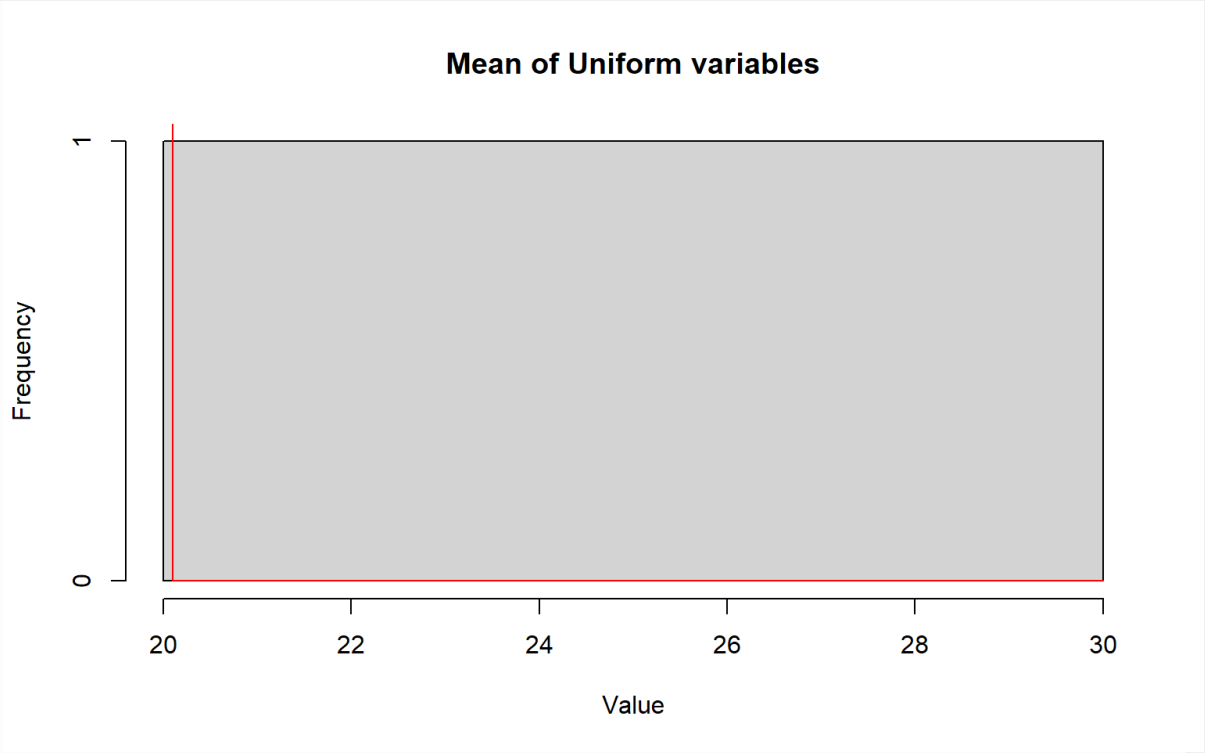

Ilustración 25: Plot the distribution of the mean of Uniform variables.

```r
# Function to generate n random variables with a Weibull distribution with
parameters shape = 2 and scale = 1

mean_weibull <- function(n){

  return(mean(rweibull(n, shape = 2, scale = 1)))

}


# Plot the distribution of the mean of Weibull variables

n <- 100000

hist(mean_weibull(n), main = "Mean of Weibull variables", xlab = "Value", ylab =
"Frequency")

curve(dnorm(x, mean = sqrt(pi/2), sd = sqrt((pi - 2)/2/n)), col = "red", add =
TRUE)
```

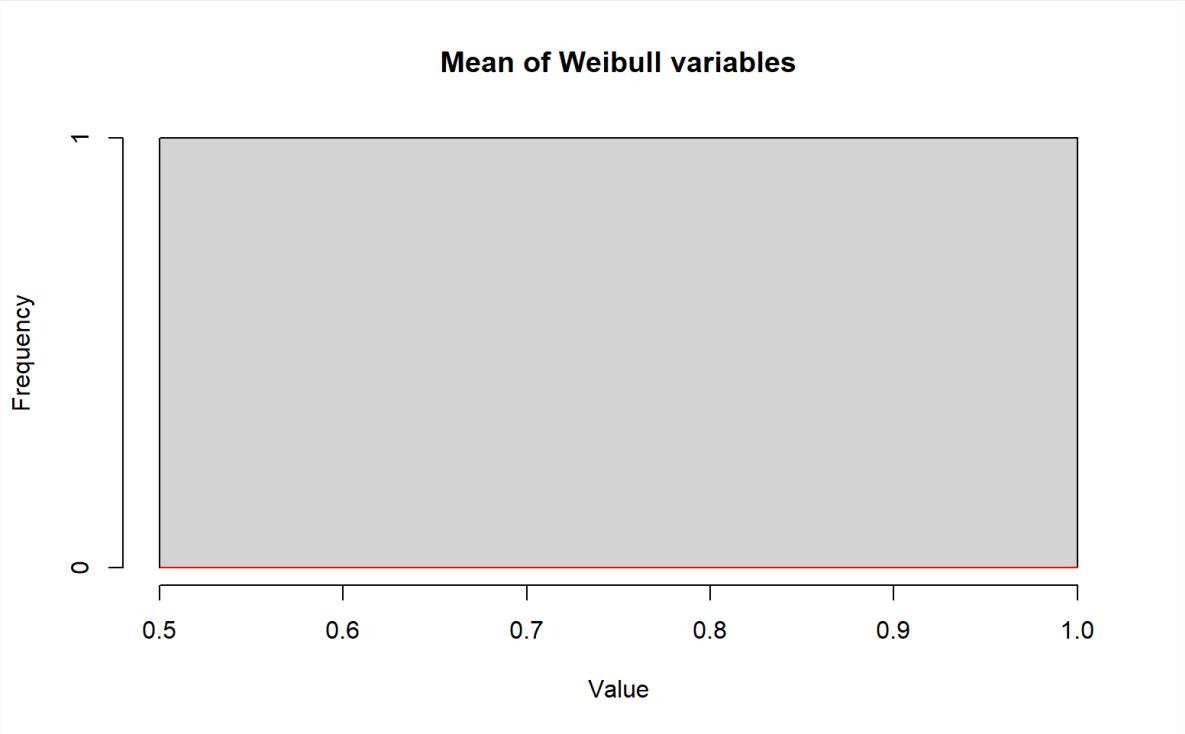

Ilustración 26: Plot the distribution of the mean of Weibull variables.

Como puedes ver en los gráficos, la distribución de la media de variables aleatorias no necesariamente sigue una distribución normal. Sin embargo, según el Teorema Central del Límite, si la cantidad de variables aleatorias aumenta (es decir, si n se hace grande), la distribución de la media de estas variables se aproxima cada vez más a una distribución normal con media igual a la media de las variables aleatorias y desviación estándar igual a la desviación.

Ejercicio N° 5 - IC y Prueba de Hipótesis

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

```
url =
"https://github.com/hllinas/DatosPublicos/blob/main/Estudiantes.Rdata?raw=false"

repmis::source_data(url)
```

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

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

```
## 6bf9d5a19779293538bd61d55d0662bdaf8100a1

## [1] "Estudiantes"
```

```
datos <- Estudiantes
```

Realizar los siguientes ejercicios. Interprete todas sus respuestas. (David M. Levine, 2023)

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.

```
# Se seleccionan las observaciones desde la 2 hasta la 35
datos2a35 <- datos[2:35, ]


# Se Verifica el tamaño del data frame
dim(datos2a35)
```

```
## [1] 34 46
```

```
# Se Verifica las variables y estructura
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" "SB11201910013623" ...

##  $ 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
desacuerdo" ...

##  $ PandemiaCat : chr  "De acuerdo" "De acuerdo" "De acuerdo" "Ni de acuerdo,
ni en desacuerdo" ...

##  $ 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" "Totalmente de acuerdo" ...

##  $ AGPEQ2      : chr  "En desacuerdo" "En desacuerdo" "Ni de acuerdo, ni en
desacuerdo" "Ni de acuerdo, ni en desacuerdo" ...

##  $ AGPEQ3      : chr  "En desacuerdo" "De acuerdo" "Totalmente de acuerdo"
"Totalmente en desacuerdo" ...

##  $ SATS1       : chr  "En desacuerdo" "En desacuerdo" "Totalmente en
desacuerdo" "Indeciso" ...

##  $ 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"
"Indeciso" ...

##  $ 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 veces" ...

##  $ IDARE2.7    : chr  "Algunas veces" "Algunas veces" "Frecuentemente"
"Algunas veces" ...

##  $ IDARE2.8    : chr  "Algunas veces" "Algunas veces" "Frecuentemente"
"Frecuentemente" ...

##  $ 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 ...
```

Todos 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.

```
# Se define el objeto "Sexo"
datos2a35$Sexo <- as.factor(datos2a35$Sexo)


# Se Verifica los niveles del objeto "Sexo"
levels(datos2a35$Sexo)
```

```
## [1] "Femenino"  "Masculino"
```

Como se observa en la salida str(datos2a35) la variable sexo era de tipo carácter y para atender la solicitud se convierte en factor y se vuelve asignar al data frame utilizado.

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

```
# Se Construye una tabla de frecuencias
table(datos2a35$Sexo)
```

```
##

##  Femenino Masculino

##        20        14
```

```
# Se Construye un diagrama de barras
```

```
barplot(table(datos2a35$Sexo), xlab="Sexo", ylab="Frecuencia", main="Diagrama de Barras para la Variable Sexo")
```

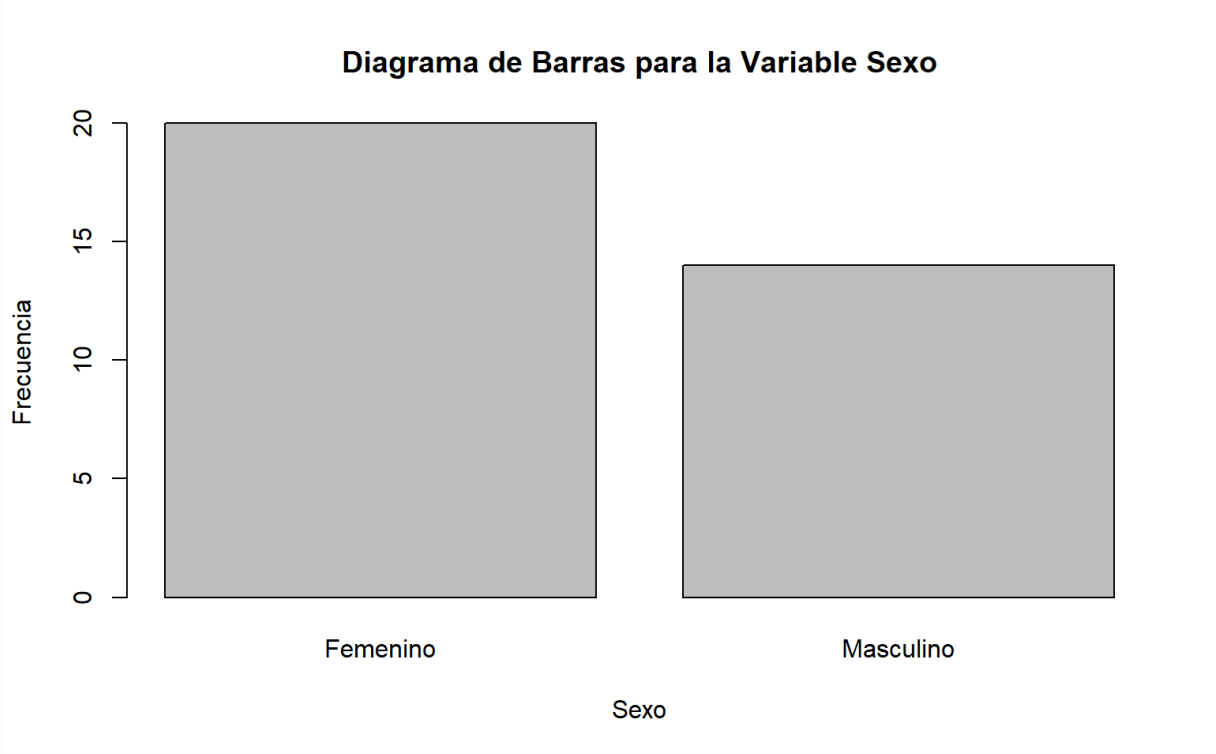

Ilustración 27: Diagrama de Barras para la Variable Sexo.

Como se observa en la salida existen un mayor número de estudias categorizados con el sexo femenino (20) que los categorizados como masculinos (14).

d) Determinar la proporción de mujeres.

```
prop_mujeres <- sum(datos2a35$Sexo == "Femenino")/length(datos2a35$Sexo)
```

Se observa que la proporción de mujeres corresponde al 58.82%.

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.

R/ Resumen del enunciado del problema: Se quiere determinar si el porcentaje poblacional de mujeres es menor o igual que el 30% al nivel del 5% de significancia.

Para realizar una prueba de hipótesis sobre la proporción poblacional de mujeres siendo menor o igual que el 30% con un nivel del 5%, se puede utilizar el método de la región crítica. Los supuestos para realizar esta prueba son:

- La muestra debe ser aleatoria y representativa de la población.
- Independencia de las muestras: La muestra de mujeres debe ser independiente de la muestra de hombres.
- Normalidad de la distribución: La distribución de la población de mujeres debe ser normal.

La hipótesis nula H0 sería que la proporción poblacional de mujeres es menor o igual que el 30%, mientras que la hipótesis alternativa Ha sería que la proporción poblacional de mujeres es mayor que el 30%.

Conclusión: Si se cumplen los supuestos, se puede utilizar el test Z para determinar si el porcentaje poblacional de mujeres es menor o igual que el 30%.

Fórmula: La fórmula para el test Z es:

$$Z = \frac{\bar{p} - p}{\sqrt{\frac{p(1-p)}{n}}}$$

donde:

p̂ es la proporción estimada de la muestra p es la proporción de la población n es el tamaño de la muestra

```
#la proporción muestral fue calculada con anterioridad y fue cargada en la
variable prop_mujeres

# n de la muestra esta determinado por 34 registros

# Se carga en la variable p_0 con el valor hipotético del 30%
```

```
p_0=0.3

# En tal sentido, la formula seria la siguiente:

z=(prop_mujeres-p_0)/sqrt(0.3*(1-p_0)/length(datos2a35$Sexo))

#Se calcula la región critica, teniendo en cuenta que debido a que h0 <= 30% la región critica está localizada en la zona derecha de la curva normal

# Se calcula el valor critico

qnorm(0.95)
```

```
## [1] 1.644854
```

```
# Valor obtenido de la distribución z

z
```

```
## [1] 3.667558
```

Con relación a los resultados obtenidos se puede concluir que hay evidencia suficiente para afirmar que el porcentaje poblacional de mujeres es mayor al 30%, por tal razón, se rechaza la hipótesis nula.

El hecho que el valor calculado sea mayor al crítico indica que se debe rechazar h0.

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).

$$P\text{-valor} = P(Z \geq z)$$

para una prueba de una cola a la izquierda. para una prueba de una cola a la derecha. para una prueba de dos colas.

Para el cálculo del P-valor en R se utiliza el siguiente código

```
pValor=1-pnorm(z)
```

```
pValor
```

```
## [1] 0.0001224391
```

Debido a que el P-valor es menor al nivel de significancia se rechaza la h0, se concluye que no hay evidencia suficiente para afirmar que el porcentaje poblacional de mujeres es menor o igual que el 30%, esta sería la misma medida obtenido a través del método de región critica.

Se recuerda la regla de decisión: - Se rechaza H0 cuando P-valor≤α - No se rechaza H0 cuando P-valor>α

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

```
ejerG <- prop.test(length(datos2a35$Sexo), n=34, p = 0.3, alternative =
"greater" )

ejerG$p.value
```

```
## [1] 1.393396e-18
```

Como se observa en el resultado obtenido, el valor del p-valor sigue siendo menor al nivel de significancia, por esta razón se rechaza h0, como en los ejercicios anteriores.

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

El intervalo de confianza se puede construir utilizando la siguiente fórmula:

prop_mujeres ± z_alfa/2 * sqrt(prop_mujeres * (1 - prop_mujeres) / n)

donde:

z_alfa/2 es la mitad del valor crítico de la distribución normal estándar para un nivel de confianza del 95%, que es aproximadamente 1.96 prop_mujeres es la estimación de la proporción de mujeres en la muestra n es el número total de estudiantes en la muestra

```r
# Calculo de la varianza de la proporción de mujeres
var_prop_mujeres <- prop_mujeres * (1 - prop_mujeres) / 34

# Calculao de la desviación estándar de la proporción de mujeres
sd_prop_mujeres <- sqrt(var_prop_mujeres)

# Calculo del intervalo de confianza del 95%
ci_lower <- prop_mujeres - 1.96 * sd_prop_mujeres
ci_upper <- prop_mujeres + 1.96 * sd_prop_mujeres

# Impresión del intervalo de confianza
ci_lower
```

```
## [1] 0.4228044
```

```r
ci_upper
```

```
## [1] 0.7536662
```

Como se observa en el resultado obtenido, el intervalo de confianza del 95% arroja que la proporción de las mujeres en la población puede estar entre el rango del 42.28 % - 75.36 %, eso con relación a lo observado en la hipótesis nula, que en los ejercicios anterior esta fue rechazada, informando que no había evidencia suficiente para afirmar que la proporción de las mujeres fuera o estuviera por debajo de 30%.

Ejercicio N° 6 - Regresión Lineal Simple

Una analista de deportes quiere saber si existe una relación entre la cantidad de bateos que realiza un equipo de béisbol y el número de runs que consigue. En caso de existir y de establecer un modelo, podría predecir el resultado del partido.

Solución del ejercicio: (David M. Levine, 2023)

```
equipos <- c("Texas","Boston","Detroit","Kansas","St.","New_S.","New_Y.",
"Milwaukee","Colorado","Houston","Baltimore","Los_An.","Chicago",
"Cincinnati","Los_P.","Philadelphia","Chicago","Cleveland","Arizona",
"Toronto","Minnesota","Florida","Pittsburgh","Oakland","Tampa",
"Atlanta","Washington","San.F","San.I","Seattle")
numero_bateos <- c(5659, 5710, 5563, 5672, 5532, 5600, 5518, 5447, 5544, 5598,
5585, 5436, 5549, 5612, 5513, 5579, 5502, 5509, 5421, 5559,
5487, 5508, 5421, 5452, 5436, 5528, 5441, 5486, 5417, 5421)
runs <- c(855, 875, 787, 730, 762, 718, 867, 721, 735, 615, 708, 644, 654, 735,
667, 713, 654, 704, 731, 743, 619, 625, 610, 645, 707, 641, 624, 570,
593, 556)
datos <- data.frame(equipos,numero_bateos,runs)
head(datos)
```

```
##   equipos numero_bateos runs
## 1   Texas          5659  855
## 2  Boston          5710  875
## 3 Detroit          5563  787
## 4  Kansas          5672  730
## 5     St.          5532  762
## 6  New_S.          5600  718
```

Se solicita responder lo siguiente,

1) Realizar una visualización gráfica que permita determinar la relación entre ambas variables.

Para esto se puede usar un scatterplot que permita identificar la distribución de los puntos y determinar la relación entre ellos, el código en R sería el siguiente:

```
plot(numero_bateos, runs)
```

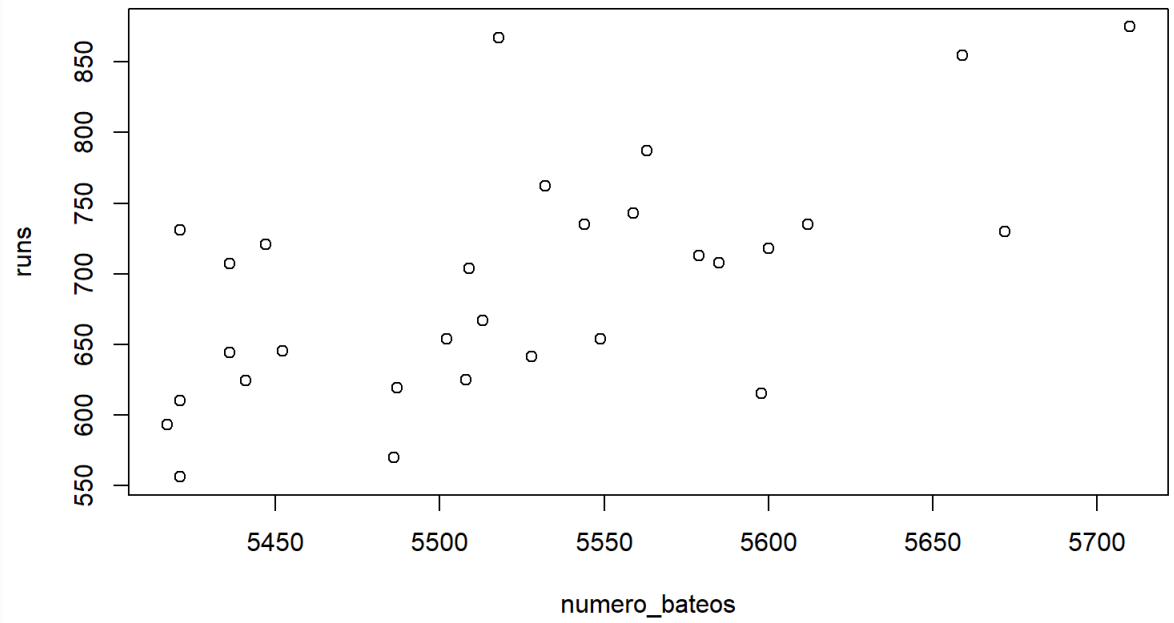

Ilustración 28: Scatterplot.

Este gráfico permite ver si existe una relación lineal entre las dos variables y su dirección. Inicialmente se puede ver que existe una relación entre las variables con pendiente positiva.

2. Poner a prueba la conjetura de significancia estadística del coeficiente de correlación lineal poblacional entre ambas variables con un nivel de significación del 5%.

Para poner a prueba la conjetura de significancia estadística del coeficiente de correlación lineal poblacional, se puede calcular el coeficiente de correlación Pearson (r) y la prueba t correspondiente. El código en R se presenta a continuación:

```
cor.test(numero_bateos, runs)
```

```
##

##  Pearson's product-moment correlation

##

## data:  numero_bateos and runs

## t = 4.0801, df = 28, p-value = 0.0003388

## alternative hypothesis: true correlation is not equal to 0

## 95 percent confidence interval:

##  0.3209675 0.7958231

## sample estimates:

##      cor

## 0.610627
```

El resultado indica el valor de r y el valor p para la hipótesis nula de que el coeficiente de correlación lineal poblacional es igual a cero. Como se observa el valor p es menor que el nivel de significación establecido (0.05), en tal sentido, se puede rechazar la hipótesis nula y concluir que existe una relación lineal significativa entre las dos variables.

3. Construir un modelo de regresión lineal simple. Identifique la variable respuesta y el regresor del modelo. Interpretar los resultados de los parámetros estimados.

Para construir un modelo de regresión lineal simple, se puede usar la función lm(). La variable numero_bateos es el regresor y runs es la variable respuesta. A continuación, se presenta el código en R:

```
modelo <- lm(runs ~ numero_bateos, data = datos)
summary(modelo)
```

```
##
```

```
## Call:

## lm(formula = runs ~ numero_bateos, data = datos)

##

## Residuals:

##     Min      1Q  Median      3Q     Max

## -125.58  -47.05  -16.59   54.40  176.87

##

## Coefficients:

##                 Estimate Std. Error t value Pr(>|t|)

## (Intercept)   -2789.2429   853.6957  -3.267 0.002871 **

## numero_bateos     0.6305     0.1545   4.080 0.000339 ***

## ---

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 66.47 on 28 degrees of freedom

## Multiple R-squared:  0.3729, Adjusted R-squared:  0.3505

## F-statistic: 16.65 on 1 and 28 DF,  p-value: 0.0003388
```

Los resultados de los parámetros estimados incluyen la intercepta y la pendiente del modelo. La interpretación de estos parámetros nos dice cómo la variable numero_bateos influye en la variable runs. Por ejemplo, un aumento de 1 en numero_bateos se asocia con un aumento en runs de la pendiente estimada, esto basado en la fórmula de regresión lineal simple (y_i = + x_i + _i.)

4. Realizar una estimación por intervalos de confianza para la predicción del modelo obtenido con una confianza del 95/. Interpretar los resultados.

Para realizar una estimación por intervalos de confianza para la predicción del modelo, se puede utilizar la función predict(). El argumento interval se usa para especificar el intervalo de confianza.

```
predicciones <- predict(modelo, interval = "confidence", level = 0.95)
```

Los resultados incluyen los valores inferiores y superiores de los intervalos de confianza para cada observación. Estos intervalos indican la incertidumbre en la predicción del modelo.

5. Incorporar al gráfico del punto 1) el modelo estimado.

Para incorporar el modelo estimado al gráfico del punto 1), se puede usar la función abline().

```
plot(numero_bateos, runs)
abline(modelo, col = "red")
```

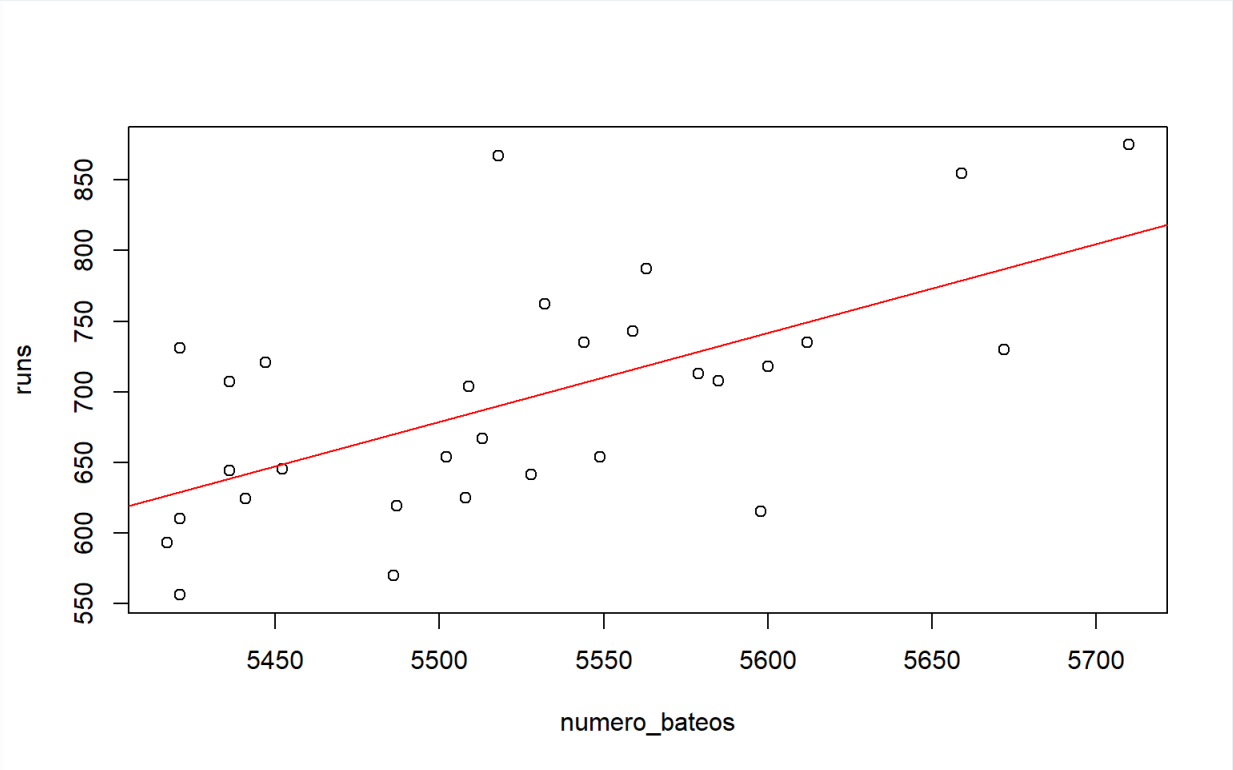

Ilustración 29: Scatterplot con linea de regresión.

Este gráfico muestra la línea de regresión superpuesta en el scatterplot.

6. Verificar los supuestos del modelo de regresión lineal.

Los supuestos del modelo de regresión lineal son importantes para garantizar que los resultados obtenidos a partir del modelo sean válidos y fiables. Estos supuestos incluyen:

- Linealidad: La relación entre la variable respuesta y el regresor debe ser lineal.
- Independencia: Las observaciones deben ser independientes entre sí.
- Homocedasticidad: La varianza de los errores debe ser constante en todo el rango del regresor.
- Normalidad: Los errores deben ser normales y tener una distribución normal.
- Ausencia de multicolinealidad: No debe existir una correlación fuerte entre el regresor y otras variables.

En los resultados obtenidos se observa: Residual standard error: 66.47 on 28 degrees of freedom Multiple R-squared: 0.3729, Adjusted R-squared: 0.3505 F-statistic: 16.65 on 1 and 28 DF, p-value: 0.0003388

Estos se pueden interpretar de la siguiente forma: Residual standard error: El error estándar residual (RSE) es una medida de la variabilidad residual, es decir, la variabilidad de la respuesta que no es explicada por el modelo. Un RSE más pequeño indica un mejor ajuste del modelo. En este caso, el RSE es de 66.47, lo que significa que los residuos tienen una variabilidad moderada.

Multiple R-squared: El R-cuadrado múltiple es una medida de qué tanto el modelo explica la variabilidad de la respuesta. Un R-cuadrado más cercano a 1 indica un mejor ajuste del modelo. En este caso, el R-cuadrado múltiple es de 0.3729, lo que significa que el modelo explica el 37.29% de la variabilidad de los datos.

F-statistic: La estadística F es una medida de qué tanto el modelo es significativo en general. Un valor más grande de F indica un modelo más significativo. En este caso, el valor F es de 16.65, con un p-value de 0.0003388, lo que significa que existe una relación significativa entre las variables en el modelo.

CONCLUSIÓN

Al finalizar el desarrollo del trabajo de investigación presentado, se observó la gran utilidad de la herramienta R en el manejo de datos, esta permite abordar grandes problemáticas de la vida diaria y a través de código encontrar respuesta por simulaciones y/o manejo de datos que ofrece este software. Todo este manejo de información basado en conceptos teóricos permite llegar a decisiones que pueden influir proyectos que terminan por afectar a muchas personas, por eso la importancia de comprender tanto el concepto teórico como el práctico, y con suficiente datos y premisas abordadas tomar decisiones soportadas en la información presentada de la data analizada.

BIBLIOGRAFÍA

CHAO, L. L. (1993). *ESTADISTICA PARA LAS CIENCIAS ADMINISTRATIVAS.* Colombia: Mc Graw Hill.

David M. Levine, T. C. (4 de 2 de 2023). *Estadística para administración.* México: PEARSON Prentice Hall.

Durand, S. L. (2008). *Inferencia estadística y análisis de datos.* Madrid: PEARSON Prentice Hall.

r-coder.com. (03 de 02 de 2023). *r-coder.com*. Obtenido de r-coder.com: https://r-coder.com/