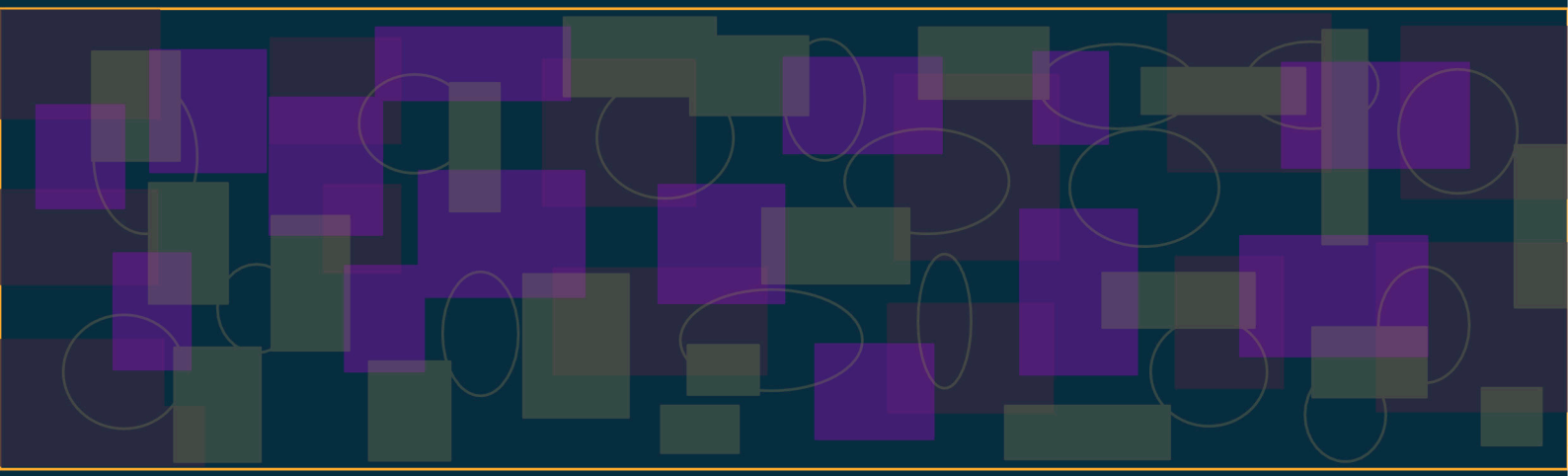
ESTADÍSTICA

DESCRIPTIVA CON R



Ariosto Vicuña Pino
Jéssica Ponce Ordóñez

“ESTADÍSTICA DESCRIPTIVA CON R”

Publicado por: Universidad Técnica Estatal de Quevedo.
Dir. Av. Quito km 1½ vía a Santo Domingo de los Tsáchilas, Quevedo, Ecuador. www.uteq.edu.ec.

Derechos reservados: © Universidad Técnica Estatal de Quevedo, Ecuador 2022.
Dirección de Investigación Ciencia y Tecnología (DICYT). Se autoriza la reproducción de esta publicación con fines educativos y otros que no sean comerciales sin permiso escrito previo detentar el derecho de autor, mencionando la cita.

Cita del libro: Vicuña A. y Ponce J. 2022. Estadística Descriptiva con R. Universidad Técnica Estatal de Quevedo, Ecuador. 252 pp.

Revisión de Pares Externos: Jorge Llaguno Vera
Magister en Métodos Matemáticos y Simulación Numérica en Ingeniería
Universidad Politécnica Salesiana.
Wilder A. Bravo Cedeño
Magister en Gerencia de Sistemas y Tecnología de la información
Universidad de las Américas.

Diseño y Diagramación: Ing. J. Bladimir Mora Macías
Diseñador Gráfico y Multimedia.

Primera Edición: Quevedo, Julio del 2022.

ISBN: 978-9978-371-30-5

ESTADÍSTICA

DESCRIPTIVA CON R

PRIMERA EDICIÓN

Ariosto Vicuña Pino

Docente de la Facultad de Ciencias de la Ingeniería
Universidad Técnica Estatal de Quevedo

Jéssica Ponce Ordóñez

Docente de la Facultad de Ciencias Empresariales
Universidad Técnica Estatal de Quevedo

PREFACIO

En un curso normal de estadística su objetivo es proporcionar los conceptos y fórmulas que se aplican a un conjunto de datos que normalmente, para fines de enseñanza, son pequeños. Sin embargo, el mundo real presenta conjuntos de datos grandes. Bajo esta realidad, realizar las operaciones estadísticas con una calculadora se vuelve un trabajo de enorme esfuerzo y en algunos casos podría ser imposible. El objetivo principal de este libro, a más de hacer una revisión conceptual de los temas tratados, es usar un software estadístico que sea sin costo y al alcance de todos, que facilite el cálculo estadístico y sobre todo que se visualicen gráficamente los resultados que permitan una mejor interpretación de los resultados. Al final del curso, esperamos que los estudiantes dominen la visualización gráfica de los procesos estadísticos y los pongan en práctica en su vida profesional preferentemente.

El texto muestra cómo las personas que trabajan en el campo estadísticos pueden reducir el tiempo de trabajo que toma el procesamiento estadístico y dedicar el mayor tiempo posible a la interpretación de los resultados. Los conjuntos de datos utilizados a lo largo del texto se comparan con modelos ideales para visualmente ver que ocurre con esos datos y llegar a conclusiones correctas.

Nuestro enfoque

La Asociación Americana de Estadística (ASA) en su informe: Pautas para la evaluación e instrucción en educación estadística (GAISE) - Informe universitario Informe universitario 2016 (https://www.amstat.org/asa/files/pdfs/GAISE/GaiseCollege_Full.pdf), establece en las metas para estudiantes en cursos introductorios de estadística que los estudiantes deben poder producir presentaciones gráficas y resúmenes numéricos e interpretar lo que estos revelan y no revelan.

Este objetivo plantea la necesidad de visualizar los resultados, en lo posible de forma gráfica. ASA de forma acertada indica que el análisis de datos implica mucho más que construir un intervalo de confianza o encontrar un valor p. Las pantallas gráficas de datos proporcionan información sobre la distribución de valores de datos, relaciones entre variables y valores atípicos. Con el advenimiento de grandes conjuntos de datos, a menudo de estudios observacionales que pueden no ser una muestra aleatoria de una población definida, lo que hace que las técnicas de inferencia estándar sean inapropiadas, el uso adecuado de las pantallas gráficas es fundamental. El uso de software para producir pantallas gráficas hace que la visualización de grandes conjuntos de datos sea relativamente fácil. Las pantallas gráficas univariadas importantes incluyen histogramas, diagramas de caja, diagramas de puntos y gráficos de barras. Las pantallas gráficas bivariados

incluyen diagramas de dispersión, gráficos de barras agrupadas y apiladas, e histogramas comparativos y diagramas de caja. A menudo se pueden agregar variables adicionales a una pantalla gráfica (por ejemplo, se pueden incluir puntos de colores y líneas de regresión por separado para hombres y mujeres en un diagrama de dispersión que relaciona la edad con la altura para niños de 3 años a 18 años).

Queda en evidencia que este libro se enfoca a visualizar los resultados de forma gráfica para una mejor comprensión del comportamiento de los datos de acuerdo a los procesos estadísticos aplicados. Incluso aquellos resultados que estamos acostumbrados a visualizarlos de forma numérica en este libro se lo hacen de forma gráfica como por ejemplo la media, mediana y moda. La revisión teórica y fórmulas matemáticas se exponen como parte del fundamento teórico; sin embargo, es el uso de R Statistic para graficar los resultados es lo sustancial en este libro.

Organización del libro

El capítulo 1 se centra en la gestión de los datos, con especial énfasis en la adquisición de datos desde fuentes externas. También trata, con igual importancia, la manipulación del conjunto de datos desde R, esto implica como obtener subconjuntos de datos a partir de un conjunto de datos. El capítulo concluye con un apartado para tratar los datos perdidos o faltantes dentro de un conjunto de datos.

El capítulo 2 se refiere a la representación gráfica de los datos desde el punto de vista de la estadística descriptiva. Lo interesante que se presenta es que se combinan varios gráficos estadísticos en una misma salida gráfica usando únicamente el core de R. Esta combinación muestra ampliamente varios detalles que facilitan la interpretación estadística de lo que está ocurriendo.

El capítulo 3 describe, explora y compara datos desde el punto de vista de las medidas de centralidad, dispersión y de posición relativa. Los gráficos de estas medidas se proponen sobre la función densidad de la distribución. Muestra gráficamente cada una de las medidas propuestas para tener un cabal conocimiento de la distribución.

El capítulo 4 nos muestra como graficar las distribuciones de probabilidad que comúnmente se trata en los cursos de estadística. Se hace énfasis en la distribución normal dado que es la distribución de mayor uso dada sus características. Se presenta también los modelos de las distribuciones Binomial, Poisson y Weibull.

El capítulo 5 nos lleva al principio de la inferencia estadística marcada por la determinación de los intervalos de confianza. Se propone el marco teórico necesario para comprender el porqué de su importancia y nos detalla cómo realizar los gráficos de intervalos de confianza de manera sencilla.

El capítulo 6 se interna en la prueba de hipótesis desde tres puntos de vista de la hipótesis nula (igual a cero, mayor o igual que cero o que sea menor o igual que cero) para su visualización. A partir de este enfoque se maneja los gráficos sobre la prueba de hipótesis. Se propone un ejemplo por cada caso.

El capítulo 7 propone la regresión lineal simple como último ítem de estudio. Inicialmente se da las pautas para que analítica y gráficamente se determine la posibilidad de aplicación del método de regresión. De ser posible su aplicación, se ilustra como obtener el modelo lineal y cómo graficarlo. Finalmente, nos da varias pautas para validar el modelo obtenido.

Autoevaluación y ejercicios propuestos

Los capítulos constan de una sección de autoevaluación que proporciona una buena pauta a los lectores para que midan su aprendizaje. Además, se propone cinco ejercicios al final de cada capítulo para que desarrolle sus habilidades. Las respuestas de estos ejercicios se encuentran al final del texto.

TABLA DE CONTENIDO

1 MANEJO DE LOS DATOS

La gestión de datos tiene notable importancia antes de iniciar el procesamiento y análisis estadístico de los mismos. Incluye la creación, adquisición y manipulación del conjunto de datos que se produce dentro del proceso de recopilación de datos. Existen diferentes formas de implementar la gestión de datos; sin embargo, este libro -en general- se enfoca al uso de las funciones nativas que incluye el núcleo de R Statistic. Se evita, en la medida de lo posible, el uso de librerías o paquetes adicionales. En este capítulo se examina los siguientes puntos:

- Tipos de datos
- Contenedores de datos
- Creando el conjunto de datos
- Adquisición del conjunto de datos
- Manipulación del conjunto de datos

Al final del capítulo usted debe ser capaz de gestionar sus datos de manera rápida y eficiente.

Tipos de datos y objetos contenedores de datos

Para la recopilación de datos existen una variedad de métodos y procedimientos disponibles. A continuación se mencionan algunos de los métodos de recopilación de datos más útiles y usados con con mayor frecuencia [1]:

- Experimentos
- Encuestas telefónicas
- cuestionarios escritos y encuestas
- Observación directa y entrevistas personales.

Cuando se aplican estos métodos inicialmente se tiene datos sin procesar, que es un término utilizado para números y etiquetas de categoría que se han recopilado pero que aún no se han procesado de ninguna manera [2].

La estadística clásica se puede dividir en dos ramas principales: la estadística descriptiva y la estadística inferencial. En ambos casos, trabajamos con un conjunto de mediciones [3]. Estas mediciones se realizan sobre las variables de interés formuladas en el estudio o investigación. Las variables pueden ser de distinta naturaleza y se asocian con un tipo de dato; es decir, pueden ser numéricas, caracter, lógicas, etcétera. Un ejemplo de esto es el peso de una persona. Se define (en R) entonces la **variable** pesoPersona que tiene un valor cuyo **tipo de dato** es numérico. El proceso

de fijar un valor a una variable se conoce como **asignación**. La asignación entre la variable pesoPersona y el valor del peso se lo realiza con el operador $< -$.

$$variable < -valor \qquad \text{pesoPersona} < -\mathbf{145.4}$$

A la **variable** pesoPersona se **asigna** (<-) el valor de **145.40** cuyo **tipo de datos** es numérico. Si se asigna entre comillas **"**145.4**"** no se trata de un valor numérico sino una cadena de caracteres que no son suceotibles de realizar operaciones de tipo cuantitativo. Por tanto, se debe tener con el tipo de datos que se asigna a una variable. La tabla 1.0 muestra los principales tipos de datos que admite R.

Tabla 1.0 *Tipo de datos básicos que admite R*

TIPO DE DATOS	DESCRIPCIÓN	EJEMPLO
Caracter	Caracter [a-z, A-z, alfanuméricos]	var1
Lógico	Valores de verdadero o falso	TRUE / FALSE
Enteros	Números sin parte decimal	231
Numérico	Valores reales o decimales	231.342
Complejo	admiten números del tipo ni+mj	3+2j

Los objetos más comunes para contener datos están en la tabla 1.1

Tabla 1.1 *Contenedores de datos más usados en R*

CONTENEDOR	DESCRIPCIÓN	EJEMPLO
Escalar	Contienen un solo valor de cualquier tipo	e<-3.4
Vectores	Colección de elementos, comúnmente de tipo: carácter, lógico, entero o numérico	v[i]<-5
Matrices	Es un conjunto de datos bidimensional	m=matrix (c('a', 'a', 'b', 'c'), nrow=2, ncol=2)
Tablas	Podemos usar tablas para todo tipo de datos. Útiles para resumir datos	t<-table(***var1***,var2)
Dataframes	Son objetos de datos tabulares. cada columna puede contener diferente tipo de datos	df<-data.frame(v1=c("2", "4"), v1=c("0.3", "0.22"))

El conjunto de datos

Cuando se recopila datos, el primer paso es introducirlos en R. Posteriormente, querrá ver sus datos y realizar resumen de estadísticas y otros análisis sobre ellos[4]. Un conjunto de datos puede estar compuesto de todos los valores observados para las variables del estudio. Sin embargo, también podría contener solo una parte de las variables y/o datos. Las preguntas que se pueden explorar y las técnicas analíticas que se pueden utilizar dependerán del tipo de datos y del número de variables [5]. Para mostrar cómo trabajar con datos, primero vamos a establecer un conjunto de datos que contendrá datos acerca de las cartillas de análisis de sangre de distintos pacientes. Las variables de estudio propuestas se encuentran en la tabla 1.2

Tabla 1.2 *Descripción de las variables del conjunto de datos análisis de sangre*

Variable	Descripción
gR	Glóbulos Rojos (millones de células / mcL)
gB	Glóbulos Blancos (10^9/L)
plq	Plaquetas (10^9/L)
Hgl	Hemoglobina (g/dl)
hto	Hematocrito (%)

Las variables, como se mencionó, están asociadas a un tipo de datos. Además, las variables intrínsicamente pertenecen a un tipo de contenedor de datos. Para aclarar esta idea, tomemos el caso de pesoPersonas que fue declarado como un escalar (por contener un solo dato). Si se necesita que almacene varios valores de pesoPersona necesitamos que ésta variable pertenezca a otro tipo de contenedor, como por ejemplo una matriz: tipoPersona$<-$**matrix**(). La tabla 1.3 presenta el conjunto de datos (variables y valores) que se usará en este apartado.

Tabla 1.3 *Conjunto de datos análisis de sangre*

Sexo	**gR**	**gB**	**plq**	**hgl**	***hto***
m	4.3	7.2	124	13.0	*80*
v	4.8	6.5	219	16.7	*70*
m	5.0	8.7	290	15.2	*100*
v	5.9	7.8	314	14.5	*50*
m	4.8	6.4	219	13.8	*70*
v	6.0	8.1	340	15.7	*60*
v	5.2	9.3	310	13.9	*70*

Variables con valores únicos

Es común, en estadística, contener un solo valor en una variable cuantitativa o cualitativa. En R se usa el nombre de la variable, el operador de asignación (<-) y la función **c**(). Para la representación gráfica de ese valor se hará uso de la función **plot**() para visualizar el valor de la variable. Para el ejemplo, se asigna el valor de 4.3 a la variable cuantitativa globuloRojo. La vista gráfica se presenta en la figura 1.0

```
globuloRojo<-c(4.3)                      # Crear  la variable glóbulo rojo
plot(globuloRojo)                        # Gráfico de la variable
text(1,4.7, "Glóbulo Rojo", col="red") # Texto dentro del gráfico
```

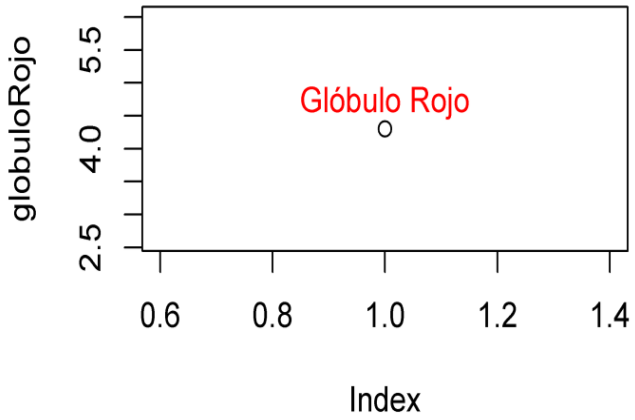


Figura 1.0 *Gráfico del valor de una variable numérica*

La función plot() automáticamente proporciona un eje vertical llamado glóbulo rojo con una escala coherente al valor de dicha variable. También proporciona un eje horizontal llamado índice (index) que provee la posición del valor dentro de la variable. Al ser un valor único tomará siempre la posición uno.

Vectores

Un vector es una estructura de datos básica y un objeto R que almacena un conjunto de valores del mismo tipo de datos usado cuando desea almacenar y modificar un conjunto de valores de una misma variable. Los tipos de datos pueden ser lógicos, enteros, dobles y de caracteres. Los vectores se pueden crear utilizando la función **c**()[6].

La función **c**() es la función predeterminada que *combina* los elementos que forman el conjunto de datos que estarán contenidos en el vector. Todos los elementos se convierten a un mismo tipo de datos, que es el tipo de datos del valor devuelto.

Observe el siguiente código que crea la variable glóbulos rojos (gR) con los valores dados en el conjunto de datos análisis de sangre. Además, se incluye el gráfico de los valores del vector por medio de la función plot().

```
#CREAR UN VECTOR EN R
gR<-c(4.3, 4.8, 5.0, 5.9, 4.8, 6.0, 5.2)
gR
# SALIDA POR CONSOLA
## [1] 4.3 4.8 5.0 5.9 4.8 6.0 5.2
#VISUALIZACIÓN ESTÁNDAR DEL VECTOR
par(mfrow=c(1,1), mar=c(5,3.8,2,2))
plot(gR)
```

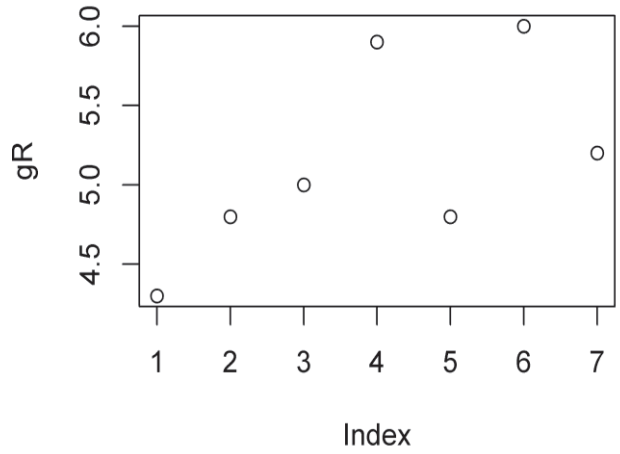


Figura 1.1 *Gráfico de un vector correspondiente a una variable numérica*

La salida de la variable gR es una combinación de números. La figura 1.1 contiene la salida gráfica que muestra en el eje vertical (y) los valores almacenados en el vector gR y en el eje horizontal (x) la posición (o índice) dentro del mismo vector.

A continuación se va a crear un vector llamado análisis de sangre 1 (as1) que va a contener la información de la primera fila del conjunto de datos análisis de sangre de la tabla 1.3. Para indicar la variable a quien pertenece cada valor, utilizaremos la función **names()** con el parámetro as1. De esta manera, se elimina la posición numérica (index) que R asigna por defecto.

```
as1<-c(4.3, 7.2, 124, 13.0, 80)
#ASIGNAR NOMBRES DE LAS COLUMNAS
names(as1)<-c("gR","gB","plq","hgl", "hto")
#VECTOR as1
as1
##    gR    gB   plq   hgl   hto
##   4.3   7.2 124.0  13.0  80.0
#ASIGNAR NOMBRES DE LOS EJES
par(mfrow=c(1,1), mar=c(5,3.8,2,2))
plot(as1, xaxt="n")
axis(side=1, at=1:5, labels = names(as1))
```

La figura 1.2 tiene en su eje x los nombres de las variables del análisis de sangre. Sin embargo, solo se han tomado en cuenta los valores numéricos dejando de lado información cualitativa del sexo de la persona (de la tabla 1.3) que se realiza el análisis de sangre.

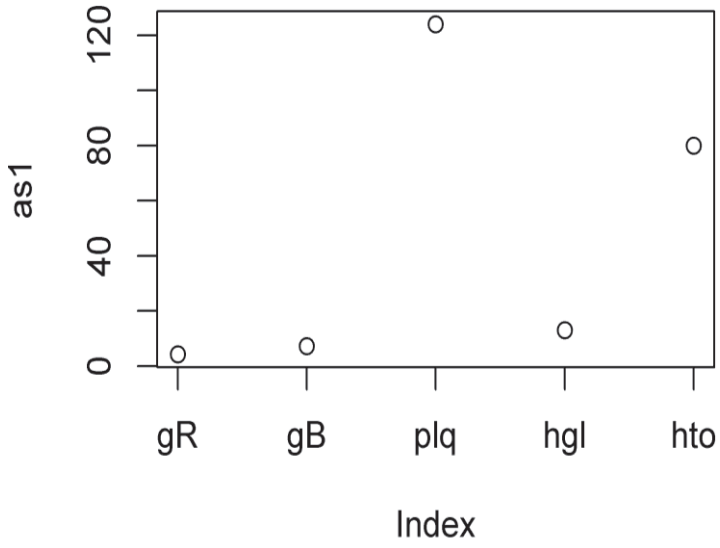


Figura 1.2 *Gráfico de un vector con variable numéricas*

Incluir el valor de la variable sexo conlleva a enfrentarse con el problema del tipo de datos. Esta variable es tipo caracter y por consiguiente transformará a todos los datos del vector en valores tipo caracter. Por definición, los vectores tienen un único tipo de datos; es decir, un vector no puede tener datos tipo caracter y tipo numérico al mismo tiempo. Para observarlo, se usará la función **str()** que muestra de forma compacta la estructura interna de un objeto R, es una función de diagnóstico.

```
as1<-c("m" ,4.3, 7.2, 124, 13.0, 80)
names(as1)<-c("Sexo","gR","gB","plq","hgl", "hto")
# RECUPERANDO UN VALOR DEL VECTOR
str(as1)
# SALIDA POR CONSOLA
##  Named chr [1:6] "m" "4.3" "7.2" "124" "13" "80"
##  - attr(*, "names")= chr [1:6] "Sexo" "gR" "gB" "plq" ...
```

La salida por consola evidencia que los datos numéricos han sido definidos como caracteres:

```
Named chr [1:6] "m" "4.3" "7.2" "124" "13" "80"
```

Esto significa que no es posible hacer operaciones matemáticas con ellos. La pregunta que surge es ¿cómo se puede tener los valores numéricos?

Suponga que el valor de la variable glóbulos blancos (gB) debe ser multiplicado por cinco (5). Para realizar esta operación siga los siguientes pasos: Recupere el valor usando el nombre asignado a las columnas con names(). Para el caso de la variable glóbulos blancos se usó “gB” y lo transforma mediante la función as.numeric(). El código de la parte inferior muestra cómo hacerlo.

```
as1<-c("m" ,4.3, 7.2, 124, 13.0, 80)
# NOMBRES DE COLUMNAS
names(as1)<-c("Sexo","gR","gB","plq","hgl", "hto")
```

```
# RECUPERANDO UN VALOR DEL VECTOR
as1["gB"]
##    gB
## "7.2"
str(as1["gB"])
##  Named chr "7.2"
##  - attr(*, "names")= chr "gB"
# TRANSFORMANDO EL DATOS DE CARACTER A NUMÉRICO
5*as.numeric(as1["gB"])
## [1] 36
```

Al inicio se detalló qué variables existen y cuál es su tipo de dato para en caso de ser necesario transformarla. Se recomienda usar siempre la función str() para verificar si el tipo de dato de la variable es correcto; y, si la variable no tiene el tipo de dato correcto se debe proceder a transformar el tipo de dato con el operador **as.tipoDeDato()**. La tabla 1.4 muestra algunas de las transformaciones de tipos de datos más usadas.

Un problema que se avizora es que se va a necesitar un vector por cada análisis de sangre. Como se tiene la información de siete exámenes de sangre se necesitaría siete vectores; es decir, desde as1 hasta as7 para tener todos los valores de los análisis de sangre. Esto no es conveniente para trabajar el conjunto de datos análisis de sangre. Lo ideal sería tener todos los datos en un mismo objeto de R. Revisemos entonces el objeto matriz, que está limitado solo para datos numéricos.

Tabla 1.4 *Uso del operador as para cambiar de tipo de datos*

Opción	Transformar a
as.numeric()	numérico
as.character	caracter
as.double	double
as.table	Tabla
as.matrix	Matriz
as.list	Lista
as.data.frame	Data.frame

Matrices

Una matriz en matemáticas es solo un arreglo bidimensional de números. Las matrices se utilizan para muchos propósitos en estadística teórica y práctica. Sin embargo, las matrices y los arreglos de dimensiones superiores también se utilizan para fines más simples, principalmente para mantener tablas, por lo que es necesaria una descripción elemental. En R, la noción de matriz se extiende a elementos de cualquier tipo[7].

Una matriz es un objeto de R de dos dimensiones que únicamente puede conener datos numéricos. Se crea mediante la función **matrix**(), se divide en n filas (nrow) y m columnas (ncol) de la siguiente manera **matrix(datos, nrow= n, ncol= m)**. Para el conjunto de datos análisis de sangre esto representa un problema porque la variable sexo es de tipo caracter y no puede ser tomada en cuenta dentro de una matriz. Por tanto, la matriz tiene cinco columnas (numéricas) y siete filas de datos (Tabla 1.3). Nótese que solo se usa variables numéricas para crear una matriz.

Tabla 1.5 *Conjunto de datos análisis de sangre sin la columna sexo*

gR	**gB**	**plq**	**hgl**	***hto***
4.3	7.2	124	13.0	*80*
4.8	6.5	219	16.7	*70*
5.0	8.7	290	15.2	*100*
5.9	7.8	314	14.5	*50*
4.8	6.4	219	13.8	*70*
6.0	8.1	340	15.7	*60*
5.2	9.3	310	13.9	*70*

Analice como se crea la Tabla 1.5 utilizando una matriz.

```
# CREAR MATRIZ
analisisSangre<-matrix(c(  4.3, 7.2, 124, 13.0, 80
, 4.8, 6.5, 219, 16.7, 70
, 5.0, 8.7, 290, 15.2, 100
, 5.9, 7.8, 314, 14.5, 50
, 4.8, 6.4, 219, 13.8, 70
, 6.0, 8.1, 340, 15.7, 60
, 5.2, 9.3, 310, 13.9, 70), nrow=7, ncol=5)
print("Matriz de análisis de sangre")
## [1] "Matriz de análisis de sangre"
analisisSangre
# SALIDA POR CONSOLA
##       [,1]  [,2]  [,3]  [,4]  [,5]
## [1,]   4.3 219.0 100.0   6.4  15.7
## [2,]   7.2  16.7   5.9 219.0  60.0
## [3,] 124.0  70.0   7.8  13.8   5.2
## [4,]  13.0   5.0 314.0  70.0   9.3
## [5,]  80.0   8.7  14.5   6.0 310.0
## [6,]   4.8 290.0  50.0   8.1  13.9
## [7,]   6.5  15.2   4.8 340.0  70.0
```

La salida por consola nos muestra una matriz con identificación numérica de filas y de columnas. En general, usar nombres en las columnas es más común que usar nombres en las filas. En las filas resulta mejor que mantener su identificación numérica, aunque existen casos que es necesario cambiar estos nombres. ¿Cómo poner los nombres de filas y columnas? En la siguiente sección de código se usa las funciones **rownames()** y **colname()** para poner los nombres de filas y columnas.

```
analisisSangre<-matrix(c(  4.3, 7.2, 124, 13.0, 80
              , 4.8, 6.5, 219, 16.7, 70
```

```
              , 5.0, 8.7, 290, 15.2, 100
              , 5.9, 7.8, 314, 14.5, 50
              , 4.8, 6.4, 219, 13.8, 70
              , 6.0, 8.1, 340, 15.7, 60
              , 5.2, 9.3, 310, 13.9, 70)
              , nrow=7
              , ncol=5)
# NOMBRE DE LAS COLUMNAS
colnames(analisisSangre)<-c("gR","gB","plq","hgl","hto")
# NOMBRE DE LAS FILAS
rownames(analisisSangre)<-c("genesis","kevin","karina","carlos","elizabeth","
marco","john")
print("Matriz de análisis de sangre con nombres de filas")
## [1] "Matriz de análisis de sangre con nombres de filas"
analisisSangre
# SALIDA POR CONSOLA
##              gR    gB   plq   hgl   hto
## genesis     4.3 219.0 100.0   6.4  15.7
## kevin       7.2  16.7   5.9 219.0  60.0
## karina    124.0  70.0   7.8  13.8   5.2
## carlos     13.0   5.0 314.0  70.0   9.3
## elizabeth  80.0   8.7  14.5   6.0 310.0
## marco       4.8 290.0  50.0   8.1  13.9
## john        6.5  15.2   4.8 340.0  70.0
```

La salida por consola muestra que la matriz *analisisSangre* contiene todos los datos de la tabla 1.5. Con matrices hay mucho por hacer, pero no es el objetivo del libro profundizar más allá de lo necesario para que pueda trabajar con R de forma básica. Es necesario en este punto saber cómo agregar una fila o columna a una matriz. Imagine que tiene un nuevo examen y es necesario agregarlo (m, 4.3, 219.0, 100.0, 6.4, 15.7) a la matriz. Para ello se usan las funciones **rbind()** y **cbind()**. Con *rbind* se agregan filas y con *cbind* las columnas.

```
# CREAR LA MATRIZ
analisisSangre<-matrix(c(  4.3, 7.2, 124, 13.0, 80
              , 4.8, 6.5, 219, 16.7, 70
              , 5.0, 8.7, 290, 15.2, 100
              , 5.9, 7.8, 314, 14.5, 50
              , 4.8, 6.4, 219, 13.8, 70
              , 6.0, 8.1, 340, 15.7, 60
              , 5.2, 9.3, 310, 13.9, 70)
              , nrow=7 # Número de filas
              , ncol=5)# Número de columnas
# PONER NOMBRES A COLUMNAS
colnames(analisisSangre)<-c("gR","gB","plq","hgl","hto")
# PONER NOMBRES A FILAS
rownames(analisisSangre)<-c("genesis","kevin","karina","carlos","elizabeth","
marco","john")
estefi<-c(4.9, 9.0, 312, 11.3, 75)           # Datos de la fila a agregar
analisisSangre<-rbind(analisisSangre, estefi) # Agregar fila
Na<-c(136,145,138,137,142,139,144,141)       # Datos de la columna a agregar
analisisSangre<-cbind(analisisSangre, Na)     # Agregar columnas
print("Matriz análisis de sangre: Agregando filas")
```

```
## [1] "Matriz análisis de sangre: Agregando filas"
analisisSangre
# SALIDA POR CONSOLA
##              gR    gB   plq   hgl   hto  Na
## genesis     4.3 219.0 100.0   6.4  15.7 136
## kevin       7.2  16.7   5.9 219.0  60.0 145
## karina    124.0  70.0   7.8  13.8   5.2 138
## carlos     13.0   5.0 314.0  70.0   9.3 137
## elizabeth  80.0   8.7  14.5   6.0 310.0 142
## marco       4.8 290.0  50.0   8.1  13.9 139
## john        6.5  15.2   4.8 340.0  70.0 144
## estefi      4.9   9.0 312.0  11.3  75.0 141
```

La salida por consola muestra que la información se encuentra perfectamente guardada e identificada. Para incrementar un poco la dificultad, sería interesante agregar el promedio de cada columna de la matriz analisisSangre ¿cómo podríamos hacerlo?

En R, se calcula el promedio con la función **mean()**. Sin embargo, es necesario emplear la función **apply()** para aplicar funciones a las variables de la matriz. La función apply() necesita como parámetro el conjunto de datos, indicar si la operación se aplica a la fila (1) o a la columna (2) y la operación que se va a realizar (mean) .

```
#SE APLICA LA FUNCIÓN mean() A TODAS LAS COLUMNAS
print("Matriz análisis de sangre: Calculando el promedio de las columnas ")
## [1] "Matriz análisis de sangre: Calculando el promedio de las columnas "
apply(analisisSangre, 2, mean)
# SALIDA POR CONSOLA
##       gR       gB      plq      hgl      hto       Na
##  30.5875  79.2000 101.1250  84.3250  69.8875 140.2500
```

Se ha obtenido de forma fácil el resultado de la media de cada columna por medio de las funciones apply() y mean(). Es momento de agregar este resultado a la matriz *analisisSangre*. Para añadir una fila a la matriz usaremos la función **rbind().** La nueva fila será almacenada en un vector llamado MEDIAS y posteriormente agregada a la matriz analisisSangre.

```
MEDIAS<-apply(analisisSangre, 2, mean)        # Cálculo de la media
analisisSangre<-rbind(analisisSangre,MEDIAS)  # Fila agregada
print("Matriz análisis de sangre: Agregando el promedio de cada columna ")
## [1] "Matriz análisis de sangre: Agregando el promedio de cada columna "
AnalisisSangre
# SALIDA POR CONSOLA
##                 gR    gB     plq     hgl      hto     Na
## genesis     4.3000 219.0 100.000   6.400  15.7000 136.00
## kevin       7.2000  16.7   5.900 219.000  60.0000 145.00
## karina    124.0000  70.0   7.800  13.800   5.2000 138.00
## carlos     13.0000   5.0 314.000  70.000   9.3000 137.00
## elizabeth  80.0000   8.7  14.500   6.000 310.0000 142.00
## marco       4.8000 290.0  50.000   8.100  13.9000 139.00
## john        6.5000  15.2   4.800 340.000  70.0000 144.00
```

```
## estefi      4.9000   9.0 312.000  11.300  75.0000 141.00
## MEDIAS     30.5875  79.2 101.125  84.325  69.8875 140.25
```

Una vez que hemos agregado la media a la matriz *analisisSangre*, vamos también a incluir los valores máximos y mínimos de cada variable contenidas en la matriz. Para hallar el valor máximo se usa la función **max()** y para el valor mínimo **min()**.

```
# CREAR LA MATRIZ DE DATOS
analisisSangre<-matrix(c(  4.3, 7.2, 124, 13.0, 80
             , 4.8, 6.5, 219, 16.7, 70
              , 5.0, 8.7, 290, 15.2, 100
              , 5.9, 7.8, 314, 14.5, 50
              , 4.8, 6.4, 219, 13.8, 70
              , 6.0, 8.1, 340, 15.7, 60
              , 5.2, 9.3, 310, 13.9, 70)
              , nrow=7
              , ncol=5)
# NOMBRE DE COLUMNAS DE LA MATRIZ
colnames(analisisSangre)<-c("gR","gB","plq","hgl","hto")
# NOMBRE DE FILAS DE LA MATRIZ
rownames(analisisSangre)<-c("genesis","kevin","karina","carlos","elizabeth","
marco","john")
# CÁLCULO DE LOS VALORES MÁXIMOS, MÍNIMOS Y PROMEDIOS
MAXIMOS<-apply(analisisSangre, 2, max) # Valor máximo
MINIMOS<-apply(analisisSangre, 2, min) # Valor mínimo
MEDIAS<-apply(analisisSangre, 2, mean) # Media
# AGREGAR LOS RESULTADOS DE LOS VALORES MÁXIMOS, MÍNIMOS Y PROMEDIOS
analisisSangre<-rbind(analisisSangre,MAXIMOS) # Agrega máximos
analisisSangre<-rbind(analisisSangre,MEDIAS)  # Agrega medias
analisisSangre<-rbind(analisisSangre,MINIMOS) # Agrega mínimos
print("Matriz análisis de sangre: Agregando los valores Max, Min y Medias")
## [1] "Matriz análisis de sangre: Agregando los valores Max, Min y Medias"
print("de cada columna")
## [1] "de cada columna"
# SALIDA POR CONSOLA
analisisSangre
# SALIDA POR CONSOLA
##                  gR        gB   plq       hgl       hto
## genesis     4.30000 219.00000 100.0   6.40000  15.70000
## kevin       7.20000  16.70000   5.9 219.00000  60.00000
## karina    124.00000  70.00000   7.8  13.80000   5.20000
## carlos     13.00000   5.00000 314.0  70.00000   9.30000
## elizabeth  80.00000   8.70000  14.5   6.00000 310.00000
## marco       4.80000 290.00000  50.0   8.10000  13.90000
## john        6.50000  15.20000   4.8 340.00000  70.00000
## MAXIMOS   124.00000 290.00000 314.0 340.00000 310.00000
## MEDIAS     34.25714  89.22857  71.0  94.75714  69.15714
## MINIMOS     4.30000   5.00000   4.8   6.00000   5.20000
```

La salida por consola muestra como se ha creado una matriz con información estadística de las medias y valores máximos y mínimos de cada columna. Como usted podrá notar que existen funciones muy fáciles de usar como mean(), max(), min(). A continuación se indicará como

recuperar los valores que se encuentran dentro de la matriz. Empecemos por recuperar filas o columnas completas.

Recuperación de valores desde una matriz

Para recuperar una fila en una matriz se utiliza el índice o etiqueta de la fila que se desea recuperar de forma completa, no se debe especificar nada para las columnas. Recordemos que el formato de una matriz creada es **nombre_matriz[fila, columna]**. Un índice es un número que identifica la fila o columna; en R, los índices empiezan desde uno. Para el ejemplo, vamos a recuperar una fila usando ambas opciones: el índice y la etiqueta. Recuperemos la fila con el índice 4 que tiene la etiqueta C**arlos**.

```
print("Recuperar una fila mediante el índice numérico")
## [1] "Recuperar una fila mediante el índice numérico"
analisisSangre[4,]
# SALIDA POR CONSOLA
##    gR    gB   plq   hgl   hto
##  13.0   5.0 314.0  70.0   9.3
print("Recuperar una fila mediante la etiqueta")
## [1] "Recuperar una fila mediante la etiqueta"
analisisSangre["Carlos",]
# SALIDA POR CONSOLA
##    gR    gB   plq   hgl   hto
##  13.0   5.0 314.0  70.0   9.3
```

De manera similar, si queremos recuperar los valores de una columna usamos el índice o la etiqueta de la columna.

```
print("Recuperar una fila mediante el índice")
## [1] "Recuperar una fila mediante el índice"
analisisSangre[,5]
# SALIDA POR CONSOLA
##   genesis     kevin    karina    carlos elizabeth     marco      john
##  15.70000  60.00000   5.20000   9.30000 310.00000  13.90000  70.00000
##   MAXIMOS    MEDIAS   MINIMOS
## 310.00000  69.15714   5.20000
print("Recuperar una fila mediante la etiqueta")
## [1] "Recuperar una fila mediante la etiqueta"
analisisSangre[,"hto"]
# SALIDA POR CONSOLA
##   Genesis     Kevin    Karina    Carlos Elizabeth     Marco      John
##  15.70000  60.00000   5.20000   9.30000 310.00000  13.90000  70.00000
##   MAXIMOS    MEDIAS   MINIMOS
## 310.00000  69.15714   5.20000
```

La salida por consola se puede mejorar transformando la matriz analisisSangre en un dataframe.

```
# RECUPERAR UNA COLUMNA MEDIANTE EL ÍNDICE
hto<-analisisSangre[,5]
hematocritos<-as.data.frame(hto)
print("Recuperar una columna mediante el índice")
```

```
## [1] "Recuperar una columna mediante el índice"
Hematocritos
# SALIDA POR CONSOLA
##                 hto
## genesis    15.70000
## kevin      60.00000
## karina      5.20000
## carlos      9.30000
## elizabeth 310.00000
## marco      13.90000
## john       70.00000
## MAXIMOS   310.00000
## MEDIAS     69.15714
## MINIMOS     5.20000
```

Resulta cómodo recuperar filas o columnas completas de una matriz, pero si queremos un valor específico del conjunto de datos contenido en la matriz analisisSangre ¿cómo se hace? Este caso se resuelve especificando la fila y la columna. Se propone extraer la información de la MEDIA de la columna hematocrito (hto). Nótese las diversas formas de obtener el mismo resultado.

```
print("Recuperar datos:")
## [1] "Recuperar datos:"
print("Usando índices numéricos de fila y columna")
## [1] "Usando índices numéricos de fila y columna"
analisisSangre[9,5]
## [1] 69.15714
print("Usando etiqueta para la fila e índice numérico columna")
## [1] "Usando etiqueta para la fila e índice numérico columna"
analisisSangre["MEDIAS", 5]
## [1] 69.15714
print("Usando índice numérico para la fila y etiqueta para la columna")
## [1] "Usando índice numérico para la fila y etiqueta para la columna"
analisisSangre[9,"hto"]
## [1] 69.15714
print("Usando etiquetas para filas y columnas")
## [1] "Usando etiquetas para filas y columnas"
analisisSangre["MEDIAS","hto"]
## [1] 69.15714
```

Otro requerimiento de recuperación de datos que con frecuencia se formula en matrices es el que incluye varias filas/columnas a partir de índices numéricos o etiquetas conocidas. Se propone el siguiente caso: obtener los datos de los pacientes Karina, Carlos, Elizabeth y Marco sobre sus plaquetas(plq), hemoglobina (hgl) y hematocritos (hto).

```
# SELECCIÓN POR FILAS
print("Recuperar datos usando índices numéricos")
## [1] "Recuperar datos usando índices numéricos"
analisisSangre[3:6,]
##              gR    gB   plq  hgl   hto
## karina    124.0  70.0   7.8 13.8   5.2
## carlos     13.0   5.0 314.0 70.0   9.3
```

```
## elizabeth  80.0   8.7  14.5  6.0 310.0
## marco       4.8 290.0  50.0  8.1  13.9
print("Recuperar datos usando etiquetas")
## [1] "Recuperar datos usando etiquetas"
filas<-c("Karina", "Carlos", "Elizabeth", "Marco")
analisisSangre[filas,]
##              gR    gB   plq  hgl   hto
## karina    124.0  70.0   7.8 13.8   5.2
## carlos     13.0   5.0 314.0 70.0   9.3
## elizabeth  80.0   8.7  14.5  6.0 310.0
## marco       4.8 290.0  50.0  8.1  13.9
# SELECCIÓN POR FILAS Y COLUMNAS
print("Recuperar datos usando etiquetas e índices numéricos")
## [1] "Recuperar datos usando etiquetas e índices numéricos"
analisisSangre[filas,3:5]
##             plq  hgl   hto
## karina      7.8 13.8   5.2
## carlos    314.0 70.0   9.3
## elizabeth  14.5  6.0 310.0
## marco      50.0  8.1  13.9
print("Recuperar datos usando etiquetas")
## [1] "Recuperar datos usando etiquetas"
columnas<-c("plq", "hgl", "hto")
analisisSangre[filas,columnas]
##             plq  hgl   hto
## karina      7.8 13.8   5.2
## carlos    314.0 70.0   9.3
## elizabeth  14.5  6.0 310.0
## marco      50.0  8.1  13.9
```

Es posible, en ocacasiones, que no se tiene exactamente conocimiento de los valores del que se desean recuperar. En su lugar, se tiene nociones de esos valores. Cuando estamos frente a este caso se requiere trabajar con operadores matemáticos para obtener las filas y columnas que posiblemente contengan el valor buscado. La tabla 1.6 contiene los operadores matemáticos que se usan con mayor frecuencia.

Tabla 1.6 *Operadores Matemáticos de mayor frecuencia de uso*

Operaciones	**Símbolo**
Y	&&
O	||
Igual	==
Diferente	!
Mayor	>
Mayor igual	>=
Menor	<
Menor que	<=

Se propone los siguientes ejemplos de para filtrar o seleccionar información:

- Caso 1: Extraer las filas cuyo valor de hematocrito sea mayor o igual a 60
- Caso 2: Extraer las filas con valor de hematocrito menor o igual a 60, seleccionando las columnas glóbulos rojos y plaquetas
- Caso 3: Extraer las filas con plaquetas mayor o igual que 100 y hematocrito menor o igual que 75, mostrar todas las columnas
- Caso 4: Extraer las filas con valores de plaquetas mayor que 10 y sodio mayor o igual que 137, seleccione las columnas glóbulos blancos, hemoglobina y sodio

```
print("Recuperar filas para si el hematocrito es mayor o igual a 60")
## [1] "Recuperar filas para si el hematocrito es mayor o igual a 60"
analisisSangre[analisisSangre,"hto"]>=60,]
##                  gR        gB   plq       hgl       hto
## kevin       7.20000  16.70000   5.9 219.00000  60.00000
## elizabeth  80.00000   8.70000  14.5   6.00000 310.00000
## john        6.50000  15.20000   4.8 340.00000  70.00000
## MAXIMOS   124.00000 290.00000 314.0 340.00000 310.00000
## MEDIAS     34.25714  89.22857  71.0  94.75714  69.15714
print("Recuperar filas para si el hematocrito es menor o igual a 60")
## [1] "Recuperar filas para si el hematocrito es menor o igual a 60"
print("Únicamente las columnas glóbulos rojos y plaquetas")
## [1] "Únicamente las columnas glóbulos rojos y plaquetas"
columnas<-c("gR", "plq")
analisisSangre[analisisSangre,"hto"]<=60, columnas]
##            gR   plq
## genesis   4.3 100.0
## kevin     7.2   5.9
## karina  124.0   7.8
## carlos   13.0 314.0
## marco     4.8  50.0
## MINIMOS   4.3   4.8
print("Recuperar filas para plaquetas mayor o igual a 100 Y")
## [1] "Recuperar filas para plaquetas mayor o igual a 100 Y"
print("hematocrito menor o igual a 75, mostrar todas las columnas")
## [1] "hematocrito menor o igual a 75, mostrar todas las columnas"
analisisSangre[(analisisSangre,"plq"]>=100) & (analisisSangre,"hto"]<=75),]
##           gR  gB plq  hgl  hto
## genesis  4.3 219 100  6.4 15.7
## carlos  13.0   5 314 70.0  9.3
print("Recuperar filas cuyas plaquetas sea mayor que 10 Y")
## [1] "Recuperar filas cuyas plaquetas sea mayor que 10 Y"
print("sodio mayor o igual que 137, con las columnas")
## [1] "sodio mayor o igual que 137, con las columnas"
print("glóbulos blancos, hemoglobina y sodio")
## [1] "glóbulos blancos, hemoglobina y sodio"
columnas<-c("gB", "hgl", "hto")
analisisSangre[(analisisSangre[1:8,"gB"]>10) & (analisisSangre[1:8,"hgl"]>219
), columnas]
```

```
##            gB hgl hto
## john     15.2 340  70
## MAXIMOS 290.0 340 310
```

El gráfico de dispersión representa puntos de pares ordenados (x,y) de dos variables cualesquiera. La función plot(x,y) cumple con este cometido. Como ejemplo se realiza el gráfico de dispersión glóbulos rojos (gR) vs hemoglobina (hgl).

```
par(mar=c(4.0,3.8,2,2))
globulosRojos<-analisisSangre[,"gR"]
globulosBlancos<-analisisSangre[,"hgl"]
plot(globulosRojos,globulosBlancos
     , main =  "Glóbulos Blancos vs Glóbulos Rojos")
```

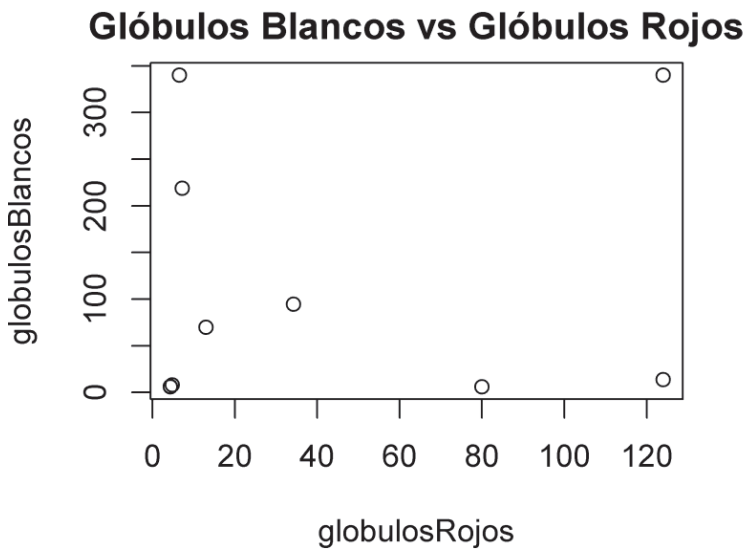


Figura 1.3 *Diagrama de dispersión*

Dataframe

Un dataframe tiene mayor capacidad para manejar de datos que una matriz, porque sus columnas pueden contener diferentes tipos de datos (numéricos, de caracteres, etcétera). Es similar al conjunto de datos que normalmente vería en SAS, SPSS y Stata. Los dataframe son la estructura de datos más común con la que tratará en R[8]. Básicamente, un dataframe es una lista de vectores que pueden considerarse como las columnas del dataframe y las longitudes de cada vector, que forma parte del dataframe, deben ser iguales. Esto facilita que se pueda tener un conjunto de datos formado por distintos tipos de datos (numérico, carácter, factor, entre otros). Las hojas de datos de Excel, tablas de bases de datos, datos en formatos de otros sistemas estadísticos (como SPSS, SAS, STATA, etcétera), archivos csv y de texto pueden ser organizados y manejados a través del objeto dataframe de R.

La forma básica de crear un dataframe a partir de datos que debemos ingresar desde R es la siguiente:

```
#VECTORES
sexo<-c("m", "v", "m", "v", "m", "v", "v")
gR<-c(4.3, 4.8, 5.0, 5.9, 4.8, 6.0, 5.2)
gB<-c(7.2, 6.5, 8.7, 7.8, 6.4, 8.1, 9.3)
plq<-c("124", "219", "290", "314", "219", "340", "310" )
hgl<-c(13.0, 16.7, 15.2, 14.5, 13.8, 15.7, 13.9)
hto<-c(80, 70, 100, 50, 70, 60, 70)
#CREACIÓN DE UN DATAFRAME A PARTIR DE VECTORES
examenes<-data.frame(sexo, gR, gB, plq, hgl, hto)
kable(examenes, caption = "Dataframe exámenes")
```

Dataframe exámenes

sexo	gR	gB	plq	hgl	hto
m	4.3	7.2	124	13.0	80
v	4.8	6.5	219	16.7	70
m	5.0	8.7	290	15.2	100
v	5.9	7.8	314	14.5	50
m	4.8	6.4	219	13.8	70
v	6.0	8.1	340	15.7	60
v	5.2	9.3	310	13.9	70

Una vez más se enfatiza el hecho de que es importante verificar los tipos de datos que se crean en los objetos contenedores. Podrían surgir errores de tipo de datos sobre todo con las variables numéricas y caracter que se obtiene desde archivos. Cuando se observa los datos del dataframe no es posible percatarse visualmente de ese detalle, usted simplemente ve números. También puede darse el caso de que una variable carácter este reconocida como tipo factor. Para resolver estos inconvenientes con los tipos de datos de un dataframe se recomienda, como parte del procedimiento de la creación de un dataframe, *observar el conjunto de datos y verificar el tipo de datos de las variables*. Para esto, puede usar las siguientes funciones: la función head(nombre del dataframe, número de filas) que deja observar dataframe según el número de filas que se considere necesario; y, la función str(nombre del dataframe) que describe la estructura del dataframe dando información como el número de observaciones y tipo de datos de las variables.

```
#VERIFICAR EL TIPO DE DATOS DE LAS COLUMNAS DEL DATAFRAME
#VECTORES
sexo<-c("m", "v", "m", "v", "m", "v", "v")
gR<-c(4.3, 4.8, 5.0, 5.9, 4.8, 6.0, 5.2)
gB<-c(7.2, 6.5, 8.7, 7.8, 6.4, 8.1, 9.3)
#INTENCIONALMENTE SE CREA ESTE VECTOR CON CARACTERES
plq<-c("124", "219", "290", "314", "219", "340", "310" )
hgl<-c(13.0, 16.7, 15.2, 14.5, 13.8, 15.7, 13.9)
hto<-c(80, 70, 100, 50, 70, 60, 70)
#CREACIÓN DE UN DATAFRAME A PARTIR DE VECTORES
#TAMBIÉN SE PUEDE USAR OTROS NOMBRES DIFERENTES AL DE LOS VECTORES
#DE LA SIGUIENTE MANERA globuloRojo=gR
```

```
examenes<-data.frame(sexo, gR, gB, plq, hgl, hto)
#VISUALIZACIÓN DE LOS DATOS
kable(head(examenes, 3))
```

sexo	gR	gB	plq	hgl	hto
m	4.3	7.2	124	13.0	80
v	4.8	6.5	219	16.7	70
m	5.0	8.7	290	15.2	100

```
#VISUALIZACIÓN DE LOS TIPOS DE DATOS
str(examenes)
# SALIDA POR CONSOLA
## 'data.frame':    7 obs. of  6 variables:
##  $ sexo: Factor w/ 2 levels "m","v": 1 2 1 2 1 2 2
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : Factor w/ 6 levels "124","219","290",..: 1 2 3 5 2 6 4
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
```

La variable sexo y plaqueta están declaradas como tipo caracter en los vectores, pero R implícitamente las declara como tipo factor. Lo correcto es que la variable sexo sea tipo caracter y la variable hemoglobina sea tipo numérico respectivamente. Verificar los tipos de datos de las variables, antes de hacer algún procedimiento estadístico, nos evitará más de un dolor de cabeza. En un dataframe es necesario usar el símbolo dólar para indicar con que columna se quiere trabajar, así señalamos que se trabajará con la variable plaquetas (plq) del dataframe exámenes: examenes$plq. A continuación, se correge los errores en los tipos de datos de las variables sexo y plaquetas (plq) usando el operador **as.***tipo_de_datos*().

```
# CORRECCIÓN DEL TIPO DE DATOS
examenes$sexo<-as.character(examenes$sexo)
examenes$plq<-as.numeric(examenes$plq)
# VISUALIZACIÓN DE LOS DATOS
head(examenes, 3)
# SALIDA POR CONSOLA
##   sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2   1 13.0  80
## 2    v 4.8 6.5   2 16.7  70
## 3    m 5.0 8.7   3 15.2 100
# VISUALIZACIÓN DE LOS TIPOS DE DATOS
str(examenes)
# SALIDA POR CONSOLA
## 'data.frame':    7 obs. of  6 variables:
##  $ sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  1 2 3 5 2 6 4
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
```

Corregido los problemas de tipo de datos puede continuar con la preparación de datos.

Importación / Exportación (I/E) del conjunto de datos

Como analista de datos normalmente se enfrentará al problema de adquirir datos que provienen de una variedad de fuentes y tipos de formatos. La tarea de importar los datos para someterlos a un proceso de análisis estadístico que concluya con un informe de los resultados es frecuente. R proporciona una amplia gama de herramientas para importar datos. Puede importar datos desde archivos de texto, MS Excel, bases de datos relacionales, otros paquetes estadísticos, sitios web y servicios en línea [9].

Importar y exportar datos es uno de los procesos más significativos al trabajar con datos. Esto conlleva a tener protocolos que faciliten y aseguren la realización de estas tareas. Se sugiere el siguiente protocolo para importar datos desde archivos:

1. Instalar librería (en caso de ser necesario) - `install.packages("nombre de la librería")`
2. Cargar librería (en caso de ser necesario) - `library("nombre de la librería")`
3. Ubicarnos en el directorio donde está el archivo que contiene los datos - `setwd("dirección del archivo")`
4. De acuerdo a tipo de archivo que se va a importar (csv, txt, xls/xlsx), usar las fuunciones: `read.csv / read.table / loadWorkbook(), readWorksheet()`
5. Explorar el conjunto de datos - `head(objeto R, número de filas)`
6. Explorar estructuralmente el conjunto de datos - `str(objeto R)`

Cuando se trabaja con datos (preprocesamiento) es prudente tener una copia del conjunto de datos original o preprocesados. En ocasiones se presentan imprevistos que alteran su contenido y tener una copia es una buena opción para recuperarlos. Una copia se obtiene exportando el conjunto de datos a un archivo o base de datos. A continuación, se muestran los pasos a seguir:

1. Instalar librería (en caso de ser necesario) - `install.packages("nombre de la librería")`
2. Cargar librería (en caso de ser necesario) - `library("nombre de la librería")`
3. Ubicarnos en el directorio donde está el archivo - `setwd("dirección del archivo")`
4. De acuerdo al tipo de archivo que se va a exportar (csv, txt, xls/xlsx), usar las funciones: `write.csv / write.table / writeWorksheetToFile().`
5. Explorar si el archivo ha sido guardado - `file.exists("nombre del archivo")`

I/E desde archivos csv

Para iniciar la práctica de importación de datos abra el bloc de notas, digite los siguientes datos separándolos con puntos y comas. La figura 1.4 muestra como debe verse el bloc de notas. Al final, guarde el archivo con el nombre examenes.csv:

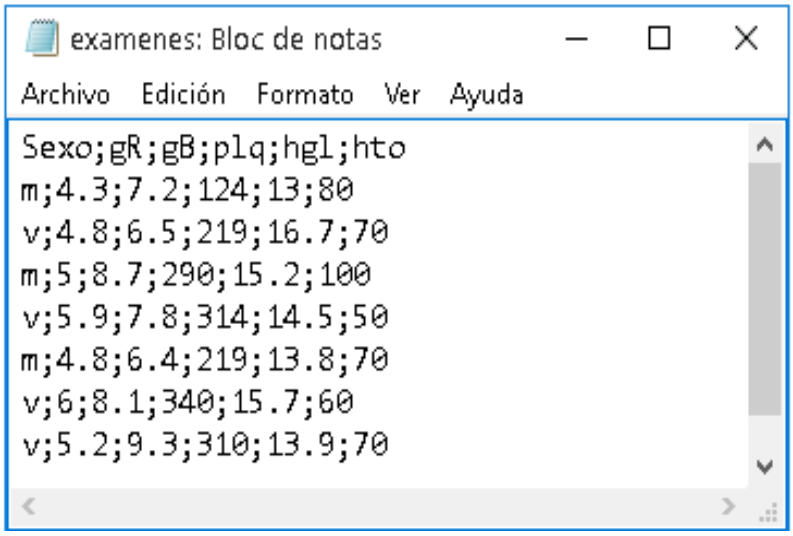


Figura 1.4 *Archivo csv: examenes.csv*

Creado el archivo examenes.csv iniciamos la práctica, para importar (leer) datos desde un archivo de valores separado por comas (.csv) se utiliza la función **read.csv**(). Es importante que se indique cual es el directorio de trabajo donde esta el archivo csv que vamos a leer. Esto se logra con la función **setwd**() y cambiando la barra diagonal "\" -que normalmente se usa- por la barra diagonal invertida"/".

Con el directorio de trabajo configurado, la función **read.csv**("nombre_archivo.csv", header=TRUE, sep=";") es la apropiada para leer los datos desde archivos csv. El parámetro header=TRUE indica que la primera fila del archivo.csv pertenece al nombre de las columnas de los datos y el parámetro sep=";" nos revela que el separador es un punto y coma (;). Otros separadores comunes son el tab (sep = " t") y la coma (sep = ","). El código de abajo lee los datos desde el archivo examenes.csv y los pone en el dataframe examenesCsv.

```
# UBICAR EL DIRECTORIO DONDE SE ENCUENTRA EL ARCHIVO CSV
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR (LEER) LOS DATOS
examenesCsv<-read.csv("examenes.csv", header=TRUE, sep = ";")
# VISUALIZAR LOS DATOS, SE MUESTRA LAS TRES PRIMERAS FILAS
head(examenesCsv, 3)
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
```

```
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenesCsv)
# SALIDA POR CONSOLA
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: Factor w/ 2 levels "m","v": 1 2 1 2 1 2 2
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : int  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : int  80 70 100 50 70 60 70
```

Observamos que la variable Sexo ha sido creada como un tipo Factor, pero sabemos que se está trabajando como tipo caracter. Se realiza la corrección:

```
# CORRECCIÓN DEL TIPO DE DATO
examenesCsv$Sexo<-as.character(examenesCsv$Sexo) # Cambio de factor a
# VISUALIZAR TRES FILAS DE LOS DATOS             # caracter
head(examenesCsv, 3)
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenesCsv)
# SALIDA POR CONSOLA
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : int  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : int  80 70 100 50 70 60 70
```

Recuerde que con la función **str**() se obtiene la cantidad de observaciones, el número de variables y la descripción del tipo de datos de las variables contenidas en el dataframe. Para exportar el archivo original a una copia usamos la función **write.csv**(). Se debe ubicar el directorio donde se desea guardar el archivo csv, el nombre de la copia y *row.names* en falso (para el caso que los nombres de las filas no tengan importancia).

```
# UBICAR EL DIRECTORIO DONDE SE VA A GUARDAR EL ARCHIVO CSV
setwd("C:/Users/tiran/Desktop/Datos")
# EXPORTAR LOS DATOS
write.csv(examenesCsv, file="examenesCopia.csv", row.names = F)
# REVISAR SI EL ARCHIVO HA SIDO GRABADO
file.exists("examenesCopia.csv")
# SALIDA POR CONSOLA
## [1] TRUE
```

I/E desde archivos de texto

Cuando se trabaja con archivos de texto usamos la función **read.table** (). Al igual que **read.csv**() los argumentos más usados para realizar la tarea son: el nombre del archivo, indicar si tiene cabecera (nombre de las columnas) y cuál es su separador.

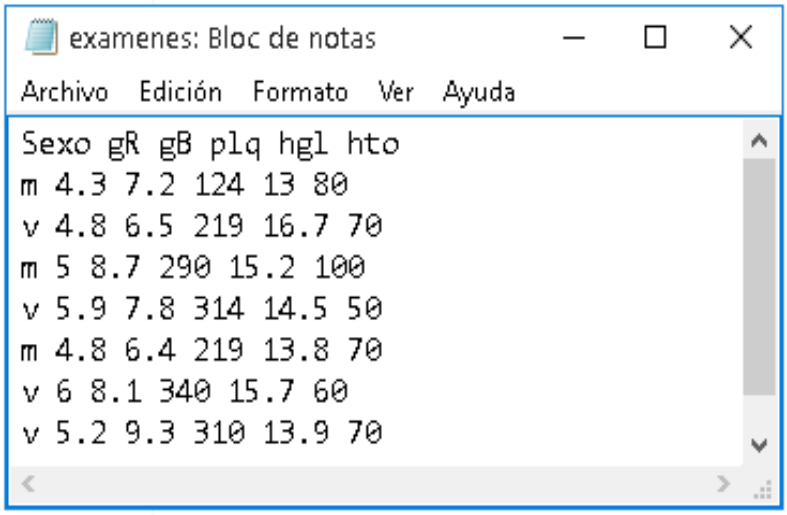


Figura 1.5 *Archivo de texto: exámenes.txt*

Para el ejemplo, el nombre del archivo fuente es "examenes.txt". Se observa en la figura 1.5 que el conjunto de datos tiene cabecera (*Header*=TRUE) y que sus columnas están separadas por espacios en blanco (*sep*="")

```
# CONFIGURAR EL DIRECTORIO DONDE SE ENCUENTRA EL ARCHIVO DE TEXTO
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LOS DATOS
examenTxt<-read.table("examenes.txt", header=TRUE, sep = "")
# VISUALIZAR TRES FILAS DE LOS DATOS
head(examenTxt,3)
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenTxt)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: Factor w/ 2 levels "m","v": 1 2 1 2 1 2 2
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : int  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : int  80 70 100 50 70 60 70
```

Aplicando la recomendación de revisar el conjunto de datos creado en R, se visualiza unas cuantas filas (tres filas) del dataframe examenTxt con la función **head**() y verificamos si los tipos de datos

son los correctos con **str**(). Hecho esto, note que la variable Sexo tiene el tipo de dato Factor, que para el ejemplo no es correcto porque debe ser de tipo character. Procedemos a corregir el inconveniente.

```
# CORRECCION DEL TIPO DE DATO
examenTxt$Sexo<-as.character(examenTxt$Sexo)
# VISUALIZAR TRES FILAS DE LOS DATOS
head(examenTxt, 3)
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenTxt)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : int  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : int  80 70 100 50 70 60 70
```

Realizada la corrección del tipo de datos Sexo tenemos listo el conjunto de datos para trabajar. En caso de necesitar exportar el dataframe examenTxt a un archivo de texto se procede de manera:

```
# UBICAR EL DIRECTORIO DONDE SE VA A GUARDAR EL ARCHIVO CSV
setwd("C:/Users/tiran/Desktop/Datos")
# EXPORTAR LOS DATOS
write.table(examenTxt, file="examenesCopia.txt", sep=';', dec='.')
# REVISAR SI EL ARCHIVO HA SIDO GRABADO
file.exists("examenesCopia.txt")
## [1] TRUE
```

I/E desde libros de Excel

Una fuente de datos muy común son las hojas de cálculo de EXCEL con extensión .xls o .xlsx. Para importar datos desde Excel existen algunas librerias; sin embargo, únicamente se hará uso de la librería **XLConnect** que está preparada para importar/exportar datos desde y hacia las hojas de Excel. En EXCEL, se debe tener claro que el principal objeto es el libro, cada libro contiene hojas. Es en las hojas donde se encuentran los diferentes conjuntos de datos con los que vamos a trabajar.

Primero, en caso de no estar instalada, se debe instalar la librería XLConnect de la siguiente manera: `install.packages("XLConnect").` Recuerde siempre ubicarse en el directorio que contiene el libro de Excel -con la función **setwd**()-. El libro de Excel será leído con la función **loadWorkBook**(), mientras que para acceder a una hoja específica del libro se usa la función **readWorksheet**(). Veamos como trabaja todo esto:

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# MOSTRAR NOMBRES DE LAS HOJAS (SHEETS) DEL LIBRO
getSheets(libroExcel)
## [1] "examenSangre"   "examenSangreA"  "examenSangreB"  "examenSangreDF"
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenSangreXls=readWorksheet(libroExcel, sheet="examenSangre")
# VISUALIZAR TRES FILAS DE LOS DATOS
head(examenSangreXls, 3)
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenSangreXls)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
```

Al revisar la estructura del dataframe examenSangreXls se observa que los tipos de datos son correctos; y, en este caso no es necesario rectificar los mismos.

Para exportar (guardar) los datos contenidos en un dataframe hacia una hoja de Excel, se usa la función **writeWorksheetToFile**() de la librería XLConnect. Los parámetros mínimos necesarios para que esta función realice su trabajo son: el nombre del libro Excel con que se va a guardar el dataframe, el nombre del dataframe que contiene los datos a guardar (*data*), el nombre de la hoja en el libro Excel (*sheet*), la fila de inicio de guardado (*startRow*) y la columna de inicio de guardado (*startCol*). El siguiente ejemplo muestra cómo hacerlo.

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# EXPORTAR DATOS DESDE LA HOJA DESEADA
writeWorksheetToFile("examenesCopia.xlsx"     #nombre del libro de Excel
                     , data = examenSangreXls #dataframe que contien los dato
s
                     , sheet = "examenSangre" #hoja del libro de Excel
                     , startRow = 1           #fila de inicio
                     , startCol = 1)          #columna de inicio
```

```
# REVISAR SI EL ARCHIVO HA SIDO GRABADO
file.exists("examenesCopia.xlsx")
## [1] TRUE
```

I/E desde la Web

Para importar datos desde la web se realiza los siguientes pasos:

1. Escribir la dirección web donde está contenido el archivo de Excel
2. Bajar el archivo donde se encuentran los datos
3. Descomprimir el archivo (en caso de ser necesario)
4. Listar los archivos obtenidos para conocer / verificar el nombre del archivo del cual se va a importar los datos
5. Leer el archivo de la manera tradicional

Cabe indicar que es posible leer directamente desde la web los datos, pero como una buena práctica es mejor seguir estos pasos que nos facilitan obtener el archivo y guardarlo en nuestra computadora. La ventaja de hacer todo el procedimiento es que podemos trabajar desconectados. Puede darse el caso de que el archivo no esté empaquetado en un archivo zip o rar, sino que la URL nos da el acceso directo al archivo csv en cuyo caso debemos usar directamente la función **read.csv**().

```
# DIRECCIÓN WEB DONDE SE ENCUENTRA NUESTRO ARCHIVO
url<-"https://www.stats.govt.nz/assets/Uploads/Food-price-index/Food-price-in
dex-June-2019/Download-data/food-price-index-june-2019-csv.zip"
# CONFIGURAR EL DIRECTORIO DONDE DESEA GUARDAR EL ARCHIVO
setwd("C:/Users/tiran/Desktop/Datos")
# BAJAR EL ARCHIVO
download.file(url,destfile = "food-price-index-june-2019.zip",mod="wb")
# COMO EL ARCHIVO QUE SE HA BAJADO TIENE EXTENCIÓN zip DEBE
# SER DESCOMPRIMIDO
unzip("food-price-index-june-2019.zip",exdir = ".")
# LISTAR LOS ARCHIVOS OBTENIDOS PARA VERIFICAR EL NOMBRE DEL ARCHIVO
# DEL CUAL VAMOS A OBTENER LOS DATOS
list.files()
# SALIDA POR CONSOLA
##  [1] "archivoCSV.png"
##  [2] "archivoTXT.png"
##  [3] "atletas.xlsx"
##  [4] "como escribir los ejercicios.png"
##  [5] "componentesQuimicos.txt"
##  [6] "componentesQuimicosTabla.txt"
##  [7] "datosFaltantesXlsx.png"
##  [8] "dnorm.png"
##  [9] "examenes.csv"
## [10] "examenes.txt"
## [11] "examenes.xlsx"
```

```
## [12] "examenesCopia.csv"
## [13] "examenesCopia.txt"
## [14] "examenesCopia.xlsx"
## [15] "examenesNuevo.csv"
## [16] "food-price-index-jun19-index-numbers-csv-tables.csv"
## [17] "food-price-index-jun19-seasonally-adjusted-csv-tables.csv"
## [18] "food-price-index-jun19-weighted-average-prices-csv-tables.csv"
## [19] "food-price-index-june-2019.zip"
## [20] "Nueva carpeta"
## [21] "planeta.png"
## [22] "pruebaHipotesis.xlsx"
## [23] "qnorm.png"
## [24] "rmarkdown-spanish.pdf"
# AHORA DEPENDIENDO DEL TIPO DE ARCHIVO SE PROCEDE
# COMO EN LOS CASOS ANTERIORES
# LISTAMOS ÚNICAMENTE LOS ARCHIVOS DE EXTENSIÓN CSV
list.files(pattern = "\\.csv")
## [1] "examenes.csv"
## [2] "examenesCopia.csv"
## [3] "examenesNuevo.csv"
## [4] "food-price-index-jun19-index-numbers-csv-tables.csv"
## [5] "food-price-index-jun19-seasonally-adjusted-csv-tables.csv"
## [6] "food-price-index-jun19-weighted-average-prices-csv-tables.csv"
# CON EL NOMBRE EXACTO DEL ARCHIVO CSV PROCEDEMOS A IMPORTAR LOS DATOS
alimentos<-read.csv("food-price-index-jun19-index-numbers-csv-tables.csv"
                    , header=TRUE, sep = ",")
head(alimentos, 5)
##   Series_reference  Period Data_value STATUS UNITS
## 1       CPIM.SE901 1960.01   45.92346  FINAL Index
## 2       CPIM.SE901 1960.02   45.49864  FINAL Index
## 3       CPIM.SE901 1960.03   45.11630  FINAL Index
## 4       CPIM.SE901 1960.04   45.15878  FINAL Index
## 5       CPIM.SE901 1960.05   45.28623  FINAL Index
##   Subject Group
## 1 Consumers Price Index - CPI Food Price Index for New Zealand
## 2 Consumers Price Index - CPI Food Price Index for New Zealand
## 3 Consumers Price Index - CPI Food Price Index for New Zealand
## 4 Consumers Price Index - CPI Food Price Index for New Zealand
## 5 Consumers Price Index - CPI Food Price Index for New Zealand
##   Series_title_1
## 1           Food
## 2           Food
## 3           Food
## 4           Food
## 5           Food
str(alimentos)
## 'data.frame':    10981 obs. of  8 variables:
##  $ Series_reference: Factor w/ 37 levels "CPIM.SE901","CPIM.SE9011",..: 1
1 1 1 1 1 1 1 1 ...
##  $ Period          : num  1960 1960 1960 1960 1960 ...
##  $ Data_value      : num  45.9 45.5 45.1 45.2 45.3 ...
##  $ STATUS          : Factor w/ 1 level "FINAL": 1 1 1 1 1 1 1 1 1 1 ...
##  $ UNITS           : Factor w/ 1 level "Index": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Subject         : Factor w/ 1 level "Consumers Price Index - CPI": 1 1
1 1 1 1 1 1 1 1 ...
```

```
##  $ Group           : Factor w/ 4 levels "Food Price Index for New Zealand"
,..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Series_title_1  : Factor w/ 37 levels "Beef and veal (fresh, chilled or
frozen)",..: 11 11 11 11 11 11 11 11 11 11 ...
```

Como se ha mencionado, existen muchas librerías para importar y exportar datos. Sin embargo, la idea principal de este texto es mostrar los métodos más sencillos y prácticos para realizar el trabajo con datos. En realidad, existe innumerable información contenida en archivos de texto (.txt), valores separados por comas (csv) y de Excel (xls o xlsx). R contiene infinidad de librerías para extraer datos de archivos con formatos de otros paquetes estadísticos y desde bases de datos. Los principios para importar / exportar datos desde ellos no distan mucho de lo expuesto en este apartado.

Dataframe: Manipulación de datos

El núcleo del trabajo con datos es la capacidad de crear, fusionar, dividir y realizar otras operaciones con subconjuntos de datos. La aplicación de varias operaciones a subconjuntos de datos al por mayor es tan importante como la aplicación de los métodos estadísticos [10]. La tarea de manipular datos se puede ejecutar de numerosas maneras. Las librerías base de R provee operadores y funciones que facilitan esta actividad; sin embargo, existen librerías que no pertenecen a las librerías base cuyas funcionalidades resultan más fácil y apropiadas para manipular datos. Estas librerías adicionales no son tratadas en este libro.

Existen varias formas de extraer un subconjunto de datos como el operador **[,]** o la función **subset**(); para operaciones de agregar filas o columnas se puede usar **merge**(), **cbind**() o **rbind**() que facilita combinar filas / columnas desde múltiples vectores o dataframes. Es poco frecuente obtener los datos que exactamente necesitamos desde la fuente, por lo que es necesario un trabajo adicional llamado **preprocesamiento de datos**. El preprocesamiento de datos inicia con la obtención de la estructura correcta. Existen otros inconvenientes como: tener un conjunto de datos con variables no necesarias para el estudio, filas que no contienen datos que sean parte de estudio, errores en la toma de datos, digitalización o formato de los mismos, entre otras razones. A menudo se necesita agregar, eliminar y reemplazar los datos. A continuación, se presenta un conjunto de operaciones que son frecuentes en el preprocesamiento de datos.

Agregar filas y/o columnas

Para realizar las operaciones de agregar filas o columnas en un dataset se procede de la siguiente manera:

1. Leer o crear el conjunto de datos en un dataframe

2. Examinar con las funciones **head**() y **str**() el contenido y la estructura de los datos
3. Agregar filas o columnas
4. Guardar la modificación en caso de ser necesario

Para el paso 3 se debe hacer lo siguiente:

- Si son filas o un dataframe que se van a unir use la función **rbind**() con igual número de columnas que el dataframe;
- Si son columnas las que se van a adjuntar use la función **cbind**() teniendo en cuenta que debe tener el mismo número de filas del dataframe.

El código del siguiente ejemplo está dividido en cuatro secciones donde primero se presenta la importación de datos desde un archivo de Excel (xlsx), luego dos secciones donde se muestra de forma sencilla como agregar datos por filas y columnas para luego finalizar con la exportación del nuevo conjunto de datos llamado examenesAB hacia el archivo examenesTratamientoDatos.xlsx.

```
#-------------------------------------------------
#IMPORTACIÓN DE DATOS DESDE UN ARCHIVO XLSX
#-------------------------------------------------
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenSangreA=readWorksheet(libroExcel, sheet="examenSangreA")
examenSangreB=readWorksheet(libroExcel, sheet="examenSangreB")
# VISUALIZAR DOS FILAS DE LOS DATOS
head(examenSangreA, 2)
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
head(examenSangreB, 2)
##   Sexo  gR  gB plq  hgl hto
## 1    v 6.0 8.0 219 14.5  50
## 2    m 4.8 6.5 340 13.8  70
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenSangreA)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
```

```
str(examenSangreB)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "v" "m" "v" "m" ...
##  $ gR  : num  6 4.8 5 5.9 4.8 6 4
##  $ gB  : num  8 6.5 8.7 7.8 6.4 8.1 6
##  $ plq : num  219 340 310 124 219 290 314
##  $ hgl : num  14.5 13.8 15.7 13.9 13 16.7 15.2
##  $ hto : num  50 70 60 70 80 70 100
#-------------------------------------------------
#UNIR DATAFRAMES O FILAS CON LA FUNCIÓN rbind()
#-------------------------------------------------
# UNIR DOS DATAFRAMES POR FILAS
examenesAB<-rbind(examenSangreA, examenSangreB)
# NOTE QUE EL NUMERO DE OBSERVACIONES HA CAMBIADO
head(examenesAB, 2)
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
str(examenesAB)
## 'data.frame':    14 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2 6 4.8 5 ...
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3 8 6.5 8.7 ...
##  $ plq : num  124 219 290 314 219 340 310 219 340 310 ...
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9 14.5 13.8 15.7 ...
##  $ hto : num  80 70 100 50 70 60 70 50 70 60 ...
# INCREMENTAR UNA FILA
fila<-c("v",5,5.7,210,17.7,80)
examenesAB<-rbind(examenesAB, fila)
# OBSERVAR QUE LA ÚLTIMA FILA ES LA QUE SE HA AGREGADO
tail(examenesAB, 3)
##    Sexo gR  gB plq  hgl hto
## 13    v  6 8.1 290 16.7  70
## 14    m  4   6 314 15.2 100
## 15    v  5 5.7 210 17.7  80
#-------------------------------------------------
#UNIR COLUMNAS CON LA FUNCIÓN cbind()
#-------------------------------------------------
columna<-c(22,25,50,20,30,25,45,22,25,50,24,35,25,45,20)
examenesAB<-cbind(examenesAB, costo=columna)
head(examenesAB, 2)
##   Sexo  gR  gB plq  hgl hto costo
## 1    m 4.3 7.2 124   13  80    22
## 2    v 4.8 6.5 219 16.7  70    25
#-------------------------------------------------
#EXPORTAR DATOS A UN ARCHIVO XLSX
#-------------------------------------------------
writeWorksheetToFile("examenesCopia.xlsx"
                     , data = examenSangreXls
                     , sheet = "examenSangre"
                     , startRow = 1
                     , startCol = 1)
# REVISAR SI EL ARCHIVO HA SIDO GRABADO
file.exists("examenesTratamientoDatos.xlsx")
## [1] FALSE
```

Eliminar filas y/o columnas

De forma básica se ha logrado añadir filas y/o columnas a un dataframe, pero puede darse el caso de que se requiera hacer el proceso inverso; es decir, eliminar filas y/o columnas. Como ejemplo consideremos el conjunto de datos que contiene las siguientes columnas: Sexo, glóbulos rojos (gR), glóbulos blancos (gB), plaquetas (plq), hemoglobina (hgl), hematocritos (hto). Se Plantea el caso de que solo se necesitan las columnas de glóbulos rojos y glóbulos blancos. Además, los datos de hombres no son necesarios. Para eliminar filas del dataframe es necesario crear subconjuntos de datos, esto será estudiado en el siguiente apartado.

Para eliminar las columnas se crea un vector que tenga únicamente las columnas con que se va a trabajar; visto de otra manera, no contiene las columnas que van a ser eliminadas. Se crea un subconjunto de datos llamado glóbulos el cual va a contener el resultado de examenesSangre[,columnas]. Es importante que usted se fije que no especifica nada para las filas, por tanto, el subconjunto de datos contendrá todas las filas del dataframe examensSangre.

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenesSangre=readWorksheet(libroExcel, sheet="examenSangreA")
#VISUALIZAR TRES FILAS DE LOS DATOS
head(examenesSangre, 5)
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenesSangre)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
# ESCOGER LAS COLUMNAS QUE SE REQUIEREN
columnas<-c("Sexo","gR", "gB")
globulos<-examenesSangre[,columnas]
head(globulos, 5)
```

```
##   Sexo  gR  gB
## 1    m 4.3 7.2
## 2    v 4.8 6.5
## 3    m 5.0 8.7
## 4    v 5.9 7.8
## 5    m 4.8 6.4
# ESCOGER LAS FILAS QUE SE REQUIEREN
globulos<-globulos[globulos$Sexo=="m",]
head(globulos, 5)
##   Sexo  gR  gB
## 1    m 4.3 7.2
## 3    m 5.0 8.7
## 5    m 4.8 6.4
```

Crear subconjuntos de datos con la función subset()

Tal vez esta sea una de las partes más importantes del manejo de datos. Un subconjunto de datos contiene una parte del conjunto de datos pero que es de primordial interés. Significa entonces que contendrá los datos con lo que se desea trabajar. La función **subset**() trabaja de la mano con los operadores lógicos, signos de igualdad y desigualdad los cuales servirán para realizar la selección de las filas que debe contener el subconjunto de datos. La función **subset**() necesita el dataframe y los criterios de selección: **subset**(*dataframe, criterios de selección*).

Tome el siguiente ejemplo para desarrollar su código en R: Una de los requerimientos típicos es cuando se establece criterios de selección que incluye los operadores igual que (==), mayor que (>), mayor igual que (>=), menor que (<), menor igual que (<=). Los siguientes ejemplos no tomarán en cuenta la selección de filas con uso de operadores lógicos.

- Seleccionar aquellos exámenes cuyo sexo sea igual a mujer (m)
- Seleccionar aquellos exámenes cuyos glóbulos rojos sean mayor igual que 4.5
- Seleccionar aquellos exámenes cuyos glóbulos blancos sean menor igual a 8.5

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenesSangre=readWorksheet(libroExcel, sheet="examenSangreA")
# VISUALIZAR TRES FILAS DE LOS DATOS
head(examenesSangre, 5)
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
```

```
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenesSangre)
# SALIDA POR CONSOLA
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
# SEXO IGUAL A MUJER M
examMujeres<-subset(examenesSangre, Sexo=="m")
examMujeres
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 3    m 5.0 8.7 290 15.2 100
## 5    m 4.8 6.4 219 13.8  70
# GLÓBULOS ROJOS MAYOR IGUAL A 4.5
Datos_1Criterio<-subset(examenesSangre, gR>=4.5)
Datos_1Criterio
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
# GLÓBULOS BLANCOS MENOR IGUAL QUE 8.5
Datos_1Criterio<-subset(examenesSangre, gB<=8.5)
Datos_1Criterio
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
```

Se ha demostrado lo fácil que es seleccionar filas cuando se usa un solo criterio de selección. Sin embargo, en la mayoría de los casos los criterios de selección suelen ser más de uno, lo que implica adicionar operadores lógicos y concatenarlos. Los tres operadores lógicos más importantes para hacer este trabajo son: Y (**&**), O (|) y diferente de -o negado de- (**!**).

Con estos operadores lógicos se puede seleccionar filas bajo criterios de selección más complejos. Iniciemos el trabajo de selección con los siguientes ejemplos:

Uso del operador lógico **Y** (Se tiene que cumplir todos los criterios para que la fila sea seleccionada)

- Seleccionar aquellos exámenes cuyos glóbulos rojos sean mayores que 4.5 **y** (&) sus glóbulos blancos sean menor a 8.5
- Seleccionar aquellos exámenes cuyos glóbulos rojos sean mayores que 4.5 **y** (&) sus glóbulos blancos sean menor a 8.5 **y** (&) hematocritos < 70

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenesSangre=readWorksheet(libroExcel, sheet="examenSangreA")
# VISUALIZAR TRES FILAS DE LOS DATOS
print("CONJUNTO DE DATOS")
## [1] "CONJUNTO DE DATOS"
ExamenesSangre
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenesSangre)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
#(gR MAYOR QUE 4.5) Y (gB MENOR A 8.5)
print("(gR MAYOR QUE 4.5) Y (gB MENOR A 8.5)")
## [1] "(gR MAYOR QUE 4.5) Y (gB MENOR A 8.5)"
Datos_2Criterios<-subset(examenesSangre, gR>4.5 &  gB<8.5)
Datos_2Criterios
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 2    v 4.8 6.5 219 16.7  70
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
```

```
#(gR MAYOR QUE 4.5) Y (gB MENOR A 8.5) Y (hto MENOR A 50)
print("(gR MAYOR QUE 4.5) Y (gB MENOR A 8.5) Y (hto MENOR A 50)")
## [1] "(gR MAYOR QUE 4.5) Y (gB MENOR A 8.5) Y (hto MENOR A 50)"
Datos_3Criterios<-subset(examenesSangre, gR>4.5 & gB<8.5 & hto<70)
Datos_3Criterios
##   Sexo  gR  gB plq  hgl hto
## 4    v 5.9 7.8 314 14.5  50
## 6    v 6.0 8.1 340 15.7  60
```

Uso del operador lógico **O** (Suficiente que se cumpla un criterio para que la fila sea seleccionada)

- Seleccionar aquellos exámenes cuyos glóbulos rojos sean mayores que 4.5 **o** (|) sus glóbulos blancos sean menor a 8.5
- Seleccionar aquellos exámenes cuyos glóbulos rojos sean mayores que 4.5 **o** (|) sus glóbulos blancos sean menor a 8.5 **o** (|) hematocritos igual a (==) 80

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenesSangre=readWorksheet(libroExcel, sheet="examenSangreA")
# VISUALIZAR TRES FILAS DE LOS DATOS
print("CONJUNTO DE DATOS")
## [1] "CONJUNTO DE DATOS"
examenesSangre
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
# USE str() PARA REVISAR QUE EL NUMERO DE OBSERVACIONES
# Y SOBRE TODO QUE LOS TIPOS DE DATOS LOS CORRECTOS
# PARA CADA VARIABLE
str(examenesSangre)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
# (gR MAYOR QUE 4.5) O (gB MENOR A 8.5)
print("(gR MAYOR QUE 4.5) O (gB MENOR A 8.5)")
## [1] "(gR MAYOR QUE 4.5) O (gB MENOR A 8.5)"
Datos_2Criterios<-subset(examenesSangre, gR>5.0 |  gB<6.5)
Datos_2Criterios
```

```
##   Sexo  gR  gB plq  hgl hto
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
# (gR MAYOR QUE 4.5) O (gB MENOR A 8.5) O (hto IGUAL A 80)
print("(gR MAYOR QUE 4.5) O (gB MENOR A 8.5) O (hto IGUAL A 80)")
## [1] "(gR MAYOR QUE 4.5) O (gB MENOR A 8.5) O (hto IGUAL A 80)"
Datos_3Criterios<-subset(examenesSangre, gR>5.0 | gB<6.5 | hto==80)
Datos_3Criterios
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
```

Uso del operador lógico **diferente de** (suficiente que sea diferente de -el criterio- para que la fila sea seleccionada)

- Seleccionar aquellos exámenes cuyos Sexo sean diferente de mujer (m)
- Seleccionar aquellos exámenes cuyos hematocritos sea diferente de 70

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenesSangre=readWorksheet(libroExcel, sheet="examenSangreA")
# VISUALIZAR TRES FILAS DE LOS DATOS
print("CONJUNTO DE DATOS")
## [1] "CONJUNTO DE DATOS"
examenesSangre
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
# REVISAR QUE EL NUMERO DE OBSERVACIONES, VARIABLES Y TIPOS DE DATOS SEAN
str(examenesSangre)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
```

```
# SEXO DIFERENTE DE MUJER
print("SEXO DIFERENTE DE MUJER")
## [1] "SEXO DIFERENTE DE MUJER"
Datos_1Criterio<-subset(examenesSangre, Sexo !="m")
Datos_1Criterio
##   Sexo  gR  gB plq  hgl hto
## 2    v 4.8 6.5 219 16.7  70
## 4    v 5.9 7.8 314 14.5  50
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
# HEMATOCRITO DIFERENTE DE 80
print("HEMATOCRITO DIFERENTE DE 70")
## [1] "HEMATOCRITO DIFERENTE DE 70"
Datos_1Criterio<-subset(examenesSangre, !(hto==70))
Datos_1Criterio
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 6    v 6.0 8.1 340 15.7  60
```

El uso del operador lógico **Y** combinado con el operador lógico **O** puede traer resultados que quizás no sean los deseados. Esto se debe, en ciertas ocasiones, porque no se agrupa de manera correcta los operadores lógicos. Como ejemplo analicemos el siguiente caso.

- Seleccionar aquellos exámenes cuyos glóbulos rojos sean mayores que 5 **o** (|) sus glóbulos blancos sean menor a 6.5 **o** (|) pero que sus hematocritos sean igual a (==) 70.

Esto significa que existe primero una selección de filas con el criterio de que los glóbulos rojos que sean mayor que 5 **o** (|) que sus glóbulos blancos sean menores a 6.5, una vez realizada esta selección únicamente se escogerán aquellos con hematocrito igual a 70.

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenesSangre=readWorksheet(libroExcel, sheet="examenSangreA")
# VISUALIZAR TRES FILAS DE LOS DATOS
print("CONJUNTO DE DATOS")
## [1] "CONJUNTO DE DATOS"
examenesSangre
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
```

```
# REVISAR QUE EL NUMERO DE OBSERVACIONES, VARIABLES Y TIPOS DE DATOS
str(examenesSangre)
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
# gR MAYOR QUE 5 O gB MENOR A 6.5 Y hto IGUAL A 70
print("MAL AGRUPADO LOS CRITERIOS DE SELECCIÓN")
## [1] "MAL AGRUPADO LOS CRITERIOS DE SELECCIÓN"
Datos_3Criterios<-subset(examenesSangre, gR>5.0 | gB<6.5 & hto==70)
Datos_3Criterios
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
# (gR MAYOR QUE 4.5) O (gB MENOR A 8.5) & (hto IGUAL A 80)
print("CORRECTAMENTE AGRUPADO LOS CRITERIOS DE SELECCIÓN")
## [1] "CORRECTAMENTE AGRUPADO LOS CRITERIOS DE SELECCIÓN"
Datos_3Criterios<-subset(examenesSangre, (gR>5.0 | gB<6.5) & hto==70)
Datos_3Criterios
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 5    m 4.8 6.4 219 13.8  70
## 7    v 5.2 9.3 310 13.9  70
```

Ejemplo con los operadores lógicos **Y**, **O** y **diferente de** para filtrar datos.

- Seleccionar las filas cuyos glóbulos rojos **no** sean mayores que 5, sus glóbulos blancos **no** sean menores que 6.5 y sus hematocritos **no** sean igual a 70.

Para resolver esto, nos percatamos de que la palabra clave NO (negado de) está en todas las condiciones. Por lo tanto, si gR>5.0 & gB<6.5 contiene las filas que no deseo entonces debo negar esa selección para obtener las filas que deseo !(gR>5.0 & gB<6.5). De igual manera con el hematocrito, dice de forma clara NO sea igual a 70, entonces con hto==70 obtengo todas las filas con hematocrito igual a 70 pero me dicen que NO SEA IGUAL A 70 entonces debo escribir !(hto==70) para obtener ese resultado.

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenesSangre=readWorksheet(libroExcel, sheet="examenSangreA")
```

```
# VISUALIZAR TRES FILAS DE LOS DATOS
print("CONJUNTO DE DATOS")
## [1] "CONJUNTO DE DATOS"
ExamenesSangre
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 2    v 4.8 6.5 219 16.7  70
## 3    m 5.0 8.7 290 15.2 100
## 4    v 5.9 7.8 314 14.5  50
## 5    m 4.8 6.4 219 13.8  70
## 6    v 6.0 8.1 340 15.7  60
## 7    v 5.2 9.3 310 13.9  70
#REVISAR QUE EL NUMERO DE OBSERVACIONES, VARIABLES Y TIPOS DE DATOS
str(examenesSangre)
# SALIDA POR CONSOLA
## 'data.frame':    7 obs. of  6 variables:
##  $ Sexo: chr  "m" "v" "m" "v" ...
##  $ gR  : num  4.3 4.8 5 5.9 4.8 6 5.2
##  $ gB  : num  7.2 6.5 8.7 7.8 6.4 8.1 9.3
##  $ plq : num  124 219 290 314 219 340 310
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
#Diferente de ((gR MAYOR QUE 4.5) O (gB MENOR A 8.5)) & diferente de (hto IGU
AL A 80)
print("CORRECTAMENTE AGRUPADO LOS CRITERIOS DE SELECCIÓN")
## [1] "CORRECTAMENTE AGRUPADO LOS CRITERIOS DE SELECCIÓN"
Datos_3Criterios<-subset(examenesSangre, !(gR>5.0 | gB<6.5) & !(hto==70))
Datos_3Criterios
# SALIDA POR CONSOLA
##   Sexo  gR  gB plq  hgl hto
## 1    m 4.3 7.2 124 13.0  80
## 3    m 5.0 8.7 290 15.2 100
```

Tratamiento de datos faltantes

En ocasiones, mientras se trabaja en datos, se puede encontrar que el conjunto de datos tiene datos faltantes (conocidos también como valores perdidos) que R etiqueta como **NA**. Los valores faltantes son un problema complejo de resolver y no se deben desestimar porque potencialmente pueden conducirnos a resultados erróneos. En este texto para principiantes se tratará de forma leve este problema y sus posibles soluciones.

En un conjunto de datos grande puede suceder que contenga un número de valores faltantes muy pequeño (generalmente menos del 5%), en este caso los valores faltantes pueden ignorarse, eliminando las filas que contienen los datos faltantes, y el análisis puede realizarse con el resto de los datos. En otros casos, puede ser que el número de valores faltantes sea demasiado grande por lo que se debe "imputar" estos valores faltantes en lugar de eliminarlos de los datos. Imputar significa la sustitución de valores no informados o faltantes en una observación por otros.

Los valores faltantes generalmente se clasifican en tres tipos: Missing Completely At Random, Missing At Random y Not Missing At Random.

Missing Completely At Random (MCAR) incluyen a aquellos valores faltantes que no están relacionados con ninguna característica de la propia variable y de las otras variables del conjunto de datos; es decir que la probabilidad de que una respuesta a una variable sea dato faltante es independiente tanto del valor de esta variable como del valor de otras variables del conjunto de datos.

Missing At Random (MAR) implica que los valores faltantes pueden explicarse completamente por los datos que ya tenemos. Entonces la probabilidad de que una respuesta sea dato faltante es independiente de los valores de la misma variable pero es dependiente de los valores de otras variables del conjunto de datos.

Not Missing At Random (NMAR) significa que existe una relación entre la propensión de un valor que falta y sus valores. Significa que la probabilidad de que una respuesta a una variable sea dato faltante es dependiente de los valores de la variable.

MCAR y MAR se consideran "ignorables" porque no tenemos que incluir ninguna información sobre los datos que faltan. MNAR se llama "no ignorable" porque el mecanismo de datos faltantes debe modelarse a medida que se manejan los valores perdidos. Debe incluir algún modelo de por qué faltan los datos y cuáles son los valores probables.

La imputación de valores faltantes puede realizarse con los datos sin valores faltantes utilizando algunas medidas estadísticas como la media y la mediana. Por lo tanto, una de las formas más fáciles de completar o "imputar" valores faltantes es completarlos de tal manera que algunas de estas medidas no cambien. Para los datos numéricos, se puede usar la media de los datos para que la media general no cambie. En este proceso, sin embargo, la varianza disminuye y cambia.

Para datos no numéricos, "imputar" con la moda es una opción común. Si hubiéramos predicho el valor probable para datos no numéricos, naturalmente predeciremos el valor que ocurre la mayor parte del tiempo (que es la moda) y es fácil de imputar.

En algunos casos, los valores se imputan con ceros o valores muy grandes para que puedan diferenciarse del resto de los datos. Del mismo modo, imputar un valor faltante con algo que está fuera del rango de valores también es una opción. Sin embargo, estos se usan solo para un análisis rápido. Para los modelos que están destinados a generar conocimiento en firme, los valores faltantes deben ser atendidos de manera razonable. Considere el archivo de Excel mostrado en la figura 1.6 que contiene datos faltantes.

El código que se muestra abajo contiene la lectura del libro de Excel llamado **examenes.xlsx** que contiene la hoja **examenSangre** la cual contiene datos faltantes (NA). Para el primer ejemplo vamos a seleccionar del conjunto de datos llamado examenSangre un subconjunto de datos denominado examenSangreDF con las columnas "Sexo" y "gR". Esto se lo hace para cumplir con la condición del porcentaje de valores faltantes menor o igual al 5%.

Iniciemos el tratamiento de valores faltantes (NA) menor al 5%, tome en cuenta los siguientes pasos:

1. Constatar la existencia de valores faltantes. Uso de la función is.na()
2. Calcular el número de filas sin valores faltantes con la función complete.cases()
3. Calcular el porcentaje de filas con valores mediante la función complete.cases()
4. Si el porcentaje de filas con valores faltantes es menor o igual al 5% entonces eliminar esas filas por medio del uso de la función complete.cases().

	Sexo	gR	gB	plq	hgl	hto
2	m	4.3		124.0	13.0	80.0
3	v	4.8	6.5		16.7	70.0
4	m	5.0	8.7	290.0	15.2	100.0
5	v	5.9		314.0	14.5	
6	m	4.8	6.4		13.8	70.0
7	v		8.1	340.0	15.7	60.0
8	v	5.2	9.3	310.0	13.9	
9	m	5.8	8.7	270.0	13.0	111.0
10	v	5.9		309.0	14.0	
11	m	5.2	7.9		13.8	75.0
12	m	5.1		289.0	15.0	90.0
13	v	5.4	7.0		15.7	70.0
14	m	5.0	9.0	220.0	13.4	87.0
15	v	5.8		250.0	16.0	
16	m	4.4	8.0		15.0	75.0
17	v	5.6	7.1	190.0	14.0	62.0
18	v	5.2	9.3	310.0	13.9	
19	m	5.8	8.7	270.0	13.0	111.0
20	m	5.2	7.9		13.8	75.0

Figura 1.6 *Dataframe datosFaltantesXlsx.png*

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
examenSangre=readWorksheet(libroExcel, sheet="examenSangreDF")
# CONJUNTO DE DATOS PARA CUMPLIR CON LA CONDICIÓN DEL PUNTO 4
columnas<-c("Sexo", "gR")
```

```
examenSangreDF<-examenSangre[,columnas]
# 1. CONSTATAR QUE EXISTEN VALORES FALTANTES
any(is.na(examenSangreDF))
## [1] TRUE
# 2. CALCULAR NÚMERO DE FILAS SIN VALORES FALTANTES
nrow(examenSangreDF[complete.cases(examenSangreDF), ])
## [1] 18
# 3. PORCENTAJE DE FILAS CON VALORES FALTANTES
nrow(examenSangreDF[!complete.cases(examenSangreDF), ])/nrow(examenSangreDF)*
100
## [1] 5.263158
# 4. ELIMINACIÓN DE FILAS- VALORES FALTANTES MENOR AL 5%
datosES<-examenSangreDF[complete.cases(examenSangreDF), ]
#CONJUNTO DE DATOS SIN VALORES FALTANTES
datosES
# SALIDA POR CONSOLA
##    Sexo  gR
## 1     m 4.3
## 2     v 4.8
## 3     m 5.0
## 4     v 5.9
## 5     m 4.8
## 7     v 5.2
## 8     m 5.8
## 9     v 5.9
## 10    m 5.2
## 11    m 5.1
## 12    v 5.4
## 13    m 5.0
## 14    v 5.8
## 15    m 4.4
## 16    v 5.6
## 17    v 5.2
## 18    m 5.8
## 19    m 5.2
```

Para el tratamiento de valores faltantes (NA) mayor al 5% realice los siguientes pasos:

1. Constatar la existencia de valores faltantes. Uso de la función **is.na**()
2. Calcular el número de filas sin valores faltantes con la función **complete.cases**()
3. Calcular el porcentaje de filas con valores mediante la función **complete.cases**()
4. Si el porcentaje de filas con valores faltantes es mayor al 5% es necesario imputar con la función mean() usando la función median() para imputar con la mediana esas filas.

```
# CARGAR LIBRERIA
library(XLConnect)
# CONFIGURAR EL DIRECTORIO DONDE SE ECUENTRA EL LIBRO DE EXCEL
setwd("C:/Users/tiran/Desktop/Datos")
# IMPORTAR LIBRO DE EXCEL
libroExcel=loadWorkbook("examenes.xlsx")
# IMPORTAR DATOS DESDE LA HOJA DESEADA
```

```
examenSangreDF=readWorksheet(libroExcel, sheet="examenSangreDF")
# OBSERVE LAS COLUMNAS CON VALORES FALTANTES gR, gB, plq y hto
examenSangreDF
# SALIDA POR CONSOLA
##    Sexo  gR  gB plq  hgl hto
## 1     m 4.3  NA 124 13.0  80
## 2     v 4.8 6.5  NA 16.7  70
## 3     m 5.0 8.7 290 15.2 100
## 4     v 5.9  NA 314 14.5  NA
## 5     m 4.8 6.4  NA 13.8  70
## 6     v  NA 8.1 340 15.7  60
## 7     v 5.2 9.3 310 13.9  NA
## 8     m 5.8 8.7 270 13.0 111
## 9     v 5.9  NA 309 14.0  NA
## 10    m 5.2 7.9  NA 13.8  75
## 11    m 5.1  NA 289 15.0  90
## 12    v 5.4 7.0  NA 15.7  70
## 13    m 5.0 9.0 220 13.4  87
## 14    v 5.8  NA 250 16.0  NA
## 15    m 4.4 8.0  NA 15.0  75
## 16    v 5.6 7.1 190 14.0  62
## 17    v 5.2 9.3 310 13.9  NA
## 18    m 5.8 8.7 270 13.0 111
## 19    m 5.2 7.9  NA 13.8  75
# 1. CONSTATAR QUE EXISTEN VALORES FALTANTES
any(is.na(examenSangreDF))
## [1] TRUE
# 2. CALCULAR NÚMERO DE FILAS SIN VALORES FALTANTES
nrow(examenSangreDF[complete.cases(examenSangreDF), ])
## [1] 5
# 3. PORCENTAJE DE FILAS CON VALORES FALTANTES
nrow(examenSangreDF[!complete.cases(examenSangreDF), ])/nrow(examenSangreDF)*
100
## [1] 73.68421
# 4.IMPUTACIÓN - VALORES FALTANTES MAYOR AL 5%
# IMPUTACIÓN USANDO LA MEDIA, COLUMNAS gR y gB
examenSangreDF$gR[is.na(examenSangreDF$gR)] <- round(mean(examenSangreDF$gR,
na.rm = TRUE),1)
examenSangreDF$gB[is.na(examenSangreDF$gB)] <- round(mean(examenSangreDF$gB,
na.rm = TRUE),1)
# OBSERVE LAS COLUMNAS CON VALORES FALTANTES plq y hto
examenSangreDF
# SALIDA POR CONSOLA
##    Sexo  gR  gB plq  hgl hto
## 1     m 4.3 8.0 124 13.0  80
## 2     v 4.8 6.5  NA 16.7  70
## 3     m 5.0 8.7 290 15.2 100
## 4     v 5.9 8.0 314 14.5  NA
## 5     m 4.8 6.4  NA 13.8  70
## 6     v 5.2 8.1 340 15.7  60
## 7     v 5.2 9.3 310 13.9  NA
## 8     m 5.8 8.7 270 13.0 111
## 9     v 5.9 8.0 309 14.0  NA
## 10    m 5.2 7.9  NA 13.8  75
## 11    m 5.1 8.0 289 15.0  90
```

```
## 12    v 5.4 7.0  NA 15.7  70
## 13    m 5.0 9.0 220 13.4  87
## 14    v 5.8 8.0 250 16.0  NA
## 15    m 4.4 8.0  NA 15.0  75
## 16    v 5.6 7.1 190 14.0  62
## 17    v 5.2 9.3 310 13.9  NA
## 18    m 5.8 8.7 270 13.0 111
## 19    m 5.2 7.9  NA 13.8  75
# IMPUTACIÓN USANDO LA MEDIANA, COLUMNAS plq y hto
examenSangreDF$plq[is.na(examenSangreDF$plq)] <- median(examenSangreDF$plq, n
a.rm = TRUE)
examenSangreDF$hto[is.na(examenSangreDF$hto)] <- median(examenSangreDF$hto, n
a.rm = TRUE)
# DATAFRAME SIN VALORES FALTANTES
examenSangreDF
# SALIDA POR CONSOLA
##    Sexo  gR  gB plq  hgl hto
## 1     m 4.3 8.0 124 13.0  80
## 2     v 4.8 6.5 289 16.7  70
## 3     m 5.0 8.7 290 15.2 100
## 4     v 5.9 8.0 314 14.5  75
## 5     m 4.8 6.4 289 13.8  70
## 6     v 5.2 8.1 340 15.7  60
## 7     v 5.2 9.3 310 13.9  75
## 8     m 5.8 8.7 270 13.0 111
## 9     v 5.9 8.0 309 14.0  75
## 10    m 5.2 7.9 289 13.8  75
## 11    m 5.1 8.0 289 15.0  90
## 12    v 5.4 7.0 289 15.7  70
## 13    m 5.0 9.0 220 13.4  87
## 14    v 5.8 8.0 250 16.0  75
## 15    m 4.4 8.0 289 15.0  75
## 16    v 5.6 7.1 190 14.0  62
## 17    v 5.2 9.3 310 13.9  75
## 18    m 5.8 8.7 270 13.0 111
## 19    m 5.2 7.9 289 13.8  75
```

El resultado obtenido es el conjunto de datos examenSangreDF sin valores faltantes. Aunque lo realizado no sea lo óptimo; sin embargo, para un curso inicial es suficiente y queda en manos del lector profundizar sobre el problema de los valores faltantes y sus posibles soluciones.

AUTOEVALUACIÓN

Autoevaluación 1-1

Una fábrica elabora tres productos A, B y C, en tamaños grandes y pequeño. Produce diariamente 1050 productos grandes y 6000 pequeños de tipo A; 7000 grandes y 4000 pequeños de tipo B, y 2000 grandes y 5000 pequeños de tipo C.

a) ¿Cuál es el código para representar la información en forma de una matriz?

b) ¿Cuáles los valores máximos y mínimos?

c) ¿Cuál es la media de la producción?

Autoevaluación 1-2

Un ingeniero civil tiene que realizar tres obras de construcción: la obra uno requiere 50 unidades de madera, 25 de ladrillos, 12 de hierro; la obra dos necesita 19, 0 y 15, y la obra tres necesita 40, 18 y 23 respectivamente.

a) ¿Cuál es el código para crear un dataframe?

b) Mostrar los tipos de datos del dataframe

Autoevaluación 1-3

Un adquiere rosa, clavel, lirio, margarita y jazmín de tres proveedores. Los precios de cada proveedor para las flores vienen dados por la matriz en dólares.

a) Crear un archivo de texto plano e importarlo en R

b) Obtener los tipos de datos.

	Rosa	***Clavel***	***Lirio***	***Margarita***	***Jazmin***
P 1	5	7	4	6	9
P 2	9	6	6	9	10
P 3	12	5	2	8	9

EJERCICIOS DEL CAPÍTULO

1. Crear dos vectores y visualizar sus datos por medio de una gráfica, el primero de nombre gRM para los glóbulos rojos de hombres con los siguientes valores 4.7, 6.1, 5.9, 5.4, 6.2 y el segundo gRF con los valores: 4.2, 4.8, 5.4, 6.3 para los glóbulos rojos de las mujeres.

2. Establecer una matriz con el siguiente conjunto de datos sobre el análisis de sangre, ¿Cuáles son los valores máximos, mínimos y la media de los datos?

3. Crear una matriz con el siguiente conjunto de datos sobre el análisis de sangre describa el código para mostrar únicamente los datos de Rosa, Romario y Ruber.

4. Realizar un DataFrame por medio de vectores y describa sus tipos de datos:

- sexo[F,M,F,M,F,M,M],
- gR[4.1,4.7,5.2,.4.9,4.6,6.1,5.9],
- gB[6.2,5.5,7.7,6.8,7.4,9.1,8.3],
- plq[124,219,290,314,219,340,310],

- hgl[13.0,16.7,15.2,14.5,13.8,15.7,13.9],
- hto[80,70,100,50,70,60,70].

5. Obtener un conjunto de datos desde un archivo Excel y describa el código para corregir los tipos de datos de las columnas gR, gB ,plq , hgl y hto que están en caracter y deben ser numéricos, el conjunto de datos se muestra a continuación.

Capítulo 1. Respuestas de las autoevaluaciones

1-1 **a)**
```
produccion<-matrix(c(1050,6000,7000,4000,2000,5000)
                   ,nrow=3
                   ,ncol=2
                   ,byrow=T)
colnames(produccion)<-c("Grande","Pequeños")
rownames(produccion)<-c("A","B","C")
produccion
```

b)
```
MAXIMOS<-apply(produccion, 2, max)
MINIMOS<-apply(produccion, 2, min)
produccion<-rbind(produccion,MAXIMOS)
produccion<-rbind(produccion,MINIMOS)
producción
RESULTADO
MAXIMOS   7000     6000
MINIMOS   1050     4000
```

c)
```
MEDIAS    3620     5000
```

1-2 **a)**
```
Obras<-c("Obra uno", "Obra dos","Obra tres")
madera<-c(50,19,40)
ladrillos<-c(25,0,18)
hierro<-c(12,15,23)
ObrasConstruccion<-data.frame(Obras, madera, ladrillos, hierro)
ObrasConstruccion
RESULTADO
     Obras      madera  ladrillos hierro
1  Obra uno     50        25        12
2  Obra dos     19         0          15
3  Obra tres     40        18          23
```

b)
```
str(ObrasConstruccion)
RESULTADO
'data.frame':   3 obs. of  4 variables:
$ Obras    : Factor w/ 3 levels "Obra dos","Obra tres",..: 3 1 2
$ madera   : num  50 19 40
$ ladrillos: num  25 0 18
$ hierro   : num  12 15 23
```

1-3 **a)**

```
setwd("C:/Users/Ariosto/Desktop")
FloresPrecioProveedor<-read.table("flores.txt", header=TRUE
                                                , sep = "")
head(FloresPrecioProveedor,3)
```

b)

```
str(FloresPrecioProveedor)
```

RESULTADO

```
'data.frame':   3 obs. of  5 variables:
$ Rosa     : int  5 9 12
$ Clavel   : int  7 6 5
$ Lirio    : int  4 6 2
$ Margarita: int  6 9 8
$ Jazmin   : int  9 10 9
```

2 REPRESENTACIÓN GRÁFICA DE LOS DATOS

Existen dos formas de presentar resultados estadísticos: con números y texto; y, como representación gráfica. Es frecuente observar patrones o detectar anomalías en los valores de los datos observando gráficos. Un gráfico bien diseñado puede ayudar a encontrar relaciones significativas entre toda la información, extrayendo patrones con más facilidad que con otros métodos.

Este capítulo revisa los gráficos de uso más frecuente como los de barras y pastel. Posteriormente se verá cómo construir el gráfico de caja y bigote a la par del arreglo de tallo y hojas. Se revisa cómo elaborar los histogramas y polígonos de frecuencia. Junto con ellos, se experimenta la configuración de los parámetros de la función **plot**() para modificar las características de la presentación de cualquier gráfico. Estas características incluyen títulos del gráfico y ejes, etiquetas, colores, líneas, símbolos y anotaciones de texto. Finalmente. se lista los gráficos que se van a contruir:

- Gráfico de Barras
- Gráfico circular o de pastel
- Gráfico de Caja y bigote
- Gráfico de puntos
- Histogramas de frecuencia
- Polígono de frecuencia
- Polígono de frecuencia acumulado
- Gráfico de la densidad de la distribución
- Tallo y hojas

La apariencia de los gráficos que se construyen en esta sección tiene el estilo que se utiliza en las publicaciones científicas. En cuanto a los datos, se utiliza el conjunto de datos **personas** (revise con atención la tabla 2.0). En primera instancia, se preparan los datos en el dataframe **personas** que contiene cuatro columnas, una de tipo carácter (Sexo) y tres de tipo numérico (Edad, Peso, Altura) y treinta filas. Para el efecto, se crea cuatro vectores para cada una de las columnas (Sexo, Edad, Peso, Altura) con la función **c**(). Con estos vectores se procede a componer el dataframe **personas** con la función **data.frame**(). Con el dataframe **personas** listo, se hace uso de las funciones gráficas **barplot**(), **pie**(), **pai3D**(), **boxplot**(), `stripchart(), hist()`. Es necesario el conocer los parámetros de estas funciones para lograr gráficos de buena calidad.

Tabla 2.0 *Conjunto de datos personas*

Sexo	Edad	Peso	Altura	Sexo	Edad	Peso	Altura
m	18	130	153	m	20	148	153
m	28	210	155	m	21	128	154
m	20	114	154	m	19	126	159
v	21	194	153	v	19	162	172
v	19	164	181	m	19	110	174
m	19	126	154	m	21	150	166
v	23	164	170	v	23	144	174
m	19	122	164	m	20	126	158
m	18	116	156	m	19	102	159
m	19	114	158	m	19	170	147
v	23	180	162	v	23	204	174
v	19	166	170	v	19	200	170
m	20	172	169	m	20	120	154
v	20	136	158	v	21	144	169
v	28	166	178	v	25	166	170

El resultado de la construcción del dataframe personas se aprecia en la tabla 2.1. La salida con formato de la tabla se realiza con la función **Kable**(). El parámetro caption se usa para poner el título a la tabla resultante.

```
sexo<-c("m", "m", "m", "m", "m", "m", "v", "v", "v", "m", "m", "m", "v", "v",
"m", "m", "m", "m", "m", "m", "v", "v", "v", "v", "m", "m", "v", "v", "v", "v
" )
edad<-c(18, 20, 28, 21, 20, 19, 21, 19, 19, 19, 19, 21, 23, 23, 19, 20, 18, 1
9, 19, 19, 23, 23, 19, 19, 20, 20, 20, 21, 28, 25)
peso<-c(130, 148, 210, 128, 114, 126, 194, 162, 164, 110, 126, 150, 164, 144,
122, 126, 116, 102, 114, 170, 180, 204, 166, 200, 172, 120, 136, 144, 166, 16
6 )
altura<-c(153, 155, 154, 154, 159, 153, 172, 181, 174, 154, 166, 170, 174, 16
4, 158, 156,   159, 158,147, 162, 174, 170, 170,169, 154, 158, 169, 178, 170,
169)
personas<-data.frame(sexo, edad, peso, altura)
kable(head(personas, 5), caption = "Tabla 2.1 Dataframe personas")
```

Tabla 2.1 *Dataframe personas*

sexo	edad	peso	altura
m	18	130	153
m	20	148	155
m	28	210	154
m	21	128	154
m	20	114	159

Gráfico de Barras

La información contenida en los datos se puede resumir con el uso de una tabla. Sin embargo, las técnicas estadísticas gráficas captan la atención más rápidamente que una tabla de números. Se pueden utilizar dos técnicas gráficas: tanto la del gráfico de barra como la de pastel para resumir los resultados de la tabla 2.1. Un gráfico de barras se usa a menudo para mostrar frecuencias. Un gráfico circular muestra gráficamente las frecuencias relativas. El gráfico de barras se crea dibujando un rectángulo que representa cada categoría. La altura del rectángulo representa la frecuencia. La base es arbitraria[11].

Vamos a crear el diagrama de barras para las variables edades y sexo del dataframe personas. Un diagrama de barras básico en R se construye con la función **barplot**(), que necesita como parámetro principal el conteo de cada valor o característica de la variable que se va a graficar. El conteo lo obtenemos usando la función **table**(). Tome en cuenta que la variable edades es tipo numérico (cuantitativo) y la variable sexo es de tipo caracter (cualitativo).

El siguiente código en R usa la función **table**() para el conteo que se guarda en las variables *resumenEdad* y *resumenSexo* y cuyo resultado se muestra en la tabla 2.2 (Edades) y la tabla 2.3 (Sexo). Con la función **barplot**() se realiza el grafico para cada variable (ver figura 2.1). La función **par**() con el parámetro *mfrow* permite una salida gráfica que contiene los dos gráficos de barras.. Se observa que estos graficos carecen de un título principal y de las etiquetas en cada uno de sus ejes. Sin embargo, ha sido posible representar el conteo de las características de la variable edad y sexo respectivamente. Por tanto, es necesario hacer cambios en los valores de los parámetros de la función **barplot**() para mejorar su presentación.

```
# CONTEO DE LAS EDADES Y DEL SEXO La figura 2.1 exhibe dos gráficos de barras que aún no representa
n de forma adecuada la información
resumenEdad<-table(personas$edad)
resumenSexo<-table(personas$sexo)
kable(resumenEdad, caption = "Tabla 2.2 Resumen de Edades")
```

Tabla 2.2 *Rresumen de Edades*

Var1	Freq
18	2
19	11
20	6
21	4
23	4
25	1
28	2

```
kable(resumenSexo, caption = "Tabla 2.3 Resumen de Sexo")
```

Tabla 2.3 *Resumen de Sexo*

Var1	Freq
m	17
v	13

```
par(mfrow=c(1,2), mar=c(1,2,2,2)) # División de la salida gráfica
                                  # mar=c (abajo, izq, arriba, der)
barplot(table(personas$edad))     # barplot de la variable
barplot(table(personas$sexo))     # barplot de la variable sexo
```

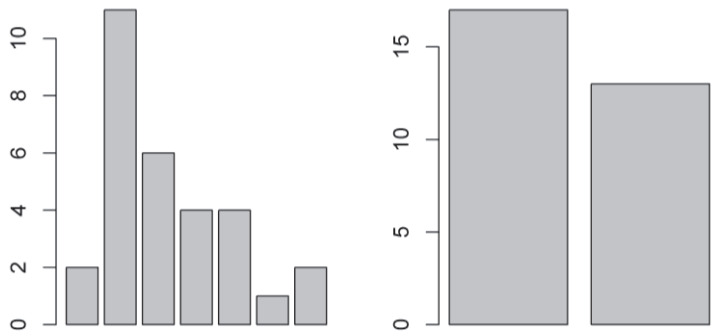


Figura 2.1 *Gráfico de barras básico*

La función **barplot**() tiene el parámetro *main* que ubica un título principal en el gráfico. Los parámetros *xlab* y *ylab* se usan para las etiquetas de los ejes (x,y). Además, el parámetro *col* ajusta el color que se desee dar al gráfico. Como se propuso dar un estilo de gráficos para artículos científicos, se usa una escala de grises. Ver figura 2.2.

```
resumenEdad<-table(personas$edad)
resumenSexo<-table(personas$sexo)
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
par(mfrow=c(1, 2), mar=c(4,3.8,2,2))
barplot(resumenEdad                 # barplot de la variable edad
        , main = "EDADES"           # Título principal
        , xlab="Edad (en años)"     # Etiqueta del eje x
        , ylab="Conteo"             # Etiqueta del eje y
        , col = escalaGrises(10)    # Color del gráfico
)
barplot(resumenSexo                 # barplot de la variable sexo
        , main = "SEXO"             # Título principal
        , xlab="Sexo"               # Etiqueta del eje x
        , ylab="Conteo"             # Etiqueta del eje y
        , col=escalaGrises(5)       # Color del gráfico
)
```

Los colores usados en los diagramas de barras de la figura 2.2 provienen de la variable escalaGrises creada con la función la función **colorRampPalette**() que forma una paleta de colores facilmente. La variable escalaGrises acepta un parámetro que significa la cantidad de tonalidades que serán usadas en el gráfico. En nuestro caso, se tiene una escala de grises que va desde gray100 que es la tonalidad más clara hasta gray50 que es la tonalidad más oscura.

En R existe muchas formas de crear paletas de acuerdo a las diferentes necesidades de apariencia de un gráfico determinado. No se pretende profundizar sobre el manejo de colores en R porque solo se usará colores en escala de grises para la escritura científica, salvo contadas excepciones. El lector puede indagar sobre la creación de paletas usando distintas librerías como **colorspace**(), **ColorBrewer**(), **viridis**(), **shades**(), entre otras. Puede consultar el siguiente enlace para la creación de sus paletas personalizadas: *http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf*

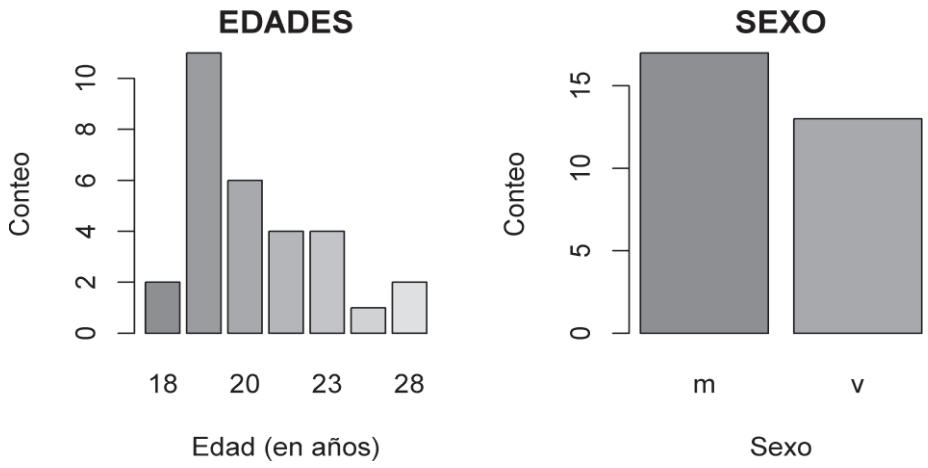


Figura 2.2 *Gráfico de barras con títulos y color*

Paulatinamente se ha mejorado la calidad de los gráficos, ahora es el turno de agregar una leyenda o letrerito que proporcione información sobre el gráfico de barras. Para esto se usa la función **legend**(), que presenta una leyenda que puede agregarse a cualquier gráfico. El primer parámetro que se configura es la posición donde se ubicará la leyenda, existen un conjunto de palabras claves como: "*bottomright*", "*bottom*", "*bottomleft*", "*left*", "*topleft*", "*top*", "*topright*", "*right*" y "*center*" que indican la posición es que se ubicará la leyenda, también se puede ubicar la leyenda por medio de las coordenadas (x, y).

El parámetro *legend* contiene la descripción de la leyenda y *col* los colores de dicha descripción. Con *lty* y *fill* se indica si va a usar líneas o cajas de identificación para los resultados del gráfico. Finalmente, se usa los parámetros *cex* (tamaño de letra), *box.lty* (tipo de línea), *box.lwd* (ancho de la línea) y *box.col* (color de la línea) para configurar la caja que contiene la leyenda. La figura 2.3 muestra una representación de barras mucho más elaborada.

```
resumenEdad<-table(personas$edad)
resumenSexo<-table(personas$sexo)
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
par(mfrow=c(1, 2))
#GRÁFICO DE EDADES
barplot(resumenEdad
        , main = "EDADES"
        , xlab="Edad (en años)"
```

```
        , ylab="Conteo"
        , col = escalaGrises(10)
)
legend("topright"
       , legend=c("18", "19", "20", "21", "23", "25", "28")
       , fill=escalaGrises(10) # Color
       , cex=0.8               # Tamaño de las letras
       , box.lty=1             # Tipo de línea del cuadro
       , box.lwd=1             # Ancho de la línea del cuadro
       , box.col="gray80")     # color del recuadro

#GRÁFICO DE SEXO
barplot(resumenSexo
        , main = "SEXO"
        , xlab="Sexo"
        , ylab="Frecuencia Absoluta"
        , col=escalaGrises(5)
)
legend("topright"
       , legend=c("Mujeres (m)", "Varones (v)")
       , fill=escalaGrises(10) # Color
       , cex=0.8               # Tamaño de las letras
       , box.lty=1             # Tipo de línea del cuadro
       , box.lwd=1             # Ancho de la línea del cuadro
       , box.col="gray80")     # color del recuadro
```

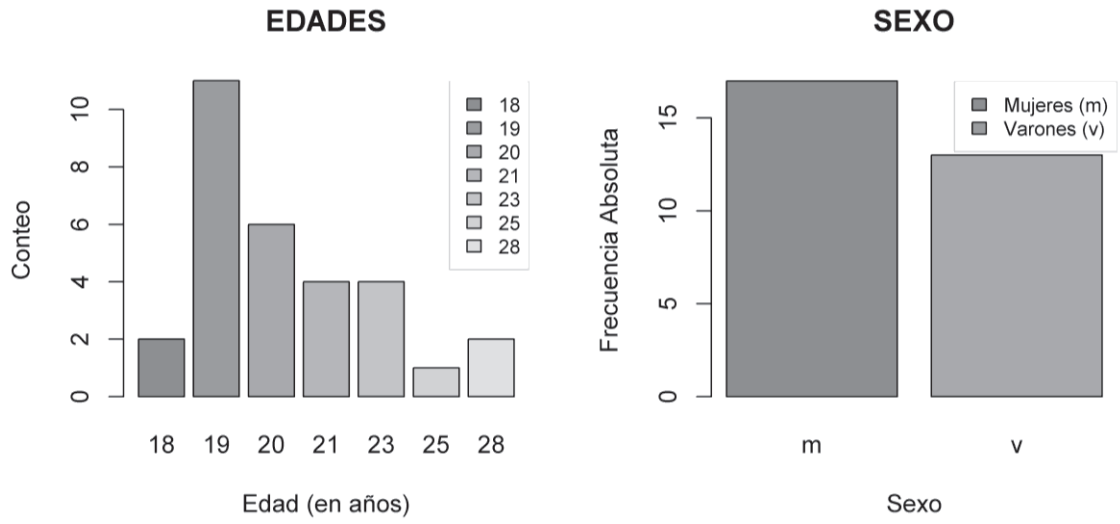


Figura 2.3 *Gráfico de barras con leyenda*

Aunque la figura 2.3 tiene un aspecto más sobrio, aún tiene un detalle que debe ser subsanado: las barras superan la escala que proporcionan el eje y. Para resolver este inconveniente se usa los parámetros *xlim* y *ylim* que redimensiona la escala de los ejes. Veamos el efecto en la figura 2.4.

```
resumenEdad<-table(personas$edad)
resumenSexo<-table(personas$sexo)
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
par(mfrow=c(1, 2))
```

```
barplot(resumenEdad
        , main = "EDADES"
        , xlab="Edad (en años)"
        , ylab="Conteo"
        , col = escalaGrises(10)
        , ylim=c(0, 15)         # Ajusta la escala del eje y
)
legend("topright"
       , legend=c("18", "19", "20", "21", "23", "25", "28")
       , col=escalaGrises(10)
       , fill=escalaGrises(10)
       , cex=0.8
       , box.lty=1
       , box.lwd=1
       , box.col="gray80")
barplot(resumenSexo
        , main = "SEXO"
        , xlab="Sexo"
        , ylab="Conteo"
        , col=escalaGrises(5)
        , ylim=c(0, 20)          # Ajusta la escala del eje y
        , las=1                  # Dirección de la escala eje x
)
legend("topright"
       , legend=c("Mujeres (m)", "Varones (v)")
       , col=escalaGrises(10)
       , fill=escalaGrises(10)
       , cex=0.8
       , box.lty=1
       , box.lwd=1
       , box.col="gray80")
```

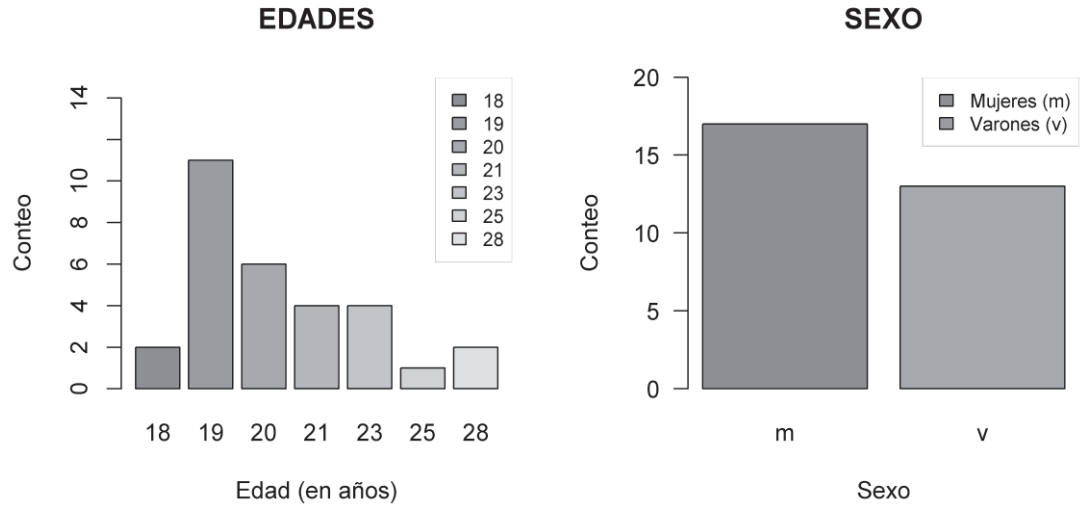


Figura 2.4 *Gráfico de barras con ajuste de ejes*

Finalmente, la figura 2.5 expone uno de los usos más frecuente del gráfico de barras: comparar categorías. Esto se realiza por medio del parámetro *beside*. En ocasiones, es necesario tener el gráfico en sentido horizontal, el parámetro *horiz* permite obtener ese efecto.

```
resumenEdad<-table(personas$edad)
resumenSexo<-table(personas$sexo)
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
par(mfrow=c(1, 2))
barplot(table(personas$sexo, personas$edad)
        , main = "EDADES"
        , xlab="Edad"
        , ylab="Frecuencias Absolutas"
        , col=c("royalblue", "grey")
        , legend.text = c("Mujer", "Varón")
        , ylim = c(0,10)
        , beside=TRUE                    # Barras horizontales
)
barplot(table(personas$sexo, personas$edad)
        , main = "EDADES"
        , xlab="Frecuencias Absolutas"
        , ylab="Edad"
        , col=c("royalblue", "grey")
        , legend.text = c("Mujer", "Varón")
        , xlim = c(0,10)
        , horiz =TRUE                    # Barras horizontales
        , beside=TRUE                    # Más de una barra
)
```

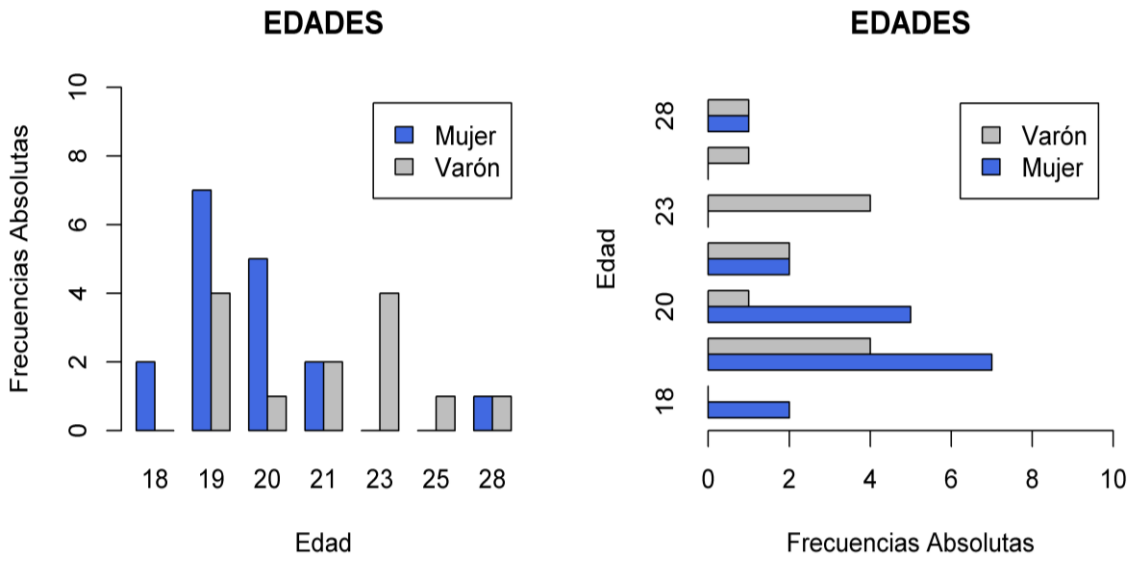


Figura 2.5 *Gráfico de barras comparativo*

La práctica en el uso de los parámetros es fundamental porque puede marcar la diferencia entre una mala o buena presentación de resultados. Las funciones **barplot**(), **par**() y **legend**() son muy utilizadas para la presentación de resultados estadísticos descriptivos.

Pastel

Un gráfico circular presenta los tamaños relativos de las categorías de una variable. El gráfico circular divide un círculo en "sectores" proporcionales a las frecuencias relativas de las categorías [12]. La Figura 2.6 contiene gráfico de pastel de la variable Sexo de tabla 2.0.

Para obtener este gráfico se utiliza tres funciones: **table**(), **prop.table**() y **pie**(). Con **table**() se adquiere la frecuencia absoluta (fa.c) de cada categoría de la variable Sexo (Varones, Mujeres). La función **prop.table**() obtiene la frecuencia relativa (fr.c) de cada categoría a partir de la frecuencia absoluta (fa.c). Por último, se obtiene el gráfico circular o de pastel con la función **pie**(). Los parámetros que se configuran en **pie**() son: *main* para el título del gráfico y *labels* para identificar cada uno de los sectores. Para obtener las etiquetas de los sectores se coloca los nombres de cada uno de ellas en el parámetro *names* y se lo une con la función **paste**() al valor guardado en la variable fr.c, la misma que se redondea a tres decimales con la función **round**().

```
# FRECUENCIAS ABSOLUTAS POR CATEGORÍA
fr.c<-prop.table(fa.c)    # Frecuencias relativas por categoría
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
etq<-paste(names(fr.c)      # Creación de las etiquetas de los sectores
           , paste(round(fr.c*100, 2), "%")
           , sep=" ")
# MÁRGENES DE LA SALIDA GRÁFICA
par(mfrow=c(1,1), mar=c(1, 2, 1, 2))
pie(fr.c                    # Gráfico circular o de pastel
    , main = "SEXO"         # Título principal
    , labels = etq          # Etiquetas
    , col=escalaGrises(5)   # Color de los sectores
    )
```

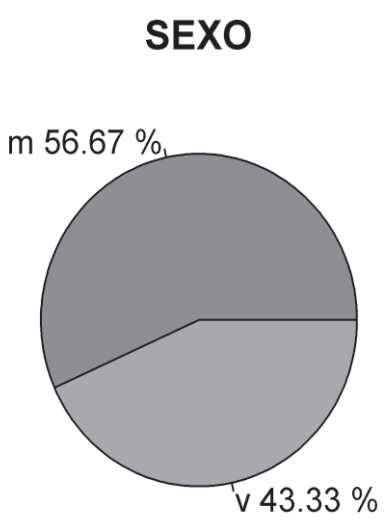


Figura 2.6 *Gráfico de pastel básico*

En ocasiones es necesario manipular el tamaño del gráfico circular para adecuarlo a las necesidades de la presentación que se debe realizar. El parámetro *radius* se encarga de ampliar o reducir el radio del pastel. Por otra parte, como se advirtió en el gráfico de barra, el uso de leyendas es se hace impresindible para identificar de forma solvente la información que se expone en los gráficos en general. La construcción de una leyenda es igual para todos los gráficos, observe el resultado en la figura 2.7.

```
fa.c<-table(personas$sexo)
fr.c<-prop.table(fa.c)
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
etq<-paste(names(fr.c)
           , paste(round(fr.c*100, 2), "%")
           , sep=" ")
par(mfrow=c(1,1), mar=c(1, 2, 1, 2))
pie(fr.c
    , main = "SEXO"
    , labels = etq
    , col=escalaGrises(5)
    , radius = 0.7
    )
legend("topright"
       , legend=c("Mujeres (m)","Varones (v)")
       , fill = escalaGrises(5)
    )
```

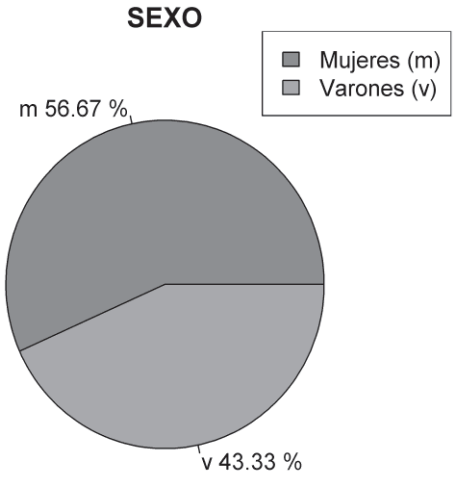


Figura 2.7 *Gráfico de pastel con leyenda*

El gráfico de pastel en tres dimensiones no se implenta en el núcleo de R. Por tanto, se hace necesario utilizar una librería para hacer ese tipo de gráficos. La librería plotrix provee las funciones **pie3d**() y **pie3d.labels**() para crear gráficos de pastel en tres dimensiones.

La función **pie3d**() grafica el pastel y se configura los siguientes parámetros: *radius* (tamaño del pastel), *height* (altura), *explode* (separación de los sectores) y *col* (colores). El gráfico se guarda en una variable (gp) para luego pasarla a la función **pie3d.labels**(). Los parámetros que se configuran son: *labels* (las etiquetas), *labelcex* (tamaño de las letras), *labelrad* (distancia entre las etiquetas y el gráfico de pastel) y *labelcol* (color de las letras). Ver figura 2.8.

```
library(plotrix)            # Cargar librería plotrix
fa.c<-table(personas$sexo)  # Crear la frec absoluta por característica
fr.c<-prop.table(fa.c)      # Crear la frec relativa por característica
escalaGrises<-colorRampPalette(c("grey50", "grey100")) # Paleta de colores
etq<-paste(names(fr.c),     # Etiquetas de los sectores
           paste(round(fr.c*100, 2), "%")
           , sep=" ")
```

```
par(mfrow=c(1,1), mar=c(1, 2, 1, 2))
gp<-pie3D(fr.c              # Se crea el gráfico de pastel
      , main = "SEXO"
      , radius = 1.0        # Radio del pastel
      , height=0.15         # Altura del pastel
      , explode = 0.2       # Separación de los sectores
      , col= escalaGrises(5)# Colores
)
pie3D.labels(gp
             , labels=etq        # Etiquetas
             , labelcex=0.9      # Tamaño de las etiquetas
             , labelrad=2        # Distancia al pastel
             , labelcol="gray20" # Color de las etiquetas
             )
legend("topright"
       , legend=c("Mujeres (m)", "Varones (v)")
       , col=escalaGrises(10)
       , fill=escalaGrises(10)
       , cex=0.8
       , box.lty=1
       , box.lwd=1
       , box.col="gray80")
```

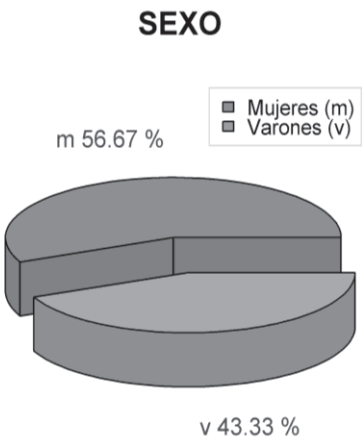


Figura 2.8 *Gráfico de pastel en tres dimensiones*

Como último ejemplo se presenta en la figura 2.9 el gráfico de barras y de pastel juntos. Tome en cuenta que el gráfico de barras, por lo general, se usa para representar las frecuencias absolutas de las categorías de una variable mientras que el gráfico de pastel se enfoca en las frecuencias relativas. Nótese lo fácil de manejar los colores con la función colorRampPalette().

```
library(plotrix)
fa<-table(personas$sexo)
fr<-prop.table(fa)
escalaGrises<-colorRampPalette(c("azure1", "azure4"))
etq<-paste(names(fr), "", paste(round(fr*100, 2), "%"), sep=" ")
par(mfrow=c(1, 2), mar=c(1,2,2,1))
#GRÁFICO DE BARRAS
barplot(fa
        , main = "SEXO"
```

```
        , xlab="Sexo"
        , ylab="Frecuencia Absoluta"
        , col=escalaGrises(5)
        , ylim=c(0, 20)
)
legend("topright"
       , legend=c("Mujeres (m)", "Varones (v)")
       , col=escalaGrises(10)
       , lty=2
       , fill=escalaGrises(10)
       , cex=0.8
       , box.lty=1
       , box.lwd=1
       , box.col="gray80")
#GRÁFICO DE PASTEL
gp<-pie3D(fr
      , main = "SEXO"
      , radius = 1.3
      , height=0.15
      , col= escalaGrises(5)
      , explode = 0.2
)
pie3D.labels(gp
             , labels=etq
             , labelcex=0.9
             , labelcol="gray20"
             , labelrad=2
             )
legend(0.4, 1.2
       , legend=c("Mujeres (m)", "Varones (v)")
       , col=escalaGrises(10)
       , fill=escalaGrises(10)
       , cex=0.8
       , box.lty=1
       , box.lwd=1
       , box.col="gray80")
```

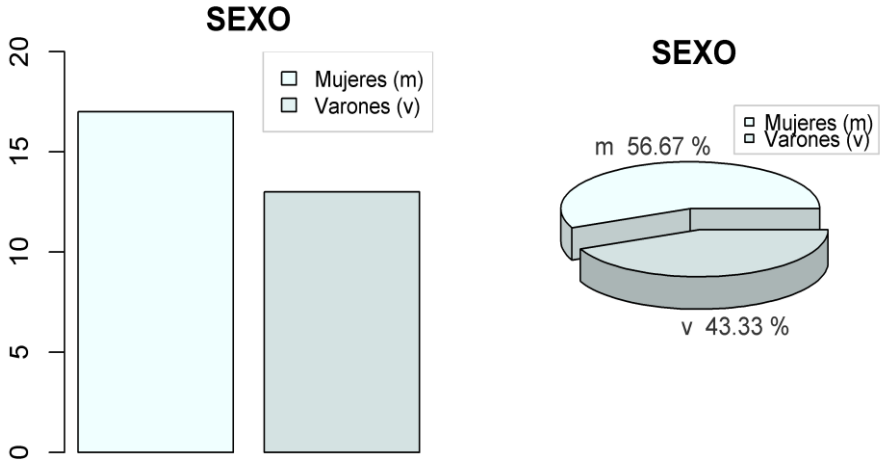


Figura 2.9 *Gráfico de barras y pastel en tres dimensiones*

Gráfico de Caja y bigote

Explica Debore [13] que “en años recientes, se ha utilizado con éxito un resumen gráfico llamado gráfica de caja para describir varias de las características más prominentes de un conjunto de datos. Estas características incluyen 1) el centro, 2) la dispersión, 3) el grado y naturaleza de cualquier alejamiento de la simetría y 4) la identificación de las observaciones “extremas o apartadas” inusualmente alejadas del cuerpo principal de los datos”.

Por otra parte, [14] indica que un diagrama de caja es un resumen de cinco números: una caja central se extiende por los cuartiles Q1 y Q3, una línea en el cuadro marca la mediana y las líneas se extienden desde el cuadro hasta las observaciones más pequeñas y más grandes llamadas bigotes. Cuando se observa un diagrama de caja, primero ubique la mediana, que marca el centro de la distribución. Luego reconozca la propagación fijándose en los cuartiles Q1 y Q2 que muestran la propagación del 50% de los datos, y los extremos (las observaciones más pequeñas y más grandes) que representan la propagación de todo el conjunto de datos.

Una descripción del diagrama de caja y bigote lo da [15] de la siguiente manera: “El cuadro central muestra la mitad central de los datos, entre los cuartiles. Como la parte superior de la caja está en el cuartil superior (Q3) y la parte inferior está en Q1, la altura de la caja es igual a Q3 - Q1, que es el IQR. La mediana se muestra como una línea horizontal. Si la mediana está aproximadamente centrada entre los cuartiles, entonces la mitad central de los datos es aproximadamente simétrica. Si no está centrado, la distribución está sesgada. Los bigotes se extienden desde la caja a los valores más extremos que no se consideran valores atípicos. El diagrama de caja designa los puntos como valores atípicos si caen más allá de 1.5*IQR más allá de cualquier cuartil. Los valores atípicos se muestran individualmente. Los diagramas de caja son especialmente útiles para comparar varias distribuciones una al lado de la otra."

Para R resulta muy fácil crear un gráfico de caja y bigote mediante la función **boxplot**(). Su forma más elemental se plasma en la figura 2.10. A la función **boxplot**() únicamente se le ha pasado el conjunto de datos (edad). También se presenta un diagrama de caja y bigote con título y etiqueta en el eje x usando los parámetros *main* y *xlab* respectivamente. Además, se manipula el tamaño de la letra del parámetro *main* y *xlab* con *cex.main* y *cex.lab*.

```
par(mfrow=c(1,2), mar=c(3,2,2,2))
boxplot(edad                # Conjunto de datos
        , data=personas)
mtext(side=1, line=2, "Boxplot básico", cex=1.2)
boxplot(edad                # Conjunto de datos
        , data=personas
        , main= "Edades"    # Título principal
```

```
        , cex.main=1.4      # Tamaño del título
        , ylab="Edad (Años)"# Etiqueta del eje y
        , cex.lab=0.9       # Tamaño de la etiqueta del eje y
        , col = "gray90"    # Color de la caja
        , frame.plot=FALSE  # Recuadro
)
mtext(side=1, line=0.5, "Estudiantes")
mtext(side=1, line=2, "Boxplot con etiquetas", cex=1.2)
```

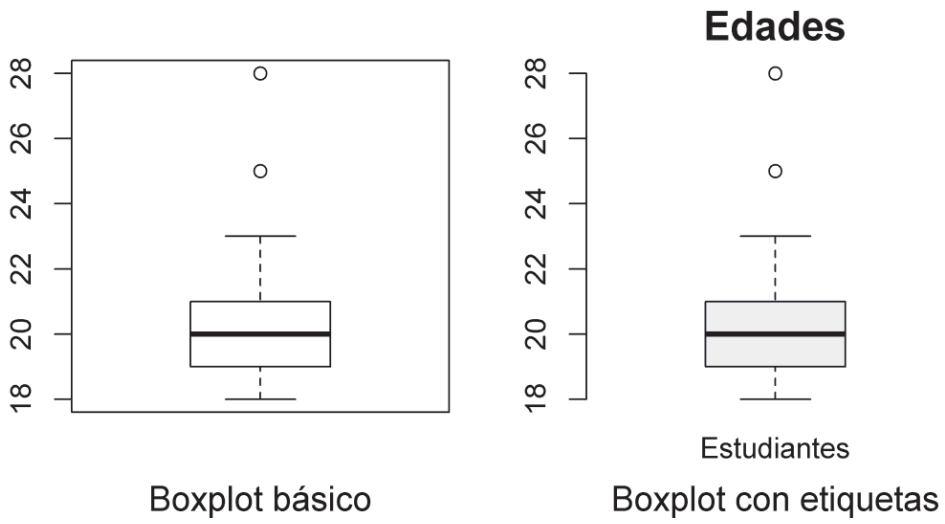


Figura 2.10 *Gráfico de Caja y Bigote elemental*

La figura 2.11 propone la visualización de las variables cualitativas del conjunto de datos personas (tabla 2.0) por medio del diagrama de caja y bigote, siendo esta característica una de las más potentes porque facilita hacer comparaciones entre variables. Se presentan en dos perspectivas: vertical y horizontal. Esto se logra configurando el parámetro *horizontal*. Otro aspecto es que a veces suele ocurrir que los nombres de los ejes podrían resultar extensos y se sobreponen por lo que es conveniente girarlos usando el parámetro las. El cambio del tamaño de las letras y/o números tanto de las etiquetas como de los ejes se logra a través de los parámetros *cex.lab* y *cex.axis* respectivamente.

```
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
par(mfrow=c(1,2), mar=c(3.5,2.5,2,2))
boxplot(personas[,-1]
        , main= "Personas"
        , horizontal=FALSE      # Giro del gráfico
        , col = escalaGrises(4) # Color de la caja
        , cex.lab=0.8           # Tamaño de letra de las etiquetas
        , cex.axis=0.8          # Tamaño de letra de los ejes
        , las=1                 # Giro de las letras del eje x
                , frame.plot = FALSE
)
mtext(side=1, line=2.5, "Boxplot vertical", cex=1.2)
boxplot(personas[,-1]
        , main= "Personas"
        #, xlab="Años-Kg-m"
```

```
        , ylab="Características"
        , horizontal=TRUE        # Giro del gráfico
        , col = escalaGrises(4)  # Color de la caja
        , cex.lab=0.8            # Tamaño de letra de las etiquetas
        , cex.axis=0.8           # Tamaño de letra de los ejes
        , frame.plot = FALSE     # Recuadro
)
mtext(side=1, line=2.5, "Boxplot horizontal", cex=1.2)
```

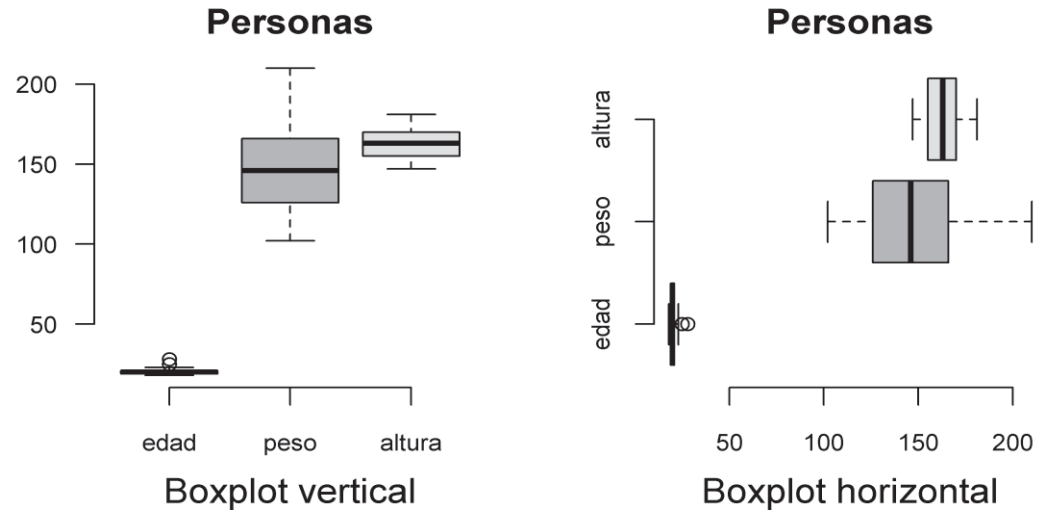


Figura 2.11 *Gráfico de Caja y Bigote vertical y horizontal*

Una característica útil del gráfico de caja y bigote es la presentación de valores atípicos. Estos valores son todos aquellos que superen a Q1-5.5*IQR y Q3+1.5*IQR. A veces no es posible observar los valores atípicos porque están más allá de los límites del gráfico que R propone, véase el gráfico 2.12, aparentenmente es una distribución sin valores atípicos. Para subsanar este inconveniente se usa el parámetro *range* para reducir la escala del gráfico y toma valores entre 0 y 1. El ancho de la caja puede ser redimensionado con el parámetro *boxwex*.

```
escalaGrises<-colorRampPalette(c("grey50", "grey100"))
par(mfrow=c(1,2), mar=c(3.5,2.5,2,2))
boxplot(personas[,-1]
        , main= "Personas"
        , ylab="Características"
        , col = escalaGrises(4)
        , cex.lab=0.8
        , cex.axis=0.8
        , las=1
        , horizontal = TRUE
        , frame.plot=FALSE
)
grid()
mtext(side=1, line=2.5, "Boxplot sin valores atípicos", cex=1.2)
boxplot(personas[,-1]
        , main= "Personas"
        , ylab="Características"
        , col = escalaGrises(4)
```

```
        , cex.lab=0.8
        , cex.axis=0.8
        , las=1
        , horizontal = TRUE
        , range = 0.5               # Rango
        , ann=TRUE                  # Habilita los títulos
        , boxwex=0.5                # Ancho de la caja
        , frame.plot=FALSE
)
grid()
mtext(side=1, line=2.5, "Boxplot con valores atípicos", cex=1.2)
```

El gráfico de caja y bigote es uno de los más importantes gráficos descriptivos; por lo tanto, trabajaremos para darle una apariencia de calidad.

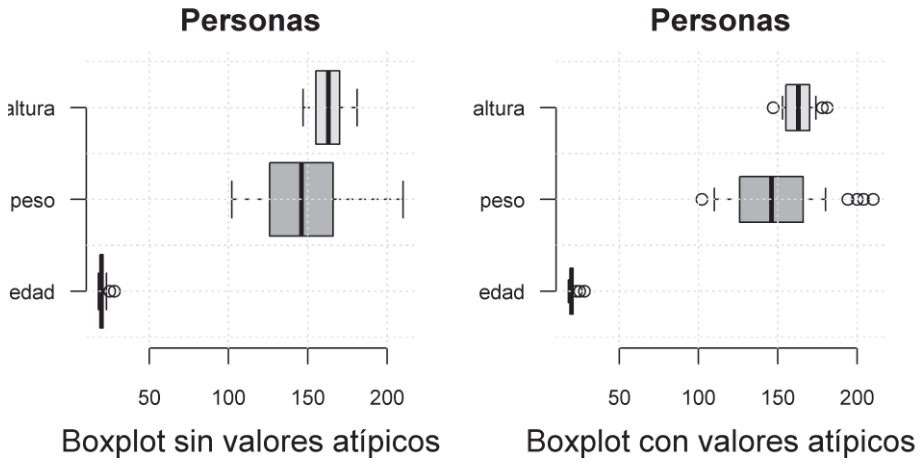


Figura 2.12 *Gráfico de Caja y Bigote con valores atípicos*

```
escalaGrises<-colorRampPalette(c("grey70", "grey100"))
par(mfrow=c(1,2), mar=c(4.1,5,2,2))
boxplot( formula =  peso ~ edad         # Formula
         , data = personas              # Conjunto de datos
         , range = 0.5                  # Rango
         , main= "Peso ~ Edad"          # Título principal
         , xlab=""                      # Etiqueta del eje x
         , ylab = "Peso (kg)"           # Etiqueta del eje y
         , cex.axis=0.8                 # Tamaño de letra en los ejes
         , frame.plot = FALSE           # Recuadro
         , col = escalaGrises(10)       # Color del recuadro
         , staplewex = 0.5              # Ancho del bigote
         , staplecol = "grey70"         # color del bigote
         , whisklty = 1                 # Línea hacia el bigote
         , whiskcol = "grey70"          # Color de la línea hacia el bigote
         , boxwex = .70                 # Ancho de la caja
         , boxcol = "grey70"            # Color de la caja
         , outpch = 20                  # Punto de los valores atípicos
         , outcex = .5                  # Tamaño de letra de valores atípicos
         , outcol = "grey60"            # Color valores atípicos
         , medlty = 3                   # Tipo de línea de la mediana
         , medpch = 21                  # Tipo de punto
         , medlwd = 0                   # Ancho de la línea
```

```
)
grid()
mtext(side=1, line=2, "Edad (Año)", cex=1)
mtext(side=1, line=3.1, "Boxplot de dos variables", cex=1.2)
boxplot( personas[,-1]                  # Formula
         , data = personas              # Conjunto de datos
         , range = 0.5                  # Rango
         , main="Personas"              # Título principal
         , cex.axis=1
         , frame.plot = FALSE           # Recuadro
         , col = escalaGrises(4)        # Color del recuadro
         , staplewex = 0.5              # Ancho del bigote
         , staplecol = "grey70"         # color del bigote
         , whisklty = 1                 # Línea hacia el bigote
         , whiskcol = "grey70"          # Color de la línea hacia el bigote
         , boxwex = .5                  # Ancho de la caja
         , boxcol = "grey70"            # Color de la caja
         , outpch = 21                  # Punto de los valores atípicos
         , outcex = 1.2                 # Tamaño de letra de valores atípicos
         , outcol = "darkorchid4"       # Color valores atípicos
         , medlty = 3                   # Tipo de línea de la mediana
         , medpch = 21                  # Tipo de punto
         , medlwd = 0                   # Ancho de la línea
)
grid()
mtext(side=1, line=3.1, "Boxplot de n variables", cex=1.2)
```

La figura 2.13 muestra el resultado de la configuración de los parámetros de la función **boxplot**(). El código está agrupado de la siguiente manera: datos (*fórmula, data, range*), etiquetas(*main, xlan, ylab*), recuadro(*frame.plot, col*), bigote(*staplewex, staplecol, whiskty, whiskcol*), caja (*boxwex, boxfill, boxcol*), valores atípicos(*outpch, outcex, outcol*) y línea de la mediana(*medlty, medpch, medlwd*). Vea el interesante resultado obtenido.

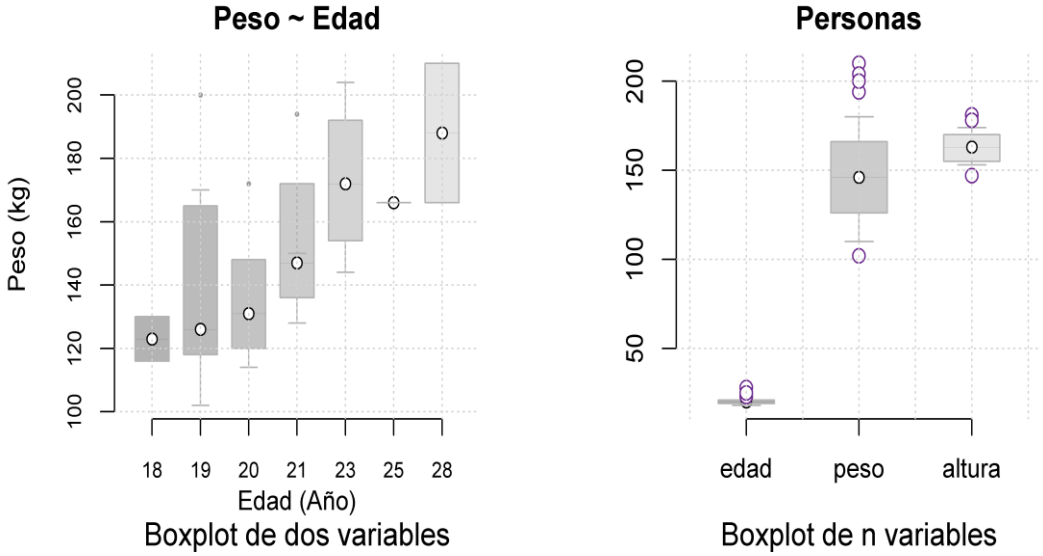


Figura 2.13 *Gráfico de Caja y Bigote con valores atípicos*

Gráfico de puntos

El grafico de puntos es una representación estadística relativamente simple que generalmente se usa para mostrar datos cuantitativos continuos. En un gráfico de puntos, cada valor de datos se representa a lo largo del eje horizontal mediante un punto. Si varios puntos tienen los mismos valores, los puntos se irán ubicando verticalmente. Si hay una gran cantidad de puntos cercanos, puede que no sea posible mostrar todos los valores a lo largo del eje horizontal. Los gráficos de puntos pueden ser especialmente útiles para observar la forma general de la distribución de puntos de datos junto con la identificación de valores de datos o intervalos para los que hay agrupaciones y espacios en los datos [16].

Lo que se busca en el diagrama de puntos básicamente es un valor representativo o típico en el conjunto de datos, la medida en que se extienden los valores de los datos, la naturaleza de la distribución de valores a lo largo de la recta numérica y la presencia de valores inusuales en el conjunto de datos. [17]

La tabla 2.0 tiene datos de sexo y edad, se formula usar el gráfico de puntos para mostrar la distribución de la edad de acuerdo al sexo. Un diagrama de puntos puede desplegarse en R con la función **stripchart**(). El código presentado para crear la figura 2.14 contiene los siguientes elementos: datos (*y~x, data*), etiquetas (*main, xlab, ylab, group.names, las*), presentación (*frame.plot, vertical*), punto (*method, offset, pch, col*). Tenga cuidado con el uso del parámetro *method* que puede el valor de stack o jitter. La figura 2.14 muestra los métodos (stack y jitter) con los que se puede visualizar un diagrama de puntos. Por lo general, el método stack se utiliza en contraste con el histograma de frecuencias y el método jitter como apoyo al diagrama de caja y bigote.

```
escalaColor<-colorRampPalette(c("darkorchid1", "darkorchid4"))
par(mfrow=c(1,2), mar=c(4.2,5,2,2))
stripchart( edad ~ sexo            # Fórmula
            , data = personas      # Conjunto de datos
            , main="Personas"      # Título principal
            , xlab=""              # Etiqueta del eje x
            , cex.axis=0.8         # Tamaño de letra en los ejes
            , group.names=c("Mujeres", "Varones") # Etiquetas del eje y
            , las=1                # Dirección de las etiquetas
            , frame.plot = FALSE   # Recuadro
            , vertical=FALSE       # Orientación de los ejes
            , method = "stack"     # Métodos: jitter, stack. Si Method=stack
            , offset=0.5           # distancia entre los puntos
            , pch=16               # Caracter usado para el punto
            , col=c("olivedrab4", "lightsteelblue4") # Color
 )
mtext(side=1, line=2, "Edad (Año)", cex=0.8)
```

```
mtext(side=1, line=3.1, "Método stack", cex=1.2)
stripchart( edad ~ sexo            # Fórmula
           , data = personas       # Conjunto de datos
           , main="Personas"       # Título principal
           , xlab=""               # Etiqueta del eje x
           , cex.axis=0.8          # Tamaño de letra en los ejes
           , group.names=c("Mujeres", "Varones") # Etiquetas del eje y
           , las=1                 # Dirección de las etiquetas
           , frame.plot = FALSE    # Recuadro
           , vertical=FALSE        # Orientación de los ejes
           , method = "jitter"     # Métodos: jitter, stack
           , pch=21                # Caracter usado
           , col=c("olivedrab4", "lightsteelblue4") # Color
 )
mtext(side=1, line=2, "Edad (Año)", cex=0.8)
mtext(side=1, line=3.1, "Método jitter", cex=1.2)
```

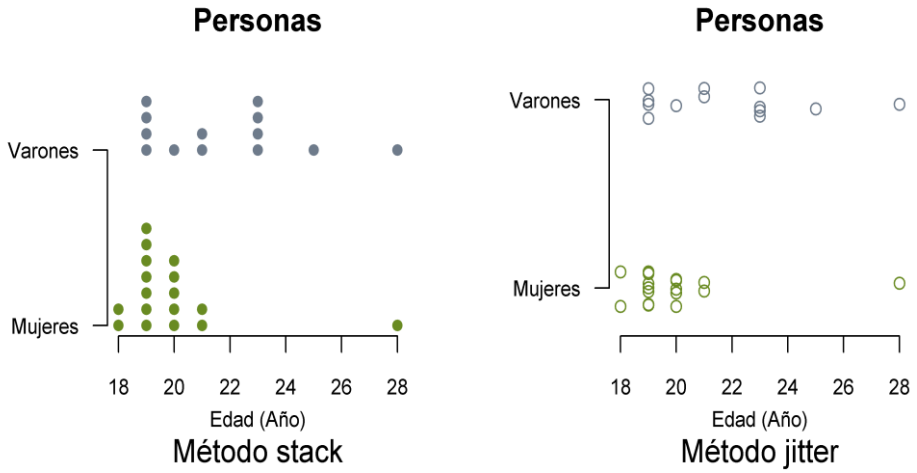


Figura 2.14 *Gráfico de puntos*

De acuerdo a lo manifestado, se presenta el uso de la función **stripchart**() en combinación con la función **boxplot**(). Entonces tenemos un gráfico de caja y bigote que muestra la distribución por medio de los cuartiles y la mediana que incluye un gráfico de puntos que ubica justamente los puntos representativos de la distribución en el gráfico de caja y bigote. El gráfico 2.15 muestra la potencia de esta combinación.

```
escalaColor<-colorRampPalette(c("darkorchid1", "darkorchid4"))
par(mfrow=c(1,1), mar=c(4,3.8,2,2))
boxplot(peso ~ edad                    # y=f(x)
        , data = personas              # Datos
        , main = "Personas"            # Título principal
        , xlab = "Edad (Años)"         # Etiqueta del eje x
        , ylab = "Peso (Kg)"           # Etiqueta del eje y
        , ylim=c(90, 230)              # Límites del eje y
        , frame.plot = FALSE           # Recuadro
        , range=0.7                    # Escala
        , staplewex = 0.5              # Ancho del bigote
        , staplecol = "grey70"         # color del bigote
        , whisklty = 1                 # Línea hacia el bigote
```

```
        , whiskcol = "grey70"          # Color de la línea hacia el bigote
        , boxwex = .5                  # Ancho de la caja
        , boxfill="gray90"             # Color de la caja
        , boxcol = "grey70"            # Color de la caja
        , medlty = 0                   # Tipo de línea de la mediana
        , medpch = 21                  # Tipo de punto de la mediana
        , medlwd = 0                   # Ancho de línea de la mediana
        )
stripchart(peso ~ edad
            , data = personas
            , frame.plot = FALSE       # Recuadro
            , col=escalaColor(20)
            , vertical=TRUE
            , method = "jitter"        # Métodos: jitter, stack
            , offset=1
            , pch=21
            , add=TRUE
 )
grid()
```

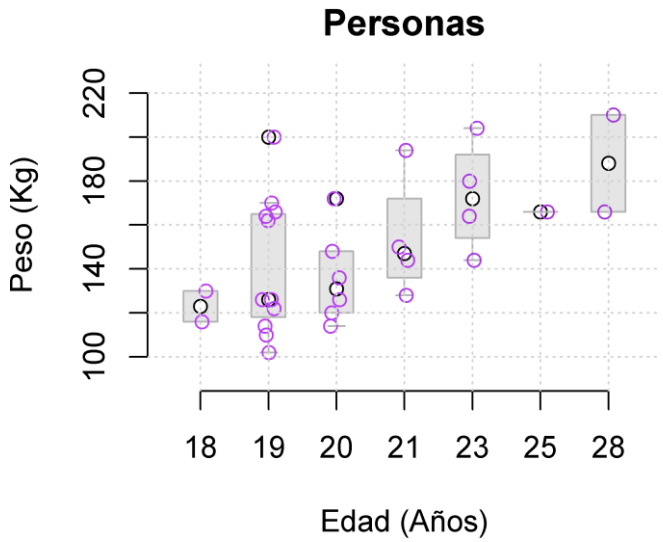


Figura 2.15 *Gráfico de Caja y Bigote aplicando el método jitter a los puntos*

Existe documentación sobre todos los parámetros de la función stripchart() en la dirección web: *https://rdrr.io/cran/EnvStats/man/stripChart.html*. Es importante que el lector manipule estos parámetros para mejorar el uso y presentación de esta función.

Tablas e Histogramas de frecuencia

Alfredo Díaz Mata[18] considera que las tablas e histogramas de frecuencia son: "La forma más conveniente de resumir conjuntos numerosos de datos es a través de tablas y gráficas, ya que permiten condensar la información y, al mismo tiempo, facilitan la apreciación de su contenido. Por lo general, para construir gráficas es necesario resumir primero los datos en una tabla y, por ello, la relación entre estas dos formas de presentación de datos es estrecha". También es

importante anotar el aporte de [19] cuando afirma que una distribución de frecuencia muestra el número (la frecuencia) de elementos en cada una de varias clases que no se superponen. Sin embargo, lo que interesa a menudo es la proporción, o el porcentaje, de elementos en cada clase. La frecuencia relativa de una clase es igual a la fracción o proporción de elementos que pertenecen a cada clase.

Para crear una tabla de frecuencia, primero se debe determinar el número de clases (divisiones del intervalo) por medio de la ecuación 2.0, siendo n el número de observaciones.

$$Número\ de\ clases\ (k) = \sqrt{n} \qquad [2.0]$$

Se representa con la letra k al número de clases con que se va a construir la tabla e histograma de frecuencia; este número se lo puede obtener mediante la raíz cuadrada de n (número de observaciones) y en ocaciones por tanteo. Es útil comprobar la inecuación 2.1 de tal forma que el número de clases (k) resulte apropiado para agrupar los datos.

$$2^k <= Total\ de\ observaciones \qquad [2.1]$$

Determinado el número de clases (k), se calcula el intervalo de las clases usando medio de la ecuación 2.2.

$$Intervalo\ de\ clase = Valor\ Máximo - Valor\ Mínimo \ [2.2]$$

Obtenido el intervalo de clases, es el momento de calcular el ancho de cada clase mediante la ecuación 2.3

$$Ancho\ de\ clase = \frac{Intervalo\ de\ clase}{Número\ de\ clases} \qquad [2.3]$$

Con el ancho de clase se determinan los límites de cada clase. La clase inicial (k=1) toma el valor mínimo y se establece como límite inferior de la clase; se suma el ancho de la clase para obtener el límite superior de la clase; y, así sucesivamente hasta alcanzar el valor máximo que estaría contenido en la clase k. Establecido los límites de cada clase se procede a ubicar los valores del conjunto de datos. La ubicación de estos valores dentro de las clases se da distribuyendo cada valor dentro de los límites de clase que correspondan, convirtiéndose en **datos agrupados**. Para datos cuantitativos, la **frecuencia de una clase** representa el número de observaciones que están dentro de cada clase.

Tablas de frecuencia

Para construir tablas de frecuencias, el core de R provee la función **cut**() que divide el intervalo de un vector (x) en clases y agrupa los valores de x en cada intervalo según corresponda. Para revisar

este apartado vamos a tomar las variables altura y peso de la tabla 2.0. La función **cut**() tiene parámetro *breaks* que indica el número de clases, *include.lowest* sirve para incluir los valores que caen en el límite inferior de la primera clase, *dig.lab* establece con cuantos decimales se necesita trabajar. Finalmente, uno de los parámetros más importantes es *right* que establece como se tomarán los intervalos: [a, b) o (a, b].

En el siguiente ejemplo, se toma la variable peso del dataframe personas con la función **cut**() para obtener la frecuencia absoluta (F.Absoluta). A partir de este resultado, se aplica la función **cumsum**() que realiza sumas acumuladas, obteniendo así la frecuencia absoluta acumulada (F.A.Acumulada). La Frecuencia relativa se la obtiene usando la función **prop.table**() pasándole la frecuencia absoluta como parámetro. El último paso es unir todo con la función **cbind**() para obtener la tabla 2.2

```
Clases <- cut(personas$peso        # Datos
                   , breaks = 6    # Número de clases
                   , include.lowest = TRUE # Incluye los límites
                   , dig.lab = 2   # No de dígitos de los intervalos
                   , right = TRUE  # Intervalos [a, b). Por defecto: (a, b]
)
F.Absoluta <- as.data.frame(table(Clases))   # Crea la tabla de frecuencias
names(F.Absoluta)[2]=c("F.Absoluta")
#LA FRECUENCIA ACUMULADA
F.A.Acumulada <- cumsum(F.Absoluta$F.Absoluta)
F.Relativa<- round(prop.table(F.Absoluta$F.Absoluta),3)
F.R.Acumulada <- round(cumsum(prop.table(F.Absoluta$F.Absoluta)),3)
TablaFrecuencias <- cbind(F.Absoluta, F.A.Acumulada, F.Relativa, F.R.Acumulad
a)
kable(TablaFrecuencias, caption = "Tabla 2.2 Frecuencias")
```

Tabla 2.2 *Frecuencias*

Clases	F.Absoluta	F.A.Acumulada	F.Relativa	F.R.Acumulada
[1e+02,1.2e+02]	6	6	0.200	0.200
(1.2e+02,1.4e+02]	7	13	0.233	0.433
(1.4e+02,1.6e+02]	4	17	0.133	0.567
(1.6e+02,1.7e+02]	8	25	0.267	0.833
(1.7e+02,1.9e+02]	1	26	0.033	0.867
(1.9e+02,2.1e+02]	4	30	0.133	1.000

Histograma de frecuencia Absoluta

Un histograma consiste en una serie de rectángulos, cuyo ancho de clase fue calculado por medio de la ecuación 2.3 y su altura es equivale al número de elementos que están dentro de cada clase. Como consecuencia, el área contenida en cada rectángulo (base por altura) ocupa un porcentaje del área total de todos los rectángulos la cual es igual a la frecuencia absoluta de esa clase

correspondiente respecto a todas las observaciones hechas[20]. De acuerdo a [21] un histograma muestra tres tipos de información:

1. Proporciona una indicación visual de dónde está el centro aproximado de los datos. El punto central a lo largo de los ejes horizontales en los histogramas identifica claramente el lugar donde se centran los datos.
2. Se obtiene una comprensión del grado de propagación (o variación) en los datos. Cuanto más se agrupan los datos alrededor del centro, menor es la variación en los datos. Si los datos se extienden desde el centro, los datos exhiben una mayor variación.
3. Podemos observar la forma de la distribución.

Para hacer un histograma de frecuencia (absoluta) con R se usa la función **hist()**, la figura 2.16 muestra los histogramas de frecuencias absolutas para el peso y la altura del dataframe personas. Los histogramas mostrados son los más elementales y toman el título que le da R por defecto. Poco a poco iremos mejorando la estética del histograma de frecuencia a medida que incrementemos la configuración de sus parámetros.

```
par(mfrow=c(1,2), mar=c(3,2,2,2))
hist(personas$peso)    # Histograma de la variable peso
hist(personas$altura)  # Histograma de la variable altura
```

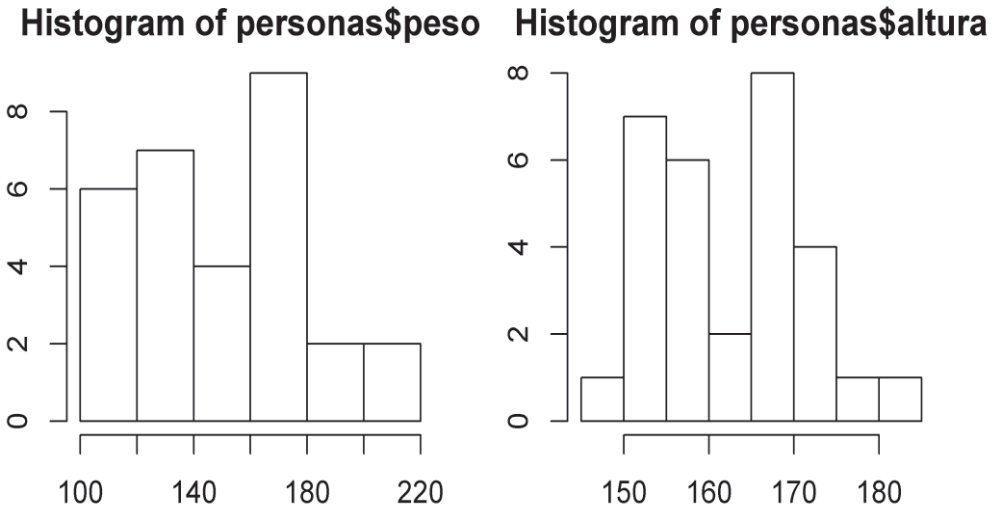


Figura 2.16 *Histograma de frecuencias*

En la figura 2.16 no se muestra información sobre lo que representan los dos histogramas. Es necesario personalizar la información sobre los histogramas para que nos oriente sobre su significado. Es momento de agregar, en ambos histogramas, el título principal (*main*) y los nombres de los ejes x (*xlab*) y (*ylab*). Observe en la figura 2.17 como cambia el resultado visual de los histogramas. Sin embargo, a pesar de su mejora, se evidencia que los ejes (x, y) no están

correctamente dimensionados. Los parámetros *xlim* y *ylim* solucionan este problema ubicando los límites que se consideren convenientes mediante un vector **c**(*límite inicial, límite final*).

```
# PREPARACIÓN DE LA SALIDA GRÁFICA
par(mfrow=c(1,2), mar=c(4,4,2,2))
# HISTOGRAMA PARA LA VARIABLE PESO
hist(personas$peso           # Datos
     , main="Personas"       # Título principal
     , cex.main=1            # Tamaño del título
     , xlab="Peso (libras)"  # Título del eje x
     , ylab = "F. Absoluta") # Título del eje y
# HISTOGRAMA PARA LA VARIABLE ALTURA
hist(personas$altura         # Datos
     , main="Personas"       # Título principal
     , cex.main=1            # Tamaño del título
     , xlab="Altura (m)"     # Título del eje x
     , ylab = "F. Absoluta") # Título del eje y
```

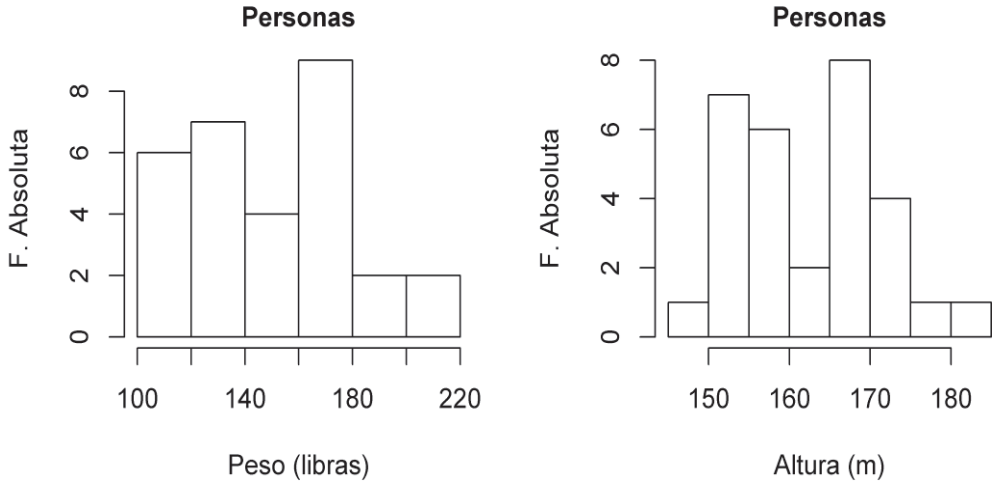


Figura 2.17 *Histograma de frecuencias*

Una vez corregidas las dimensiones de los ejes apreciamos su cambio en la figura 2.18

```
par(mfrow=c(1,2), mar=c(4,4,2,2))
hist(personas$peso
     , main="Personas"
     , cex.main=1           # Tamaño del título
     , xlab="Peso (libras)"
     , ylab ="F. Absoluta"
     , xlim=c(90,230)       # Ajuste del eje x
     , ylim=c(0,10)         # Ajuste del eje y
)
hist(personas$altura
     , main="Personas"
     , cex.main=1           # Tamaño del título
     , xlab="Altura (m)"
     , ylab ="F. Absoluta"
     , xlim=c(141,189)      # Ajuste del eje x
     , ylim=c(0,10)         # Ajuste del eje y
)
```

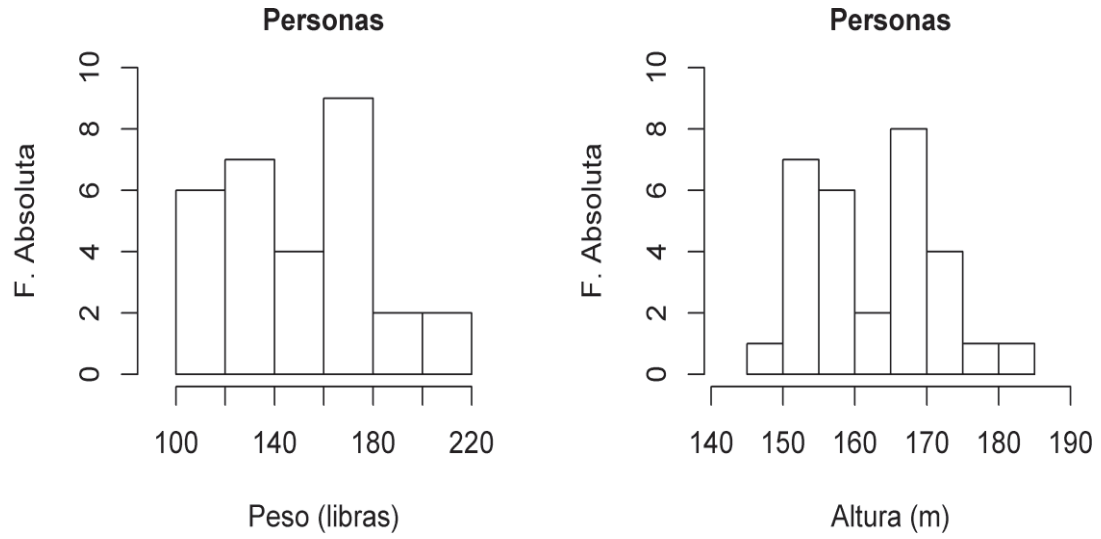


Figura 2.18 *Histograma de frecuencias con ajuste de ejes*

El lector conoce como crear un histograma de frecuencia y un gráfico de caja y bigote por separado. Lo siguiente será unir ambos gráficos para obtener una visualización más potente de los datos. Previamente se usa las funciones **layout**() y **par**() para configurar la salida gráfica. La función **layout**() concibe una división de la salida gráfica mientras que la función **par**() en unión del parámetro *mar* pone los límites de la división realizada. También es importante eliminar el eje x del histograma con el parámetro *xaxt* y su valor en "none". La función **boxplot**() se configura *ylim* y *add*. El parámetro *ylim* del **boxplot**() debe ser igual a *xlim* de la función **hist**(), la razón de esta igualdad es que el histograma está en sentido vertical y el boxplot en sentido horizontal y ambos límites deben ser iguales para que se acoplen correctamente. Por último, en la función **boxplot**() el parámetro *add* debe tomar el valor de TRUE para que se integre al histograma creado con la función **hist**(). Vea la figura 2.19.

```
#PRIMERA FORMA
layout(matrix(c(1,2,1,3), 2, 2, byrow = TRUE)
       , widths=c(2,2)
       , heights=c(3.5,2.2)) # División de la salida gráfica
par(mar=c(4.7,3.8,2,2))      # Márgenes
hist(personas$peso           # Datos
     , main="Personas"
     , cex.main=1            # Tamaño del título
     , xlab=""
     , ylab = "Frecuencia"
     , col="grey90"          # Color del histograma
     , xlim=c(90,230)        # Límite del eje x
     , ylim=c(0,10)          # Límite del eje y
     , border = "darkgrey"   # Líneas del histograma
)
grid()
boxplot(personas$peso        # Gráfico de caja y bigote
     , horizontal=TRUE       # Caja en el sentido horizontal
```

```
     , ylim=c(90,230)        # Límite de y(boxplot)=x(hist)
     , frame=F               # Recuadro
     , col = "grey80"        # Color de la caja
     , width = 10            # Ancho de la caja
     , add=TRUE)             # Habilitado para insertarse
                             # en el histograma
#Texto inferior
mtext(side=1, line=2.1, "Peso (libras)", cex = 0.8)
mtext(side=1, line=3.6, "Histograma con gráfico de caja y bigote integrado")
#SEGUNDA FORMA
par(mar=c(0,3.8,2,2))        # Márgenes
hist(personas$peso           # Datos
     , main="Personas"
     , cex.main=1            # Tamaño del título
     , xlab=""
     , ylab = "Frecuencia"
     , xlim=c(100,220)       # Límite del eje x
     , ylim=c(0,10)          # Límite del eje y
     , col="grey90"          # Color del histograma
     , border = "darkgrey"   # Color del borde del histograma
     , xaxt="none")          # Deshabilitar el eje x
grid()
par(mar=c(4.5,3.8,0,2))      # Márgenes
boxplot(personas$peso        # Datos
     , horizontal=TRUE       # Caja en el sentido horizontal
     , xlab=""
     , ylim=c(100,220)       # Límite de y(boxplot)=x(hist)
     , frame=F               # Sin marco
     , col = "grey80"        # Color de la caja
     , boxwex = 0.8)         # Ancho de la caja
#Texto inferior
mtext(side=1, line=2.1, "Peso (libras)", cex = 0.8)
mtext(side=1, line=3.3, "Histograma con gráfico de caja y bigote separado")
```

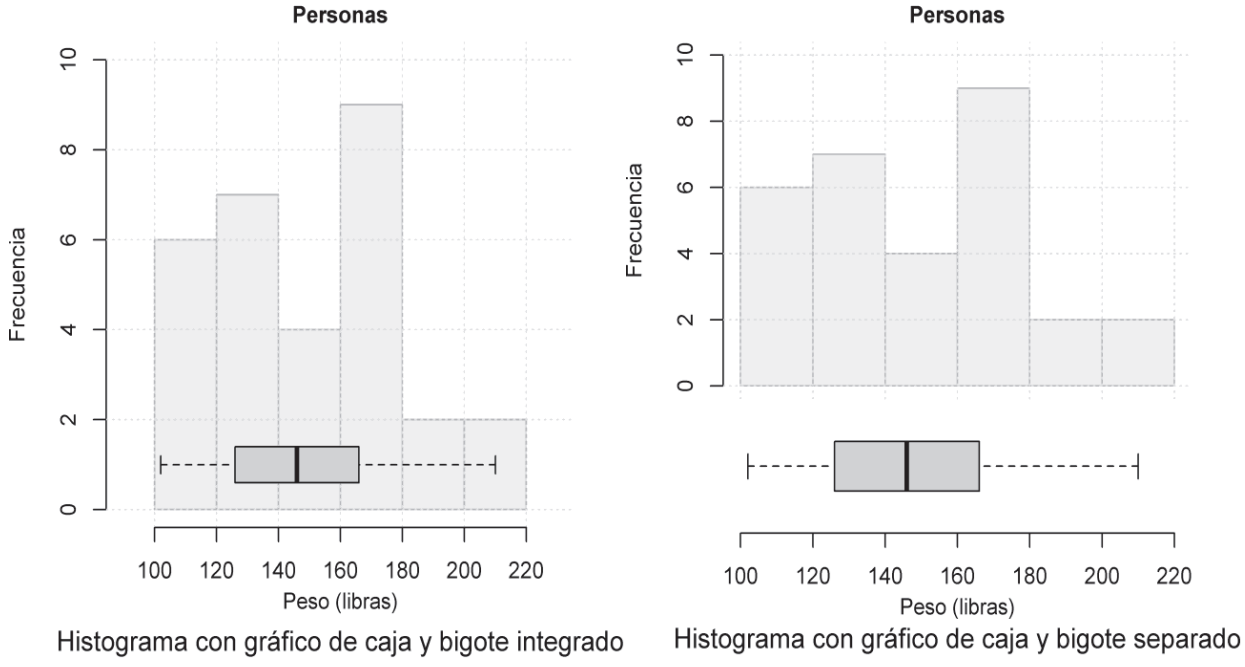


Figura 2.19 *Gráfico de Caja y Bigote con valores atípicos*

Histograma de Frecuencia Relativa

La frecuencia relativa indica por su parte el valor porcentual que representa la frecuencia absoluta de cada clase de la distribución de frecuencia. La ecuación 2.4 resume como se realiza el cálculo de la frecuencia relativa.

$$Frecuencia\ relativa\ de\ la\ clase = \frac{Frecuencia\ absoluta\ de\ la\ clase}{Suma\ de\ todas\ las\ frecuencias} \qquad [2.4]$$

La función **hist**() muestra de forma predeterminada la frecuencia absoluta de un conjunto de datos en el eje y. El histograma de frecuencia relativa se consigue configurando el parámetro *probability* con el valor de TRUE de la función **hist**(). La figura 2.20 muestra las frecuencias relativas de las variables peso y altura del dataframe personas.

```
par(mfrow=c(1,2), mar=c(4, 4, 2, 2))
hist(personas$peso           # Datos
     , probability = "TRUE"  # Frecuencia relativa
     , main="Personas"       # Título principal
     , cex.main=1            # Tamaño del título
     , xlab="Peso (lb)"      # Etiqueta del eje x
     , ylab = "Porcentaje"   # Etiqueta del eje y
     , cex.lab=0.8           # Tamaño de letra de las etiquetas
     , xlim = c(80,240)      # Límite del eje x
     , ylim = c(0.00, 0.02)  # Límite del eje y
     , col="grey90"          # Color del histograma
     , border = "darkgrey"   # Color del borde del histograma
)
grid()
hist(personas$altura         # Datos
     , probability = "TRUE"  # Frecuencia relativa
     , main="Personas"       # Título principal
     , cex.main=1            # Tamaño del título
     , xlab="Altura (m)"     # Etiqueta del eje x
     , ylab = "Porcentaje"   # Etiqueta del eje y
     , cex.lab=0.8           # Tamaño de letra de las etiquetas
     , xlim = c(140,200)     # Límite del eje x
     , ylim = c(0.00, 0.06)  # Límite del eje y
     , col="grey90"          # Color del histograma
     , border = "darkgrey"   # Color del borde del histograma
)
grid()
```

Otra forma de observar los resultados de la figura 2.20 es utilizando dos histogramas en un solo gráfico, vea la figura 2.21. Para lograr este efecto es necesario que los parámetros *xlim* y *ylim* de ambos histogramas sean iguales, analice el siguiente código y observe que en ambos histogramas se configura *xlim* con el límite (80,250) y *ylim* con (0.00, 0.06). Como se trata de frecuencias relativas el parámetro *probability* debe estar en TRUE.

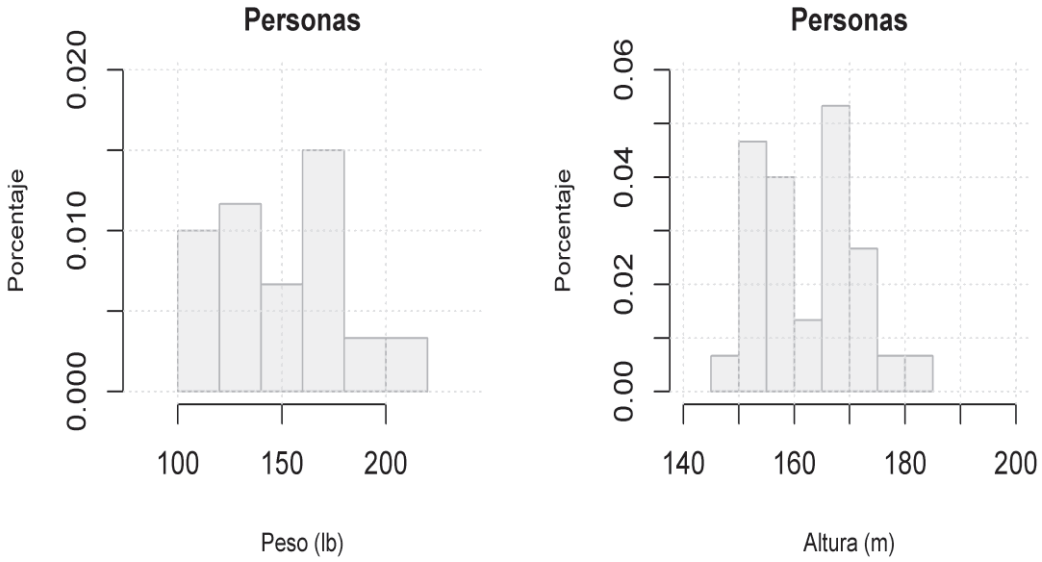


Figura 2.20 *Histograma de frecuencias relativas*

```
par(mfrow=c(1,1), mar=c(2, 4, 2, 2))
hist(personas$peso           # Datos
     , probability = "TRUE"  # Frecuencia relativa
     , main="Personas"       # Título principal
     , cex.main=1            # Tamaño del título
     , xlab=""               # Etiqueta del eje x
     , ylab = "Porcentaje"   # Etiqueta del eje y
     , cex.lab=0.8           # Tamaño de letra de las etiquetas
     , xlim = c(80,250)      # Límite del eje x
     , ylim = c(0.00, 0.06)  # Límite del eje y
     , col="azure1"          # Color del histograma
     , border = "darkgrey"   # Color del borde del histograma
)
grid()
hist(personas$altura         # Datos
     , probability = "TRUE"  # Frecuencia relativa
     , cex.lab=0.8           # Tamaño de letra de las etiquetas
     , xlim = c(140,250)     # Límite del eje x
     , ylim = c(0.00, 0.06)  # Límite del eje y
     , col="grey70"          # Color del histograma
     , border = "darkgrey"   # Color del borde del histograma
     , add=TRUE              # Habilitado para mostrarse en un
)                            # gráfico previo
legend("topright"
       , legend=c("Peso (lb)", "Altura (m)")
       , col=c("azure1", "gray70")
       , fill=escalaGrises(10)
       , cex=0.8
       , box.lty=1
       , box.lwd=1
       , box.col="gray80")
```

El último elemento que se debe configurar para acoplar ambos histogramas de frecuencia en un solo gráfico esta relacionado con el parámetro *add*. Este parámetro agrega un gráfico sobre otro

que este disponible en la salida gráfica, de tal forma que no se elimine. Debe configurarse en el segundo histograma con el valor de TRUE.

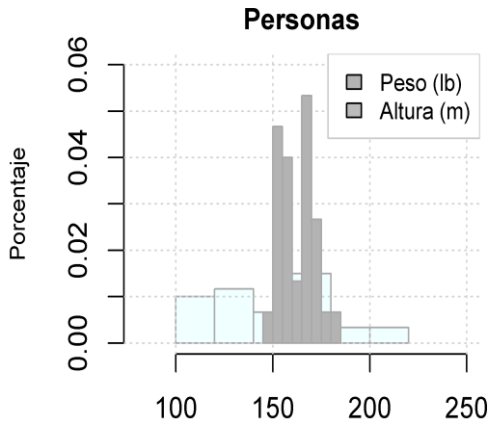


Figura 2.21 *Frecuencias relativas de dos variables*

Polígono de frecuencia

Los polígonos de frecuencias son otra forma de representar gráficamente distribuciones de frecuencia tanto absolutas como relativas [22]. El polígono de frecuencia es un gráfico que exhibe los datos mediante el uso de una línea que va conectando puntos los puntos medios establecidos para cada frecuencias de las clases de la distribución; las frecuencias están representadas por las alturas de los puntos [23].

Para construir el polígono de frecuencias relativas primero se calcula los puntos medios de cada clase. Se dibuja el histograma de frecuencia y se rotula el eje x con el punto medio de cada clase, es importante usar una escala adecuada en el eje y de las frecuencias. Con los puntos medios para los valores de x y con los valores del eje y para las frecuencias relativas se ubica los puntos en la cima de cada clase. Se dibuja una línea de desde (limiteInferior, 0) hasta el punto medio de la primera clase, lo siguiente es conectar los puntos medios de cada clase adyacente con segmentos de línea, por último se traza un segmento de línea desde el punto medio de la última clase hasta el punto (limiteSuperior, 0).

Para obtener los datos necesarios para la construcción del polígono se debe conocer el objeto que retorna la función **hist**(). Para el caso de estudio, vamos a capturar el objeto que retorna la función **hist**() en una variable h.peso de la siguiente manera:

```
h.peso<-hist(personas$peso,…)
```

Luego, se tiene las siguientes líneas:

```
x<- c(min(h.peso$breaks),h.peso$mids,max(h.peso$breaks))
y<- c(0, h.peso$counts, 0)
```

La variable x es un vector que contiene: el límite inferior (`min(h.peso$breaks)`), todos los puntos medios (`h.peso$mids`) y el límite superior (`max(h.peso$breaks)`). La variable y es un vector que contiene: la altura (0) que corresponde al límite inferior, la frecuencia de cada clase (`h.peso$counts`) y la altura (0) del límite superior.

Con estos puntos se forman los pares ordenados (x,y) enviados a la función **lines**() para crear el polígono de frecuencias :

- Punto inicial: (min(h.peso$breaks),0)
- Puntos intermedios: (h.peso$mids, h.peso$counts)
- Punto final: (max(h.peso$breaks),0)

El valor mínimo del eje x concuerda con el valor de cero en el eje y siendo coherente con el inicio del gráfico de cualquier polígono de frecuencia que inicia en el punto (Xmin,0) siendo x el valor mínimo de la distribución de frecuencias. Luego se debe trazar líneas hasta los puntos medios con una altura igual a las frecuencias (punto medio de la clase, frecuencia de la clase) y se concluye el gráfico en el punto (Xmax,0) siendo Xmax el valor máximo de la distribución de frecuencias. Veamos el polígono de frecuencia para las variables peso y altura del dataframe personas en la figura 2.22.

```
par(mfrow=c(1,2), mar=c(4, 4, 1, 2)
h.peso<-hist(personas$peso
        , probability = FALSE
        , main="Personas"
        , cex.main=1            # Tamaño del título
        , xlab="Peso (lb)"
        , ylab = "Porcentaje"
        , cex.lab=0.8
        , xlim = c(50,250)
        , ylim = c(0.00, 10)
        , col="grey90"
        , border = "darkgrey"
        , lty=1
        , lwd=1)
x<- c(min(h.peso$breaks),h.peso$mids,max(h.peso$breaks))
y<- c(0, h.peso$counts, 0)
lines(x, y
      , type = "b"
      , pch = 1
      , col = "blue"
      , lwd = 1)
grid()
h.altura<-hist(personas$altura
        , probability = FALSE
        , main="Personas"
        , cex.main=1            # Tamaño del título
```

```
        , xlab="Altura (m)"
        , ylab = "Porcentaje"
        , cex.lab=0.8
        , xlim = c(140,190)
        , ylim = c(0.00, 10)
        , col="grey90"
        , border = "darkgrey"
        , lty=1
        , lwd=1)
x<- c(min(h.altura$breaks),h.altura$mids,max(h.altura$breaks))
y<- c(0, h.altura$counts, 0)
lines(x, y
      , type = "b"
      , pch = 1
      , col = "blue"
      , lwd = 1)
grid()
```

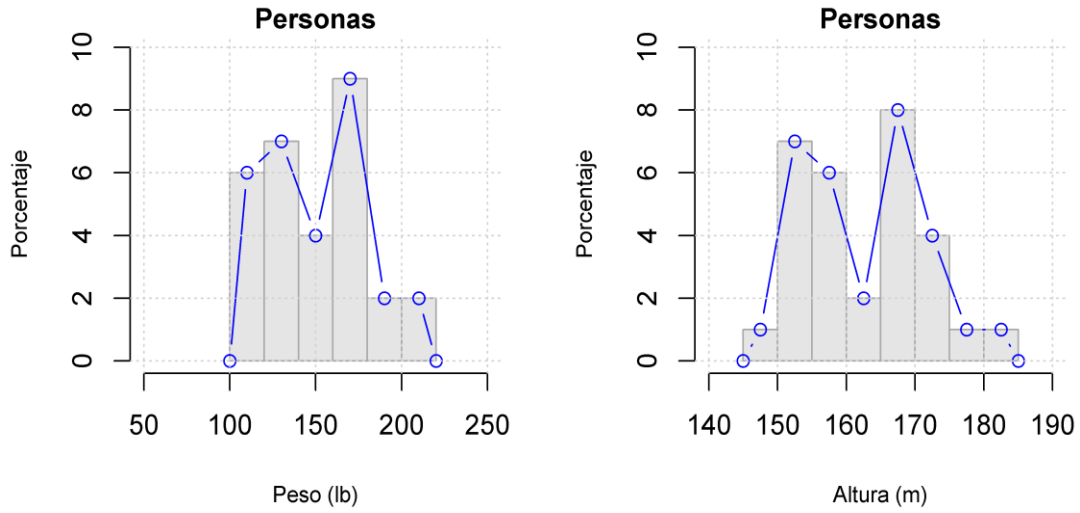


Figura 2.22 *Histograma y Polígono de frecuencias*

Histograma de frecuencia acumulado

Una distribución de frecuencia acumulativa da el número total de valores que se encuentran por debajo del límite superior de cada clase. En una tabla de distribución de frecuencia acumulativa, cada clase tiene el mismo límite inferior pero un límite superior diferente[24]. Usando el core de R, el gráfico del histograma de frecuencias acumulado se lo construye en dos fases. La primera fase es crear una variable que contenga el objeto creado por la función **hist**(), en nuestro caso será h.peso. En h.peso$counts se encuentra los valores de la frecuencia absoluta que vamos a transformar con el uso de la función **cumsum**() que realiza la suma acumulativa de forma muy simple: `h.peso$counts<-cumsum(h.peso$counts)`. Se tiene en este momento la suma acumulativa guardada en el objeto histograma h.peso que mediante el uso de la función **plot**(h.peso,…) vamos a graficar para obtener el histograma de frecuencias acumuladas. Revise el código de abajo y su resultado en la figura 2.23.

```
par(mfrow=c(1,2), mar=c(4, 4, 1, 2))
h.peso<-hist(personas$peso
        , probability = FALSE
        , main="Personas"
        , cex.main=1            # Tamaño del título
        , xlab=""
        , ylab = "Porcentaje"
        , cex.lab=0.8
        , xaxt="n"
        , yaxt="n"
        , xlim = c(80,240)
        , ylim = c(0.00, 10)
        , col="grey90"
        , border = "darkgrey"
        , lty=1
        , lwd=1)
grid()
axis(1, cex.axis=0.8, col.axis="blue")
axis(2, cex.axis=0.8, col.axis="blue")
mtext(side=1, line=2, "Peso (lb)")
mtext(side=1, line=3, "Histograma de frecuencias", cex=0.95)
h.peso$counts<-cumsum(h.peso$counts)
plot(h.peso
     , main="Personas"
     , cex.main=1           # Tamaño del título
     , xlab = ""
     , ylab = "Porcentaje"
     , cex.lab=0.8
     , border = "grey"
     , type="h"
     , lty=1
     , xaxt="n"
     , yaxt="n"
     , col="grey90")
grid()
axis(1, cex.axis=0.8, col.axis="blue")
axis(2, cex.axis=0.8, col.axis="blue")
mtext(side=1, line=2, "Peso (lb)")
mtext(side=1, line=3, "Histograma de frecuencias acumulado", cex=0.95)
```

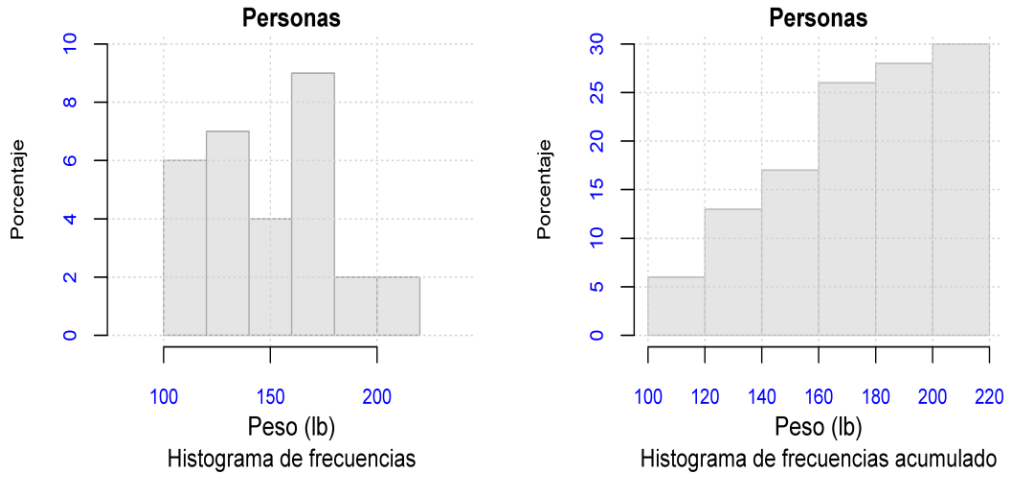


Figura 2.23 *Histograma de frecuencia acumulado*

Polígono de frecuencia acumulado

La gráfica de una distribución acumulada, llamada ojiva, muestra los valores de datos sobre el eje horizontal, y las frecuencias acumuladas, las frecuencias relativas acumuladas o las frecuencias porcentuales acumuladas, sobre el eje vertical[25]. La ojiva tiene como característica que sus líneas se unen en los puntos (límite superior de cada clase, frecuencia acumulada de cada clase), con dos excepciones: la primera línea que nace en el punto (limite inferior, 0) y la última línea que finaliza en el punto (límite superior, frecuencia acumulada de la última clase).

Para crear la ojiva vamos a utilizar la variable x que va a contener todos los límites de la clase: `x<-c(h.peso$breaks),` previo a graficar el histograma de frecuencias acumuladas se transforma las frecuencias absolutas de h.peso$counts en frecuencias acumuladas mediante `h.peso<-cumsum(h.peso$counts)`. La variable y va a contener las frecuencias acumuladas de cada clase mediante `y<-c(0, h.peso$counts).` El polígono de frecuencias y la ojiva se muestra en la figura 2.24

```
par(mfrow=c(1,2), mar=c(5, 4, 1, 2))
h.peso<-hist(personas$peso
        , probability = FALSE
        , main="Personas"
        , cex.main=1          # Tamaño del título
        , xlab=""
        , ylab = "Porcentaje"
        , cex.lab=0.8
        , xaxt="n"
        , yaxt="n"
        , xlim = c(80,240)
        , ylim = c(0.00, 10)
        , col="grey90"
        , border = "darkgrey"
        , lty=1
        , lwd=1)

x<- c(min(h.peso$breaks),h.peso$mids,max(h.peso$breaks))
y<- c(0, h.peso$counts, 0)
lines(x, y, type = "b", pch = 16, col = "blue", lwd = 1)
grid()
axis(1, cex.axis=0.8, col.axis="blue")
axis(2, cex.axis=0.8, col.axis="blue")
mtext(side=1, line=2, "Peso (lb)", cex=0.8)
mtext(side=1, line=3, "Histograma y polígono", cex=0.95)
mtext(side=1, line=4, "de frecuencias", cex=0.95)
h.peso$counts<-cumsum(h.peso$counts)
plot(h.peso
     , main="Personas"
     , cex.main=1           # Tamaño del título
     , xlab = ""
     , ylab = "F.Acumulada"
```

```
     , border = "grey"
     , type="h"
     , lty=1
     , xaxt="n"
     , yaxt="n"
     , col="grey90"
     , cex.lab=0.8)
x<- c(h.peso$breaks)
y<- c(0, h.peso$counts)
lines(x, y, type = "b", pch = 16, col = "blue", lwd = 1)
grid()
axis(1, cex.axis=0.8, col.axis="blue")
axis(2, cex.axis=0.8, col.axis="blue")
mtext(side=1, line=2, "Peso (lb)", cex=0.8)
mtext(side=1, line=3, "Histograma y polígono", cex=0.95)
mtext(side=1, line=4, "de frecuencias acumulada", cex=0.95)
```

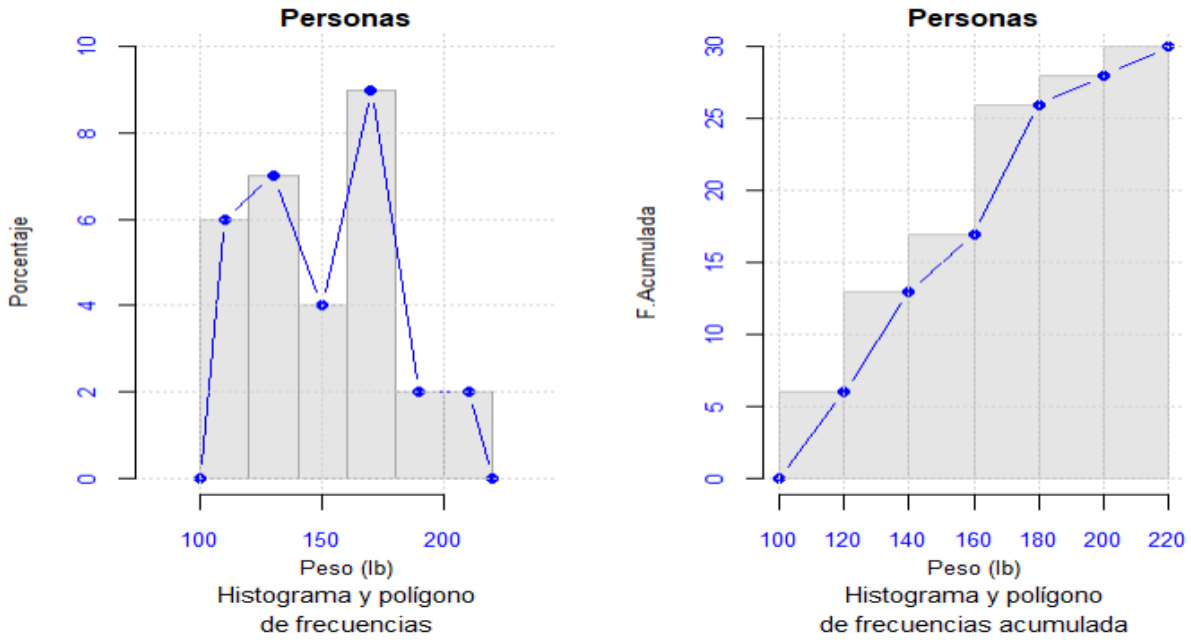


Figura 2.24 *Histograma de frecuencia acumulado*

Función densidad y curva normal

Cuando los intervalos de clase no son del mismo ancho, las frecuencias o frecuencias relativas no deben usarse en el eje vertical, en su lugar se utiliza la altura de cada rectángulo, llamada densidad para el intervalo de clase. El uso de la escala de densidad para construir el histograma asegura que el área de cada rectángulo en el histograma será proporcional a la frecuencia relativa correspondiente[26]. El cálculo de la densidad viene dado por la ecuación 2.5:

$$densidad = \frac{Frecuencia\ relativa\ del\ intervalo\ de\ clase}{Ancho\ del\ intervalo\ de\ clase} \qquad [2.5]$$

De acuerdo a [27] si el histograma se dibuja con una gran cantidad de pequeños intervalos, podemos suavizar los bordes de los rectángulos para producir una curva suave como se muestra

en la figura 8.25. En muchos casos, es posible determinar una función f (x) que se aproxima a la curva. La función f(x) se llama función de densidad de probabilidad. Los siguientes requisitos se aplican a una función de densidad de probabilidad f (x) cuyo rango es a ≤ x ≤ b:

1. f (x) ≥ 0 para todas las x entre a y b.
2. El área total bajo la curva entre a y b es 1.0

El gráfico de la función densidad de la distribución nos muestra como es esa distribución y, en algunos casos, resulta útil compararla con la curva normal para analizar visualmente si la distribución representada por la función f(x) se ajusta a la distribución normal.

Una distribución normal tiene una curva en forma de campana (simétrica). Su media se denota por μ y su desviación estándar por σ. Una variable aleatoria continua x que tiene una distribución normal se denomina variable aleatoria normal. Tenga en cuenta que no todas las curvas en forma de campana representan una curva de distribución normal. Solo un tipo específico de curva en forma de campana representa una curva normal[28].

Para realizar el gráfico de densidad se usa dos funciones: **lines**() y **density**(). La función **density**() toma el conjunto de datos de la variable a gráficar, los transforma y estos deben pasarse a la función **lines**() de la siguiente manera:

```
lines(density(personas$peso) , col="blue" , lwd=2)
```

Para crear la curva normal se usa la función **curve**() que requiere los datos de la curva normal que se obtiene por medio de la función **dnorm**(), la misma que recibe en un vector un conjunto de valores entre los valores mínimo y máximo de la distribución en el eje x, la media y la desviación estándar. La media se la calcula con la función **mean**() y la desviación estándar con la función **sd**(). Recuerde configurar el parámetro *add* con TRUE para que la curva se agregue al gráfico del histograma que se ha dibujado previamente. La obtención de esos requisitos se lo hace de la siguiente manera:

Datos de la distribución : `x<- c(min(h$breaks),h$mids,max(h$breaks))`

Media : `mean(personas$peso)`

Desviación estándar : `sd(personas$peso)`

De esta manera, la función curve() grafica la curva normal para la distribución que es su objeto de estudio:

```
curve(dnorm(x, mean(personas$peso), sd(personas$peso)), col="red", lty=1,
lwd=2, add=T)
```

```
par(mfrow=c(1,2), mar=c(5, 4, 1, 2))
h<-hist(personas$peso
        , probability = TRUE
        , main="Personas"
        , cex.main=1            # Tamaño del título
        , xlab=""
        , ylab = "Densidad"
        , cex.lab=0.8
        , xaxt="n"
        , yaxt="n"
        , xlim = c(80,240)
        , ylim = c(0.00, 0.015)
        , col="grey90"
        , border = "darkgrey"
        , lty=1
        , lwd=1)

x<- c(min(h$breaks),h$mids,max(h$breaks))
y<- c(0, h$density, 0)
lines(x, y
      , type = "b"
      , pch = 16
      , col = "gray31"
      , lwd = 1)
lines(density(personas$peso)
      , col="blue"
      , lwd=2)
grid()
axis(1, cex.axis=0.8, col.axis="blue")
axis(2, cex.axis=0.8, col.axis="blue")
mtext(side=1, line=2, "Peso (lb)", cex=0.8)
mtext(side=1, line=3, "Densidad de la distribución", cex=0.95)
mtext(side=1, line=4, "de frecuencias", cex=0.95)
# Curva normal
h<-hist(personas$peso
        , probability = TRUE
        , main="Personas"
        , cex.main=1            # Tamaño del título
        , xlab=""
        , ylab = "Densidad"
        , cex.lab=0.8
        , xaxt="n"
        , yaxt="n"
        , xlim = c(80,240)
        , ylim = c(0.00, 0.015)
        , col="grey90"
        , border = "darkgrey"
        , lty=1
        , lwd=1)
x<- c(min(h$breaks),h$mids,max(h$breaks))
y<- c(0, h$density, 0)
lines(x, y
      , type = "b"
      , pch = 16
      , col = "gray31"
```

```
      , lwd = 1)
lines(density(personas$peso)
      , col="blue"
      , lwd=2)

curve(dnorm(x, mean(personas$peso)
    , sd(personas$peso))
    , col = "red"
    , lty = 1
    , lwd = 2
    , add=T)
grid()
axis(1, cex.axis=0.8, col.axis="blue")
axis(2, cex.axis=0.8, col.axis="blue")
mtext(side=1, line=2, "Peso (lb)", cex=0.8)
mtext(side=1, line=3, "Densidad de la distribución", cex=0.95)
mtext(side=1, line=4, "contrastada con la curva normal", cex=0.95)
```

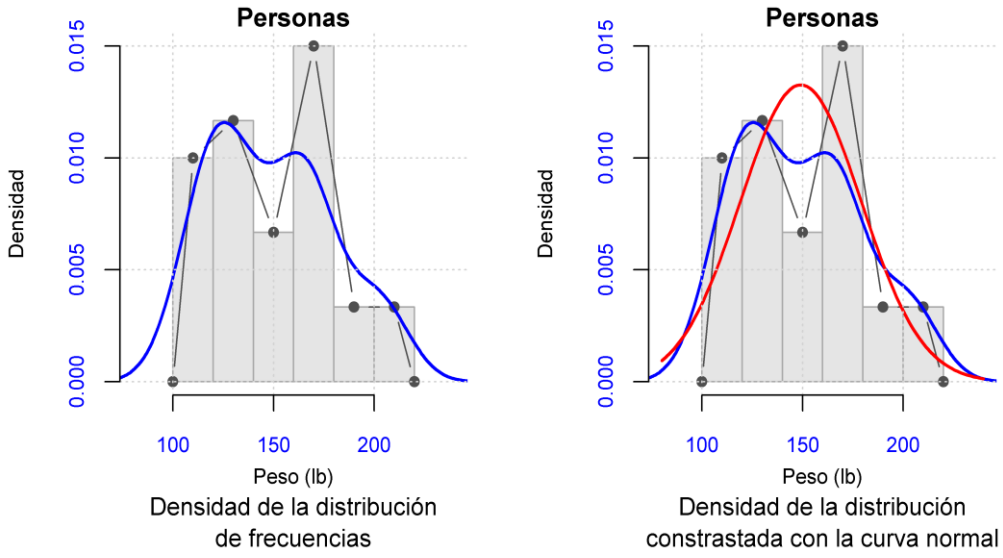


Figura 2.25 *Histograma de frecuencia acumulado*

Tallo y hojas

John Tukey (1977), como parte de su enfoque general para el análisis de datos, conocido como análisis exploratorio de datos (EDA), desarrolló una variedad de métodos para mostrar datos de manera visualmente significativa. Uno de los métodos más simples es la visualización de tallo y hoja. Los histogramas, las distribuciones de frecuencia y las funciones de densidad son métodos utilizados para presentar datos, cada uno tiene sus inconvenientes. Los histogramas representan observaciones dentro de intervalos por lo que se pierden los valores numéricos reales. Las distribuciones de frecuencia retienen los valores de las observaciones individuales, pero pueden ser difíciles de usar cuando no resumen los datos lo suficiente. Un enfoque alternativo que evita ambas críticas es la visualización de tallo y hojas.

```
par(mfrow=c(1,2))
stem(personas$peso, scale=1)
```

```
#  SALIDA POR CONSOLA
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   10 | 20446
##   12 | 02666806
##   14 | 4480
##   16 | 24466602
##   18 | 04
##   20 | 040
stem(personas$peso, scale=2)
# SALIDA POR CONSOLA
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   10 | 2
##   11 | 0446
##   12 | 026668
##   13 | 06
##   14 | 448
##   15 | 0
##   16 | 244666
##   17 | 02
##   18 | 0
##   19 | 4
##   20 | 04
##   21 | 0
```

AUTOEVALUACIÓN

Autoevaluación 2-1

Se recogen datos de los gastos económicos de 10 adolecentes en temporada de vacaciones visualice la información en un histograma y polígono de frecuencia los datos son: $24, $10, $12, $28, $19, $10, $17, $14, $12, $10

Autoevaluación 2-2

Una fábrica calculó el promedio de duración que tienen sus productos de plásticos en años, visualice esta información a partir de la densidad contrastada con la curva normal, la información se muestran a continuación: 1, 3, 1, 5, 2, 3, 7, 3, 2, 6, 3, 4, 2, 1, 1, 6, 7, 9, 5

Autoevaluación 2-3

En una ciudad se obtienen 20 datos sobre las edades de personas que usan zapatos deportivos, represente las edades mediante un diagrama de tallos y hojas en escala de 1. A continuacion se presenta la siguiente distribución de frecuencia de los datos 32, 23, 5, 4, 12, 55, 17, 8, 29, 10, 49, 29, 39, 56, 71, 22, 17, 19, 21, 70

EJERCICIOS DEL CAPÍTULO

1. El personal de calidad de una empresa ha recolectado datos sobre el tiempo que tardan en brindar dos tipos de servicios a sus clientes, para el servicio 1 se recolecto una muestra en 12 clientes con los siguientes tiempos en minutos: 8, 8, 3, 1, 9, 7 , 5, 5, 12, 7, 11, 9 para el servicio 2 se obtienen los siguientes datos en minutos: 12, 11, 10, 6, 8, 9, 9, 10, 11, 9, 8, 10
   a) Realizar un gráfico de barra de manera vertical, con leyendas para visualizar los datos del servicio 1 y servicio 2 en color naranja y azul respectivamente.
   b) Realizar un gráfico de barra de manera horizontal, con leyendas para visualizar los datos del servicio 1 y servicio 2 en color amarillo y magenta respectivamente.
2. En una muestra de 20 personas sobre el uso de tres repelentes se obtuvo la siguiente información. Repelentes: r1, r2, r3, r1, r1, r3, r2, r2, r3, r1, r2, r1, r1, r3, r2, r2, r3, r1, r2, r3
   a) Realice un gráfico de barra en 3 D con colores goldenrod3 y darkolivegreen1
   b) Agregue etiquetas sobre los sectores y leyendas en el gráfico.
3. Un estudio de mercado busca saber a través de encuestas cuales son las marcas de cuadernos que usan los universitarios usualmente. En la siguiente tabla se muestran los resultados.

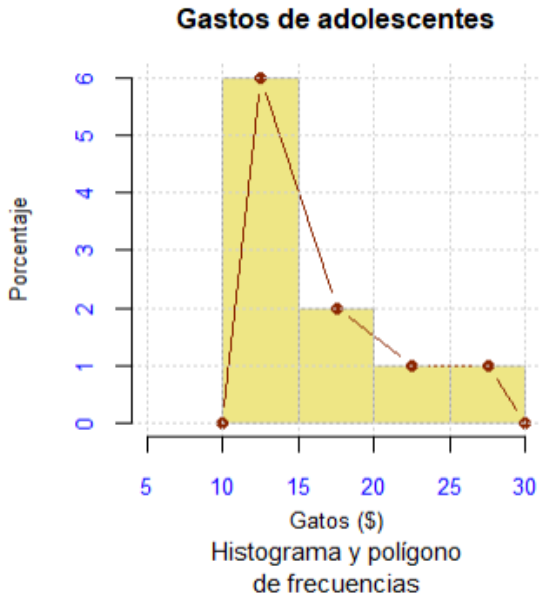
Marca	Cantidad
Estilo	56
Escribe	120
Norma	220
Ideal	140
Kiut	90
Jean Book	70
Artesco	100
Estilo	102
Norma	105
Ideal	105
Kiut	203
Artesco	108

   a) Realice un gráfico de caja y bigote básico
   b) Visualice el gráfico de caja y bigote de color darkolivegreen, con tamaño de letra 2 para el título.
4. Con los datos anteriores sobre el uso de las marcas de cuadernos del ejercicio 2 realice un gráfico de puntos con el método stack, con colores coral y goldenrod1 para conocer el comportamiento de las marcas
5. Realice un Histograma con gráfico de caja y bigote integrado con colores khaki3 para el histograma y goldenrodpara el color de la caja que represente el uso del celular , en una muestra de 20 personas se obtuvieron los siguientes datos en horas: 1, 1.5,2.3,5.2,6.7,2.3, 5.2,6,9,1.6,4.4,2,3,1.2,4.3,5,3.3,7,5.2,2

Respuestas de las autoevaluaciones

2-1

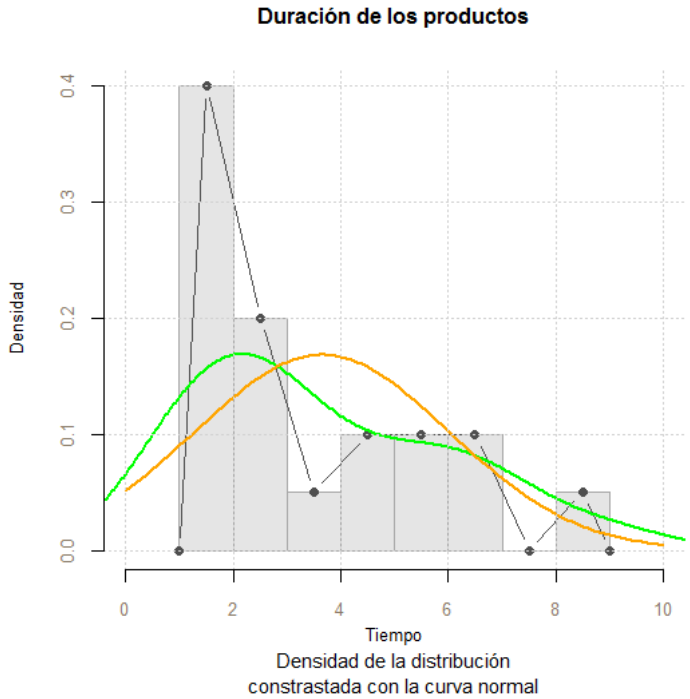
```
gastos<-c(24,10,12,28,19,10,17,14,12,10)
h.gasto<-hist(gastos
        , probability = FALSE
        , main="Gastos de adolescentes"
        , cex.main=1            # Tamaño del título
        , xlab=""
        , ylab = "Porcentaje"
        , cex.lab=0.8
        , xaxt="n"
        , yaxt="n"
        , xlim = c(10,30)
        , ylim = c(0.00, 10)
        , col=" khaki2"
        , border = "darkgrey"
        , lty=1
        , lwd=1)
x<- c(min(h.gasto$breaks), h.gasto$mids,max(h.gasto$breaks))
y<- c(0, h.gasto$counts, 0)
lines(x, y, type = "b", pch = 16, col = " orangered4", lwd = 1)
grid()
axis(1, cex.axis=0.8, col.axis="blue")
axis(2, cex.axis=0.8, col.axis="blue")
mtext(side=1, line=2, "Gatos ($)", cex=0.8)
mtext(side=1, line=3, "Histograma y polígono", cex=0.95)
mtext(side=1, line=4, "de frecuencias", cex=0.95)
```



2-2

```
duracion<-c(1, 3,1, 5,2,3,7,3,2,6,3,4,2,1,1,6,7,9,5,2)
h<-hist(duracion
        , probability = TRUE
        , main="Duración de los productos"
        , cex.main=1            # Tamaño del título
        , xlab=""
        , ylab = "Densidad"
        , cex.lab=0.8
```

```
        , xaxt="n"
        , yaxt="n"
        , xlim = c(0,10)
        , ylim = c(0.00,0.40)
        , col="grey90"
        , border = "darkgrey"
        , lty=1
        , lwd=1)
x<- c(min(h$breaks),h$mids,max(h$breaks))
y<- c(0, h$density, 0)
lines(x, y, type = "b", pch = 16, col = "gray31", lwd = 1)
lines(density(duracion), col="green", lwd=2)
curve(dnorm(x, mean(duracion)
    , sd(duracion))
    , col = "orange"
    , lty = 1
    , lwd = 2
    , add=T)
grid()
axis(1, cex.axis=0.8, col.axis="bisque4")
axis(2, cex.axis=0.8, col.axis="bisque4")
mtext(side=1, line=2, "Tiempo", cex=0.8)
mtext(side=1, line=3, "Densidad de la distribución", cex=0.95)
mtext(side=1, line=4, "contrastada con la curva normal", cex=0.95)
```



2-3

```
edad<-c(32, 23, 5,4,12,55,17,8,29,10,49,29,39,56,71,22,17,19,21,70)
stem(edad,scale=1)
# SALIDA POR CONSOLA
The decimal point is 1 digit(s) to the right of the |
  0 | 45802779
  2 | 1239929
  4 | 956
  6 | 01
```

3 DESCRIBIR, EXPLORAR Y COMPARAR DATOS

La recopilación, procesamiento y el análisis de datos son fundamentales en cualquier entorno del desarrollo humano. Transformar datos recopilados en información útil que estampe algún significado importante es esencial para el perfeccionamiento de toda labor que deba ser respaldada en evidencias. En este capitulo se expone algunos métodos descriptivos para el análisis de datos tales como:

- Medidas de tendencia central
- Medidas de variación
- Visualización de las medidas de centralidad y dispersión
- Medidas de posición relativa
- Visualización de las medidas de posición relativa

En principio se crea el conjunto de datos deportista (ver tabla 3.0), que contendrá los datos de entrenamiento de tres deportistas que practican heterofilia.. Los datos están en kilogramos.

```
pesista1<-c(349, 350, 350, 350, 360, 370, 370, 370, 371, 372, 372, 384, 391,
391, 392)
pesista2<-c(350, 354, 359, 363, 365, 368, 369, 372, 373, 374, 376, 380, 383,
388, 394)
pesista3<-c(351, 366, 362, 364, 364, 365, 366, 368, 371, 377, 377, 379, 380,
390, 396)
deportista<-data.frame(pesista1, pesista2, pesista3)
kable(head(deportista, 15), caption="Tabla 3.0 Conjunto de datos deportistas"
)
```

Tabla 3.0 *Conjunto de datos deportistas*

pesista1	pesista2	pesista3
349	350	351
350	354	366
350	359	362
350	363	364
360	365	364
370	368	365
370	369	366
370	372	368
371	373	371
372	374	377
372	376	377
384	380	379
391	383	380
391	388	390
392	394	396

Los métodos gráficos son un medio excelente para obtener una visión general rápida de los datos, pero no son particularmente precisos ni se prestan a un análisis posterior. Para esto debemos recurrir a medidas numéricas como el promedio[29]. El promedio es una medida que prácticamente todas las personas conocen, al menos a nivel intuitivo, y se calcula sumando el total de los datos o valores de la variable para luego dividir esa suma entre el número de datos sumados. A esa medida común se le llama indistintamente media, promedio o media aritmética. La mediana es el valor que ocupa el lugar central en una serie ordenada. La moda es el valor que más se repite, es decir, el que tiene mayor frecuencia[30].

La **media**, o valor medio, es quizá la medida de ubicación más importante para una variable, pues proporciona una medida de la ubicación central de los datos. Si los datos son para una muestra, la media se denota por $\overline{x}$; si son para una población, se denota por la letra griega μ[31]. La expresión matemática para calcular la media se muestra en la ecuación 3.0

$$\overline{x} = \frac{\sum x_i}{n} \qquad [3.0]$$

De acuerdo a [32] la **mediana** es el valor medio de las observaciones cuando las observaciones se ordenan de menor a mayor (o de mayor a menor). A continuación, tenemos la descripción de cómo encontrar la mediana en una distribución de datos:

- Poner las n observaciones en orden de su tamaño. Se calcula la posición n donde se encuentra la mediana (Ver ecuación 3.1).

$$Posición_n = \frac{Total\ de\ observaciones}{2} \qquad [3.1]$$

- Cuando el número de observaciones n es impar, la mediana es el valor de la observación en la posición n. Ver ecuación 3.2

$$Mediana = Valor(Posición_n) \qquad [3.2]$$

- Cuando el número de observaciones n es par, la mediana queda en la mitad dos observaciones de la muestra ordenada, y la mediana es su promedio. Analice la ecuación 3.3 que expresa matemáticamente el razonamiento anterior.

$$Mediana = \frac{Valor(Posición_{n-0.5}) + Valor(Posición_{n+0.5})}{2} \qquad [3.3]$$

La **moda** es una medida de repetición de los valores de una variable. Se considera a la moda como el valor que se repite con mayor frecuencia dentro del conjunto de datos de la variable.

Las funciones para calcular la media y la mediana son **mean()** y **median()**. R no tiene una función para calcular la moda por lo que se crea una función para encontrarla. En ocasiones, es necesario reducir la cantidad de decimales con la función round() . En el ejemplo, se construye el dataframe centralidad que va a contener la media, mediana y moda para los tres pesistas. La tabla 3.1 muestra las medidas de centralidad para cada uno de los pesistas.

```
#-----------------------------
#           Media mean()
#-----------------------------
MP1<-round(mean(deportista$pesista1), 2)  # Calculo de la media
MP2<-round(mean(deportista$pesista2), 2)  # Calculo de la media
MP3<-round(mean(deportista$pesista3), 2)  # Calculo de la media
M<-data.frame(Pesista1=MP1, Pesista2=MP2, Pesista3=MP3)
#-----------------------------
#           Mediana median()
#-----------------------------
Me1<-round(median(deportista$pesista1), 2)  # Calculo de la mediana
Me2<-round(median(deportista$pesista2), 2)  # Calculo de la mediana
Me3<-round(median(deportista$pesista3), 2)  # Calculo de la mediana
Me<-data.frame(Pesista1=Me1, Pesista2=Me2, Pesista3=Me3)
#-----------------------------
#           Moda
#-----------------------------
# FUNCIÓN PARA CALCULAR LA MODA.
moda <- function(dataframe) {
   uvalor <- unique(dataframe)
   uvalor[which.max(tabulate(match(dataframe, uvalor)))]
}
Mo1<-round(moda(deportista$pesista1), 2)  # Calculo de la moda
Mo2<-round(moda(deportista$pesista2), 2)  # Calculo de la moda
Mo3<-round(moda(deportista$pesista3), 2)  # Calculo de la moda
Mo<-data.frame(Pesista1=Mo1, Pesista2=Mo2, Pesista3=Mo3)
# RESUMEN DE LAS MEDIDAS DE CENTRALIDAD
centralidad<-rbind(M, Me, Mo)
rownames(centralidad) <- c("Media (M)","Mediana (Me)", "Moda (Mo)")
kable(centralidad, caption = "Tabla 3.1 Cálculo de las medidas de centralidad
")
```

Tabla 3.1 *Cálculo de las medidas de centralidad*

	Pesista1	Pesista2	Pesista3
Media (M)	369.47	371.2	371.73
Mediana (Me)	370.00	372.0	368.00
Moda (Mo)	350.00	350.0	366.00

La tabla 3.1 exhibe la información de la media, mediana y moda que se podría interpretar como el trabajo diario con pesas de cada deportista. Observe que se revela que el pesista 2 tiene la media y la mediana más alta pero el pesista 3 tiene una mejor moda. Para comprender mejor los resultados del entrenamiento diario se necesita conocer que diferencia hay entre sesión y sesión cada día. Para esto se debe recurrir a las medidas de dispersión.

Medidas de Dispersión o Variación

Los conjuntos de datos pueden tener el mismo centro (media, mediana y moda), pero con aspecto diferente por la forma en que los números se dispersan desde el centro. Las medidas de variabilidad pueden ayudarle a crear una imagen mental de la dispersión de los datos y se considera una muy importante característica de datos [33]. Esto quiere decir que las medidas de centralidad no revelan la imagen completa de la distribución de un conjunto de datos. Es pertinente revisar varios de los conceptos de las medidas de dispersión como el rango, varianza, desviación estándar y coeficiente de variación antes de entrar al estudio de cómo se calculan estas medidas con R.

El **rango** es la medida más simple de dispersión para calcular. Se obtiene tomando la diferencia entre los valores más grandes y más pequeños en un conjunto de datos. Ver ecuación 3.4.

$$Rango = Valor\ máximo\ -\ Valor\ mínimo \qquad [3.4]$$

La **varianza** muestral para una muestra de n mediciones es igual a la suma de las desviaciones al cuadrado de la media, dividida por (n - 1). El símbolo S^2 se usa para representar la varianza muestral[34]. La ecuación 3.5 representa lo manifestado.

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1} \qquad [3.5]$$

$$S^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1} \qquad [3.6]$$

La **desviación estándar** se define como la raíz cuadrada positiva de la varianza. Siguiendo la notación que adoptamos para una varianza muestral y una varianza poblacional, usamos s para denotar la desviación estándar de la muestra y σ para denotar la desviación estándar de la población [35]. Las ecuaciones 3.7 y 3.8 indican como la desviación estándar se deriva de la varianza, de la siguiente manera

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{N}} \qquad [3.7]$$

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \qquad [3.8]$$

Coeficiente de Variación

Las medidas de dispersión, explica [36], como el rango, varianza y la desviación estándar son todas medidas de dispersión absoluta y, en particular, sus valores dependen de las unidades en las que se mide la variable. Por lo tanto, es difícil comparar los grados de dispersión de dos variables que se miden en diferentes unidades. La solución es usar una medida de dispersión relativa, que

es independiente de las unidades de medida. Una de esas medidas es el **coeficiente de variación**, la ecuación 3.9 la define como:

$$Coeficiente\ de\ variación = \frac{Desviación\ Estándar}{Media} \qquad [3.9]$$

Al igual que las medidas de centralidad, R dispone de funciones para realizar este tipo de cálculos estadísticos de forma simple. Empecemos con el **rango**, de acuerdo a la definición el rango es el resultado de la resta entre el valor máximo y valor mínimo del conjunto de datos. Las funciones **max**() y **min**() consiguen esos valores, luego haciendo la resta de ambos obtenemos el rango. Para hallar la **varianza** y la **desviación estándar** usamos las funciones **var**() y **sd**(). Para hallar el coeficiente de variación se ha escrito una pequeña función donde se halla la media (mean) y la desviación estándar (sd), luego dividimos de acuerdo a la ecuación 3.9 para obtener dicho coeficiente. El código de abajo enseña como obtener las medidas de dispersión y presenta los resultados en la tabla 3.2.

```
#------------------------------
#         Rango max()-min()
#------------------------------
r1=round(max(deportista$pesista1)-min(deportista$pesista1), 2)
r2=round(max(deportista$pesista2)-min(deportista$pesista2), 2)
r3=round(max(deportista$pesista3)-min(deportista$pesista3), 2)
Rangos<-cbind(Pesista1=r1,Pesista2=r2,Pesista3=r3)
rownames(Rangos)=c("Rango")
#------------------------------
#           Varianza var()
#------------------------------
var1=round(var(deportista$pesista1), 2)
var2=round(var(deportista$pesista2), 2)
var3=round(var(deportista$pesista3), 2)
Varianzas<-cbind(Pesista1=var1,Pesista2=var2,Pesista3=var3)
rownames(Varianzas)=c("Varianza")
#------------------------------
#     Desviación Estándar sd()
#------------------------------
sd1=round(sd(deportista$pesista1), 2)
sd2=round(sd(deportista$pesista2), 2)
sd3=round(sd(deportista$pesista3), 2)
desv.Stand<-cbind(Pesista1=sd1,Pesista2=sd2,Pesista3=sd3)
rownames(desv.Stand)=c("SD")
#------------------------------
#     Coeficiente de variación
#------------------------------
cv<-function(ds){
  media<-mean(ds)
  desv.stand<-sd(ds)
  cv<-round(desv.stand/media,2)
  return(cv)
}
```

```
cv1=round(cv(deportista$pesista1), 2)
cv2=round(cv(deportista$pesista2), 2)
cv3=round(cv(deportista$pesista3), 2)
Coef.variacion<-cbind(Pesista1=cv1, Pesista2=cv2, Pesista3=cv3)
rownames(Coef.variacion)=c("Coef.variacion")
#-----------------------------
#     Medidas de dispersión
#-----------------------------
dispersion<-rbind(Rangos, Varianzas, desv.Stand, Coef.variacion)
kable(dispersion, caption="Tabla 3.2 Cálculo de las medidas de dispersión")
```

Tabla 3.2 *Cálculo de las medidas de dispersión*

	Pesista1	Pesista2	Pesista3
Rango	43.00	44.00	45.00
Varianza	236.27	147.74	133.50
SD	15.37	12.15	11.55
Coef.variacion	0.04	0.03	0.03

Se puede mejorar la presentación del resultado tanto de las medidas de centralidad como las de dispersión uniendo los dataframe centralidad y dispersión mediante la función de rbind() y consolidando todos los resultados. Ver tabla 3.3.

```
medidasDescriptivas<-rbind(centralidad, dispersion)
# SALIDA CON FORMATO
kable(medidasDescriptivas, caption="Tabla 3.3 Medidas de centralidad y disper
sión")
```

Tabla 3.3 *Medidas de centralidad y dispersión*

	Pesista1	Pesista2	Pesista3
Media (M)	369.47	371.20	371.73
Mediana (Me)	370.00	372.00	368.00
Moda (Mo)	350.00	350.00	366.00
Rango	43.00	44.00	45.00
Varianza	236.27	147.74	133.50
SD	15.37	12.15	11.55
Coef.variacion	0.04	0.03	0.03

Visualización de las medidas de centralidad y dispersión

Para la representación gráfica de las medidas de centralidad y dispersión las funciones **hist()**, **lines()** y **density()** son apropiadas. De acuerdo a lo revisado en el capítulo 2, estas funciones grafican el histograma de frecuencias y la función densidad de la distribución, respectivamente. Como el objetivo es observar el comportamiento de las medidas de centralidad y de dispersión vamos a utilizar la función **abline()** con su parámetro *v* para trazar una línea vertical en la ubicación de cada una de estas medidas. Para nuestro caso `v=centralidad[“Media (M)”, “Pesista1”]`. El resultado se observa en la figura 3.0 donde se presenta por separado el histograma, la función densidad y la curva normal para las medidas de centralidad y dispersión respectivamente.

```
#--------------------------------
#     Medidas de centralidad
#--------------------------------
par(mfrow=c(1,2), mar=c(3.2,3.8,1,2))
#     HISTOGRAMA DE FRECUENCIAS
hist(deportista$pesista1
     , probability = TRUE
     , main = "Medidas de Centralidad"
     , cex.main=1
     , xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
     , xlim = c(330, 420)
     , ylim = c(0.000, 0.040)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Pesista 1 (Kg)", cex=0.8)
#     GRÁFICO DE LA FUNCIÓN DENSIDAD
lines(density(deportista$pesista1), col="blue", lwd=2)
#     LÍNEA DE LA MEDIA
abline(v=centralidad["Media (M)", "Pesista1"], col="red", lty=1, lwd=2)
#     LÍNEA DE LA MEDIANA
abline(v=centralidad["Mediana (Me)", "Pesista1"], col="blue", lty=2, lwd=2)
#     LÍNEA DE LA MODA
abline(v=centralidad["Moda (Mo)", "Pesista1"], col="brown", lty=3, lwd=2)
#     CURVA NORMAL
curve(dnorm(x, centralidad["Media (M)", "Pesista1"]
  , dispersion["SD", "Pesista1"])
    , col = "darkorchid1"
    , lty = 1
    , lwd = 1
    , add=T)
#     LEYENDA
l1<-paste("M :", as.character(centralidad["Media (M)", "Pesista1"]))
l2<-paste("Me :", as.character(centralidad["Mediana (Me)", "Pesista1"]))
l3<-paste("Mo :", as.character(centralidad["Moda (Mo)", "Pesista1"]))
legend("topright"
    , col=c("red","blue","brown")
    , lty=1:3
    , legend =c(l1, l2, l3)
    , lwd=2
    , bty = "n"
    , cex=0.7)
#--------------------------------
#     Medidas de dispersión
#--------------------------------
#     HISTOGRAMA DE FRECUENCIAS
hist(deportista$pesista1
     , probability = TRUE
     , main = "Medidas de Dispersión"
     , cex.main=1
     , xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
```

```
     , cex.axis=0.8
     , xlim = c(330, 420)
     , ylim = c(0.000, 0.040)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Pesista 1 (Kg)", cex=0.8)
#     FUNCIÓN DENSIDAD
lines(density(deportista$pesista1), col="blue", lwd=2)
#     MEDIA
abline(v=centralidad["Media (M)", "Pesista1"], col="red", lty=1, lwd=2)
#     MEDIANA
abline(v=centralidad["Media (M)", "Pesista1"] + dispersion["SD", "Pesista1"]
       , col="blue", lty=2, lwd=2)
#     MODA
abline(v=centralidad["Media (M)", "Pesista1"] - dispersion["SD", "Pesista1"]
       , col="blue", lty=2, lwd=2)
#     CURVA NORMAL
curve(dnorm(x, centralidad["Media (M)", "Pesista1"]
    , dispersion["SD", "Pesista1"])
    , col = "darkorchid1"
    , lty = 1
    , lwd = 1
    , add=T)
#     LEYENDA
l1<-paste("M :", as.character(centralidad["Media (M)", "Pesista1"]))
l2<-paste("SD :", as.character(dispersion["SD", "Pesista1"]))
l3<-paste("CV :", as.character(dispersion["Coef.variacion", "Pesista1"]))
legend("topright"
    , col=c("red","blue", "white")
    , lty=1:3
    , legend =c(l1, l2, l3)
    , lwd=2
    , bty = "n"
    , cex=0.7)
```

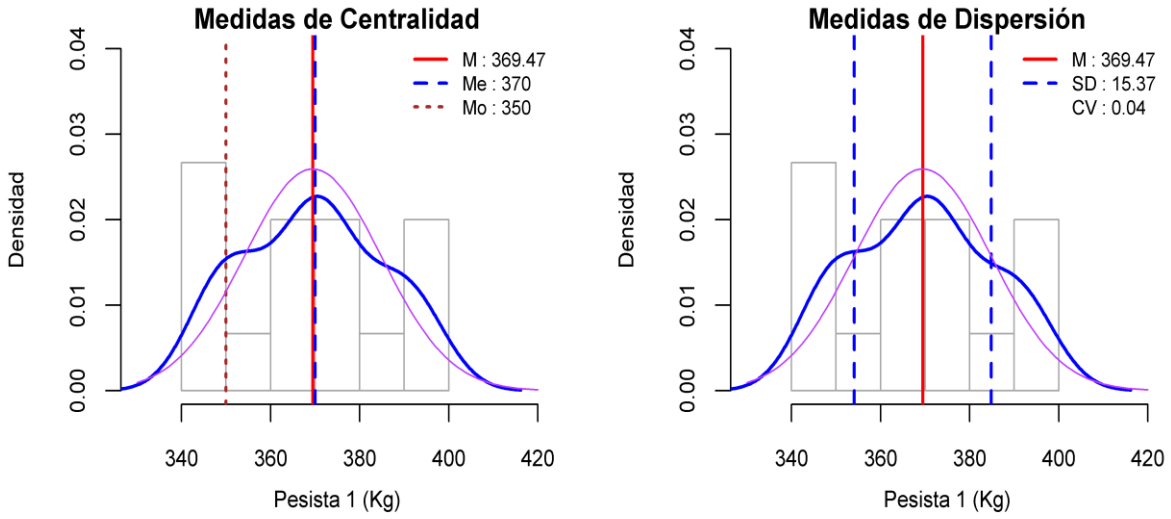


Figura 3.0 *Gráfico de las Medidas de Centralidad y Dispersión*

Como se observa en la figura 3.0 se ha visualizado las medidas de centralidad y de dispersión en el histograma de frecuencias y densidad. Un análisis breve de esto puede ser que la distribución

de los alzamientos no corresponde a una distribución normal y su desviación estándar indica que los levantamientos son muy variables.

Medidas de posición relativa

Las medidas de posición son importantes dentro del análisis estadísticos, su uso es frecuente porque se divide el conjunto de datos en cuatro partes denominados **cuartiles**, cada una de las partes contiene aproximadamente el 25% de las observaciones. Los **deciles** dividen la distribución de datos en diez partes. Un **percentil** proporciona información sobre cómo se distribuyen los datos en el intervalo del valor menor al valor mayor; es decir, el percentil p-ésimo los divide en dos partes. Alrededor de p por ciento de las observaciones tiene valores menores que el percentil p-ésimo y cerca de (100 p) cien por ciento de las observaciones tiene valores mayores que el percentil p-ésimo.

Para hallar el valor del cuartil, decil, o percentil seguimos el siguiente procedimiento:

1. Ordenar la lista de observaciones de menor a mayor
2. Calcular la posición del: cuartil (Ec. 3.10), decil (Ec. 3.11) o percentil (Ec. 3.12)
3. En la lista ordenada, busque en la posición calculada el valor correspondiente.

$$Posición(cuartil) = \frac{Cuartil}{4}(n) \qquad [3.10]$$

$$Posición(decil) = \frac{Decil}{10}(n) \qquad [3.11]$$

$$Posición(percentil) = \frac{Percentil}{100}(n) \qquad [3.12]$$

El **rango intercuartil** es otra medida de frecuente uso en el análisis estadístico, mide la propagación del 50% medio de las observaciones. Los valores grandes de esta estadística significan que el primer y el tercer cuartil están muy separados, lo que indica un alto nivel de variabilidad[37]. Por definición, su ecuación es:

$$Rango\ Intercuartil = cuartil_3 - cuartil_1 \qquad [3.13]$$

R provee una sola función para calcular los cuartiles, deciles y percentiles llamada **quantile()**. Su acción es muy sencilla de configurar pues necesita solo dos parámetros principales que son: El conjunto de datos que es el primer parámetro y el parámetro *prob*. Recordemos que los cuartiles dividen el todo en 4 partes, el decil en 10 partes y el percentil en 100 partes. Si quiero el cuartil 1, significa el 25% de los datos, que en el parámetro *prob* se configura como `prob=c(0.25)` o si quisiera el decil 60 debo pasarlo como `prob=c(0.60)` o si quiere calcular el percentil 88 usted

pasaría como `prob=c(0.88).` Para hallar el rango intercuartil se usa la función **IQR()** la cual necesita únicamente que se le pase como parámetro el conjunto de datos.

El siguiente ejemplo calcula los tres primeros cuartiles, deciles y percentiles. Además, vamos a calcular el rango intercuartil.

```
qt1<-quantile (deportista$pesista1, prob = c(0.25, 0.50, 0.75))
print("Cuartiles")
## [1] "Cuartiles"
qt1
## 25% 50% 75%
## 355 370 378
dc1<-quantile (deportista$pesista1, prob = c(0.10, 0.20, 0.30))
print("Deciles")
## [1] "Deciles"
dc1
## 10% 20% 30%
## 350 350 362
pc3<-quantile (deportista$pesista1, prob = c(0.01, 0.02, 0.03))
print("Percentiles")
## [1] "Percentiles"
pc3
##     1%     2%     3%
## 349.14 349.28 349.42
print("Rango Intercuartil")
## [1] "Rango Intercuartil"
IQR(deportista$pesista1)
## [1] 23
```

Visualización de las medidas de posición relativa

Para que tenga sentido la visualización de los cuartiles, deciles y percentiles vamos a graficar el histograma de frecuencia (**hist**()), la función densidad (**lines(density**(df)**)**) y su curva normal (**curve(dnorm**())). Con la función **abline()** y su parámetro *v* enviaremos el valor del cuartil, decil o percentil que corresponda. La figura 3.1 muestra el resultado del siguiente código

```
par(mfrow=c(2,2), mar=c(3,2,2,2))
hist(deportista$pesista1                                    # CUARTILES
     , probability = TRUE
     , main = "Cuartiles Q1, Q2, Q3"
     , cex.main=1
     , xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
     , xlim = c(330, 420)
     , ylim = c(0.000, 0.030)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Pesista 1 (Kg)", cex=0.8)
lines(density(deportista$pesista1), col="blue", lwd=2)     # FUNCIÓN DENSIDAD
q<-quantile(deportista$pesista1, probs = c(0.25, 0.50, 0.75))# MEDIA
```

```
abline(v=q[1], col="black", lty=1, lwd=1)
abline(v=q[2], col="red", lty=1, lwd=1)
abline(v=q[3], col="magenta", lty=1, lwd=1)
curve(dnorm(x, centralidad["Media (M)", "Pesista1"]          # CURVA NORMAL
    , dispersion["SD", "Pesista1"])
    , col = "seagreen"
    , lty = 1
    , lwd = 1
    , add=T)
hist(deportista$pesista1                                     # DECILES
     , probability = TRUE
     , main = "Deciles D10, D30, D50"
     , cex.main=1
     , xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
     , xlim = c(330, 420)
     , ylim = c(0.000, 0.030)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Pesista 1 (Kg)", cex=0.8)
lines(density(deportista$pesista1), col="blue", lwd=2)     # FUNCIÓN DENSIDAD
d<-quantile(deportista$pesista1, probs = c(0.10, 0.30, 0.50))
abline(v=d[1], col="black", lty=1, lwd=1)
abline(v=d[2], col="magenta", lty=1, lwd=1)
abline(v=d[3], col="red", lty=1, lwd=1)
curve(dnorm(x, centralidad["Media (M)", "Pesista1"]        # CURVA NORMAL
    , dispersion["SD", "Pesista1"])
    , col = "seagreen"
    , lty = 1
    , lwd = 1
    , add=T)
hist(deportista$pesista1                                   # PERCENTILES
     , probability = TRUE
     , main = "Percentiles P18, P55, P84"
     , cex.main=1
     , xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
     , xlim = c(330, 420)
     , ylim = c(0.000, 0.030)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Pesista 1 (Kg)", cex=0.8)
lines(density(deportista$pesista1), col="blue", lwd=2)    # FUNCIÓN DENSIDAD
p<-quantile(deportista$pesista1, probs = c(0.18, 0.55, 0.84))
abline(v=p[1], col="black", lty=1, lwd=1)
abline(v=p[2], col="magenta", lty=1, lwd=1)
abline(v=p[3], col="blue", lty=1, lwd=1)
curve(dnorm(x, centralidad["Media (M)", "Pesista1"]      # CURVA NORMAL
    , dispersion["SD", "Pesista1"])
    , col = "seagreen"
    , lty = 1
```

```
    , lwd = 1
    , add=T)
hist(deportista$pesista1                                    # RANGO INTERCUARTIL
     , probability = TRUE
     , main = "Rango Intercuartil"
     , cex.main=1
     , xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
     , xlim = c(330, 420)
     , ylim = c(0.000, 0.030)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Pesista 1 (Kg)", cex=0.8)
lines(density(deportista$pesista1), col="blue", lwd=2)    # FUNCIÓN DENSIDAD
abline(v=q[1], col="magenta", lty=1, lwd=1)
abline(v=q[3], col="magenta", lty=1, lwd=1)
curve(dnorm(x, centralidad["Media (M)", "Pesista1"]       # CURVA NORMAL
    , dispersion["SD", "Pesista1"])
    , col = "seagreen"
    , lty = 1
    , lwd = 1
    , add=T)
```

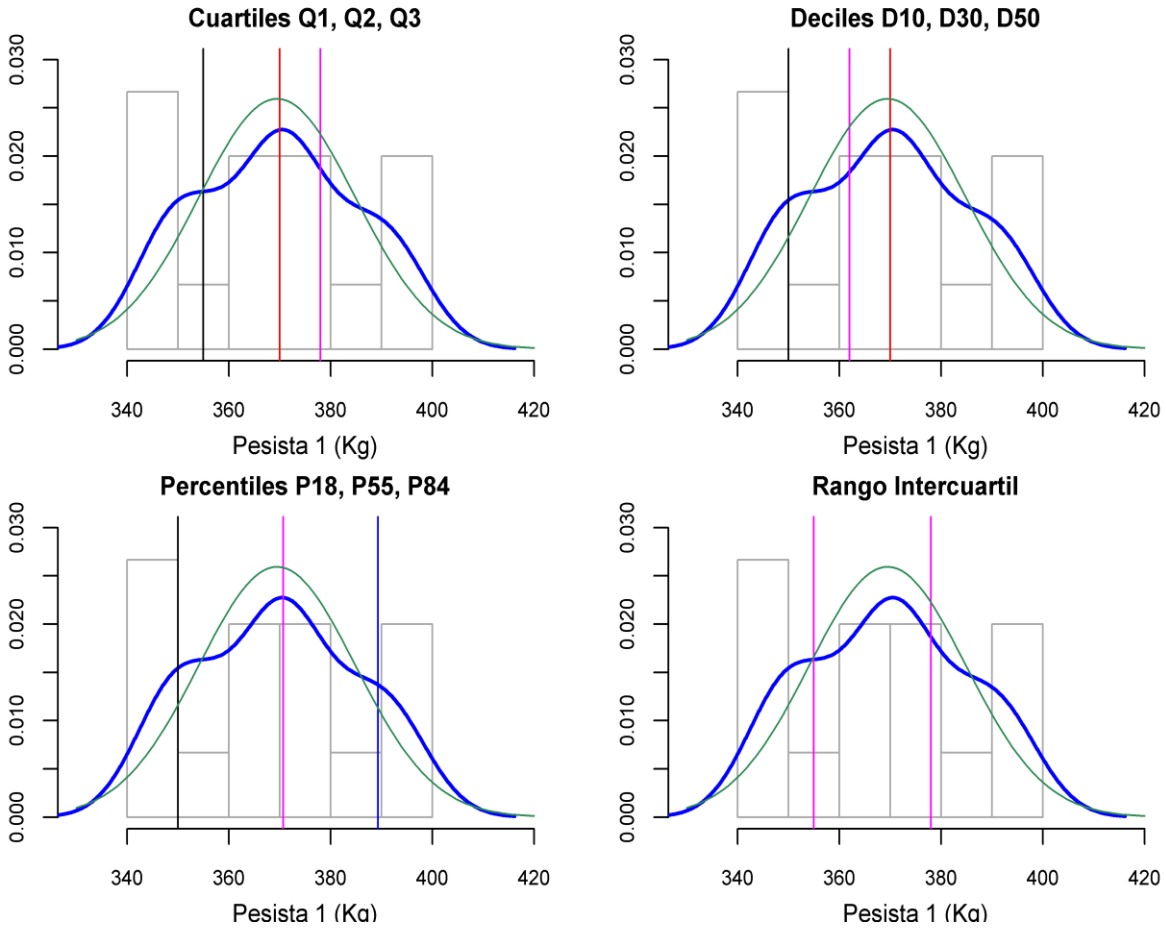


Figura 3.1 *Gráfico de las medidas de posición*

AUTOEVALUACIÓN

Autoevaluación 3-1

Se toma una muestra de 15 empleados sobre sus salarios en dólares, los datos son: 400, 800,400,750,900,1200,2100,3000,2100,500,500,1200,900,600,3000, obtenga las medidas de tendencia central.

Autoevaluación 3-2

Con una muestra de 15 estudiantes se tiene las calificaciones finales de semestre, los resultados son: 7,8,9,7,10,6,7,8,9,10,10,6,8,9,9, obtenga las medidas de dispersión.

Autoevaluación 3-3

Una muestra de 15 docentes sobre las calificaciones finales del curso de estadística, los resultados son: 7,10,9,7,10,6,7,8,7,10,10,6,8,6,9, obtenga la gráfica del cuartil con el color de línea green.

EJERCICIOS DEL CAPÍTULO

En un estudio que se realizó a dos niveles de un curso de estadística, se tomó las edades de los participantes que obtuvieron calificaciones mayores a 8.

Curso nivel 1:50, 27,38,44,20,35,25,19,55,29,44,19,20,25,30,38

Curso nivel 2: 34,32,50,32,29,27,26,19,29,55,20,44,42,38,32,49

1. Obtener la media, la mediana y la moda de las siguientes edades
2. Obtenga el rango, la varianza, la desviación estándar y el coeficiente de variación
3. Calcular los tres primeros cuartiles, deciles y percentiles. Además, calcular el rango intercuartil del nivel 2 del curso de estadística.
4. Realizar la gráfica de los deciles con el color darkgreen para la línea de la densidad.
5. Realizar la gráfica de los percentiles con el color brown4 para la línea del percentil.

Capítulo 3. Respuestas de las autoevaluaciones

3-1

```
salario<-c(400,
800,400,750,900,1200,2100,3000,2100,500,500,1200,900,600,3000)
MS<-round(mean(salario), 2)
M<-c(salario=MS)
Me1<-round(median(salario), 2)
Me<-c(salario=Me1)
moda <- function(sueldo) {
   uvalor <- unique(sueldo)
   uvalor[which.max(tabulate(match(sueldo, uvalor)))]}
Mo1<-round(moda(salario), 2)
```

```
Mo<-c(salario=Mo1)
centralidad<-rbind(M, Me, Mo)
RESULTADO
      salario
M     1223.33
Me    900.00
Mo    400.00
```

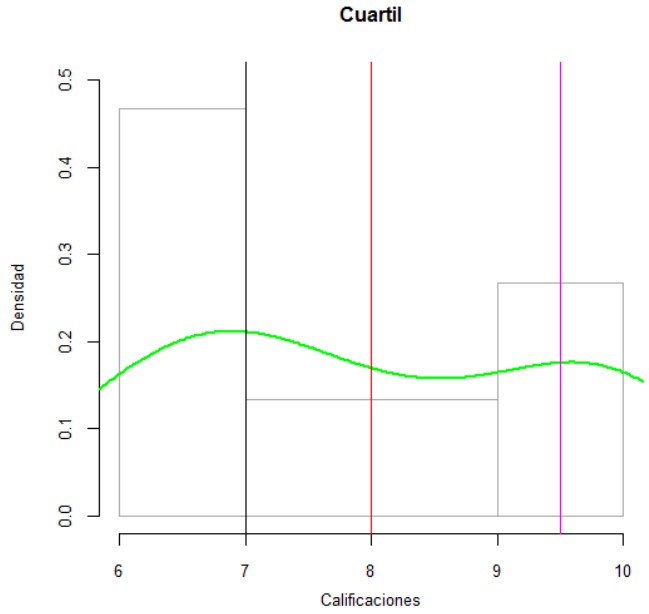
3-2

```
calificaciones<-c(7,8,9,7,10,6,7,8,9,10,10,6,8,9,9)
r=round(max(calificaciones)-min(calificaciones), 2)
Rangos<-cbind(r)
rownames(Rangos)=c("Rango")
var=round(var(calificaciones), 2)
Varianzas<-cbind(calificaciones=var)
rownames(Varianzas)=c("Varianza")
sd=round(sd(calificaciones), 2)
desv.Stand<-cbind(calificaciones=sd)
rownames(desv.Stand)=c("SD")
cv<-function(ds){
               media<-mean(ds)
               desv.stand<-sd(ds)
               cv<-round(desv.stand/media,2)
        return(cv)
      }
cv=round(cv(calificaciones), 2)
Coef.variacion<-cbind(calificaciones=cv)
rownames(Coef.variacion)=c("Coef.variacion")
      dispersion<-rbind(Rangos, Varianzas, desv.Stand, Coef.variacion)
RESULTADO
                      Calificaciones
      Rango                 4.00
      Varianza              1.89
      SD                          1.37
      Coef.variacion    0.17
```

3-3

```
calificaciones<-c(7,10,9,7,10,6,7,8,7,10,10,6,8,6,9)
# HISTOGRAMA
hist(calificaciones
     , probability = TRUE
     , main = "Cuartil"
     , cex.main=1
     , xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
     , xlim = c(6, 10)
     , ylim = c(0.000, 0.50)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Calificaciones", cex=0.8)
# FUNCIÓN DENSIDAD
lines(density(calificaciones), col="green", lwd=2)
# MEDIA
q<-quantile(calificaciones, probs = c(0.25, 0.50, 0.75))
abline(v=q[1], col="black", lty=1, lwd=1)
```

```
abline(v=q[2], col="red", lty=1, lwd=1)
abline(v=q[3], col="magenta", lty=1, lwd=1)
```



4 DISTRIBUCIONES DE PROBABILIDAD

Un concepto importante en el estudio de probabilidades es el que corresponde a variable aleatoria. Una variable aleatoria puede definirse como una característica que puede medirse y tomar valores determinados dentro de un intervalo. Si la variable aleatoria es discreta habrá un conjunto de valores determinados dentro de un intervalo y tiene una distribución de probabilidad que describe su comportamiento. Si la variable aleatoria es continua, puede tomar un infinito número de valores dentro de un intervalo, su distribución de probabilidad intuye las probabilidades correspondientes a esos valores. Algunos usos prácticos de las distribuciones de probabilidad son: Calcular intervalos de confianza para parámetros y calcular regiones críticas para pruebas de hipótesis, entre otros. Los intervalos estadísticos y las pruebas de hipótesis se basan a menudo en supuestos para una distribución específica. En esta sección, aprenderá:

- La distribución normal
- La distribución normal estándar
- Aplicaciones de las distribuciones normales
- Distribuciones muestrales y estimadores
- El teorema del límite central
- La distribución normal como aproximación de la distribución binomial
- Determinación de la normalidad

Gráficos de funciones de probabilidad con R

Para cada distribución de probabilidad, R dispone de cuatro funciones. Se puede acceder a cada función utilizando un prefijo (d, p, q, r) que define lo que hace la función. A continuación se describe el significado de cada uno de ellos:

d: Densidad

p: Distribución acumulada,

q: Cálculo de cuantiles,

r: Generar números aleatorios para la distribución.

En este libro se revisa el tratamiento de las siguientes funciones de probabilidad.

- Contínuas: Normal y Weibull
- Discretas: Binomial y Poisson.

Distribución Normal

La distribución normal es una de las distribuciones de probabilidad más importante en estadística porque se adapta a muchos fenómenos naturales. Una distribución de probabilidad normal se define mediante dos parámetros, la media (μ) y la desviación estándar (σ). A menudo se denota N(μ, σ). El dominio de una variable aleatoria normal es $-\infty < x < \infty$. Sin embargo, como cuestión práctica, el intervalo [μ - 3σ, μ + 3σ] incluye casi toda el área. Además de μ y σ, la función de densidad de probabilidad normal f (x) depende de las constantes e (aproximadamente 2.71828) y π (aproximadamente 3.14159). El valor esperado de una variable aleatoria normal es μ y su varianza es σ^2 [38]. A continuación, de acuerdo con [39], se presenta las principales características de la distribución normal.

- Una curva de distribución normal tiene forma de campana.
- La media, la mediana y la moda son iguales y se encuentran en el centro de la distribución.
- Una curva de distribución normal es unimodal (es decir, solo tiene un modo).
- La curva es simétrica respecto a la media, lo que equivale a decir que su forma es la misma en ambos lados de una línea vertical que pasa por el centro.
- La curva es continua; es decir, no hay huecos ni agujeros. Para cada valor de X, hay un valor correspondiente de Y.
- La curva nunca toca el eje x. Teóricamente, no importa cuán lejos en cualquier dirección se extienda la curva, nunca se encuentra con el eje x, pero se acerca cada vez más.
- El área total bajo una curva de distribución normal es igual a 1.00, o 100%. Este hecho puede parecer inusual, ya que la curva nunca toca el eje x, pero se puede demostrar matemáticamente mediante el cálculo.
- El área bajo la parte de una curva normal que se encuentra dentro de 1 desviación estándar de la media es aproximadamente 68%; dentro de 2 desviaciones estándar, aproximadamente 95%; y dentro de 3 desviaciones estándar, aproximadamente 99.7%.

Una curva de densidad es un modelo matemático para la distribución de una variable cuantitativa. Las curvas de densidad dan una imagen compacta del patrón general de datos. Ignoran las irregularidades menores y los valores atípicos. Para algunas situaciones, podemos capturar todas las características esenciales de una distribución con una curva de densidad. Para otras situaciones, nuestro modelo idealizado pierde algunas características importantes. Al igual que con muchas cosas en las estadísticas, se necesita su juicio cuidadoso para decidir qué es importante y qué tan

cerca es lo suficientemente bueno [40]. La distribución normal se describe mediante la **función de densidad de probabilidad** de aspecto bastante complicado, Ecuación 3.0. Para graficar la distribución normal, necesitamos conocer la media (μ) y la desviación estándar (σ). Al colocar μ, σ y un valor de la variable, x, en la función de densidad de probabilidad, podemos calcular una altura, f (x), de la función de densidad. Además, es necesario el número de Euler e (2.71828) y π (3.1416) que son constantes [41].

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-1/2(\frac{x-\mu}{\sigma})^2} \qquad [3.0]$$

Para calcular la probabilidad de algún evento dentro de los límites a y b pertenecientes a la función de densidad de la distribución normal (ecuación 3.0) se debe integrar f(x) tal como lo muestra la ecuación 3.1. La integración de esta función no es sencilla porque que es imposible encontrar su antiderivada por los métodos de integración tradicionales. Sin embargo, utilizando otros métodos como series de Taylor, series asintóticas, función gamma, entre otras es posible obtener los valores de probabilidad.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{-1/2(\frac{x-\mu}{\sigma})^2} dx \qquad [3.1]$$

Para resolver este inconveniente, los estadísticos pensaron como simplificar el proceso de integración para encontrar probabilidades de una distribución normal. Se introduce para el efecto el concepto de distribución normal estándar que facilita la tarea de integración.

La distribución normal estándar

Como cada variable normalmente distribuida tiene su propia media y desviación estándar, como se indicó anteriormente, la forma y la ubicación de estas curvas variarán. En aplicaciones prácticas, entonces, tendría que integrar para hallar el área debajo de la curva para cada variable. Para simplificar esta situación, los estadísticos usan lo que se llama la distribución normal estándar [42]. Como sugiere la regla 68–95–99.7, todas las distribuciones normales comparten muchas propiedades comunes. De hecho, todas las distribuciones normales son iguales si medimos en unidades de tamaño σ sobre la media μ como centro. Cambiar a estas unidades se llama estandarización. Para estandarizar un valor, reste la media de la distribución y luego divida por la desviación estándar [43].

La distribución normal estándar es la distribución normal N (0, 1) con media 0 y desviación estándar 1. Si una variable x tiene alguna distribución normal N (μ, σ) con media μ y desviación estándar σ [44], entonces la variable estandarizada queda de la siguiente manera:

$$z = \frac{x - \mu}{\sigma} \qquad [3.2]$$

Reemplazando el valor de z (ecuación 3.2) en la ecuación 3.0 se obtiene:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{\left(\frac{-z^2}{2}\right)} \qquad [3.3]$$

La nueva función f(z) - ecuación 3.3 - se la conoce como **función densidad de la distribución normal estándar** (PDF, por sus siglas en inglés). Es evidente que f(z) es más sencilla de integrar, ver ecuación 3.4 . Para facilitar los resultados de los valores (de probabilidad) de la integral de la función f(z) se ha creado la **tabla de distribución normal estándar** también conocida como **tabla Z**. En esta tabla se obtiene los valores de la integral de f(z) que pueden ser convertidos fácilmente a valores de la integral de f(x).

$$f(z) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{\frac{-z^2}{2}} dz \qquad [3.4]$$

En resumen, se puede decir que la distribución normal estándar es la distribución normal N (0, 1) con media 0 y desviación estándar 1. Es importante puntualizar que la regla empírica se presenta en las curvas normales con el 68%, 95% y 99,7% de los datos dentro x igual a σ, 2σ y 3σ desviaciones estándar respectivamente, mientras que en la distribución normal estándar se encuentran en z igual a 1, 2 y 3 respectivamente.

Otra función importante para el estudio de las distribuciones de frecuencia es la **función de distribución acumulativa** (CDF, por sus siglas en inglés) FX(x) describe la probabilidad de que una variable aleatoria X con una distribución de probabilidad dada se encuentre en un valor menor o igual a x. Esta función se presenta en la ecuación 3.5

$$F(X) = P[X \leq x] = \int_{-\infty}^{x} f\,x(u)du \qquad [3.5]$$

Es decir, para un valor dado x, FX (x) es la probabilidad de que el valor observado de X sea menor o igual que x. Si fx es continuo en x, entonces la función de densidad de probabilidad es la derivada de la función de distribución acumulativa.

$$\lim_{x \to -\infty} F(x) = 0 \qquad [3.6]$$

$$\lim_{x \to \infty} F(x) = 1 \qquad [3.7]$$

La función de distribución acumulativa aumenta de forma monótona, lo que significa que x1 ≤ x2 implica F (x1) ≤ F (x2). Esto se deduce simplemente del hecho de que {X ≤ x2} = {X ≤ x1} ∪ {x1 ≤ X ≤ x2} y la suma de las probabilidades de eventos disjuntos. Además, si X toma valores entre

−∞ y ∞, como la variable aleatoria gaussiana, entonces F (−∞) = 0 y F (∞) = 1, de acuerdo a las ecuaciones 3.6 y 3.7. Si la variable aleatoria X es continua y posee una densidad, p (x) , como lo hace la variable aleatoria gaussiana, se deduce inmediatamente de la definición de F, y dado que F (−∞) = 0.

Gráficos de la función densidad y acumulativa de la distribución normal

R tiene cuatro funciones nativas incorporadas para generar la distribución normal, recuerde que la primera letra es quien define lo que hace la función. A continuación se presentas dichas funciones y sus principales parámetros.

- **dnorm**(*x, mean* = 0, *sd* = 1, *log* = FALSE)
- **pnorm**(*q, mean* = 0, *sd* = 1, *lower.tail* = TRUE, *log.p* = FALSE)
- **qnorm**(*p, mean* = 0, *sd* = 1, *lower.tail* = TRUE, *log.p* = FALSE)
- **rnorm**(*n, mean* = 0, *sd* = 1)

El significado de cada uno de los parámetros de estas funciones se describe a continuación:

- *x, q*: vector de cuantiles.
- *p* : vector de probabilidades.
- *n* : número de observaciones. Si length (n)> 1, la longitud se toma como el número requerido. mean: vector de medias.
- *sd* : vector de desviaciones estándar.
- *log, log.p*: lógico; si es verdadero, las probabilidades p se dan como log (p).
- *lower.tail*: lógico; si es verdadero, las probabilidades son P[X ≤ x] sino P[X> x].

El siguiente procedimiento explica cómo conseguir hacer la gráfica de la *función densidad de probabilidad para la distribución normal estándar* en R.

1. La función **set.seed()** genera el mismo número aleatorio n veces. Esto significa que el gráfico siempre será el mismos.
2. La variable.aleatoria.x contiene una secuencia de números usando la función **sec()**. Esta secuencia de números inicia en (-4) hasta (4) en pasos de (0.1)
3. Se usa la función **dnorm()** para genera la función densidad de probabilidad con un valor para la media de cero y uno para la desviación estándar, de acuerdo a la definición de la función normal estándar.

4. Con los datos obtenidos, se emplea la función **plot()**, donde en el eje x se distribuyen los valores de variable.aleatoria.x y la altura está dada por la variable Normal.densidad.

La gráfica de cuantiles-cuantiles es una herramienta que nos ayuda a evaluar si un conjunto de datos proviene plausiblemente de alguna distribución teórica como la distribución Normal. A menudo suponemos que los datos de la muestra proceden de una distribución normal y en otras ocasiones, al contrario, se tiene el pálpito que no responde a una distribución normal. ¿Cómo podemos verificar esto?

El gráfico cuantil - cuantil es una de las formas más sencillas para verificar si una muestra pertenece a un tipo distribución específico. Nos permite ver de un vistazo si nuestra suposición es aceptable y, de no ser así, cómo se viola la suposición y qué puntos de los datos contribuyen a la violación. R tiene dos funciones para el análisis de una distribución de datos mediante cuantíles: **qqnorm**() y **qqline**(). Ambas toman los valores que tiene la función **dnorm()** quien recibe los valores de la variable aleatoria x que se desea explorear. El ejemplo simula los datos de la variable aleatoria x, de una muestra, mediante la función **seq**(). El código de abajo implementa los gráficos de una distribución normal y cuantil – cuantil. Ver figura 4.1.

```
# SE CONFIGURA LOS NUMEROS ALEATORIOS
set.seed(2345)
# SE GENERA LOS VALORES DE LA MUESTRA
# Para fines prácticos, en este vector se debe
# ubicar los valores de las observaciones o mediciones
variable.aleatoria.x<-seq(-4,4,.01)
# SE CREA LOS VALORES DE DENSIDAD PARA LA VARIABLE ALEATORIA
Normal.densidad<-dnorm(variable.aleatoria.x, 0,1)
par(mfrow=c(1,2), mar=c(5,4,2,1))
# GRÁFICO DE LA FUNCIÓN NORMAL
plot(variable.aleatoria.x, Normal.densidad
     , main="Curva Normal Estándar"
     , cex.main=0.8
     , cex=2
     , col="gray30"
     , xlab=""
     , ylab="Densidad"
     , cex.axis=.8
     , frame.plot = FALSE
     , type="l"
     , lwd=1)
# GRÁFICO CUANTIL - CUANTIL
qqnorm(Normal.densidad
       , main="Q-Q plot Curva Normal"
       , cex.main=0.8
       , pch = 1
       , col = "gray50"
       , frame = FALSE)
qqline(Normal.densidad, col = "gray20", lwd = 2)
```

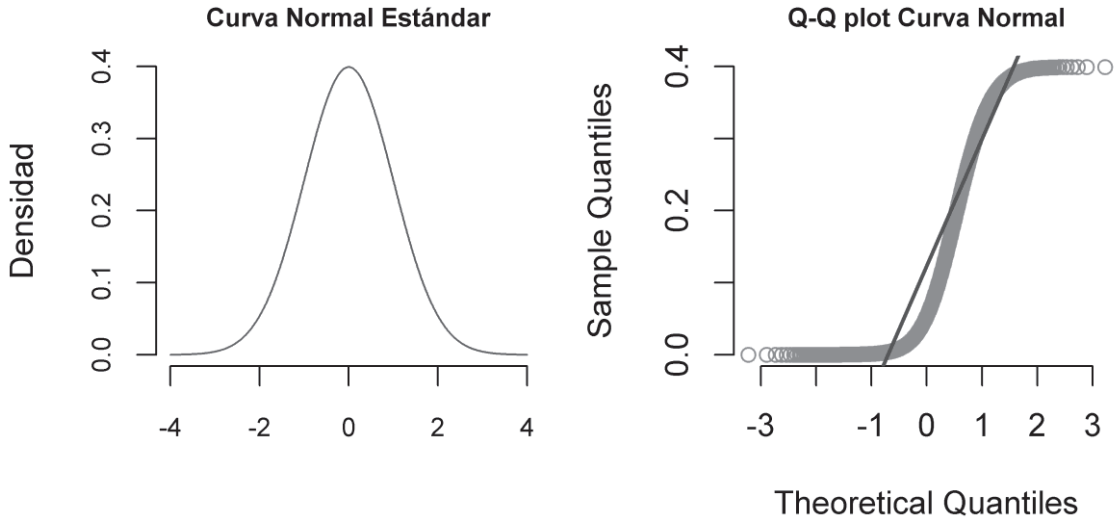


Figura 4.1 *Curva normal y gráfico cuantil - cuantil para la distribución normal*

También se puede graficar función de distribución acumulativa (CDF) para la distribución normal estándar:

1. La función **set.seed()** genera el mismo número aleatorio n veces. Esto facilita la reproducción del gráfico porque siempre se generarán los mismos datos aleatorios.
2. Se crea la variable.aleatoria.x con la función **seq**() y se obtiene una secuencia de números que inicia en (-4) hasta (4) en pasos de (0.1).
3. La función **pnorm**() produce los valores para la la función densidad de probabilidad con una media igual a cero y desviación estándar de uno. La función **pnorm**() permite calcular la proporción (probabilidad) de valores de una muestra que se encuentran antes o después de un valor Xi, siempre que conozcamos la media y desviación estándar de la muestra.
4. Con estos resultados se usa la función **plot**(), donde en el eje x se distribuyen los valores de variable.aleatoria.x y la altura está dada por la variable Normal.densidad.

Con estos simples pasos, es posible tener la representación gráfica de la función acumulada de probabilidad para la distribución normal estándar.

```
# SE CONFIGURA LOS NUMEROS ALEATORIOS
set.seed(2345)
# SE GENERA LOS VALORES DE LA MUESTRA
variable.aleatoria.x<-seq(-4,4,.01)
# SE CREA LOS VALORES DE DENSIDAD PARA LA VARIABLE ALEATORIA
Normal.acumulada<-pnorm(variable.aleatoria.x, 0, 1)
par(mfrow=c(1,1), mar=c(3,4,2,1))
# GRÁFICO DE LA FUNCIÓN NORMAL ACUMULADA
plot(variable.aleatoria.x, Normal.acumulada
     , main="Densidad Acumulada"
     , cex.main=0.8
```

```
     , col="gray30"
     , xlab=""
     , ylab="Probabilidad Acumulada"
     , cex=2
     , cex.axis=.8
     , frame.plot = FALSE
     , type="l"
     , lwd=2)
```

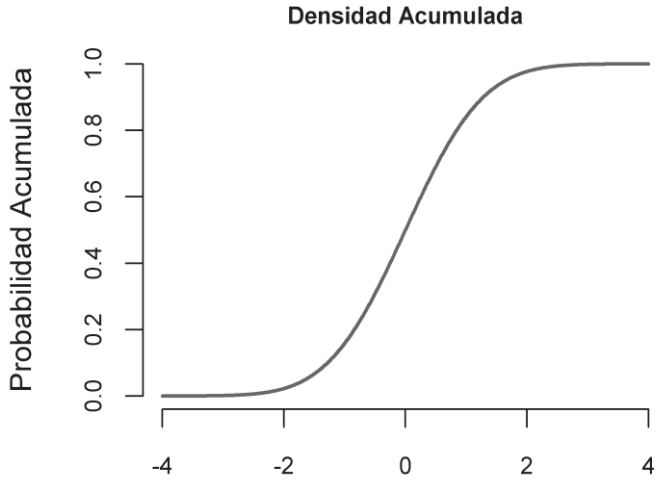


Figura 4.2 *Gráfico de Densidad Acumulada parala distribución Normal*

La figura 4.2 pertenece a la función de distribución acumulativa para la distribución normal estándar. El eje vertical tiene los valores de las probabilidades correspondientes a los valores de z que se muestran a lo largo del eje horizontal. La curva aumenta gradualmente de una probabilidad de 0.0 para valores de z alrededor de −3 a una probabilidad de 0.5 cuando z es cero y de probabilidades cercanas a 1.0 para valores de z de 3 o mayores. Podemos calcular varias probabilidades asociadas con una distribución normal usando su función de distribución acumulativa sin recurrir directamente a su función densidad de la distribución normal estándar.

Para familiarizarce con las curvas de la función densidad y función acumulada de la distribución normal, analice como se construye a partir del uso combinado de las funciones de R `curve(dnorm(x, μ, σ ),…)` y `curve(pnorm(x, μ, σ),…).` Recuerde que la distribución normal está definida N (μ, σ) para una muestra aleatoria x.

Revise el siguiente ejemplo que construye la representación gráfica de las curvas función densidad de la distribución normal estándar y función de distribución acumulativa para distribuciones normales con diferentes valores de la media y desviación estándar.

```
# SE GENERA LOS VALORES DE LA MUESTRA
set.seed(2345)
variable.aleatoria.x<-seq(-4,4,.01)
par(mfrow=c(1,2), mar=c(5,4,1.5,0))
```

```r
#-------------------------------------------------------
# Gráfico de la función acumulada usado curve() y dnorm()
#-------------------------------------------------------
curve(dnorm(x, 0, sqrt(0.2))  # Función dnorm a evaluar
      , main="Función densidad"
      , xlab = "Variable aleatoria x"
      , -5, 5, 1000           # Límites de x y nº de valores a evaluar
      , col = "gray20"        # Color de la línea
      , las = 1               # Etiquetas alineadas horizontalmente
      , ann = T               # Sin títulos en los ejes
      , xaxp = c(-5, 5, 10)   # Marcas del eje x
      , ylim = c(0,1)         # Límites del eje
      , yaxs = "i"            # Estilo del eje y, ajustado a los límites
      , frame=FALSE)
# AÑADIMOS EL RESTO DE CURVAS
curve(dnorm(x)                # Variable aleatoria
      , add = TRUE
      , col = "gray40")
curve(dnorm(x, 0, sqrt(5))    # Variable aleatoria con media=0 y
      , add = TRUE            # desviación estándar = sqrt(5)
      , col = "gray60")
curve(dnorm(x, -2, sqrt(0.5)) # Variable aleatoria con media=-2 y
      , add = TRUE            # desviación estándar = sqrt(5)
      , col = "gray80")
# NOMBRES PARA MOSTRAR EN LA LEYENDA
nombres <- expression(paste(mu, "= 0, ", sigma^2, "= 0.2")
                , paste(mu, "= 0, ", sigma^2, "= 1.0")
                , paste(mu, "= 0, ", sigma^2, "= 5.0")
                , paste(mu, "=-2, ", sigma^2, "= 0.5"))
# PUBLICAR LA LEYENDA EN EL GRÁFICO
legend("topright"             # Posición
       , legend = nombres     # Expression vector anterior
       , lty = c(1, 1, 1, 1)  # Líneas sólidas
       , bty = "n"            # Sin bordes
       , col = c("gray20", "gray40", "gray60", "gray80")
       , inset = .05          # Espaciado del margen
       , cex = 0.7            # Tamaño de letra
       , y.intersp = .75)     # Interlineado
#-------------------------------------------------------
# Gráfico de la función acumulada usado curve() y pnorm()
#-------------------------------------------------------
curve(pnorm(x, 0, sqrt(0.2))  # Función dnorm a evaluar
      , main="Función acumulativa"
      , xlab = "Variable aleatoria x"
      , -5, 5, 1000            # Límites de x y nº de valores a evaluar
      , col = "gray20"         # Color de la línea
      , las = 1                # Etiquetas alineadas horizontalmente
      , ann = T                # Sin títulos en los ejes
      , xaxp = c(-5, 5, 10)    # Marcas del eje x
      , ylim = c(0,1)          # Límites del eje
      , yaxs = "i"             # Estilo del eje y, ajustado a los límites
      , frame=FALSE)
curve(pnorm(x)                 # Función pnorm a evaluar
      , add = TRUE
      , col = "gray40")
```

```
curve(pnorm(x, 0, sqrt(5))     # Función pnorm a evaluar con media=0 y
      , add = TRUE             # desviación estándar = sqrt(5)
      , col = "gray60")
curve(pnorm(x, -2, sqrt(0.5))  # Función pnorm a evaluar con media=-2 y
      , add = TRUE             # desviación estándar = sqrt(5)
      , col = "gray80")
# LEYENDA
nombres <- expression(paste(mu, "= 0, ", sigma^2, "= 0.2")
                , paste(mu, "= 0, ", sigma^2, "= 1.0")
                , paste(mu, "= 0, ", sigma^2, "= 5.0")
                , paste(mu, "=-2, ", sigma^2, "= 0.5"))
legend("bottomright"
       , legend = nombres
       , lty = c(1, 1, 1, 1)
       , bty = "n"
       , col = c("gray20", "gray40", "gray60", "gray80")
       , inset = .05
       , cex = 0.7
       , y.intersp = 0.75)
```

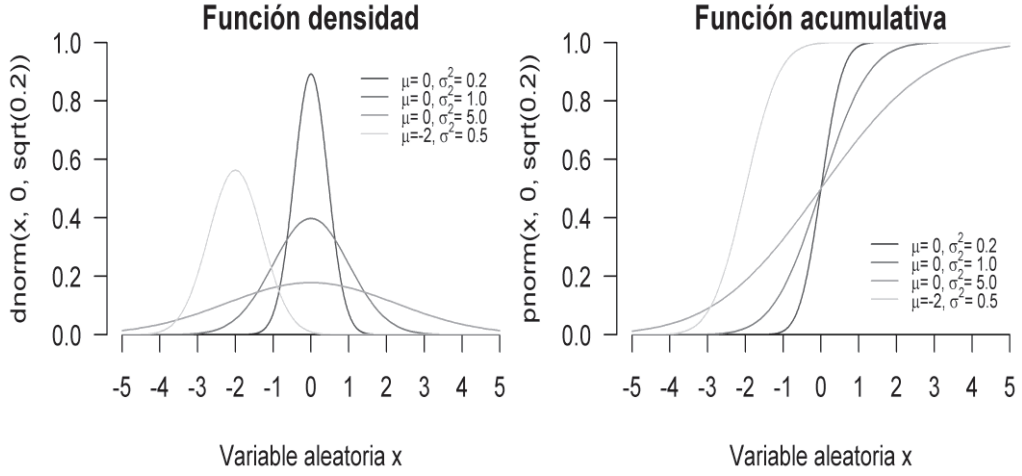


Figura 4.3 *Distribuciones Normales, función de distribución Densidad y Acumulada*

La figura 4.3 enseña un conjunto de curvas de la función densidad y función acumulada para distribuciones de probabilidad normal. El código en R resulta simple porque únicamente necesita las siguientes funciones: **hist**() para crear los histogramas, la combinación de las funciones **curve**() y **dnorm**() para construir la distribución normal teórica y **qqnorm**() y **qqline**() para obtener el gráfico qqplot.

La figura 4.4 consta del histograma de frecuencia de la variable aleatoria x que se ha medido u observado; es decir, es la representación gráfica de los datos reales. A continuación se presenta la curva de la distribución normal teórica sobre el histograma de datos reales. Finalmente, se tiene el gráfico cuantil – cuantil que compara la distribución de datos real contra la distribución de datos teórica de la distribución normal. De esta forma es posible verificar de forma gráfica si los datos de la muestra tienen la característica de una distribución normal.

```
# SE GENERA LOS VALORES DE LA MUESTRA
set.seed(3545)
variable.aleatoria.x<-seq(-4,4,.01)
Normal.desviacion.aleatoria<-rnorm(100,0,1)
par(mfrow=c(1,3), mar=c(3,4,2,1))
# HISTOGRAMA DE FRECUENCIAS
hist(Normal.desviacion.aleatoria
     , main="Histograma de Frecuencia"
     , ylab="Frecuencia"
     , cex.axis=.8
     , xlim=c(-4,4)
     , border = "gray50")
# HISTOGRAMA DE FRECUENCIAS CON LA FUNCIÓN DENSIDAD
hist(Normal.desviacion.aleatoria
     , main="Curva Normal"
     , ylab="Probabilidad"
     , probability=TRUE
     , cex.axis=.8
     , xlim=c(-5,5)
     , border = "gray50")
curve(dnorm(x
      , mean(Normal.desviacion.aleatoria)
      , sd(Normal.desviacion.aleatoria))
      , col = "gray20"
      , lty = 1
      , lwd = 2
      , add=T)
# GRÁFICO CUANTIL - CUANTIL
qqnorm(Normal.densidad
       , main="QQ plot Distribución Normal"
       , cex.main=1.1
       , pch = 1
       , col = "gray50"
       , frame = FALSE)
qqline(Normal.densidad, col = "gray20", lwd = 2)
```

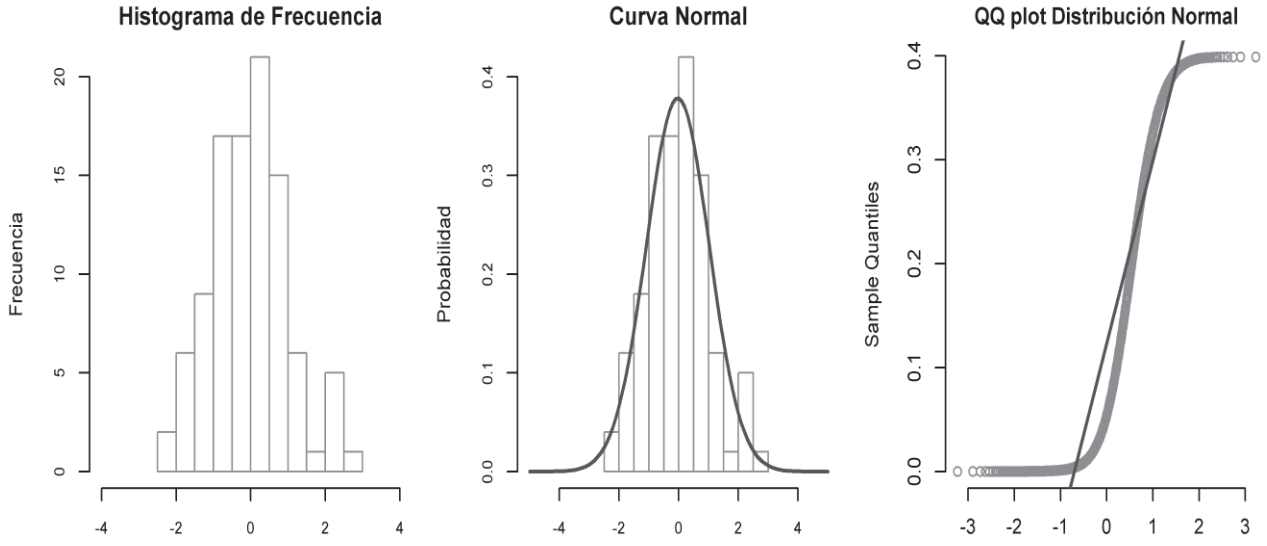


Figura 4.4 *Histograma de Frecuencias con su distribución normal*

Distribución Binomial

El modelo de distribución binomial se enfoca en la probabilidad de éxito de un evento que tiene únicamente dos resultados posibles en una serie de experimentos. En el año 2008, Jay L. Devore[45] sobre la distribución binomial expone que muchos experimentos implican una secuencia de ensayos independientes para los cuales existen más de dos resultados posibles en cualquier ensayo. Entonces, un experimento binomial puede crearse dividiendo los posibles resultados en dos grupos.

1. El experimento consta de una secuencia de n experimentos más pequeños llamados en- sayos, donde n se fija antes del experimento.
2. Cada ensayo puede dar por resultado uno de los mismos dos resultados posibles (ensayos dicotómicos), los cuales se denotan como éxito (E) y falla (F).
3. Los ensayos son independientes, de modo que el resultado en cualquier ensayo particular no influye en el resultado de cualquier otro ensayo.
4. La probabilidad de éxito es constante de un ensayo a otro; esta probabilidad se denota por **p**.

Por ejemplo, lanzar una moneda siempre da una cara o un sello. Entonces se plantea que probabilidad existe de encontrar exactamente 3 caras al lanzar una moneda 10 veces. Esto experimento se ajusta a la distribución binomial. La ecuación 3.8 es el modelo matemático que describe la distribución binomial.

$$P(x = k) = \frac{n!}{k!\,(n-k)!} p^k q^{(n-k)} \qquad [3.8]$$

De acuerdo a [46], la distribución de probabilidad binomial es una distribución de probabilidad discreta que tiene muchas aplicaciones. Está asociado con un experimento de varios pasos que llamamos experimento binomial. Un experimento binomial exhibe cuatro propiedades.

Gráfico de la función densidad para la distribución binomial

R tiene cuatro funciones nativas incorporadas para generar la distribución Binomial, las mismas que se describen a continuación.

- **dbinom**(*x, size, prob, log* = FALSE)
- **pbinom**(*q, size, prob, lower.tail* = TRUE, log.p = FALSE)
- **qbinom**(*p, size, prob, lower.tail* = TRUE, log.p = FALSE)
- **rbinom**(*n, size, prob*)

El significado de los parámetros de estas funciones es:

- *x, q*: vector de cuantiles.
- *p* : vector de probabilidades.
- *n* : número de observaciones. Si length (n)> 1, la longitud se toma como el número requerido.
- *size*: número de ensayos prob: probabilidad de éxito de cada ensayo
- *log, log.p*: lógico; si es VERDADERO, las probabilidades p se dan como log (p).
- *lower.tail*: lógico; si es VERDADERO, las probabilidades son P [X ≤ x] de lo contrario, P [X> x].

En el ejemplo, se genera para x una secuencia de números de 0 a 12. Luego, mediante la función **dbinom()** obtenemos los valores de *y* que se grafican con la función **plot**(). Configuramos el parámetro *size* con un valor de 12 con una probabilidad *prob* de 0.5 que representa el 50%. Ver figura 4.5.

```
par(mfrow=c(1,1), mar=c(4,4,2,1))
# CREAR 12 NÚMEROS EN SECUENCIA DE 1
x<-seq(0:12)
# CREAR LA DISTRIBUCIÓN BINOMIAL
y<-dbinom(x,12,0.5)
# VISUALIZAR LA DISTRIBUCIÓN BINOMIAL
plot(x,y
     , main="Distribución binomial"
     , xlab="k"
     , ylab="P(X=k)"
     , type = "h"
     , lwd=1
     , lty=1
     , col="gray50"
     , frame=F)
points(x,y, cex=1.4, col="gray30", pch=19)
text(10,0.22,expression(dbinom(x= 0:12, size=10, prob = 0.5)), col="gray10",
cex=0.8)
```

Función densidad acumulada binomial

Una distribución de probabilidad binomial acumulativa significa que cuanto mayor sea el número de ensayos, mayor será la probabilidad general de que ocurra un evento. La ecuación 3.9 modela la función de probabilidad acumulativa para la distribución binomial.

$$F(x;p,n) = \sum_{i=0}^{x} \binom{x}{i} p^i (1-p)^{(n-i)} \qquad [3.9]$$

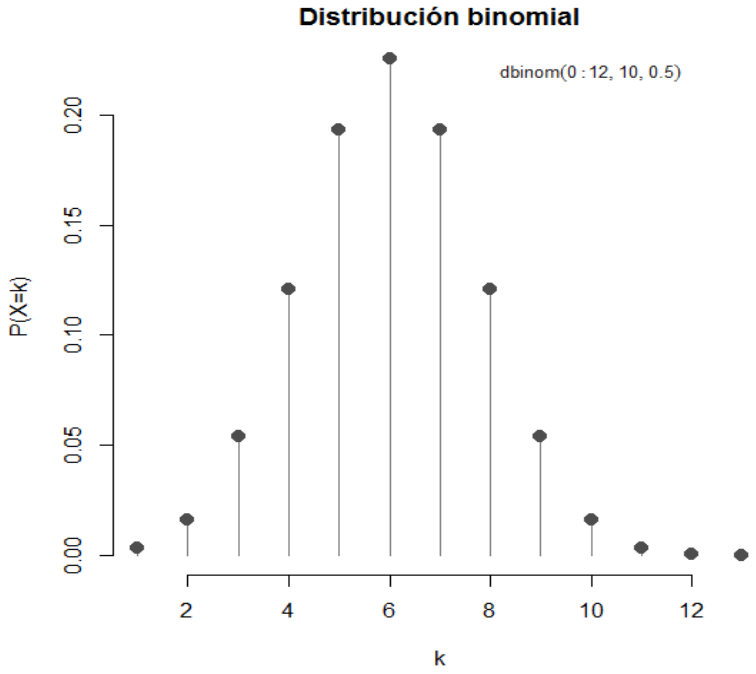


Figura 4.5 *Función densidad para la distribución binomial*

La función densidad de la distribución binomial se grafica con la función **pbinom()** que toma los parámetros: *x*, *size* y *prob*. La variable *x* resulta de aplicar la función **seq**() para obtener el vector de cuantiles. Luego, *size* es el número de ensayos y *prob* la probabilidad de éxito. Pasamos los valores de (x, y) a la función **plot()** para obtener el gráfico de la figura 4.6.

```
par(mfrow=c(1,1), mar=c(4,4,2,1))
# CREAR 12 NÚMERO EN SECUENCIA DE 1
x <- seq(0:12)
# CREAR LA DISTRIBUCIÓN BINOMIAL
y <- pbinom(x,10,0.2)
# VISUALIZAR LA DISTRIBUCIÓN BINOMIAL
plot(x,y
     , main="Distribución Binomial Acumulada"
     , xlab="k"
     , ylab="P(X=k)"
     , type = "h"
     , lwd=1
     , lty=1
     , col="gray50"
     , frame=F)
# AGREGAR PUNTOS
points(x,y, cex=1.5, col="gray30", pch=19)
```

Aproximación de la distribución Normal a la distribución Binomial

Una variable aleatoria binomial se definió como el número de éxitos observados en n ensayos independientes de un experimento aleatorio en el que cada ensayo resultó en un éxito (S) o un fracaso (F) y P (S) p para todos n ensayos.

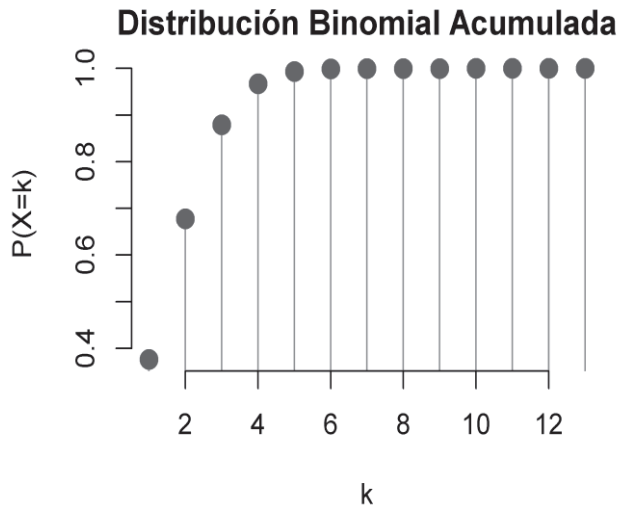


Figura 4.6 *Función acumulada para la distribución binomial*

Mediante el Teorema del Límite Central veremos cómo calcular las probabilidades para una variable aleatoria binomial usando una curva normal como una aproximación a la distribución binomial. Las probabilidades asociadas con los valores de la variable aleatoria se pueden calcular para un experimento binomial, dado cualquier valor de n o p, pero la tarea se vuelve más difícil cuando n aumenta debido a sus factoriales [47]. Revisemos el método general propuesto por [48] para aproximar las probabilidades binomiales por áreas bajo una curva normal:

Paso 1: Encuentra n, el número de intentos yp, la probabilidad de éxito.

Paso 2: Continúe solo si np y n(1 - p) son 5 o más.

Paso 3: Encuentra μ y σ, usando las ecuaciones 3.10 y 3.11 respectivamente.

Paso 4: Realice la corrección para la continuidad y encuentre el área requerida bajo la curva normal con los parámetros μ y σ.

$$\mu = np \qquad [3.10]$$

$$\sigma = \sqrt{np(1-p)} \qquad [3.11]$$

El gráfico de la distribución binomial en R no tiene ningún cambio a lo revisado en los apartados previos. Lo que resta por hacer es calcular la media y la desviación estándar por medio de las ecuaciones 3.10 y 3.11 para transformar los valores de la distribución binomial a una distribución normal, con los valores obtenidos se procede a usar la función **curve()**. Revise el siguiente código y la figura 4.7.

```
# CREAR 50 NÚMERO EN SECUENCIA DE 1
x<-seq(0:12)
# CREAR LA DISTRIBUCIÓN BINOMIAL
y<-dbinom(x,10,0.4)
```

```
# DIVISIÓN DE LA SALIDA GRÁFICA
par(mfrow=c(1,1), mar=c(4,4,2,1))
# VISUALIZAR LA DISTRIBUCIÓN BINOMIAL
plot(x,y
     , main="Cambio de Binomial a Normal"
     , xlab="k"
     , ylab="P(X=k)"
     , type = "h"
     , lwd=1
     , lty=1
     , col="gray50"
     , frame=F)
points(x,y, cex=1.5, col="gray30", pch=19)
media   = 10*0.4
desviacionEstandar  =sqrt(10*0.4*0.6)
# GRÁFICO DE LA CURVA NORMAL
curve(dnorm(x,media,desviacionEstandar), lwd=1, lty=1, col="gray20", add=T)
```

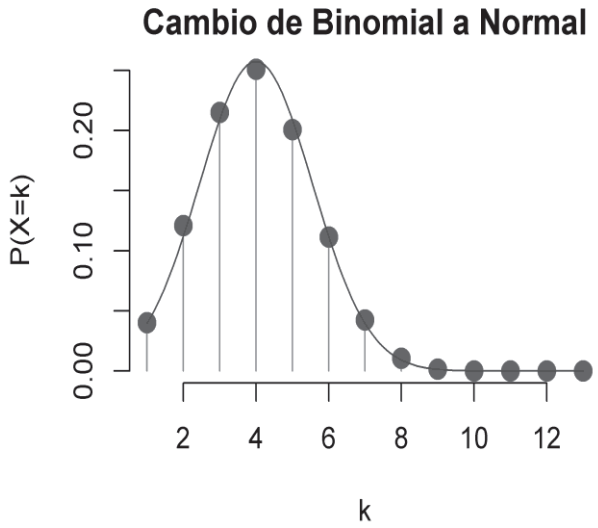


Figura 4.7 *Aproximación de la distribución binomial a la distribución normal*

Distribución de Poisson

Walpole[49] define que los experimentos que producen valores numéricos de una variable aleatoria X, el número de resultados que ocurren durante un intervalo de tiempo determinado o en una región específica, se denominan experimentos de Poisson. El número X de resultados que ocurren durante un experimento de Poisson se llama variable aleatoria de Poisson y su distribución de probabilidad se llama distribución de Poisson.

El experimento debe cumplir las siguientes propiedades:

1. El número de resultados que ocurren en un intervalo o región específica es independiente del número que ocurre en cualquier otro intervalo de tiempo o región del espacio disjunto. De esta forma vemos que el proceso de Poisson no depende de otros resultados previos.

2. La probabilidad de que ocurra un solo resultado durante un intervalo de tiempo muy corto o en una región pequeña es proporcional a la longitud del intervalo o al tamaño de la región, y no depende del número de resultados que ocurren fuera de este intervalo de tiempo o región.
3. La probabilidad de que ocurra más de un resultado en tal intervalo de tiempo corto o que caiga en tal región pequeña es insignificante.

Es importante conocer las propiedades de cada experimento para seleccionar la distribución de probalidad apropiada. La ecuación 3.12 modela la probabilidad de la distribución Poisson.

$$P(x;\lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!} \qquad [3.12]$$

e : La base del logaritmo natural igual a 2.71828

x : El número de ocurrencias de un evento; cuya probabilidad viene dada por la función x!

λ : Es el parámetro de forma que indica el número promedio de eventos en el intervalo de tiempo.

Gráfico de función densidad para la distribución de Poisson

R tiene cuatro funciones nativas incorporadas para generar la distribución de Poisson las mismas que se describen a continuación.

- **dpois**(*x, lambda, log* = FALSE)
- **ppois**(*q, lambda, lower.tail* = TRUE, *log.p* = FALSE)
- **qpois**(*p, lambda, lower.tail* = TRUE, *log.p* = FALSE)
- **rpois**(*n, lambda*)

Los parámetros necesarios para generar la distribución se presentan en la siguiente descripción:

- *x* = vector de cuantiles (enteros no negativos).
- *q* = vector de cuantiles.
- *p* = vector de probabilidades.
- *n* = número de valores aleatorios que se devolverán.
- *Lambda* = vector de medias (no negativas).
- *log.p* = lógico; si es verdadero, las probabilidades p se dan como log (p).
- *lower.tail* = lógico; si es verdadero (predeterminado), las probabilidades son P [X ≤ x], de lo contrario, P [X> x].

Para graficar la función densidad de la probabilidad de Poisson se usa la función **dpois()** de R. Primero generamos la variable independiente x con **seq()**, ésta variable se pasa como parámetro a la función **dpois**(*x, lambda*) y se configura el valor de *lambda*; el resultado de la función **dpois**() se guarda en la variable y. Los valores de (x,y) pasan a la función **plot**() para obtener el gráfico deseado, ver figura 4.8.

```
# CREAR 12 NÚMEROS EN SECUENCIA DE 1
x<-seq(0:12)
# CREAR LA DISTRIBUCIÓN POISSON
y<-dpois(0:12, 2.1)
# VISUALIZAR LA DISTRIBUCIÓN POISSON
par(mfrow=c(1,1), mar=c(4,4,2,1))
plot(x,y
     , main="Función densidad Poisson"
     , xlab="k"
     , ylab="P(X=k)"
     , type = "h"
     , lwd=1
     , lty=1
     , col="gray50"
     , frame=F)
points(x,y, cex=1.4, col="gray30", pch=19)
text(8,0.15,expression(dpois(x= 0:12, size=10, prob = 0.2)), col="gray10", ce
x=0.8)
```

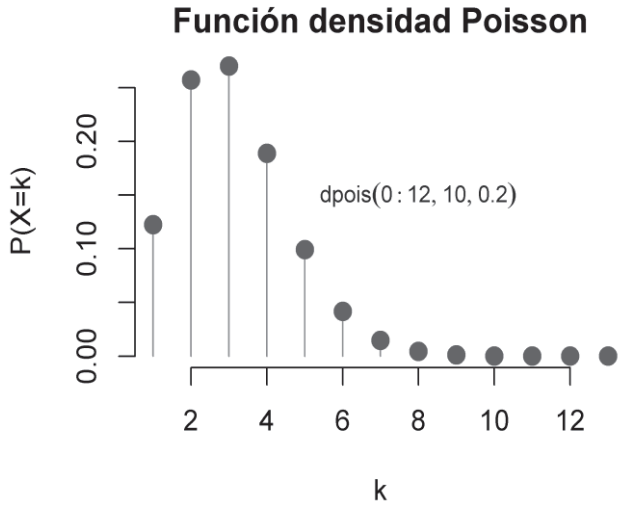


Figura 4.8 *Función densidad de la distribución de Poisson*

Función acumulada de la distribución de Poisson

La función de distribución acumulativa de Poisson le permite obtener la probabilidad de que ocurra un evento dentro de un intervalo de tiempo o espacio dado menor o igual que x veces si, en promedio, el evento ocurre λ veces dentro de ese intervalo.

La función de distribución acumulativa de Poisson para los valores dados x y λ se muestra en la ecuación 3.13:

$$F(x,\lambda) = \sum_{k=0}^{x} \frac{e^{-\lambda}\lambda^{k}}{k!} \qquad [3.13]$$

Para graficar la función acumulada de la probabilidad de Poisson se usa la función **ppois()** de R. De igual forma, generamos la variable independiente x con **seq()** y la pasamos como parámetro a la función **ppois**(*x, lambda*) y se configura el valor de *lambda*; el resultado de la función **ppois**() se guarda en la variable y. Con los valores de (x,y) pasan a la función **plot**() para obtener el gráfico de la figura 4.9.

```
# CREAR 50 NÚMERO EN SECUENCIA DE 1
x<-seq(0:12)
# CREAR LA DISTRIBUCIÓN POISSON
y<-ppois(0:12, 2.1)
# VISUALIZAR LA DISTRIBUCIÓN POISSON
par(mfrow=c(1,1), mar=c(4,4,2,1))
plot(x,y
     , main="Distribución de Poisson Acumulada"
     , xlab="k"
     , ylab="P(X=k)"
     , type = "h"
     , lwd=1
     , lty=1
     , col="gray50"
     , frame=F)
points(x,y, cex=1.4, col="gray30", pch=19)
```

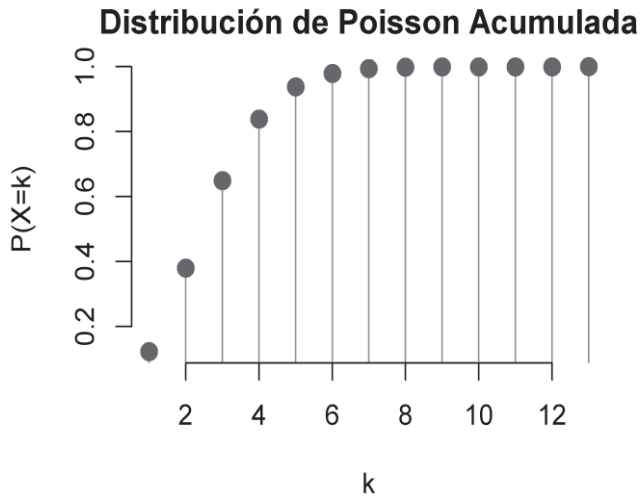


Figura 4.9 *Función acumulada de la distribución de Poisson*

Aproximación de la distribución Poisson a la distribución Normal

Cuando la media de la distribución de Poisson es grande, puede ser difícil calcular las probabilidades de Poisson usando una calculadora. Afortunadamente, cuando μ es grande, las probabilidades de Poisson se pueden aproximar usando la distribución Normal con media μ y desviación estándar σ [50].

Esto significa que para graficar la distribución normal sobre la distribución de Poisson tenemos que $\mu = \lambda$ y $\sigma = \sqrt{\lambda}$. Calculado estos valores procedemos a usar **curve**(**dnorm**(*x, μ, σ*),…) y seguir los procedimientos explicados en el apartado de la distribución normal.

```
# CREAR 90 NÚMERO EN SECUENCIA DE 1
x<-seq(0:80)
# CREAR LA DISTRIBUCIÓN POISSON
y<-dpois(0:80, 60)
# VISUALIZAR LA DISTRIBUCIÓN POISSON
par(mfrow=c(1,1), mar=c(4,3,2,1))
plot(x,y
     , main="Función densidad Poisson"
     , xlab="k"
     , ylab="P(X=k)"
     , type = "h"
     , lwd=1
     , lty=1
     , col="gray50"
     , frame=F)
points(x,y, cex=1.1, col="gray40", pch=21)
curve(dnorm(x, mean = 60, sd = sqrt(60)),add = TRUE, col = 'gray20', lwd=2)
```

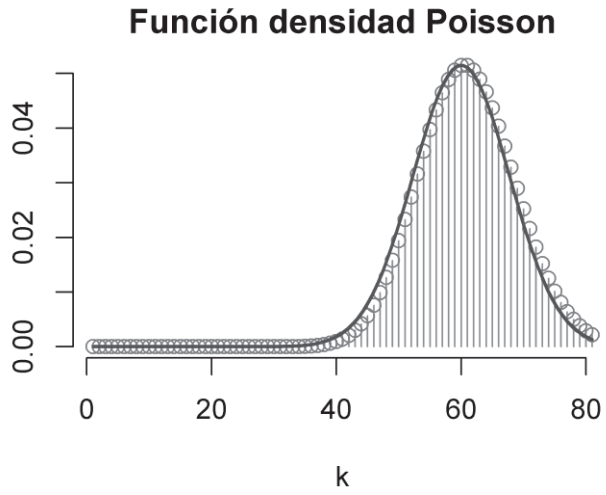


Figura 4.9 *Función acumulada de la distribución de Poisson*

La figura 4.9 muestra la aproximación de la curva de la distribución normal con la distribución de Poisson. Note el sesgo hacia la izquierda, pese a ello se considera una buena aproximación.

Distribución de Weibull

Waloddi Weibull(1951) se interesa por el estudio de la confiabilidad que los ingenieros hoy en día necesitan calcular dado el desarrollo de sistemas complicados cuya operación y seguridad dependen justamente de la confiabilidad de los diversos componentes que conforman los sistemas[51]. La distribución de Weibull es una distribución de probabilidad continua que originalmente propuso la distribución como un modelo para la resistencia a la rotura de materiales,

pero se reconoció su potencial de amplia aplicabilidad. La forma más general de la densidad de Weibull está dada por la ecuación 3.14:

$$f(t) = \frac{\alpha}{\beta}(\frac{t}{\beta})^{\alpha-1}e^{(-\frac{t}{\alpha})^{\beta}} \qquad [3.14]$$

Hoy en día, se usa comúnmente para evaluar la confiabilidad del producto, analizar los datos de vida y los tiempos de falla del modelo. El Weibull también puede adaptarse a una amplia gama de datos de muchos otros campos, incluidos: biología, economía, ciencias de la ingeniería e hidrología.

Función densidad de la distribución de Weibull

Al igual que las anteriores distribuciones, R tiene cuatro funciones nativas para trabajar con la distribución de Weibull, a saber:

- **dweibull**(*x, shape, scale* = 1, *log* = FALSE)
- **pweibull**(*q, shape, scale* = 1, *lower.tail* = TRUE, *log.p* = FALSE)
- **qweibull**(*p, shape, scale* = 1, *lower.tail* = TRUE, *log.p* = FALSE)
- **rweibull**(*n, shape, scale* = 1)

Los parámetros necesarios para generar la distribución se presentan en la siguiente descripción:

- *x*: q vector de cuantiles.
- *p*: vector de probabilidades.
- *n*: número de observaciones. Si length (n)> 1, la longitud se toma como el número requerido.
- *shape, scale*: forma y parámetros de escala, este último predeterminado a 1.
- *log, log.p*: lógico: si es VERDADERO, las probabilidades p se dan como log (p).
- *lower.tail*: si es VERDADERO (predeterminado), las probabilidades son P [X <= x], de lo contrario, P [X> x].

Para graficar la función densidad de la probabilidad de Weibull se usa la función **dweibull()** de R. Primero generamos la variable independiente x con **seq()**, ésta variable se pasa como parámetro a la función dweibull(x, shape, scale) y se configura el valor de shape y scale; el resultado de la función dweibull se guarda en la variable y. Con los valores de (x,y) pasan a la función plot() para obtener el gráfico de la función de Weibull, ver figura 4.10.

```
# CREAR 100 NÚMERO EN SECUENCIA DE 1
x<-seq(0,100,by=1)
# CREAR LA DISTRIBUCIÓN WEIBULL
y<-dweibull(x,shape = 4,scale=10)
# VISUALIZAR LA DISTRIBUCIÓN WEIBULL
par(mfrow=c(1,1), mar=c(4,3,2,1))
plot(x,y, main="Función densidad de Weibull"
     , type = "l"
     , lwd=1
     , lty=1
     , col="gray15"
     , frame=F)
text(35,0.10,expression(dweibull(x,shape = 4,scale=10)), col="gray10", cex=0.
8)
```

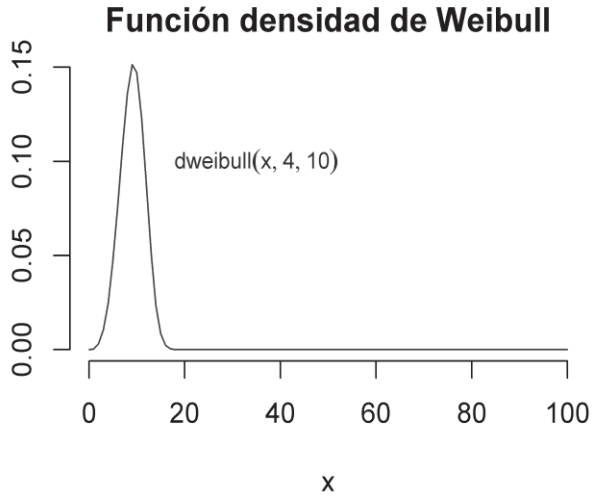


Figura 4.10 *Función densidad de la distribución de Weibull*

Función acumulada de la distribución de Weibull

F(t) es la probabilidad acumulativa de falla desde el tiempo cero hasta el tiempo t. Muy útil al estimar la proporción de unidades que fallarán durante un período de garantía. La función de distribución acumulativa de Weibull se muestra en la ecuación 3.15:

$$F(t) = 1 - e^{(-\frac{t}{\alpha})^{\beta}} \qquad [3.15]$$

Para graficar la función acumulada de la probabilidad de Poisson se usa la función **pweibull** de R. De igual forma, generamos la variable independiente x con **seq()** y la pasamos como parámetro a la función **pweibull**(*x, shape, scale*) y se configura los valores de shape y scale de acuerdo a lo que requiera; el resultado de la función **pweibull**() se guarda en la variable y. Con los valores de (x,y) pasan a la función **plot**() para obtener el gráfico deseado, ver figura 4.11.

```
# CREAR 100 NÚMERO EN SECUENCIA DE 1
x<-seq(0,100,by=1)
# CREAR LA DISTRIBUCIÓN WEIBULL
y<-pweibull(x,shape = 4,scale=10)
```

```
# VISUALIZAR LA DISTRIBUCIÓN DE WEIBULL
par(mfrow=c(1,1), mar=c(4,3,2,1))
plot(x,y, main="Función acumulada de Weibull"
#     , xlab="k"
#     , ylab="P(X=k)"
     , type = "l"
     , lwd=1
     , lty=1
     , col="gray15"
     , frame=F)
text(35,0.60,expression(pweibull(x,shape = 4,scale=10)), col="gray10", cex=0.
8)
```

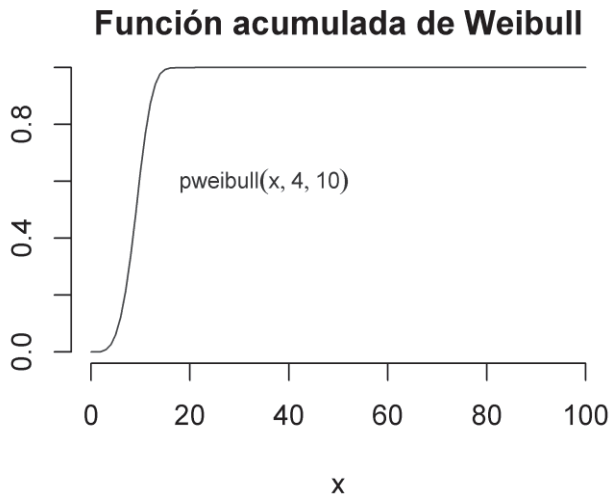


Figura 4.11 *Función densidad de la distribución de Weibull*

Histograma de frecuencia para la función densidad Weibull

La velocidad del viento cambia continuamente, por lo que es necesario describirlo de forma estadística. Es conveniente establecer un modelo de las frecuencias de las velocidades del viento. Hay varias funciones que se pueden utilizar para describir la frecuencia de la distribución de velocidades del viento pero las más utilizadas son las funciones de Weibull y Rayleigh. En ocasiones es importante graficar el histograma de frecuencia junto con la distribución de Weibull para apreciar de mejor manera lo que está ocurriendo con el conjunto de datos. El histograma muestra las ocurrencias de las variables agrupadas dentro de clases en un rango de tiempo (t).

En R, es basta simple combinar ambos gráficos. Lo primero que vamos hacer es usar la función **rweibull()** para generar la distribución de Weibull de forma aleatoria, este resultado se guarda en la variable x. A continuación, utilizamos la función **hist()** tomando x como parámetro que guarda el conjunto de datos que guarda la distribución y configurando el parámetro *probability* con TRUE. Con esto graficamos el histograma de frecuencias de la distribución. Para graficar la distribución usamos la función **lines()** que toma como parámetro el resultado de la función **density()** aplicado a x. Ver figura 4.12.

```
set.seed(2511)
# CREAR LA DISTRIBUCIÓN WEIBULL
x<-rweibull(n =100,shape = 4,scale = 10)
# VISUALIZAR LA DISTRIBUCIÓN WEIBULL
par(mfrow=c(1,1), mar=c(4,5,2,1))
hist(x
     , probability = TRUE
     , main="Histograma y distribución Weibull"
     , ylim = c(0.0, 0.22)
     , border="gray"
     , col="gray90")
lines(density(x), col="gray40", lwd=2)
```

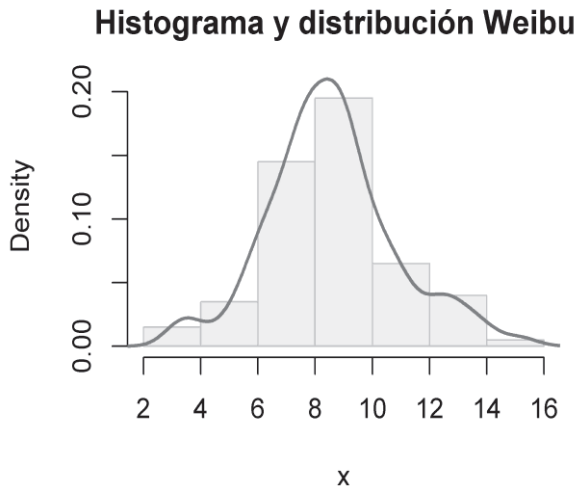


Figura 4.12 *Histograma de Frecuencia para la distribución de Weibull*

En este capítulo se ha dado las pautas necesarias para trabajar con gráficos para distribuciones de probabilidad. R tiene un buen número de funciones nativas para este cometido y existen muchas librerías que facilitan realizar visualizaciones de distribuciones de probabilidad más complejas.

AUTOEVALUACIÓN

Autoevaluación 4-1

Genere 70 números en secuencia de 1 y grafique la función densidad de Poisson, el color de la curva debe ser indianred3 y el de los puntos lightgoldenrod3.

Autoevaluación 4-2

Obtenga 90 números en secuencia de 1 y grafique la función densidad de Weibull, el color de la curva debe ser chocolate.

Autoevaluación 4-3

Crear un histograma y distribución de Weibull por medio de una distribución aleatoria con el color de línea orangered2, color del histograma wheat2, grosor de la línea de 4 para la distribución de Weibull

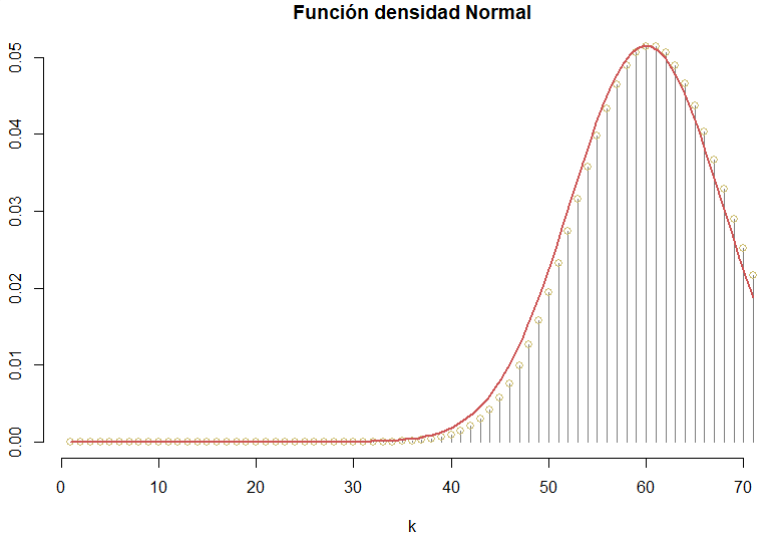
EJERCICIOS DEL CAPÍTULO

1. Realice la gráfica de la distribución normal con valores desde -3 a 3 con el color de línea magenta.
2. Realice la gráfica de la densidad acumulada con valores desde -3 a 3 con el color de línea darkorange3.
3. Con la secuencia de 0 a 24 cree el gráfico de la función binomial color de línea Brown , color de puntos darkgoldenrod3.
4. Cambie la función binomial a normal con una secuencia de 0 a 24 ubique el color turquoise4 para la normal y tan2 para el color de los puntos seagreen4.
5. Con la secuencia de 0 a 24 cree el gráfico de la función densidad de Poisson color de línea green4, color de puntos orange3.

Respuestas de las autoevaluaciones

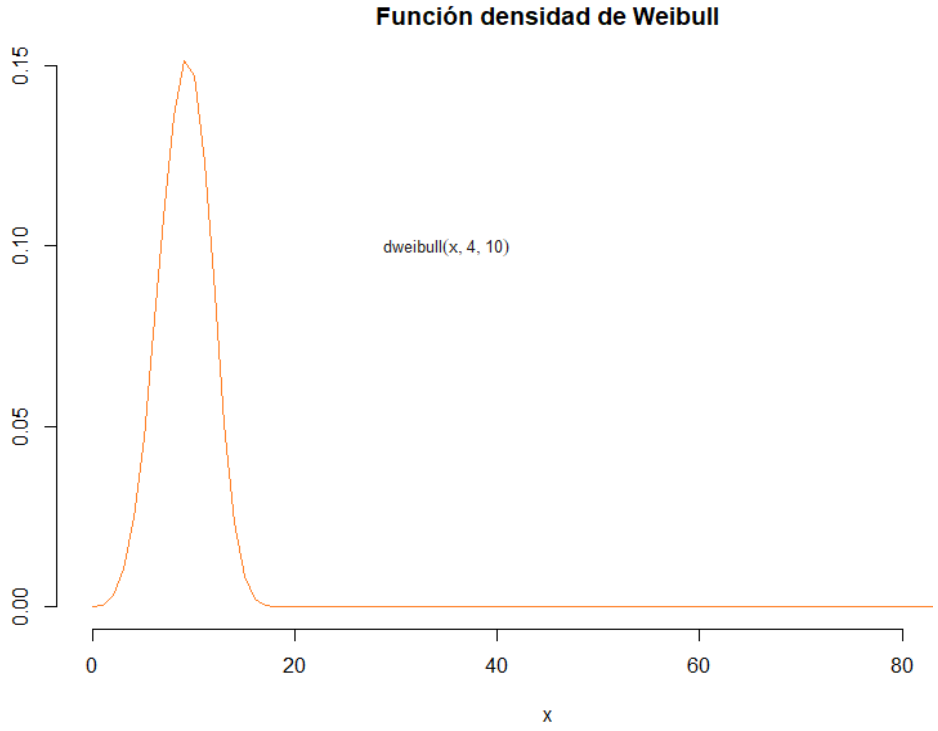
4-1

```
x<-seq(0:70)                          # Crear 70 número en secuencia de
1
y<-dpois(0:70, 60)                    # Crear la Distribución Poisson
par(mfrow=c(1,1), mar=c(4,3,2,1))
plot(x, y                            # Visualizar la Distribución Normal
      , main="Función densidad Poisson"
      , xlab="k"
      , ylab="P(X=k)"
      , type = "h"
      , lwd=1
      , lty=1
      , col="gray50"
      , frame=F)
points(x,y, cex=1.1, col=" lightgoldenrod3", pch=21)
curve(dnorm(x,mean=60, sd=sqrt(60)), add=TRUE, col='indianred3', lwd=2)
```



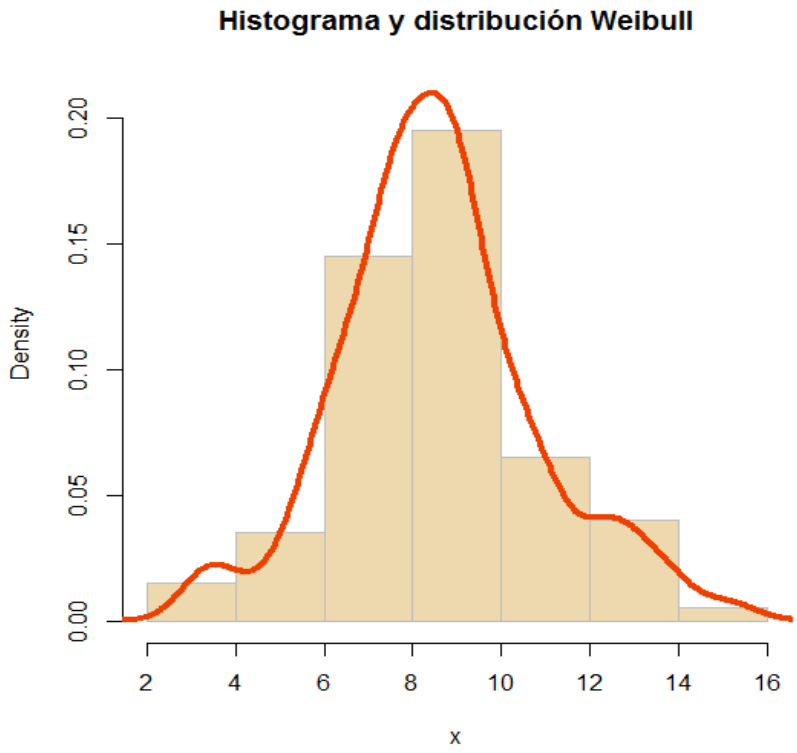
4-2.

```
# CREAR 90 NÚMERO EN SECUENCIA DE 1
x<-seq(0,90,by=1)
# CREAR LA DISTRIBUCIÓN WEIBULL
y<-dweibull(x,shape = 4,scale=10)
# VISUALIZAR LA DISTRIBUCIÓN WEIBULL
par(mfrow=c(1,1), mar=c(4,3,2,1))
plot(x,y
      , main="Función densidad de Weibull"
      , type="l"
      , lwd=1
      , lty=1
      , col="chocolate1"
      , frame=F)
text(35,0.10
      , expression(dweibull(x,shape=4,scale=10))
      , col="gray10"
      , cex=0.8)
```



4-3

```
set.seed(2511)
# CREAR LA DISTRIBUCIÓN WEIBULL
x<-rweibull(n =100,shape = 4,scale = 10)
# VISUALIZAR LA DISTRIBUCIÓN WEIBULL
par(mfrow=c(1,1), mar=c(4,5,2,1))
hist(x
          , probability = TRUE
          , main="Histograma y distribución Weibull"
          , ylim = c(0.0, 0.22)
          , border="gray"
          , col=" wheat2")
lines(density(x), col=" orangered2", lwd=4)
```


Histograma y distribución Weibull
Density
0.00
0.05
0.10
0.15
0.20
2
4
6
8
10
12
14
16
x

5 ESTIMACIÓN E INTERVALOS DE CONFIANZA

Un aspecto que estudia la estadística inferencial es la estimación, que resulta ser el proceso de estimar el valor de un parámetro poblacional a partir de la información obtenida de una muestra. Los procedimientos estadísticos para estimar la media de la población se explicarán en este capítulo. Las técnicas estadísticas inferenciales tienen varios supuestos que deben cumplirse para realizar la estimación de la media antes de obtener conclusiones válidas. Se aplicarán métodos para verificar los supuestos como la normalidad de la distribución, la no existencia de valores atípicos, entre otros.

- Distribuciones muestrales
- El teorema del límite central
- Estimadores puntuales y de intervalo
- Prueba de una aseveración respecto de una desviación estándar o de una varianza

Distribuciones muestrales

Una de las principales preocupaciones de las estadísticas es obtener conclusiones de un conjunto de datos observados. Estos datos generalmente consistirán en una muestra de ciertos elementos de una población, y el objetivo será utilizar la muestra para sacar conclusiones sobre toda la población [52]. De acuerdo a esto, una muestra estadística es una variable aleatoria cuyo valor depende de qué elementos de la población se incluyan en la muestra aleatoria. Algunas muestras pueden representar bien a la población, mientras que otras pueden diferir mucho de la población particularmente si el tamaño de la muestra es pequeño [5]. Cuando la población de la que se extraen las muestras se distribuye normalmente con una media igual a μ y una desviación estándar igual a σ, entonces:

1. La media de x, $\mu_{\overline{X}}$, es igual a la media de la población,μ
2. La desviación estándar de x, $\sigma_{\overline{X}}$, es igual a $\frac{\sigma}{\sqrt{n}}$, asumiento que n/N<=0.5
3. La forma de la distribución de muestras de x es normal, cualquiera que sea el valor de n.

La idea es analizar hasta qué punto las medias muestrales se aproximan a la media poblacional. Se formula el siguiente ejemplo para el efecto: se tiene la población (p) de estudiantes, los resultados para la variable altura son: 156, 155, 178, 178, 173, 155, 176, 174, 170, 180, 166, 183, 172, 165, 165, 162, 170, 180, 155, 193. De esta población se toman tres muestras (m1, m2 y m3) y se calcula sus medias.

Con la función **c()** conseguimos los vectores que contienen las muestras m1, m2 y m3 de la población p (usted puede escoger valores diferentes para cada muestra). Aplicamos la función **mean()** para obtener las medias y creamos un dataframe para la presentación de resultados.

```
# ALTURA DE LOS ESTUDIANTES DE UN PARALELO
p<-c(156, 155, 178, 178, 173, 155, 176, 174, 170, 180, 166, 183, 172, 165, 16
5, 162, 170, 180, 155, 193)
# MUESTRAS TOMADAS EN EL PARALELO
m1<-c(156, 155, 178, 178, 173, 180)
m2<-c(156, 174, 170, 180, 166)
m3<-c(156, 183, 172, 165, 162, 193)
# MEDIA POBLACIONAL
Xp<-mean(p)
# MEDIAS MUESTALES
Xm1<-mean(m1)
Xm2<-mean(m2)
Xm3<-mean(m3)
# RESULTADO
medias<-data.frame(Mp=Xp, Mm1=Xm1, Mm2=Xm2, Mm3=Xm3 )
medias
##      Mp Mm1   Mm2      Mm3
## 1 170.3 170 169.2 171.8333
```

Surgen dos preguntas: cómo seleccionar correctamente los elementos de la muestra y cuál debería ser el tamaño adecuado de la muestra para obtener la precisión deseada y la probabilidad de hacer una estimación correcta.

Si la población de donde se extraen las muestras tiene características de una distribución normal con media μ y desviación estándar σ, entonces la distribución de las medias muestrales tomadas de la población en estudio, x, también se distribuirá normalmente con independecia del tamaño de la muestra. La ecuación 5.0 caracteriza la media la media muestral.

$$\mu_{\overline{x}} = \mu \qquad [5.0]$$

La dispersión de las medias muestrales se calcula con el error estándar de las medias muestrales dada por la ecuación 5.1.

$$\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}} \qquad [5.1]$$

Para obtener la figura 5.0, primero se grafica el histograma con la función **hist()** pasando el conjunto de datos p (altura de la población de estudiantes) y configurando el parámetro *probability* con TRUE para poder ajustar la curva de la función densidad. Luego con **lines()** se dibuja la curva, **lines**() toma como parámetro el resultado de la función **density**() que a su vez toma el conjunto de datos de la población p. Finalmente, con la función **abline()** trazo las líneas verticales que representan a las medias muestrales m1, m2 y m3.

```
par(mfrow=c(1,1), mar=c(4,5,2,1))
hist(p, probability = TRUE                    # Histograma para p
     , main="Histograma de frecuencias"
     , xlab = "Altura (cm)"
     , xlim = c(145, 200)
     , border = "gray")
lines(density(p)                              # Curva de la distribución
      , col="gray20"                          # de frecuencias p
      , lwd=2)
abline(v=mean(p), col="red", lwd=2)           # línea de la media p
abline(v=mean(m1), col="gray20", lwd=2)       # línea de la muestra m1
abline(v=mean(m2), col="gray55", lwd=2)       # línea de la muestra m2
abline(v=mean(m3), col="gray80", lwd=2)       # línea de la muestra m3
legend("topright"
       , legend=c("μ p","μ m1", "μ m2", "μ m3")
       , col=c("red","gray20", "gray55", "gray80")
       , lty=1:2
       , cex=0.7
       , bty = "n")
```

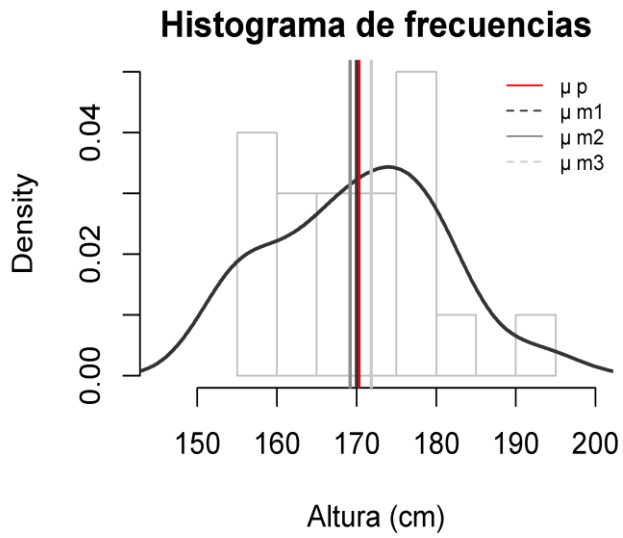


Figura 5.0 *Gráfico de la media poblacional y medias muestrales*

La figura 5.0 contiene el gráfico de las medias muestrales de m1, m2 y m3 e incluye la media poblacional para comprender mejor problema de la inferencia de la media poblacional.

Teorema del límite central

Cuando se tiene varias muestras y sus medias, la distribución muestral de las medias se apega a una distribución Normal, sobre todo cuando el tamaño de la muestra es grande. El teorema del límite central manifiesta que: Si una muestra aleatoria de tamaño n extraídas de una población con media μ y desviación estándar σ, la distribución de las muestras de las medias $\overline{X}$ se acerca a una distribución normal con media μ y desviación estándar $\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}$ a medida que aumenta el tamaño de la muestra.

Norean Sherpe[53] con acierto apunta que nos hemos deslizado suavemente entre el mundo real, en el que extraemos muestras aleatorias de datos, y un mundo de modelos matemáticos, en el que describimos cómo se comportarían las medias y las proporciones de la muestra que observamos en el mundo real. Ahora tenemos dos distribuciones con las que lidiar. La primera, que podríamos mostrar con un histograma de frecuencia (para datos cuantitativos) o con un gráfico de barras (en el caso de datos categóricos). La segunda, que modelamos con un modelo Normal basado en el Teorema del Límite Central.

El siguiente código en R grafica lo expresado por Norean Sherpe; es decir, tenemos la curva de la distribución poblacional (real de la variable altura) y la curva de la distribución normal ideal.

```
# CÁLCULO DE LA MEDIA Y DESVIACIÓN ESTÁNDAR MUESTRAL
Xm1<-mean(m1)                                # Muestra 1
Xm2<-mean(m2)                                # Muestra 2
Xm3<-mean(m3)                                # Muestra 3
mediasMuestrales<-c(Xm1, Xm2, Xm3)           # Medias muestrales
MmediasMuestrales<-mean(mediasMuestrales)    # Media de las medias muestrales
SDmediasMuestrales<-sd(mediasMuestrales)     # Desviación estándar de las
                                             # medias muestrales
data.frame(mediasMuestrales=MmediasMuestrales, SDmediasMuestrales=SDmediasMue
strales)
##   mediasMuestrales SDmediasMuestrales
## 1         170.3444           1.350034
par(mfrow=c(1,1), mar=c(4,5,2,1))
hist(p, probability = TRUE                   # Histograma para p
     , main="Media de las medias muestrales"
     , xlab = "Altura (cm)"
     , xlim = c(145, 200)
     , border = "gray")
lines(density(p)                             # Curva de la distribución
      , col="gray40"                         # de frecuencias p
      , lwd=2)
x<-p
curve(dnorm(x, mean(p), sd(p))
    , col = "gray20"
    , lty = 2
    , lwd = 2
    , add=T)
abline(v=mean(p), col="red", lwd=2)              # línea de la media p
abline(v=MmediasMuestrales, col="gray20", lwd=1) # línea de la media Muestras
abline(v=MmediasMuestrales+SDmediasMuestrales/sqrt(20), col="gray20", lwd=1,
lty=2)
abline(v=MmediasMuestrales-SDmediasMuestrales/sqrt(20), col="gray20", lwd=1,
lty=2)
legend("topleft"
       , legend=c("D. poblacional","D. Normal")
       , col=c("gray20", "gray40")
       , lty=1:2
       , cex=0.7
       , bty = "n")
```

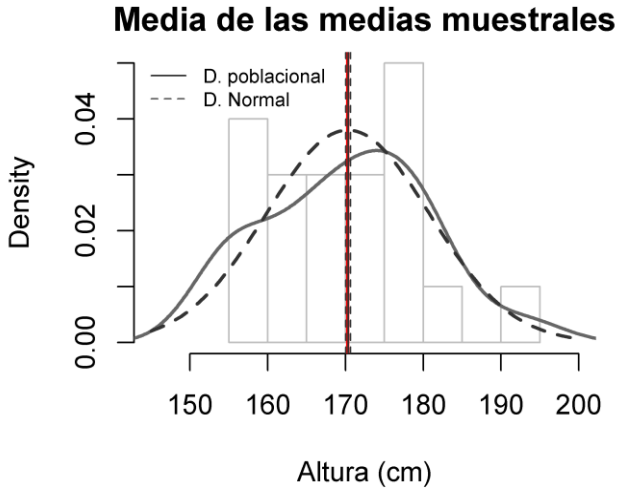


Figura 5.1 *Gráfico de la media poblacional y medias muestrales*

La figura 5.1 muestra la distribución normal ideal y real de la altura de la población de estudiantes. Observe como quedó la media poblacional está ubicada dentro del límite de $\overline{x} \pm \frac{\sigma}{\sqrt{n}}$.

Teorema de Chebyshev

El teorema de Chebyshev aplica a cualquier conjunto de mediciones y se puede usar para describir ya sea una muestra o una población. La idea que encierra el teorema de Chebyshev es que trata de construir un intervalo al medir una distancia k a cualquier lado de la media μ. El número k puede ser cualquier número mientras sea mayor o igual a 1. Entonces el teorema de Chebyshev expresa que al menos [1 - (1/k2)] del número total n de mediciones está en el intervalo construido. En resumen, cualquier conjunto de observaciones (muestra o población), la proporción de valores que se encuentran a k desviaciones estándares de la media es de por lo menos 1 - 1/k2, siendo k cualquier constante mayor que 1 [54]. La ecuación 5.2 representa matemáticamente el teorema de Chebyshev.

$$Proporción\ de\ Datos(\%) = 1 - \frac{1}{k^2} \qquad [5.2]$$

Esta regla, por lo tanto, ayuda a definir el porcentaje de datos dentro de k desviaciones estándar.

- Al menos ninguna de las mediciones está en el intervalo μ - σ a μ + σ.
- Al menos 3/4 de las mediciones están en el intervalo μ - 2σ a μ + 2σ.
- Al menos 8/9 de las mediciones están en el intervalo μ - 3σ a μ + 3σ.

Regla Empírica

En cualquier distribución de frecuencias simétrica con forma de campana, aproximadamente 68% de las observaciones se encontrarán entre más y menos una desviación estándar de la media; cerca

de 95% de las observaciones se encontrarán entre más y menos dos desviaciones estándares de la media y, de hecho todas (99.7%), estarán entre más y menos tres desviaciones estándares de la media. Ver figura 5.2

Para lograr crear el gráfico de la regla empírica iniciaremos creando la variable aleatoria x con la función **seq**() tomado como límites de -4 a 4 en pasos de 0.01; también se crea la variable normal densidad con la función **dnorm**() pasando como parámetro la variable aleatoria x y configurando la media=0 y la desviación estándar=1, que corresponden a una distribución normal estándar. Con esto se procede a graficar la curva normal estándar con la función **plot**() con los valores de la variable aleatoria x y la normal densidad. Para colorear el área de la campana de Gauss se usa la función **polygon**(). Luego se dibuja las líneas que corresponden a los intervalos de $\mu - \sigma$ a $\mu + \sigma$, $\mu - 2\sigma$ a $\mu + 2\sigma$ y $\mu - 3\sigma$ a $\mu + 3\sigma$. Como se trata de una curva normal estándar, el valor de la media es igual a $\mu=0$ y de la desviación estándar es $\sigma=1$. Estas líneas se dibujan con la función **abline()** configurando el parámetro v con los intervalos previstos.

```
# DISTRIBUCIÓN NORMAL
set.seed(3345)
variable.aleatoria.x<-seq(-4,4,0.01)
Normal.densidad<-dnorm(variable.aleatoria.x, 0, 1)
par(mfrow=c(1,1), mar=c(2,5,2,1))
plot(variable.aleatoria.x, Normal.densidad
     , col="gray20"
     , xlab=""
     , ylab="Densidad de Probabilidad"
     , type="l"
     , lwd=2
     , cex=2
     , main="Regla Empírica"
     , cex.axis=.8
     , xlim = c(-4,4))
# COLOREAR EL ÁREA DE LA CURVA NORMAL
polygon(c(-4,variable.aleatoria.x,4)
        , c(0,Normal.densidad,0)
        , col="gray95"
        , border = "gray20")
# CALCULAR LA MEDIA POBLACIONAL
media<-mean(variable.aleatoria.x)
# REGLA EMPIRICA 1SD / 68.3%
abline(v=media-1, col="gray30", lty=2)
abline(v=media+1, col="gray30", lty=2)
x<-c(media-1, media+1)
y<-c(0.3, 0.3)
lines(x,y, lwd=1, col="gray20")
text(x = 0, y = 0.32
     , "68.3% de datos dentro de 1 SD"
     , col = "gray20"
     , cex = 0.8)
# REGLA EMPIRICA 2SD / 95.5%
```

```
abline(v=media-2, col="gray30", lty=2)
abline(v=media+2, col="gray30", lty=2)
x<-c(media-2, media+2)
y<-c(0.2, 0.2)
lines(x,y, lwd=1, col="gray20")
text(x = 0, y = 0.22
     , "95.5% de datos dentro de 2 SD"
     , col = "gray20"
     , cex = 0.8)
# REGLA EMPIRICA 3SD / 99.7%
abline(v=media-3, col="gray30", lty=2)
abline(v=media+3, col="gray30", lty=2)
x<-c(media-3, media+3)
y<-c(0.1, 0.1)
lines(x,y, lwd=1, col="gray20")
text(x = 0, y = 0.12
     , "99.7% de datos dentro de 3 SD"
     , col = "gray20"
     , cex = 0.8)
```

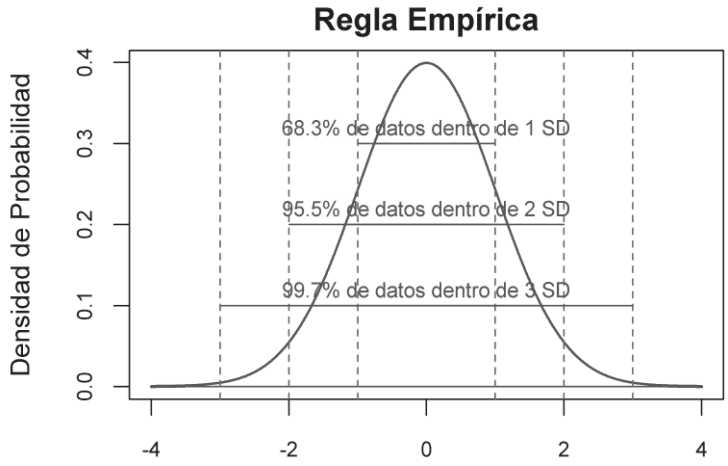


Figura 5.2 *Gráfico de la Regla Empírica*

Estos conceptos previos son necesarios para aplicarlos en el cálculo de los estimadores puntuales y de intervalo.

Según [55] la inferencia estadística se ocupa de tomar decisiones o predicciones acerca de parámetros; es decir, las medidas numéricas descriptivas que caracterizan a una población. Dos parámetros encontramos han sido ya tratados: la media poblacional μ, la desviación poblacional estándar σ pero también existe un tercer parámetro como la proporción binomial p.

Los métodos de inferencia sobre parámetros poblacionales están en una de estas dos categorías:

- Estimación: Estimar o predecir el valor del parámetro
- Prueba de hipótesis: Tomar una decisión acerca del valor de un parámetro, con base en alguna idea preconcebida acerca de cuál podría ser su valor. Estas pruebas serán revisadas en extenso en el capítulo 6.

Estimadores puntuales

Keller [56] define de manera sencilla lo que representa un estimador puntual: Un estimador puntual hace inferencias sobre una población al estimar el valor de un parámetro desconocido utilizando un único valor o punto. Al mismo tiempo, dice que un estimador de intervalo extrae inferencias sobre una población al estimar el valor de un parámetro desconocido utilizando un intervalo. Por lo tanto, una estimación puntual de un parámetro poblacional es cuando se utiliza un único valor para estimar ese parámetro, es decir, se usa un punto en concreto de la muestra para estimar el valor deseado. Un estimador debe tener ciertas propiedades que le acerquen a inferir de forma acertada el parámetro poblacional. Mencionamos las siguientes propiedades básicas de un estimador:

- **Insesgadas**: Un estimador es insesgado cuando la esperanza matemática es que este sea igual al parámetro que se desea estimar. Por tanto, la diferencia entre el parámetro a estimar y la esperanza de nuestro estimador tendría que ser 0.
- **Eficiente**: Un estimador es más eficiente o tiene la capacidad de estimar de forma precisa cuando su varianza es reducida. Por lo tanto ante 2 estimadores, siempre elegiremos el que tenga una varianza menor.
- **Consistencia**: Un estimador consistente es aquel que a medida que la medida de la muestra crece se aproxima cada vez más al valor real del parámetro. Por lo tanto, cuantos más valores entran en la muestra, el parámetro estimado será más preciso

La media poblacional (μ) puede inferirse a partir del estimador puntual de la media poblacional ($\hat{u}$) que es igual a la media muestral ($\overline{x}$). Este estimador se calcula de acuerdo a la ecuación 5.3:

$$\mu = \hat{u} = \overline{x} = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n} \qquad [5.3]$$

La desviación estándar poblacional (σ) puede inferirse a partir del estimador puntual de la desviación estándar ($\hat{S}$) que es igual a la desviación estándar de la muestra (S). Este estimador se calcula de acuerdo a la ecuación 5.4:

$$\sigma = \hat{S} = S = \sqrt{\frac{\sum_{i=1}^{i=n}(x_i + \overline{x})^2}{n}} \qquad [5.4]$$

La proporción poblacional (p) puede inferirse a partir del estimador puntual de proporción poblacional ($\hat{p}$) que es igual a la proporción de la muestra. Este estimador se calcula de acuerdo a la ecuación 5.5:

$$p = \hat{p} = \frac{x}{n} \qquad [5.5]$$

En este caso, la desviación estándar del estimador puntual de proporción se calcula con la ecuación 5.6:

$$ES(\hat{p}) = \sqrt{\frac{p(1-p)}{n}} \qquad [5.6]$$

Estimadores de intervalo

Destaca [57] que un estimador puntual es un estadístico de muestra utilizado para estimar un parámetro de población. Apunta también que la media muestral es un estimador puntual ($\hat{\mu}$) de la media poblacional μ y la proporción muestral es un estimador puntual ($\hat{p}$) de la proporción poblacional p. Debido a que no se puede esperar que un estimador puntual proporcione el valor exacto del parámetro de población, a menudo se calcula una estimación de intervalo sumando y restando un valor, llamado margen de error, a la estimación puntual. La forma general de una estimación de intervalo es la siguiente: Estimador puntual $\pm$ Margen de error. Un intervalo de confianza es un conjunto de valores que se forma a partir de una muestra de datos de forma que exista la posibilidad de que el parámetro poblacional ocurra dentro de dicho conjunto con una probabilidad específica. La probabilidad específica recibe el nombre de nivel de confianza [58]. El procedimiento de estimación planteado por [59] implica seguir estos pasos.

1. Seleccione una muestra.
2. Recopile la información requerida de los miembros de la muestra.
3. Calcule el valor de la estadística de muestra.
4. Asigne valores al parámetro de población correspondiente.

Dos conceptos importantes ligados a los intervalos de confianza y prueba de hipótesis son el nivel de confianza y el nivel de significancia: El coeficiente de confianza (cc) es el grado de certeza, expresado en porcentaje, con el que vamos a realizar la estimación de un parámetro a través de un estadístico muestral; el nivel de significacia (α) es la diferencia que existe entre la certeza y el nivel de confianza. Tome en cuenta que se debe cumplir que: $cc + \alpha = 1$

La forma general de una estimación de intervalo para una media poblacional μ y una proporción poblacional p es:

$$\hat{x} + Margen\ de\ Error$$
$$\hat{p} + Margen\ de\ Error$$

Dado que se trabajará con la distribución normal estándar, debemos "normalizar" el estimador puntal $\overline{x}$. Por lo tanto, es necesario pasar al plano z mediante la ecuación 5.7 y 5.8:

$$z = \frac{\overline{x} - \mu}{\sigma_{\overline{x}}} \qquad [5.7]$$

$$donde:$$

$$\sigma_{\overline{x}} = \frac{\sigma}{\sqrt{n}} \qquad [5.8]$$

Estimador de intervalo de la media poblacional con σ conocida

Vamos a iniciar el cálculo de los estimadores de intervalo (o intervalo de confianza, IC) mostrando el procedimiento para estimar el valor de una media poblacional desconocida cuando se conoce la desviación estándar de la población. Para esto se usa la ecuación 5.9.

$$IC(\mu) = \overline{x} \pm Z.\frac{\sigma}{\sqrt{n}} \qquad [5.9]$$

Para resolver este tipo de problemas con R, aclaremos la nomenclatura correspondiente:

1- Anotar los datos que da el problema

Tabla 5.1 Variables utilizadas para graficar el IC

Variable	Significado
xm	Media de la muestra
SDp	Desviación estándar poblacional
SDm	Desviación estándar poblacional
n	Número de observaciones
CC	Coeficiente de confianza
alfa	Nivel de significancia
ls.IC	Límite superior del intervalo de confianza
li.IC	Límite inferior del Intervalo de confianza

2- Calcular el nivel de significancia (α) alfa con: `alfa=1-CC`

3- Calcular Z con la función qnorm() y el valor del nivel de significancia que se proporcione. El argumento *lower.tail* se refiere al uso de la cola de la izquierda (-Z) o de la derecha (Z) que desea aplicar: `Z<-qnorm(alfa/2, mean = 0, sd = 1, lower.tail = FALSE)`

4- Calcular la desviación estándar: `SD<-SDp/sqrt(n)`

5- Por último, calcular el intervalo de confianza IC (LIC, LSC) con: `ls.IC=round(xm+ZSD, 2)` y `li.IC=round(xm-ZSD,2)`

Revisemos el siguiente ejemplo para examinar cómo se cálcula intervalos de confianza con R: La empresa Corona va a incluir los gastos por consumo de internet. Estos deben ser estimados para incluirlos en el plan operativo anual, sabiendo que la desviación estándar poblacional de 87 USD. Se tiene que con datos de 97 planillas la media de 3540 USD por concepto de gasto de internet. Hallar el intervalo de confianza (IC) si se establece un coeficiente de confianza del 95%.

```
# DATOS
Xm=3540
SDp=87
n=97
CC=0.95
# CALCULAR EL NIVEL DE SIGNIFICANCIA
alfa=1-CC
# CALCULAR Z
Z<-qnorm(alfa/2, mean = 0, sd = 1, lower.tail = FALSE)
# CALCULAR SD
SD<-SDp/sqrt(n)
# CALCULAR EL LÍMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Z*SD, 2)
# CALCULAR EL LÍMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Z*SD,2)
# PRENSENTACION DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
IC
##        LI      LS
## 1 3522.69 3557.31
```

Gráfico del Intervalo de Confianza

Para graficar el intervalo de confianza, hacemos uso de Z que fue calculado con la función **qnorm()**. La curva de la campana de Guaus se la logra mediante las funciones **seq()**, **dnorm()** y **plot()**. Con la función *seq()* creamos la variable aleatoria x con valores que van de -3 a 3 en pasos de 0.1 mientras que con *dnorm()* se generan los valores de y para x. La función *plot()* se encarga de realizar la gráfica para (x,y). El área pintada de la campana de Gauss se la obtiene con la función **polygon().** Se crea un polígono con el conjunto de puntos: x=(-z,x,z), y=(0,y,0). La línea media de la campana de Gauss se traza con abline() con su parámetro V en 0. Los valores de los límites del intervalo (li.IC, ls.IC) se los ubica utilizando la función **axis()**.

```
# GRÁFICO DE LA CURVA NORMAL
x <- seq(-3, 3, by=0.001)
y <- dnorm(x)
par(mar=c(4,3,1,1))
plot(x, y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , type = "l"
     , main = "Distribución Normal"
     , xlab= "IC(z)"
     , ylab=""
```

```
     , frame=F)
grid()
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-Z, Z, by=0.001)
y <- dnorm(x)
# GRÁFICO DE LA REGIÓN SOMBREADA
polygon(c(-Z,x,Z), c(0,y,0), col="gray95")
# LÍNEA CENTRAL
abline(v=0,  lty=2, col="gray40")
# VALORES DEL INTERVALO
axis(1, at = c(-1*Z, 0, 1*Z)
     , font = 8
     , cex=0.8
     , labels = c(li.IC,Xm, ls.IC))
```

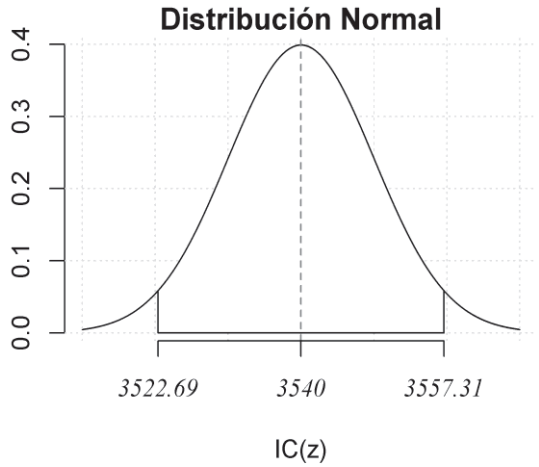


Figura 5.3 *Gráfico del intervalo de confianza con s conocida*

La figura 5.3 nos muestra la distribución normal acompañado del intervalo de confianza requerido. Observe en la mitad de la campana de Gauss se ha ubicado la media muestral. A partir de la media muestral se construye el intervalo según la ecuación 5.9

Intervalo de confianza para μ, si σ es desconocida y n>=30

En la práctica, el valor de la desviación estándar poblacional σ no se conoce, por lo que debe estimarse utilizando la desviación estándar de la muestra S. El intervalo de confianza (IC) se lo puede construir utilizando la ecuación 5.10

$$IC(\mu) = \overline{x} \pm Z * S_{\overline{x}} \qquad [5.10]$$

El error estándar se calcula a partir de la desviación estándar de la muestra:

$$S_{\overline{x}} = \frac{S}{\sqrt{n}} \qquad [5.11]$$

El siguiente ejemplo clarifica como debe utilizarse las ecuaciones 5.10 y 5.11 para hallar intervalos de confianza mediante R.

Benito Ventura registra en el SRI sus gastos personales. El desea estimar la cantidad promedio que le representará ese rubro. De las 50 facturas que seleccionó en su muestra, la cantidad promedio de gastos era de USD 541.23, desconoce la desviación estándar de todas sus facturas. Calcula la desviación estándar de la muestra recogida y resulta ser de 113.70 USD. Se desea asegurar el cálculo con un coeficiente de confianza del 99%.

```
# DATOS
Xm=541.23
n=50
SDm=113.7
CC=99/100
# CALCULAR EL NIVEL DE SIGNIFICANCIA ALFA
alfa=1-CC
alfaMedio=alfa/2
# CALCULAR Z
Z<-qnorm(alfaMedio, mean = 0, sd = 1, lower.tail = FALSE)
# CALCULAR EL LIMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Z*SD/sqrt(n), 2)
# CALCULAR EL LIMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Z*SD/sqrt(n),2)
# PRESENTACIÓN DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
IC
##       LI     LS
## 1 538.01 544.45
```

La figura 5.4 presenta el gráfico del intervalo de confianza para la declaración de gastos personales en el SRI.

```
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-3, 3, by=0.1)
y <- dnorm(x)
par(mar=c(4,3,1,1))
plot(x, y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , type = "l"
     , main = "Distribución Normal"
     , xlab= "IC(z)"
     , ylab=""
     , frame=F)
grid()
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-Z, Z, length= 200)
y <- dnorm(x)
# GRÁFICO DE LA REGIÓN SOMBREADA
polygon(c(-Z,x,Z), c(0,y,0), col="gray95")
# LÍNEA CENTRAL
abline(v=0,  lty=2, col="gray40")
# VALORES DEL INTERVALO
axis(1, at = c(-1*Z, 0, 1*Z)
     , font = 8
```

```
     , cex=0.8
     , labels = c(li.IC,Xm, ls.IC))
```

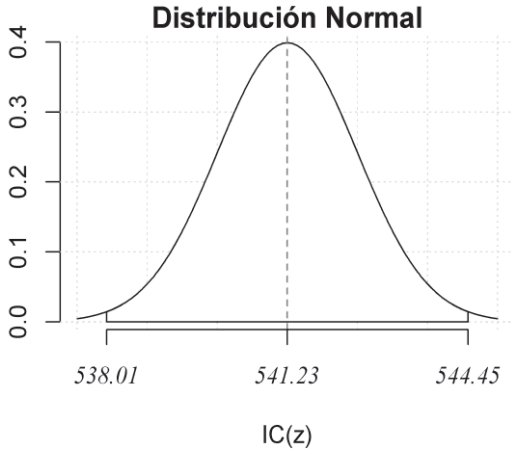


Figura 5.4 *Gráfico del intervalo de confianza con s desconocida*

Intervalo de confianza para μ, si n<30

Cuando el tamaño de la muestra es pequeño y se usa la desviación estándar s, los valores críticos suelen ser mayores que los valores de $z_{\frac{\alpha}{2}}$ que se usan en intervalos de confianza para mantener el intervalo en un nivel dado. Estos valores se toman de la distribución t de Student.

La distribución t de Student se asemeja a la distribución normal estándar porque tiene forma de campana, es simétrica respecto a la media, la curva nunca llega a tocar el eje horizontal y la media, la mediana y la moda son iguales a 0. Difiere de la distribución normal estándar porque la varianza es mayor que 1, introduce el concepto de grados de libertad que relaciona a la forma de la distribución con el tamaño de la muestra. Si se aumenta el tamaño de la muestra, la distribución t más aproxima a la distribución normal estándar. Las ecuaciones 5.12 y 5.13 proponen como establecer el intervalo de confianza para muestras pequeñas.

$$IC(\mu) = \overline{x} \pm t * S_{\overline{x}} \qquad [5.12]$$

$$S_{\overline{x}} = \frac{S}{\sqrt{n}} \qquad [5.13]$$

Revisemos el siguiente ejercicio: La fiscalía inculpa a una empresa de alterar los comprobantes de un contrato donde se especifica que un trabajo específico debiera promediar 1.150 USD. Se llaman a 11 ingenieros para testificar sobre este ítem del contrato. Las facturas presentadas tienen una media de 1.275 USD y una desviación estándar de 235 USD. Que ocurre si la fiscalía establece un intervalo de confianza con un 95% de coeficiente de confianza ¿La empresa saldría libre de la acusación? Veamos el código en R que resuelve este problema.

```
# DATOS
n=11
```

```
Xm=1150
SD=2.35
CC=95/100
GL=n-1
# CALCULAR EL NIVEL DE SIGNIFICANCIA ALFA
alfa=1-CC
alfaMedio=alfa/2
# CALCULAR Z
Tstudent<-qt(1-alfaMedio, GL)
# CALCULAR EL LIMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Tstudent*SD/sqrt(n), 2)
# CALCULAR EL LIMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Tstudent*SD/sqrt(n),2)
# PRESENTACIÓN DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
IC
# SALIDA POR CONSOLA
##        LI      LS
## 1 1144.07 1155.93
```

Con los resultados obtenidos para el límite inferior (1144.07) y el límite superior (1155.93) se procede a realizar el gráfico del intervalo de confianza.

```
x <- seq(-4,4, by=0.001)
y <- dt(x, 1)
par(mar=c(4,3,1,1))
plot(x,y
    , xaxt = "n"                          # Sin etiquetas el eje z
    , main="Distribución TStudent"
    , type = "l"
    , lty = 1
    , xlab = "IC(t)"
    , ylab = "f(t)"
    , ylim = c(0, 0.4)
    , frame=F)
# CAMPANA DE GAUUS SOMBREADA
ICt <- seq(-Tstudent, Tstudent, by = 0.001)
ICd <- dt(ICt, 1)
polygon(c(-Tstudent,ICt,Tstudent), c(0,ICd,0), col="gray95")
# LÍNEA CENTRAL
abline(v=0,  lty=2, col="gray20")
# VALORES DEL INTERVALO
axis(1
     , at = c(-1*Tstudent, 0, 1*Tstudent)
     , font = 8
     , col.axis = "gray20"
     , labels = c(li.IC, Xm, ls.IC))
```

La figura 5.5 muestra el intervalo de confianza utilizando la distribución TStudent que se usa cuando el número de muestras es pequeño (n<30). Nótese que la forma de la distribución TStudent es similar a la distribución normal. Sin embargo, la altura de la campana difiere de los anteriores ejemplos.

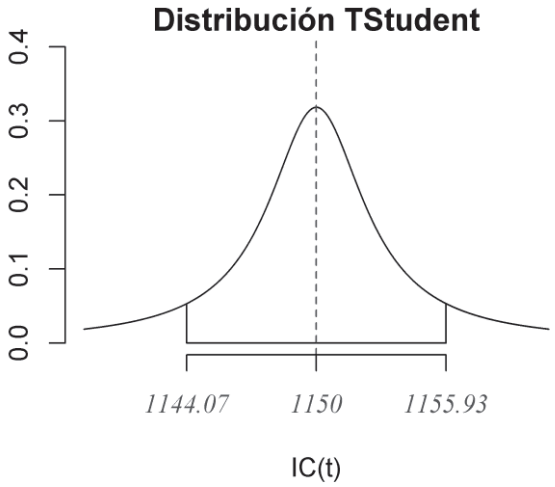


Figura 5.5 *Gráfico del intervalo de confianza para n<30*

Intervalo de confianza para la proporción poblacional

Para intervalos de confianza de una proporción el objetivo es estimar y contrastar la proporción p de elementos de una población que presentan una determinada característica. Eso significa que existe una variable dicotómica, contenida en una muestra de tamaño n, que toma el valor 1 si existe la característica de interés y 0 si no existe. Entonces, la proporción poblacional p no es otra cosa que la media poblacional de dicha variable, estimándose con la correspondiente proporción muestral o media muestral.

El intervalo de confianza para una proporción se determina con las ecuaciones 5.14 y 5.15.

$$IC(\pi) = p \pm Z * S_p \qquad [5.14]$$

$$S_p = \sqrt{\frac{p(1-p)}{n}} \qquad [5.15]$$

Demos un vistazo al siguiente ejemplo: El gerente de la radio debe determinar qué porcentaje de casas que tiene más de un radio en la ciudad de la emisora. Una muestra aleatoria de 490 casas revela que 250 tienen dos o más radios. ¿Determinar el intervalo de confianza, con un coeficiente de confianza del 90%, para estimar la proporción de todas las casas que tienen dos o más radios?

```
# TAMAÑO DE LA MUESTRA
n<-490
# TAMAÑO DE LA MUESTRA QUE CUMPLE LA CONDICIÓN
nCumple<-250
# PROPORCIÓN
p<-nCumple/n
# CÁLCULO DEL ERROR ESTANDAR
sp<-sqrt(p*(1-p)/n)
# CONFIANZA
CC=90/100
```

```
# CALCULAR EL NIVEL DE SIGNIFICANCIA ALFA
alfa=1-CC
alfaMedio=alfa/2
# CALCULAR Z
Z<-qnorm(alfaMedio, mean = 0, sd = 1, lower.tail = FALSE)
# CALCULAR EL INTERVALO DE CONFIANZA
ICs<-p+Z*sp
ICi<-p-Z*sp
# PRESENTACIÓN DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=ICi, LS=ICs)
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-3, 3, by=0.1)
y <- dnorm(x)
par(mar=c(4,3,1,1))
plot(x, y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , type = "l"
     , main = "Distribución Normal para Proporciones"
     , xlab= "IC(p)"
     , ylab=""
     , frame=F)
grid()
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-Z, Z, length= 200)
y <- dnorm(x)
# GRÁFICO DE LA REGIÓN SOMBREADA
polygon(c(-Z,x,Z), c(0,y,0), col="gray95")
# LÍNEA CENTRAL
abline(v=0,  lty=2, col="gray40")
# VALORES DEL INTERVALO
axis(1, at = c(-1*Z, 1*Z)
     , font = 8
     , cex=0.8
     , labels = c(IC$LI, IC$LS))
```

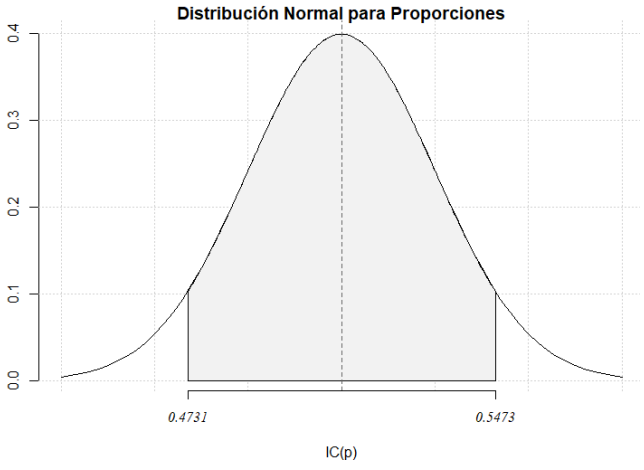


Figura 5.6 *Intervalo de confiaza para proporciones*

Con la figura 5.6 se aprecia que es script de R difiere muy poco de los casos anteriores. Justamente uno de los beneficios de usar R es que a partir de un script bien elaborado se puede usarlo repetidamente con varios conjuntos de datos obtenido resultados de la misma calidad.

AUTOEVALUACIÓN

Autoevaluación 5-1

Para una muestra, de tamaño 71, de estudiantes universitarios se obtuvo un promedio en ciencias químicas de 8. Por estudios anteriores se sabe que la desviación estándar del promedio 5.2, con el 90% del coeficiente de confianza.

a) Hallar el intervalo de confianza
b) Realizar la gráfica con palegreen1 como color de relleno para mostrar el intervalo

Autoevaluación 5-2

Se ha obtenido una muestra de 10 bailarines de un estudio de danza para conocer sus presentaciones anuales, la media es de 4 presentaciones con una desviación estándar de la muestra de 2 y con un coeficiente de confianza del 80%

a) Hallar el intervalo de confianza
b) Realizar la gráfica con lightblue2 como color de relleno para mostrar el intervalo.

Autoevaluación 5-3

A partir de una muestra de 22 personas seleccionadas al azar del barrio residencial Bracamonte, se intenta inferir la media salarial de Bracamonte. La muestra tiene una media salarial de 1200 euros y una desviación estándar de 321 euros. El coeficiente de confianza para la estimación que se solicita es de 91.

a) Hallar el intervalo de confianza.
b) Realizar la gráfica con khaki3 como color de relleno para mostrar el intervalo

EJERCICIOS DEL CAPÍTULO

1. Realice un histograma de frecuencia que represente la media poblacional (color chocolate3), la media muestral1(color darkolivegreen3) y para la media muestral2 (goldenrod2) usando los siguientes datos: 12, 45, 50, 60, 11, 7, 3, 9, 42, 28, 19, 17, 32, 39, 50, 18, 43, 66, 71, 20.
2. Realice la gráfica de la regla empírica para el 68% pinte de color olivedrab4, para el 95,5% de color orange y para el 99,7% de color tomate, genere números aleatorios desde el -3 al 3.
3. Se ha obtenido una muestra de 35 productos de una fábrica de electrodomésticos para estimar el rendimiento de dichos productos. Se sabe por otros estudios anteriores que la desviación estándar es de 2.01, La media de la muestra es de 4.9 y el coeficiente de confianza del 90%.
   a) Hallar el intervalo de confianza.
   b) Realizar la gráfica con wheat1 como color de relleno para mostrar el intervalo

4. El número de competencias de carreras mensuales sigue una distribución normal con una desviación estándar de 6 competencias. Con una muestra de 576 competidores, la media ha resultado ser de 12 competencias, con un coeficiente de confianza del 95%.
   a) Hallar el intervalo de confianza.
   b) Realizar la gráfica con caqui como color de relleno para mostrar el intervalo
5. El número de MB/s descargados por un grupo de clientes de una compañía de internet se aproxima a una distribución normal con media muestral 2,5MB/s y una desviación estándar igual a 1,8MB/s. Se toma una muestra aleatoria de tamaño 24 con un coeficiente de confianza del 98%.
   a) Hallar el intervalo de confianza
   b) Realizar la gráfica con rosybrown1 como color de relleno para mostrar el intervalo

Respuestas de las autoevaluaciones

5-1 a)

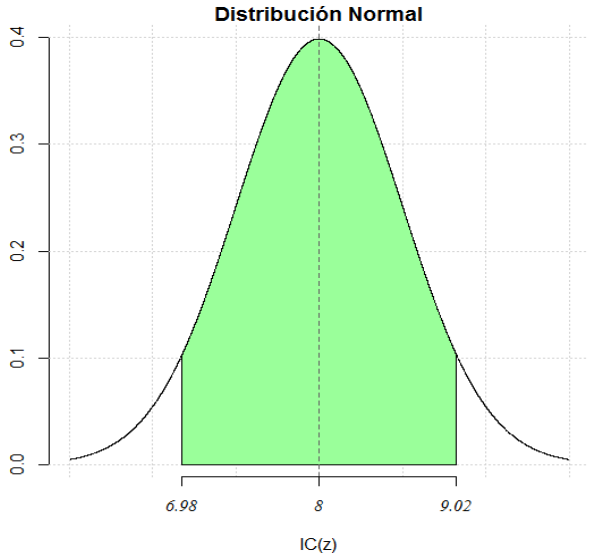
```
# DATOS
Xm=8
SDp=5.2
n=71
CC=0.90
# CALCULAR EL NIVEL DE SIGNIFICANCIA
alfa=1-CC
# CALCULAR Z
Z<-qnorm(alfa/2, mean = 0, sd = 1, lower.tail = FALSE)
# CALCULAR SD
SD<-SDp/sqrt(n)
# CALCULAR EL LÍMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Z*SD, 2)
# CALCULAR EL LÍMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Z*SD,2)
# PRENSENTACION DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
IC
RESULTADO
         LI        LS
      1 6.98     9.02
```

b)

```
# CALCULAR EL LIMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Tstudent*SD/sqrt(n), 2)
# CALCULAR EL LIMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Tstudent*SD/sqrt(n),2)
# PRESENTACIÓN DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
IC
RESULTADO
         LI      LS
      1 3.13   4.87

# GRÁFICO DE LA CURVA NORMAL
x <- seq(-3, 3, by=0.001)
```

```
y <- dnorm(x)
par(mar=c(4,3,1,1))
      plot(x, y
           , xaxt = "n" # SIN ETIQUETAS EL EJE Z
           , type = "l"
           , main = "Distribución Normal"
           , xlab= "IC(z)"
           , ylab=""
           , frame=F)
grid()
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-Z, Z, by=0.001)
y <- dnorm(x)
# GRÁFICO DE LA REGIÓN SOMBREADA
polygon(c(-Z,x,Z), c(0,y,0), col="palegreen1")
# LÍNEA CENTRAL
abline(v=0,  lty=2, col="gray40")
# VALORES DEL INTERVALO
axis(1, at = c(-1*Z, 0, 1*Z)
           , font = 8
           , cex=0.8
           , labels = c(li.IC,Xm, ls.IC))
```



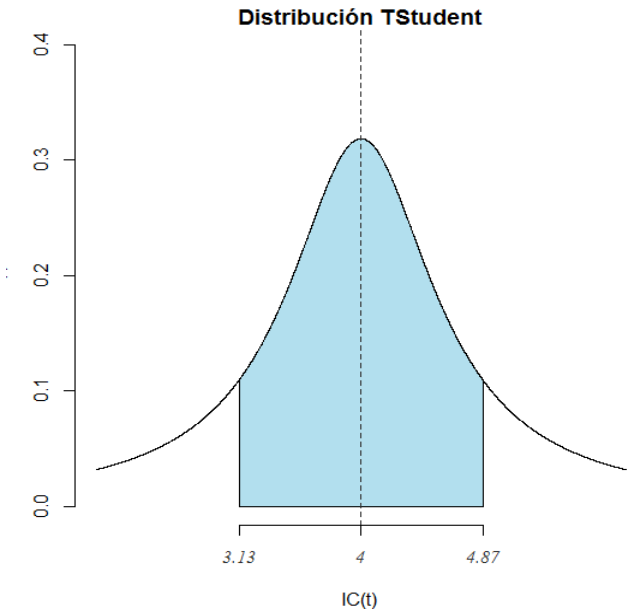
5-2 **a)**

```
# DATOS
n=10Xm=4
SD=2
CC=80/100L=n-1
# CALCULAR EL NIVEL DE SIGNIFICANCIA ALFA
alfa=1-CC
alfaMedio=alfa/2
# CALCULAR Z
Tstudent<-qt(1-alfaMedio, GL)
```

b)

```
x <- seq(-3,3, by=0.001)
y <- dt(x, 1)
par(mar=c(4,3,1,1))
plot(x,y
```

```
        , xaxt = "n" # SIN ETIQUETAS EL EJE Z
        , main="Distribución TStudent"
        , type = "l"
        , lty = 1
        , xlab = "IC(t)"
        , ylab = "f(t)"
        , ylim = c(0, 0.4)
        , frame=F)
# CAMPANA DE GAUUS SOMBREADA
ICt <- seq(-Tstudent, Tstudent, by = 0.001)
ICd <- dt(ICt, 1)
polygon(c(-Tstudent,ICt,Tstudent), c(0,ICd,0), col=" lightblue2")
# LÍNEA CENTRAL
abline(v=0,  lty=2, col="gray20")
# VALORES DEL INTERVALO
axis(1
           , at = c(-1*Tstudent, 0, 1*Tstudent)
           , font = 8
           ,  col.axis = "gray20"
           , labels = c(li.IC, Xm, ls.IC))
```



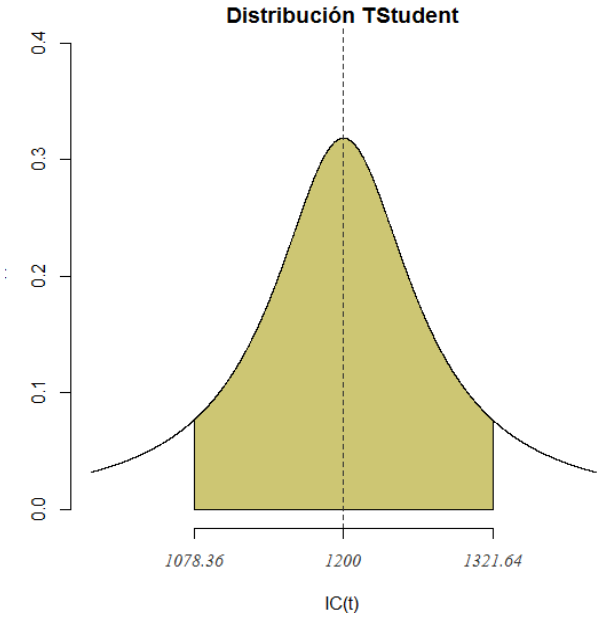
5-3 a)

```
# DATOS
n=22
Xm=1200
SD=321
CC=91/100
GL=n-1
# CALCULAR EL NIVEL DE SIGNIFICANCIA ALFA
alfa=1-CC
alfaMedio=alfa/2
#CALCULAR Z
Tstudent<-qt(1-alfaMedio, GL)
# CALCULAR EL LIMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Tstudent*SD/sqrt(n), 2)
# CALCULAR EL LIMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Tstudent*SD/sqrt(n),2)
```

```
# PRESENTACIÓN DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
IC
RESULTADO
          LI             LS
      1 1078.36    1321.64
```

b)

```
x <- seq(-3,3, by=0.001)
y <- dt(x, 1)
par(mar=c(4,3,1,1))
plot(x,y
       , xaxt = "n" # SIN ETIQUETAS EL EJE Z
       , main="Distribución TStudent"
       , type = "l"
       , lty = 1
       , xlab = "IC(t)"
       , ylab = "f(t)"
       , ylim = c(0, 0.4)
       , frame=F)
# CAMPANA DE GAUUS SOMBREADA
ICt <- seq(-Tstudent, Tstudent, by = 0.001)
ICd <- dt(ICt, 1)
polygon(c(-Tstudent,ICt,Tstudent), c(0,ICd,0), col=" khaki3")
# LÍNEA CENTRAL
abline(v=0,  lty=2, col="gray20")
# VALORES DEL INTERVALO
axis(1
     , at = c(-1*Tstudent, 0, 1*Tstudent)
     , font = 8
     , col.axis = "gray20"
     , labels = c(li.IC, Xm, ls.IC))
```



6 PRUEBA DE HIPÓTESIS

Una hipótesis estadística es una suposición que se realiza sobre los datos de la población recolectada en cualquier experimento. La hipótesis no es necesariamente un enunciado válido; de hecho, la prueba de hipótesis trata de obtener evidencia suficiente para no rechazarlo. La forma óptima de realizar una prueba de hipótesis es considerar la población, pero por diversas razones esto no suele ser posible. En su lugar se usan muestras aleatorias representativas de la población. En este capítulo se revisará los siguientes ítems:

- Fundamentos de la prueba de hipótesis
- Prueba de una aseveración respecto de una proporción
- Prueba de una aseveración respecto de una media con sd conocida
- Prueba de una aseveración respecto de una media con sd desconocida
- Prueba de una aseveración respecto de una desviación estándar o de una varianza

El objetivo principal de este capítulo visualizar los gráficos para pruebas de hipótesis de una y dos colas con datos que se sigan la forma de la Distribución Normal, TStudent y Proporciones.

Fundamentos de la prueba de hipótesis

La prueba de hipótesis es un hecho que prueba una suposición con respecto a un parámetro de población. Se utiliza para evaluar la plausibilidad de una hipótesis se mide y examina una muestra aleatoria de la población que se analiza. Existen dos hipótesis diferentes: la hipótesis nula y la hipótesis alternativa. La hipótesis nula suele ser una hipótesis de igualdad entre parámetros poblacionales. La hipótesis alternativa es efectivamente lo opuesto a una hipótesis nula. Por lo tanto, son mutuamente excluyentes y solo una no será rechazada. En el análisis estadístico se establece una afirmación, una hipótesis, se recogen datos que posteriormente se utilizan para probar la aserción. Entonces, una hipótesis estadística es ** Afirmación relativa a un parámetro de la población sujeta a verificación**[60]. Para realizar la prueba de hipótesis vamos a definir el proceso de cuatro pasos para rechazar o no rechazar la hipótesis propuesta:

1. Proponer la hipótesis: Establecer la hipótesis nula (Ho) y alternativa (Ha).
2. Formular un plan de análisis: Planificar cómo haremos el análisis.
3. Analizar datos de muestra: Cálculo e interpretación del estadístico de prueba.
4. Interpretar resultados: Aplicación de la regla de decisión.

Proponer la hipótesis

La prueba de hipótesis pretende probar la validez de una hipótesis propuesta sobre la población a través de una muestra. Consta de dos partes: La hipótesis nula, y; la hipótesis alternativa.

Hipótesis nula: Afirmación que relaciona atributos para el juicio acerca de la suposición propuesta, se denota con Ho. Siempre esta ligada a un signo de igualdad.

Hipótesis alternativa: Se consideraría válida si la hipótesis nula resulta rechazada, se denota con H1 o Ha. La hipótesis alternativa (Ha) se contrapone a Ho y se valida cuando se rechaza Ho.

Es posible plantear la prueba de hipótesis de dos colas donde el valor crítico estará entre $-Z_{\alpha/2}$ y $+Z_{\alpha/2}$. Ecuación 6.0:

$$Hipótesis = \begin{cases} H_0: & \mu = \mu_0 \\ H_1: & \mu \neq \mu_0 \end{cases} \qquad [6.0]$$

También se conciben pruebas de hipótesis de una cola donde el valor crítico se localiza en $+Z_{\alpha}$, en cuyo caso se utiliza la ecuación 6.1:

$$Hipótesis = \begin{cases} H_0: & \mu \geq \mu_0 \\ H_1: & \mu < \mu_0 \end{cases} \qquad [6.1]$$

Otro caso de la prueba de hipótesis de una cola es cuando el valor crítico está en $-Z_{\alpha}$ para lo cual se utiliza la ecuación 6.2:

$$Hipótesis = \begin{cases} H_0: & \mu \leq \mu_0 \\ H_1: & \mu > \mu_0 \end{cases} \qquad [6.2]$$

Cabe indicar que una vez definido el planteamiento de hipótesis, se establece el coeficiente de confianza (CC) y el nivel de significancia(α), donde debe cumplirse con la ecuación 6.3:

$$CC + \alpha = 1 \qquad [6.3]$$

El nivel de significancia ($\alpha/2$) define la zona de no rechazo de Ho.

Para crear los gráficos de prueba de hipótesis con R se deben aplicar conceptos previamente ilustrados. Lo primero que vamos hacer es que basados en la teoría del límite central, *la distribución normal estándar* irá de (-3,3) por esa razón tenemos que `z=seq(-3,3, by=0.1 )` lo que significa que se genera z desde -3 a 3 en pasos de 0.1. Obtenido los valores de z aplicamos la función **dnorm**(z) para obtener los valores de y de cada z, con esto podemos dibujar la distribución normal estándar.

El valor crítico (vc) y el valor de z de la muestra (zm) se calculan mediante el uso de la función **qnorm()** y la fórmula 6.4, respectivamente.

Suponga que el valor crítico es vc= 2 y zm no será tomado en cuenta. El valor crítico vc se marca con un punto color negro y zm con un punto color gris usando la función **point**(vc, 0, …) o **point**(zm, 0, …). El valor del eje x puede ser vc o zm mientras que para el eje y es cero. El valor de vc se escribe en el eje x usando la función **axis**(), el parámetro *at* indica en qué posición (*at*=c(-vc, 0, vc)) del eje x se ubican los *labels* (*labels*=c("VC=-α/2","0", "VC=α/2")).

Por último, tenemos que sombrear la zona *de rechazo* de la prueba de hipótesis. Para ello se usa una variable x que define el o los segmento (s) del eje x que estará (n) sombreado (s). Cuando se sombrea el eje negativo de z, los valores vienen desde -3 hasta el valor crítico (x=**seq**(-3, vc, *by*=01)) mientras que si se trata del eje positivo de z vamos desde el valor crítico hasta 3 (x=**seq**(-3, vc, *by*=01)). La densidad para cada x se lo calcula con y=**dnorm**(x).

Para ubicar el polígono que irá sombreado se usa la función **polygon**(c(-3,x,-abs(vc)), c(0,y,0), col="grey95"). Recuerde que para el eje negativo de z iniciamos el polígono con el punto (-3,0), avanzamos con (x,y) y terminamos el polígono con (-3,-vc), para el eje positivo iniciamos el polígono con el punto (vc,0), avanzamos con (x,y) y terminamos el polígono con (3,0). De esta manera configuramos la función **polygon()**. La zona de no rechazo siempre la escribiremos dentro de la campana del gráfico con la función **text()**.

```
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.1)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
vc=2
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n"       # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="grey95")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.1)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
```

```
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c("VC=-α/2","0", "VC=α/2"))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
points(vc,0, cex=1, pch=19, col="black")
```

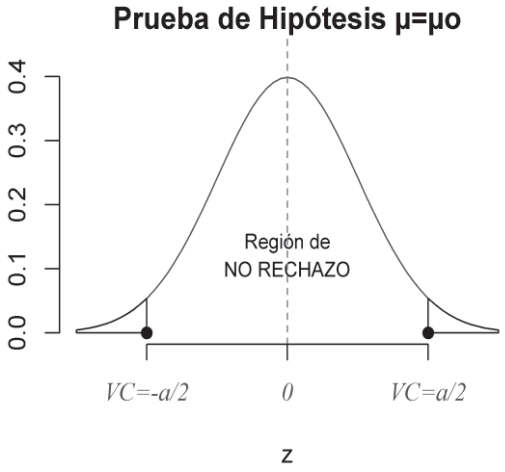


Figura 6.0 *Pruebas de hipótesis de dos colas*

La figura 6.0 presenta el gráfico para una prueba de hipótesis para la media de dos colas (μ=μo). Sin embargo, el código puede adaptarse perfectamente para pruebas de hipótesis de una cola por izquierda (μ>=μo) y por derecha (μ<=μo).

Tipos de errores

- Error tipo I: El error tipo I se produce cuando el investigador rechaza una hipótesis nula cuando es verdadera. El término nivel de significancia (α) se usa para expresar la probabilidad de error de Tipo I mientras se prueba la hipótesis.
- Error tipo II: Aceptar una hipótesis nula falsa H0 se conoce como error tipo II. El término potencia de la prueba (β) se usa para expresar la probabilidad de error de Tipo II mientras se prueba la hipótesis.

En la toma decisiones podría aparecer uno de los dos tipos de error al intentar validar la prueba de hipótesis. Por lo tanto, es necesario tener la suficiente evidencia estadística para no incurrir en ellos. Los errores de tipo I y II se resumen en la tabla 6.1

Tabla 6.1 *Tipos de errores en la prueba de hipótesis*

Toma de decisión	Ho es verdadera	Ha es falsa
No rechazar Ho	Decisión correcta	Error tipo II
Rechazar Ho	Error tipo 1	Decisión correcta

Pruebas de hipótesis para muestras grandes (n>30)

Pruebas de hipótesis para la media con σ conocida

Para hacer una prueba de hipótesis con la desviación estándar poblacional conocida definimos el estadístico de prueba que se va a usar, en este caso z. Cabe destacar que en la práctica este caso es poco frecuente que ocurra. La ecuación 6.4 define calcular z para la muestra.

$$Z = \frac{\overline{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \qquad [6.4]$$

En donde:

$\overline{x}$= Media de la muestra

μ= Media poblacional

σ= Desviación estándar poblacional

n= Tamaño de la muestra

Pruebas de hipótesis de dos colas para media (μ=μo) con σ conocida

Un fabricante de mangueras para tractores especifica que las mangueras pueden manejar 7100 kg/cm^2 en pruebas de presión. La desviación estándar calculada de la resistencia de sus mangueras es 430 kg/cm^2. El fabricante selecciona una muestra de 80 mangueras para realizar pruebas de resistencia a la presión y encuentra que la capacidad de soportar la presión de la muestra es 7020 kg/cm^2 . Si el fabricante de mangueras utiliza un nivel de significancia α=0.05 en la prueba, ¿Las mangueras soportaran la presión requerida?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu = 7100 \\ H_1: & \mu \neq 7100 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=7100

Desviación estándar poblacional=430

Media de la muestra=7020

Muestra=80

Coeficiente de Confianza=0.95

Nivel de confianza=0.05

Dos colas, α/2=0.025

Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{\sigma}$$
$$\sigma = \frac{\sigma_{\overline{x}}}{\sqrt{n}}$$

Para el script de R, recordemos que la prueba de hipótesis depende del valor de z que se obtiene de la muestra (zm) y el valor crítico (vc) que se desglosa del nivel de significancia que es calculado mediante CC+α=1 dado un coeficiente de confianza (CC). Se implementa el cálculo de z de la muestra Zm=((xm-xp)/(SDp/**sqrt**(n))) y del valor crítico mediante vc=**round**(**qnorm**(*alfa*/2,0,1) , *digits*=2). Observe en la figura 6.1 en que sector se encuentra el valor de zm para rechazar o no rechazar Ho.

```
# DATOS
xp<-7100        #Media poblacional
SDp<-430        #Desviación estándar poblacional
xm<-7020        #Media muestral
n=80            #Muestra
alfa=0.05       #Nivel de significancia
# GENERAR LOS VALORES DE Z
z=seq(-3,3,length=200)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm<-((xm-xp)/(SDp/sqrt(n)))
vc=round(qnorm(alfa/2,0,1) , digits=2)
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mfrow=c(1,1), mar=c(4,3,2,1))
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ=μo"
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE NO RECHAZO
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="grey95")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
```

```
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0", paste("VC=", as.charac
ter(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
points(vc,0, cex=1, pch=19, col="black")
# PUNTOS DEL VALOR DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm, 0.03, paste("Z=", as.character(round(Zm,2))) , cex=0.7, col="gray20"
)
```

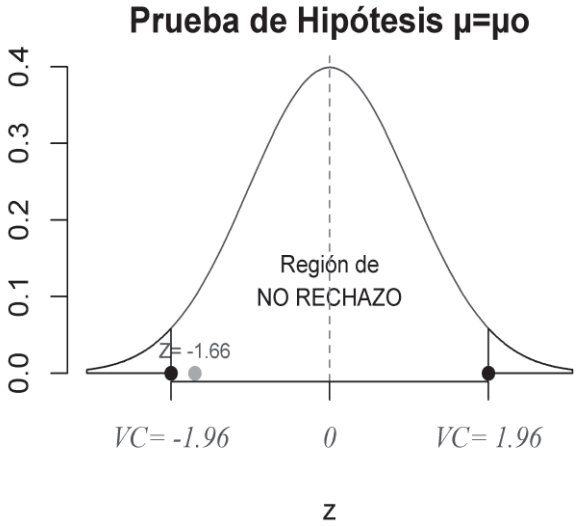


Figura 6.1 *Prueba de hipótesis de dos colas*

Interpretar resultados

La figura 6.1 expone que el estadístico z de la muestra (zm) esta en la región de no rechazo. Esto significa que la evidencia estadística no es suficiente para rechazar Ho pero no significa que Ho sea verdadera.

Prueba de hipótesis de una cola para la media (μ>=μo) con σ conocida

Para una prueba de una cola para una media μ>=μo, suponga que una clínica utiliza dosis del antiviral Veklury para curar el coronavirus. La dosis normal que se suministra es a partir de 95 cc. Las dosis reducidas interrumpen el tratamiento. La clínica sabe por experiencia que la desviación

estándar de las dosis es 2 cc. La clínica establece para un lote grande una muestra de 45 dosis, encontrándose que la media de estas dosis es 94.75 cc. Para un nivel de significancia de 0.10 ¿Las dosis son demasiado pequeñas?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \geq 95 \\ H_1: & \mu < 95 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=95

Desviación estándar poblacional=2

Media de la muestra=94.75

Muestra=45

Coeficiente de Confianza=0.90

Nivel de confianza=0.1

Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{\sigma}$$

$$\sigma = \frac{\sigma_{\overline{x}}}{\sqrt{n}}$$

El cálculo de z de la muestra se usa Zm=((xm-xp)/(SDp/**sqrt**(n))) y para el valor crítico vc=**round**(**qnorm**(*alfa*,0,1) , *digits*=2).

```
# DATOS
xp<-95       #Media poblacional
SDp<-2       #Desviación estándar poblacional
xm<-94.75    #Media muestral
n=45         #Tamaño de la Muestra
alfa=0.1     #Nivel de significancia

# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm<-((xm-xp)/(SDp/sqrt(n)))            #Z de la muestra
vc<-round(qnorm(alfa,0,1) , digits=2) #Z Teórica
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
```

```
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ>=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTOS DEL Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm, 0.03, paste("Z=", as.character(round(Zm,2))) , cex=0.7, col="gray20"
)
```

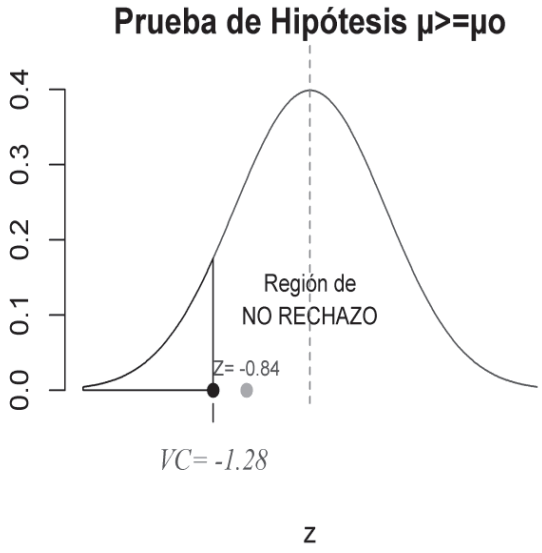


Figura 6.2 *Pruebas de hipótesis de una cola con s conocida*

Interpretar resultados

La figura 6.2 nos muestra el resultado del estadístico z de la muestra que se encuentra en la región de no rechazo. Por tanto, no es posible estadísticamente rechazar Ho.

Prueba de hipótesis de una cola para la media (μ<=μo) con σ conocida

Para la prueba de una cola con media μ<=μo suponga que una escuela de fútbol infla sus pelotas con una bomba electrónica. La presión normal que se suministra es de 11 psi, con una desviación estándar de 0.8 psi. La bomba electrónica sufre un desperfecto y las pelotas que se evidencian falta presión inflan manualmente. Se toman 18 pelotas y se obtienen que la media es 11.35 psi. Si el nivel de significancia es 0.10 ¿La presión manual es menor que la presión introducida con la bomba electrónica?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \leq 11 \\ H_1: & \mu > 11 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=11

Desviación estándar poblacional=0.8

Media de la muestra=11.35

Muestra=18

Coeficiente de Confianza=0.90

Nivel de confianza =0.1

Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{\sigma}$$

$$\sigma = \frac{\sigma_{\overline{x}}}{\sqrt{n}}$$

Se adiciona el cálculo de z de la muestra Zm=((xm-xp)/(SDp/**sqrt**(n))) y del valor crítico mediante vc=**round**(**qnorm**(*alfa*, 0, 1, *lower.tail* = FALSE) , *digits*=2).

```
# DATOS
xp<-11
SDp<-0.8
xm<-11.35
n=18
alfa=0.1
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO Y Z DE LA MUESTRA
```

```
Zm<-((xm-xp)/(SDp/sqrt(n)))
vc<-round(qnorm(alfa,0,1, lower.tail = FALSE) , digits=2)
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ<=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ<=μo
# HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)) )
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTOS DEL Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm, 0.03, paste("Z=", as.character(round(Zm,2))) , cex=0.7, col="black")
```

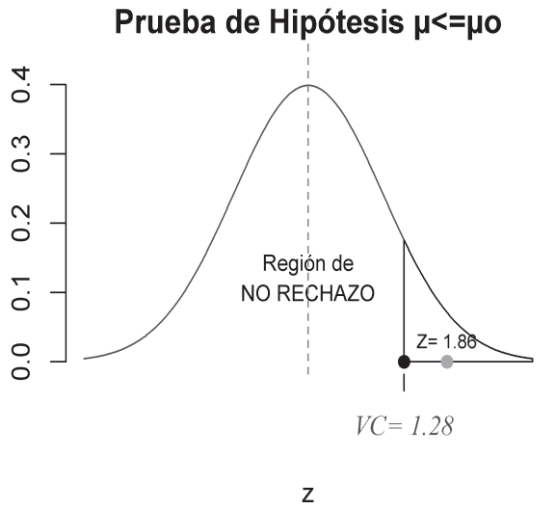


Figura 6.3 *Pruebas de hipótesis de una cola con s conocida*

Interpretar resultados

La figura 6.3 revela que Z de la muestra cae en la región de rechazo, el valor crítico vc es menor que el Z de la muestra. Por tanto, Ho se rechaza.

Prueba de hipótesis para la media con σ desconocida

Para hacer una prueba de hipótesis con la desviación estándar poblacional conocida definimos el estadístico de prueba que se va a usar, en este caso z. Debe indicarse que en la práctica este caso es poco frecuente que ocurra.

$$Z = \frac{\overline{x} - \mu}{\frac{S}{\sqrt{n}}}$$

En donde:

$\overline{x}$= Media de la muestra

μ= Media poblacional

S= Desviación estándar de la muestra

n= Tamaño de la muestra

Prueba de hipótesis de una cola para la media (μ=μo) con σ desconocida

Una cantera de materiales pétreos ha producido un promedio diario de 880 toneladas de grava en los últimos años. El departamento de control de calidad se pregunta si este promedio ha cambiado en los últimos meses. Elige al azar 50 días de la base de datos y calcula el promedio y desviación estándar de la producción. Obtiene una media de $\overline{x}$= 871 toneladas con una desviación estándar s=21 toneladas. Realice una prueba de hipótesis con un nivel de significancia de α=0.05.

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu = 880 \\ H_1: & \mu \neq 880 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=880

Media de la muestra=871

Tamaño de la muestra=50

Desviación estándar=21

Coeficiente de confianza =0.95

Nivel de significancia=0.05

Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

Aplicamos nuestro script de R y veamos el resultado.

```
# DATOS
xp<-880       #Media poblacional
xm<-871       #Media de la muestra
n=50          #Tamaño de la muestra
SD<-21        #Desviación estándar de la muestra
alfa=0.05     #Nivel de significancia
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO Y Z DE LA MUESTRA
Zm<-((xm-xp)/(SD/sqrt(n)))              #Z de la muestra
vc<-round(qnorm(alfa/2,0,1), digits=2)  #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="grey95")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0", paste("VC=", as.charac
ter(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
```

```
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm, 0.05, paste("Z=", as.character(round(Zm, 2))), cex=0.7, col="black")
```

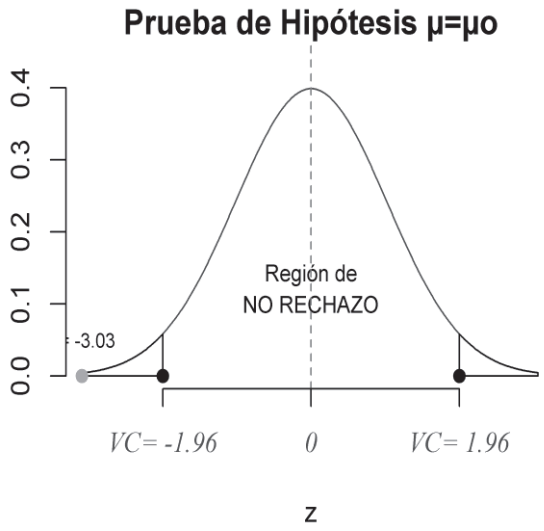


Figura 6.4 *Pruebas de hipótesis de dos colas con s desconocida*

Interpretar resultados

La figura 6.4 da como resultado que el z de la muestra esta ubicado en la región de rechazo. Por tanto, Ho se rechaza.

Prueba de hipótesis de una cola para la media (μ>=μo) con σ desconocida

El promedio semanal de ganancias de las unidades de tricimotos es 560 USD. Una muestra aleatoria de n = 40 nuevas unidades revelaron ganancias de 550 USD con una desviación estándar de 82 USD. ¿Las unidades de tricimotos nuevas que tienen el mismo recorrido generan mejores ganancias? Realice la prueba de hipótesis con un nivel de significancia α=0.01.

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \geq 560 \\ H_1: & \mu < 560 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=560

Media de la muestra=550

Tamaño de la muestra=40

Desviación estándar de la muestra=82

Coeficiente de confianza =0.95

Nivel de significancia=0.05

Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

Aplicamos nuestro script de R y veamos el resultado.

```
# DATOS
xp<-560          #Media poblacional
xm<-550          #Media de la muestra
n=40             #Tamaño de la muestra
SD<-82           #Desviación standard
alfa=0.01        #Nivel de significancia

# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO y Z DE LA MUESTRA
Zm<-((xm-xp)/(SD/sqrt(n)))           #Z de la muestra
vc<-round(qnorm(alfa,0,1), digits=2) #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ>=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
```

```
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm-0.3, 0.05, paste("Z=", as.character(round(Zm, 2))), cex=0.7, col="bla
ck")
```

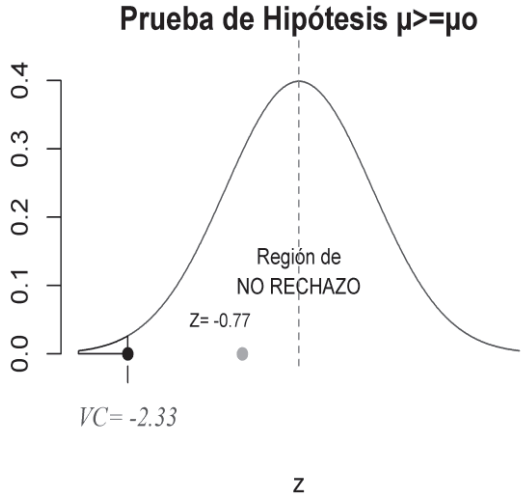


Figura 6.5 *Pruebas de hipótesis de una cola con s desconocida*

Interpretar resultados

La figura 6.5 presenta que Z de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza.

Prueba de hipótesis de una cola para la media (μ<=μo) con σ desconocida

El propietario de un restobar sabe que el promedio de encebollados vendidos por noches es por lo menos de 200. Se toma una muestra de 140 noches, dando una media de 209 encebollados por noche con una desviación estándar de 44.5 platos. Si el nivel de significancia es del 1% ¿Los encebollados no son bien recibidos por los clientes?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \leq 200 \\ H_1: & \mu > 200 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=200

Media de la muestra=209

Tamaño de la muestra=140

Desviación estándar de la muestra=44.5

Coeficiente de confianza =0.90

Nivel de significancia=0.10

Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

Aplicamos nuestro script de R y veamos el resultado.

```
# DATOS
xp<-200         #Media poblacional
xm<-209         #Media de la muestra
n=140           #Tamaño de la muestra
SD<-44.5        #Desviación standard
alfa=0.01       #Nivel de significancia
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm<-((xm-xp)/(SD/sqrt(n)))                                    #Z de la muestra
vc<-round(qnorm(alfa, 0, 1, lower.tail = FALSE ), digits=2) #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ<=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO PARA μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(-vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm+0.2, 0.05, paste("Z=", as.character(round(Zm, 2))), cex=0.7, col="bla
ck")
```

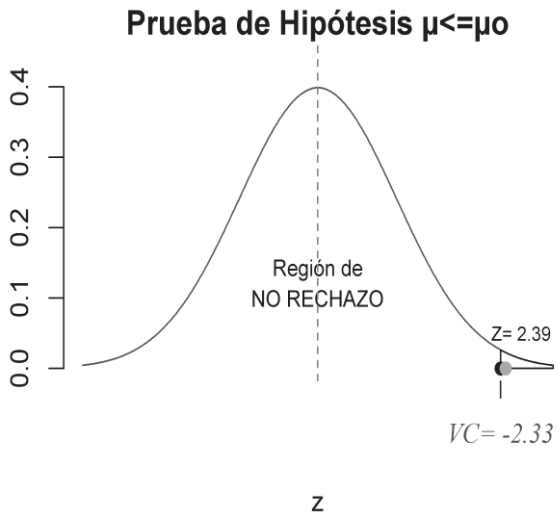


Figura 6.6 *Pruebas de hipótesis de una cola con s desconocida*

Interpretar resultados

La figura 6.6 nos muestra que el Z de la muestra está ubicado en la región de rechazo. Por lo tanto, Ho se rechaza.

Pruebas de hipótesis para muestras pequeñas (n<30)

Suponiendo que se hayan considerado todos los demás factores, es racional usar la prueba t cuando n< 30, el estadístico t y el estadístico z están cerca, y lo que es más importante, la influencia de los grados de libertad en la forma de distribución gradualmente se vuelve nula. La prueba t "Student" propuso superar la incapacidad de la prueba z para muestras pequeñas.

Pruebas de hipótesis de la media con σ desconocida

Para probar la hipótesis en muestras pequeñas se utiliza el estadístico t que se calcula mediante la ecuación 6.5

$$t = \frac{\overline{x} - \mu}{\frac{S}{\sqrt{n}}} \qquad [6.5]$$

Prueba de hipótesis de dos colas para la media (μ=μo) con σ desconocida y n<30

El proceso con el que se fabrican las rosquillas chocolatadas requiere que sus equipos estén calibrados para alcanzar una media del espesor sin chocolate de 39 mm para en lo posterior llegar a un espesor de 41 mm con el chocolate integrado. Se aparta una muestra de seis rosquillas sin chocolate y se mide su espesor obteniendo las siguientes medidas: 39.030, 38.997, 39.012, 39.008, 39.019 y 39.002. Con esta información determine si el proceso necesita recalibración de los equipos.

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu = 39 \\ H_1: & \mu \neq 39 \end{cases}$$

Formular un plan de análisis Datos

Muestra=39.030, 38.997, 39.012, 39.008, 39.019, 39.002 Media poblacional=39

Media de la muestra=xm<-mean(Muestra)

Tamaño de la muestra=n=6

Desviación standard=sd(Muestra)

Nivel de significancia=alfa=0.05

Grados de libertad=GL=n-1

Selección del estadístico

Como n<30 se recomienda utilizar el estadístico t

$$t = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

Veamos el código en R.

```
# DATOS
xp<-39          #Media poblacional
muestra<-c(39.030, 38.997, 39.012, 39.008, 39.019, 39.002)
xm<-mean(muestra) #Media de la muestra
n=6               #Tamaño de la muestra
SD<-sd(muestra)   #Desviación standard
alfa=0.05         #Nivel de significancia
GL=n-1            #Grados de libertad
# GENERAR LOS VALORES DE t
t=seq(-4,4,by=0.001)
#APLICAR LA FUNCIÓN DENSIDAD PARA t
y=dt(t,1)
# VALOR CRÍTICO
tm=round((xm-xp)/(SD/sqrt(n)), 3) #t de la muestra
vc<-round(qt(1-alfa/2, GL, lower.tail = TRUE),3)     #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(t,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
```

```
# HIPÓTESIS μ>=μo
x=seq(-4,-abs(vc),by=0.001)
y=dt(x,1)
polygon(c(-4,x,-abs(vc)), c(0,y,0), col="grey95")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),4,by=0.001)
y=dt(x,1)
polygon(c(abs(vc),x,4),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1, at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0", paste("VC=", as.charac
ter(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(tm,0, cex=1, pch=19, col="gray60")
text(tm-0.5, 0.03,paste("t=",as.character(round(tm,2))),cex=0.7, col="black")
```

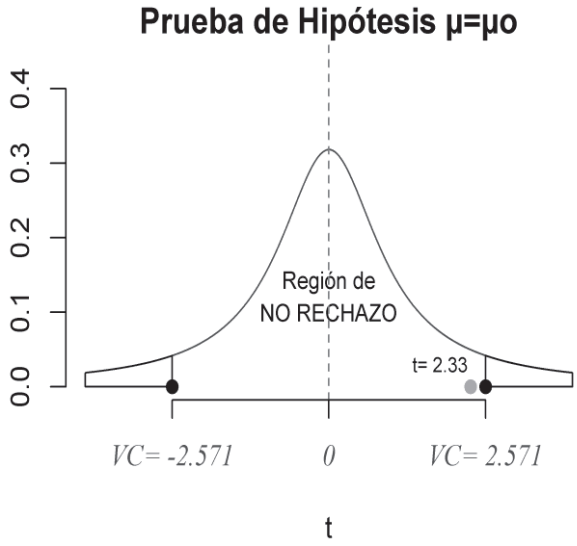


Figura 6.7 *Pruebas de hipótesis de dos colas con s desconocida y n<30*

Interpretar resultados

La figura 6.7 muestra que el t de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza.

Prueba de hipótesis de una cola para la media (μ>=μo) con σ desconocida y n<30

La compañía de jabones de tocador Pradera Verde se preocupa constantemente por la composición química de sus productos para evitar potenciales denuncias por daño de la piel o cabello por uso

de de los mismos. Por tanto, sus fábricas monitorean constantemente el ph de la glicerina usada en la fabricación de los jabones. Se retiene seis jabones luz medieval y se obtiene que la media del ph de la glicerina es 6.68 con desviación estándar de 0.20. ¿Se puede concluir que la media del pH de la glicerina es menor de 7.0?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu >= 7 \\ H_1: & \mu < 7 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional: μ=7

Media de la muestra: xm=6.68

Desviación estándar de la muestra: SDm=0.20

Coeficiente de confianza: CC=0.95

Nivel de significanciaα=0.05

α/2=0.025

Selección del estadístico Vamos a utilizar en este caso el estadístico t

$$t = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

```
# DATOS
xp<-7              #Media poblacional
xm<-6.68           #Media de la muestra
n=6                #Tamaño de la muestra
SDm<-0.20          #Desviación standard
alfa=0.01          #Nivel de significancia
GL=n-1             #Grados de libertad
# GENERAR LOS VALORES DE t
t=seq(-5,5,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA t
y=dt(t,1)
# VALOR CRÍTICO Y T DE LA MUESTRA
tm<-round((xm-xp)/(SDm/sqrt(n)), 3)                    #t de la muestra
vc<-round(qt(alfa/2, GL, lower.tail = TRUE),2)         #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(t,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ>=μo"
     , ylim = c(0,0.45)
     , type="l"
```

```
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
t=seq(-5,-abs(vc),by=0.01)
y=dt(t,1)
polygon(c(-5,t,-abs(vc)), c(0,y,0), col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(tm,0, cex=1, pch=19, col="gray60")
text(tm, 0.05, paste("t=", as.character(round(tm, 2))), cex=0.7, col="black")
```

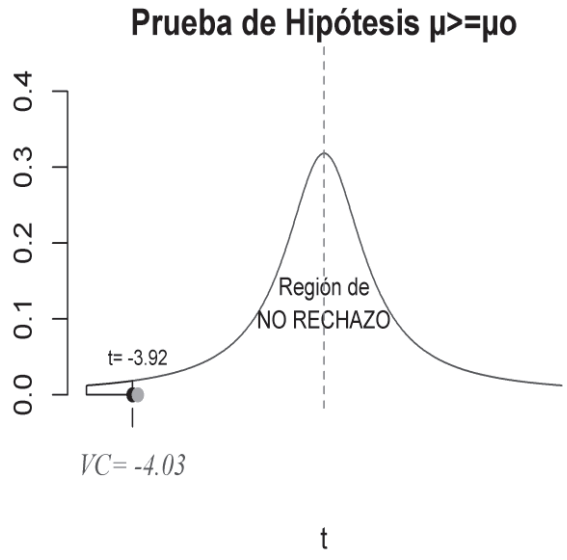


Figura 6.8 *Pruebas de hipótesis de una cola con s desconocida y n<30*

Interpretar resultados

La figura 6.8 indica que el t de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza

Prueba de hipótesis de una cola para la media (μ<=μo) con σ desconocida y n<30

Un proceso para procesar brocolis puede ser rentable si el peso promedio es mayor a 0.5 Kg. Para evaluar la rentabilidad de una nueva cosecha, se toman siete brocolis que registran pesos de 0.55,

0.46, 0.61, 0.52, 0.48, 0.57 y 0.54 Kg. ¿Esta muestra de brocolis indican que el peso promedio de la nueva cosecha es de más de 0.5 kg?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \leq 0{,}5 \\ H_1: & \mu > 0{,}5 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=0.5

muestra=0.55, 0.46, 0.61, 0.52, 0.48, 0.57, 0.54

Media de la muestra=mean(muestra)

Tamaño de la muestra=7

Desviación standard=sd(muestra)

Nivel de significancia=0.05

Grados de libertad=n-1

Selección del estadístico Vamos a utilizar en este caso el estadístico t

$$t = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

Revisemos el código en R.

```
# DATOS
xp<-0.5           #Media poblacional
muestra<-c(0.55, 0.46, 0.61, 0.52, 0.48, 0.57, 0.54)
xm<-mean(muestra) #Media de la muestra
n=7               #Tamaño de la muestra
SDm<-sd(muestra)  #Desviación standard
alfa=0.05         #Nivel de significancia
GL=n-1            #Grados de libertad
# GENERAR LOS VALORES DE t
t=seq(-5,5,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA t
y=dt(t,1)
# VALOR CRÍTICO Y T DE LA MUESTRA
tm<-round((xm-xp)/(SDm/sqrt(n)), 3)              #t de la muestra
vc<-round(qt(alfa/2, GL, lower.tail = FALSE),2)  #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(t,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ<=μo"
```

```
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ<=μo
t=seq(abs(vc),5,by=0.01)
y=dt(t,1)
polygon(c(abs(vc),t,5),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(tm,0, cex=1, pch=19, col="gray60")
text(tm, 0.05, paste("t=", as.character(round(tm, 2))), cex=0.7, col="black")
```

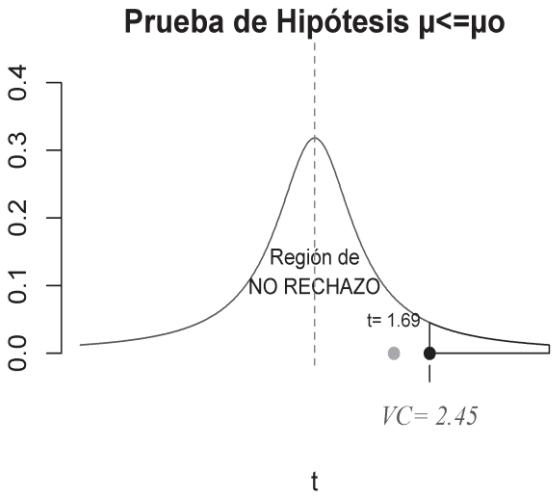


Figura 6.9 *Pruebas de hipótesis de una cola con s desconocida y n<30*

Interpretar resultados

La figura 6.9 indica que el t de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza.

Pruebas relacionadas con proporciones

Las pruebas de proporciones se aplican cuando los datos presentan frecuencias de dos o más clases. Se evalúa las hipótesis pertinentes a una proporción de la población. Las pruebas se fundamentan en el enunciado que una proporción muestral será igual a la proporción de la población. Por lo

general, estas pruebas intentan intuir sobre la diferencia entre un número esperado de ocurrencias y el número real de ocurrencias. Por ejemplo, podríamos querer saber la proporción de hombres dentro de una población de estudiantes cuando realizamos una encuesta. Una prueba de proporción evaluará si una muestra de una población representa la proporción real de toda la población.

Prueba de Hipótesis de dos colas para proporciones

Para calcular el estadístico Z para pruebas de hipótesis con proporciones de una población se utiliza la ecuación 6.6

$$z = \frac{(p - \pi_H)}{\sigma_p} \qquad [6.6]$$

p: Proporción π_H: Valor de la hipótesis σ_p: Desviación estándar de la proporción

Para obtener σ_p debemos aplicar la ecuación 6.7

$$\sigma_p = \sqrt{\frac{\pi_H(1 - \pi_h)}{n}} \qquad [6.7]$$

Apliquemos estos conceptos al siguiente ejemplo.

El director de una escuela asume que el 60% de sus alumnos poseen perros chiguaguas como mascotas. Una muestra de 700 alumnos desvela que 455 alumnos tienen chiguaguas en sus casas. A un nivel de significancia del 5% ¿Se puede inferir que la población efectivamente tiene un 60% de perros chiguaguas como mascotas?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu <= 60\% \\ H_1: & \mu >= 60\% \end{cases}$$

Formular un plan de análisis

Datos

Proporción= 0.6

Tamaño de la muestra= 700

Casos de éxito= 455

Proporción muestral= casosExito/n

Desviación standard= round(sqrt((pi*(1-pi))/n), 3)

Nivel de significancia= 0.05

Selección del estadístico Vamos a utilizar en este caso el estadístico z

$$z = \frac{(p - \pi_H)}{\sigma_p}$$

Implementando el código en R.

```
# DATOS
pi<-0.6           #Proporción
n<-700            #Tamaño de la muestra
casosExito<-455   #Casos de éxito
p<-casosExito/n   #Proporción muestral
SDp<-round(sqrt((pi*(1-pi))/n), 3)   #Desviación standard
alfa=0.05         #Nivel de significancia

# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm=round((p-pi)/(SDp), 3)                  #Z Muestra
vc<-round(qnorm(1-alfa/2,0,1) , digits=3) #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="grey95")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0", paste("VC=", as.charac
ter(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
```

```
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm, 0.05, paste("Z=", as.character(round(Zm, 2))), cex=0.7, col="black")
```

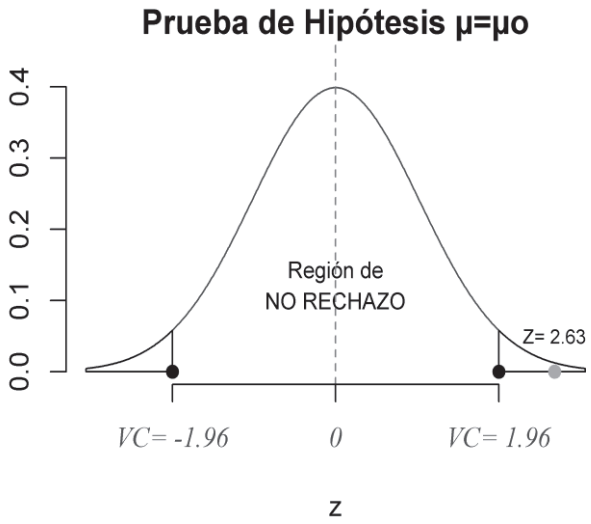


Figura 6.10 *Pruebas de hipótesis de dos colas para proporciones*

Interpretar resultados

La figura 6.10 nos muestra que el z de la muestra está ubicado en la región de rechazo. Por lo tanto, Ho se rechaza.

Prueba de Hipótesis de una cola para proporciones

El propietario de un teatro internacional quiere estar seguro que por lo menos el 70% de sus actores han concluido con su capacitación de actuación escénica. Selecciona 1100 actores y de ellos 750 han concluido las capacitaciones. Con un nivel de significancia del 5% ¿Se puede inferir que todos los actores han concluido sus cursos de capacitación en un 70%?

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \geq 0.7 \\ H_1: & \mu < 0.7 \end{cases}$$

Formular un plan de análisis

Datos

Proporción= 0.7

Tamaño de la muestra= 1100

Casos de éxito= 750

Proporción muestral= casosExito/n

Desviación standard= round(sqrt((pi*(1-pi))/n), 3)

Nivel de significancia= 0.05

Selección del estadístico Vamos a utilizar en este caso el estadístico z

$$z = \frac{(p - \pi_H)}{\sigma_p}$$

Codificando en R el problema.

```
# DATOS
pi<-0.70          #Proporción
n<-1100           #Tamaño de la muestra
casosExito<-750   #Casos de éxito
p<-round(casosExito/n, 3)          #Proporción muestral
SDp<-round(sqrt((pi*(1-pi))/n), 4) #Desviación standard
alfa=0.05         #Nivel de significancia
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm=round((p-pi)/(SDp), 3)             #Z Muestra
vc<-round(qnorm(alfa,0,1) , digits=3) #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
# CURVA NORMAL
plot(z,y
     , xaxt = "n" # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ>=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="grey95")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
```

```
points(Zm,0, cex=1, pch=19, col="gray60")
text(Zm, 0.05, paste("Z=", as.character(round(Zm, 2))), cex=0.7, col="black")
```

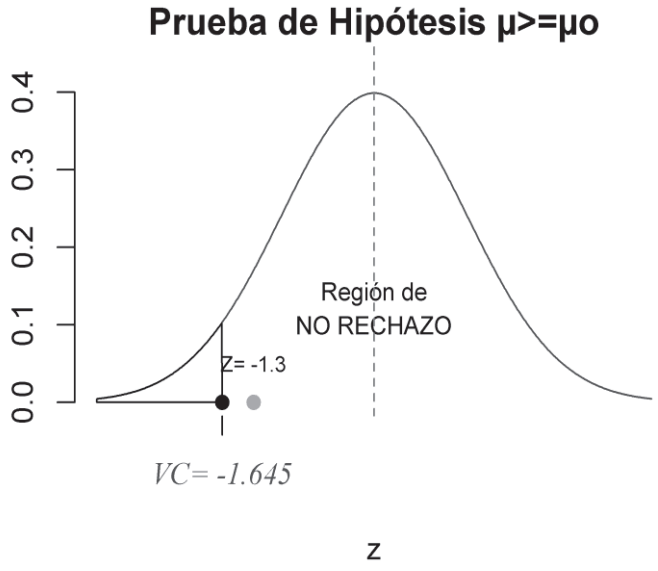


Figura 6.11 *Pruebas de hipótesis de una cola para proporciones*

Interpretar resultados

La figura 6.11 nos muestra que el z de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza.

AUTOEVALUCIÓN

Autoevaluación 6-1

Se desea conocer si ha existido cambio en la proporción del 40% de las mujeres que ingresan a la carrera de Ingeniería en software. En el ingreso para el primer semestre de 200 estudiantes matriculados el 74 son mujeres, use el nivel de significancia de 0,05.

Autoevaluación 6-2

La longitud media de una columna es de 45 milímetros , un ingeniero civil quiere saber si los albañiles han cambiado las medidas, por lo que solicita que se seleccione una muestra de 10 columnas de la obra actual, los resultados se muestran en milímetro; 41,30,44,40,35,37,43,33,31,48, ¿Cambio la longitud media de las columnas? Utilice el nivel de significancia de 0,01.

Autoevaluación 6-3

Un fabricante de gomitas dulces afirma que la media del peso de las gomitas es de 96 gramos con una desviación estándar de 11 gramos. Se toma una muestra de 52 gomitas se tiene una media de

92,7 gramos, ¿el peso medio de las gomitas dulces es menor a 96 gramos? Use un nivel de significancia de 0,10

EJERCICIOS DEL CAPÍTULO

1. Suponga que el diámetro medio de un terreno donde se construyen pequeños cuartos es 20 cm con desviación estándar de 12,1cm. Un inversionista desea comprar más terreno y quiere saber si el terreno es mejor que los de la zona vecina. Se selecciona al azar 45 de los terrenos con diámetro medio de 15,20 cm, con un nivel de significancia de 0,01 ¿muestran los datos de los terrenos vecinos suficiente evidencia como para afirmar que son diferentes?, pinte la zona de peligro con el color brown3
2. El peso de adultos mayores en un centro gerontológico sigue una distribución normal con una media de 120 lb y una desviación estándar de 50 lb, se experimenta un nuevo tipo de alimentación con 60 adultos mayores, en un determinado tiempo se los pesa y se obtiene como resultando una media de 24lb, puede afirmarse que el peso aumentó o por el contrario se ha mantenido a un nivel de significancia del 2%, pinte la zona de peligro con el color darkorange1
3. La junta de profesores de un colegio toma una muestra de 16 estudiantes acerca del tiempo que llegan tarde al plantel educativo resultando una media de 6,2 minutos de retraso y una desviación estándar de 3,4 minutos de retraso, la junta de profesores supone que le tiempo medio de retraso es de 5.5 minutos, Apoya está información la hipótesis con un nivel de significancia del 5%. Pinte la zona de peligro con brown1
4. Se ajusta una máquina procesadora de jugos con 9,0 gramos de contenido de jugo, en una muestra de 9 botellas se llenan las siguientes cantidades por frasco: 8;8,2;8,4; 8,8;8,9; 9; 9,1;9,2;9,5, ¿ se puede concluir con un nivel de significancia de 0,02 que el peso medio es igual a 9,0?, pinte la zona de peligro con darkred
5. La cantidad liquido consumido por niños de una escuela tiene una media de 1,2 litros un nutricionista recomienda que los niños deben tomar como mínimo 2 litros de agua por la actividad física que ahí se realiza, por lo que propone tomar muestras del consumo de agua de los niños después de la recomendación dando como resultados en litros los siguientes datos en una muestra de 8 niños: 1.2,1.5,1.7,1.3,1.4,1.8,1.1,1.6, A un nivel de significancia de 0.02, ¿se puede concluir que se ha elevado el consumo de agua en los niños? , pinte la zona de peligro con coral4.

Respuestas de las autoevaluaciones

Autoevaluación 6-1

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu <= 40\% \\ H_1: & \mu >= 40\% \end{cases}$$

Formular un plan de análisis

Datos

Proporción= 0.4

Tamaño de la muestra= 200

Casos de éxito= 74

Proporción muestral= casosExito/n

Desviación standard= round(sqrt((pi*(1-pi))/n), 3)

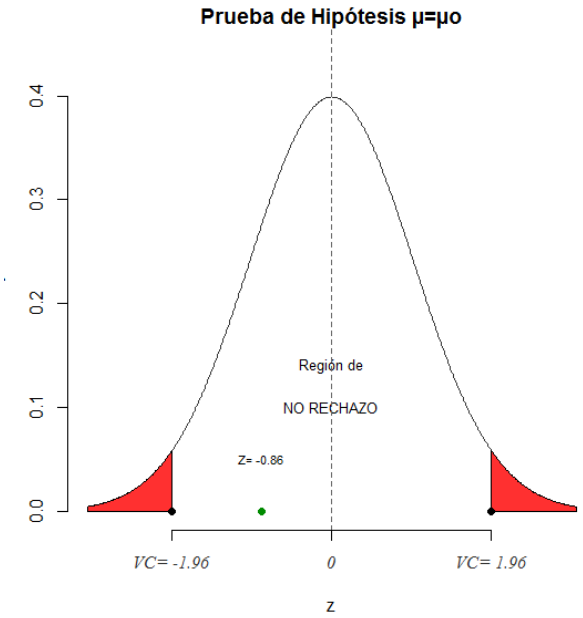
Nivel de significancia= 0.05

Selección del estadístico Vamos a utilizar en este caso el estadístico z

$$z = \frac{(p - \pi_H)}{\sigma_p}$$

```
# DATOS
pi<-0.4                                    #Proporción
n<-200                                     #Tamaño de la muestra
casosExito<-74                             #Casos de éxito
p<-casosExito/n                            #Proporción muestral
SDp<-round(sqrt((pi*(1-pi))/n), 3)         #Desviación standard
alfa=0.05                                  #Nivel de significancia
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm=round((p-pi)/(SDp), 3)                   #Z Muestra
vc<-round(qnorm(1-alfa/2,0,1) , digits=3) #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n"                          # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="firebrick1")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
```

```
polygon(c(abs(vc),x,3),c(0,y,0),col="firebrick1")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0", paste("VC=",
as.character(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="green4")
text(Zm, 0.05, paste("Z=", as.character(round(Zm, 2))), cex=0.7, col="black")
```



Interpretar resultados

El valor z de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza

Autoevaluación 6-2

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu = 45 \\ H_1: & \mu \neq 45 \end{cases}$$

Formular un plan de análisis Datos

Muestra=41,30,44,40,35,37,43,33,31,48 Media poblacional=45

Media de la muestra=xm<-mean(Muestra)

Tamaño de la muestra=n=10

Desviación standard=sd(Muestra)

Nivel de significancia=alfa=0.01

Grados de libertad=GL=n-1

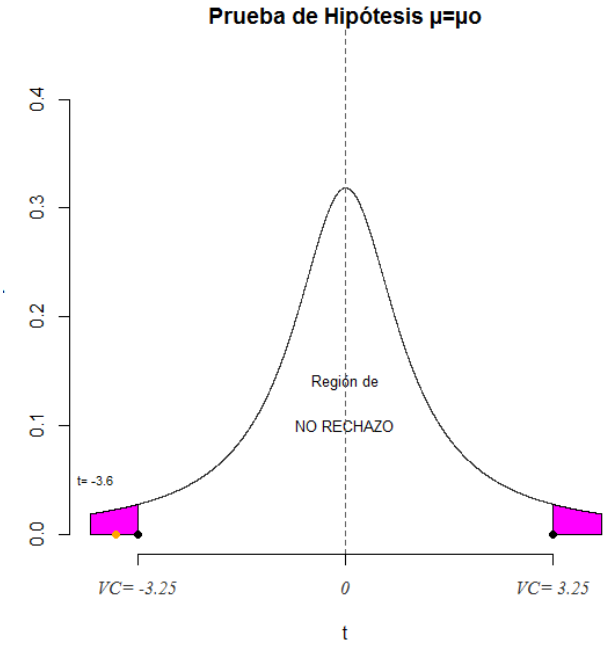
Selección del estadístico

Como n<30 se recomienda utilizar el estadístico t

$$t = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

```
# DATOS
xp<-45                                          #Media poblacional
muestra<-c(41,30,44,40,35,37,43,33,31,48)
xm<-mean(muestra)                               #Media de la muestra
n=10                                            #Tamaño de la muestra
SD<-sd(muestra)                                 #Desviación standard
alfa=0.01                                       #Nivel de significancia
GL=n-1                                          #Grados de libertad
# GENERAR LOS VALORES DE t
t=seq(-4,4,by=0.001)
# APLICAR LA FUNCIÓN DENSIDAD PARA t
y=dt(t,1)
# VALOR CRÍTICO
tm=round((xm-xp)/(SD/sqrt(n)), 3)               #t de la muestra
vc<-round(qt(1-alfa/2, GL, lower.tail = TRUE),3)#Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(t,y
     , xaxt = "n"                               # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-4,-abs(vc),by=0.001)
y=dt(x,1)
polygon(c(-4,x,-abs(vc)), c(0,y,0), col="magenta")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),4,by=0.001)
y=dt(x,1)
polygon(c(abs(vc),x,4),c(0,y,0),col="magenta")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
```

```
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0", paste("VC=",
as.character(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(tm,0, cex=1, pch=19, col="orange")
text(tm-0.3, 0.05, paste("t=", as.character(round(tm, 2))), cex=0.7,
col="black")
```



Interpretar resultados

El valor t de la muestra está ubicado en la región de rechazo. Por lo tanto, Ho se rechaza.

Autoevaluación 6-3

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \geq 96 \\ H_1: & \mu < 96 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=96

Desviación estándar poblacional=11

Media de la muestra=92,7

Muestra=52

Coeficiente de Confianza=0.90

Nivel de confianza=0.1

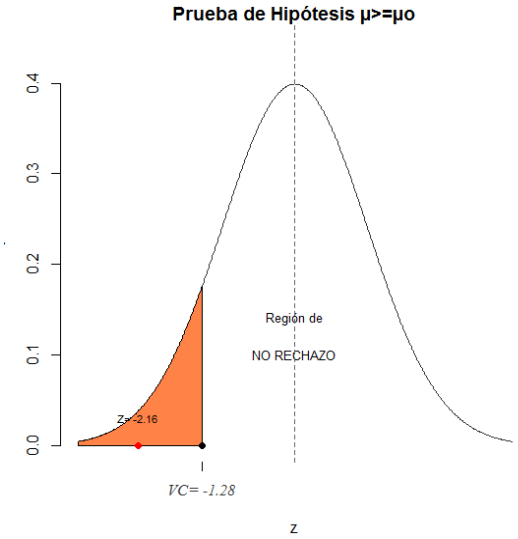
Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{\sigma}$$
$$\sigma = \frac{\sigma_{\overline{x}}}{\sqrt{n}}$$

```
# DATOS
xp<-96                                      #Media poblacional
SDp<-11                                     #Desviación estándar poblacional
xm<-92.7                                    #Media muestral
n=52                                        #Tamaño de la Muestra
alfa=0.1                                    #Nivel de significancia
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm<-((xm-xp)/(SDp/sqrt(n)))                 #Z de la muestra
vc<-round(qnorm(alfa,0,1) , digits=2)       #Z Teórica
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
par(mar=c(4,3,1,1))
plot(z,y
     , xaxt = "n"                           # SIN ETIQUETAS EL EJE Z
     , main="Prueba de Hipótesis μ>=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo
# HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="sienna1")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0,  lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
```

```
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTOS DEL Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="red")
text(Zm, 0.03, paste("Z=", as.character(round(Zm,2))) , cex=0.7, col="black")
```



Interpretar resultados

El valor z de la muestra está ubicado en la región de rechazo. Por lo tanto, Ho se rechaza

7 REGRESIÓN LINEAL SIMPLE

La regresión lineal se usa para predecir el valor de una variable dependiente Y (conocida como variable de respuesta) en función de una o más variables independientes X (conocida también como predictoras). El objetivo es establecer una relación o modelo lineal entre las variables independientes (predictivas) y la variable dependiente (de respuesta), de tal manera que sea posible estimar el valor de la respuesta (y) a través de la relación o modelo establecido en función de variables predictoras (x), cuando solo se conocen los valores de los predictores.

En esta sección vamos a aprender:

- Análisis gráfico de las variables
- Análisis de correlación entre las variables
- Creación del modelo lineal
- Comprobación analítica de la significación estadística
- Comprobación gráfica de la significación estadística
- Predicción de una variable de respuesta (y) en función de una variable predictora (x)

El capítulo desarrolla un modelo de regresión simple para determinar el valor actual de un vehículo a partir de dos variables predictoras: el valor de compra y el recorrido del vehículo. Para ello se construye el conjunto de datos vehículo que estará almacenado en un dataframe del mismo nombre. La tabla 5.0 muestra los datos que usaremos para aplicar la regresión lineal simple. El precio de compra y el actual están en miles de dólares.

```
library("knitr")
modelo<-c("city", "city", "city", "city", "creta", "creta", "verna", "elantra
", "grand-i10", "innova", "etios-gd", "corolla", "ciaz", "ertiga", "ertiga",
"ertiga", "ciaz", "ciaz", "ciaz", "s cross", "vitara")

anio<-c(2017, 2016, 2015, 2016, 2016, 2015, 2015, 2015, 2015, 2015, 2015, 201
6, 2015, 2016, 2015, 2016, 2015, 2015, 2016, 2015, 2018)

pCompra<-c(11.5, 9.5, 8.55, 8.35, 12.9, 11.25, 8.25, 11.75, 4.85, 12.5, 4.75,
14.73, 7.45, 7.75, 7.25, 7.75, 6.85, 7.45, 8.75, 6.5, 9.25)

recorrido<-c(9000, 33988, 60076, 19434, 35934, 68000, 61381, 43535, 21125, 38
000, 40000, 23000, 45000, 43000, 41678, 43000, 51000, 42367, 20273, 33429, 20
71)

pActual<-c(12.5, 11.6, 13.09, 9.4, 13.6, 13.6, 9.4, 14.79, 5.7, 13.46, 7.85,
14.89, 10.38, 10.79, 10.79, 10.79, 10.38, 8.92, 8.89, 8.61, 9.83)
```

```
vehiculos<-data.frame(modelo, pCompra, recorrido, pActual)
kable(head(vehiculos, 5), caption="Tabla 6.0 Conjunto de datos vehiculos")
```

Tabla 6.0 *Conjunto de datos vehiculos*

modelo	pCompra	recorrido	pActual
city	11.50	9000	12.50
city	9.50	33988	11.60
city	8.55	60076	13.09
city	8.35	19434	9.40
creta	12.90	35934	13.60

Fundamentos de la regresión lineal

El trabajo de desarrollar una ecuación matemática puede ser bastante complejo, porque necesitamos tener una idea sobre la naturaleza de la relación entre cada una de las variables independientes y la variable dependiente. El número de modelos matemáticos diferentes que podrían proponerse es prácticamente infinito [61]. La variable independiente es la variable en regresión que se puede controlar o manipular. La variable dependiente es la variable en regresión que no puede ser controlada o manipulada. La variable independiente también se conoce como variable explicativa o predictora, y la variable dependiente también se llama variable de respuesta. La determinación de las variables x e y no siempre es clara y a veces es una decisión arbitraria [62].

La relación entre dos variables en un análisis de regresión se expresa mediante una ecuación matemática llamado modelo de regresión. Una ecuación de regresión, cuando se traza, puede asumir una de las muchas formas posibles, incluida una línea recta. Una ecuación de regresión que proporciona una relación de línea recta entre dos variables se denomina modelo de regresión lineal; de lo contrario, el modelo se llama modelo de regresión no lineal [63]. La finalidad de la regresión lineal es crear un modelo para una variable continua Y en función de una o más variables X que pueda ser útil para predecir Y cuando sólo se conoce la X. La ecuación matemática 7.0 que generaliza el modelo lineal a encontrarse es:

$$Y = \beta 0 + \beta 1X + \epsilon \qquad [7.0]$$

En el plano cartesiano o de coordenadas (x,y), β0 es la intersección con el eje y, β1 es la pendiente o inclinación de la recta. Ambas se conocen como **coeficientes de regresión** (ver figura 7.0). Además, ϵ es el error de la parte de Y que el modelo de regresión no puede explicar.

Análisis gráfico de las variables

Antes de proceder a encontrar el modelo lineal, es necesario conocer cómo actúan las variables predictoras. Intentemos comprender estas variables gráficamente. Por lo general, para cada una de

las variables independientes (predictores), se usan gráficos de dispersión, caja y bigote y de densidad para visualizar su comportamiento.

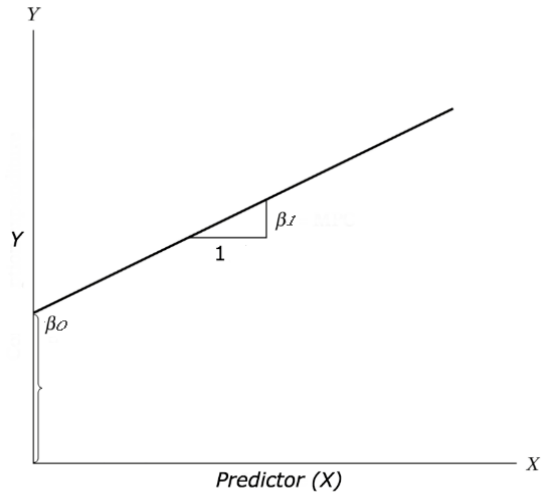


Figura 7.0: *Coeficientes de regresión*

Diagrama de dispersión: El diagrama de dispersión muestra un patrón lineal y la dispersión en los valores de y parece ser similar en el rango de valores de x. Esto apoya la adecuación del modelo de regresión lineal simple [64]. En otras palabras, los gráficos de dispersión facilitan la visualización cualquier relación lineal entre la variable dependiente (respuesta) y las variables independientes (predictoras). Teóricamente es posible que múltiples variables predictoras se observen en un diagrama de dispersión para cada una de ellas contra la respuesta, junto con la línea de mejor ajuste.

De acuerdo con lo anterior, vamos a visualizar dos gráficos de dispersión correspondiente a: precio actual ~ precio de compra y precio actual ~ recorrido. R Statistics provee la función **scatter.smooth**() que presenta un gráfico de dispersión y le agrega una curva de ajuste suavizada, los parámetros más importantes son *x*=variable predictora y *y*=variable respuesta. Los demás parámetros son similares a los trabajados en la función **plot**(). La figura 7.1 da los gráficos de dispersión resultantes para las variables predictoras precio compra y recorrido versus la variable de respuesta precio actual.

```
par(mfrow=c(1,2), mar=c(3.8,4,2,2))
scatter.smooth(x=vehiculos$pCompra      # Variable predictora
               , y=vehiculos$pActual    # Variable de respuesta
               , main="Precio Actual ~ Precio de Compra"
               , cex.main=0.8
               , xlab="Precio de Compra (en miles)"
               , ylab="Precio Actual (en miles)"
               , cex.lab=0.8)
scatter.smooth(x=vehiculos$recorrido   # Variable predictora
               , y=vehiculos$pActual   # Variable de respuesta
```

```
                , main="Precio Actual ~ Recorrido"
                , cex.main=0.8
                , xlab="Recorrido (Km)"
                , ylab="Precio Actual (en miles)"
                , cex.lab=0.8)
```

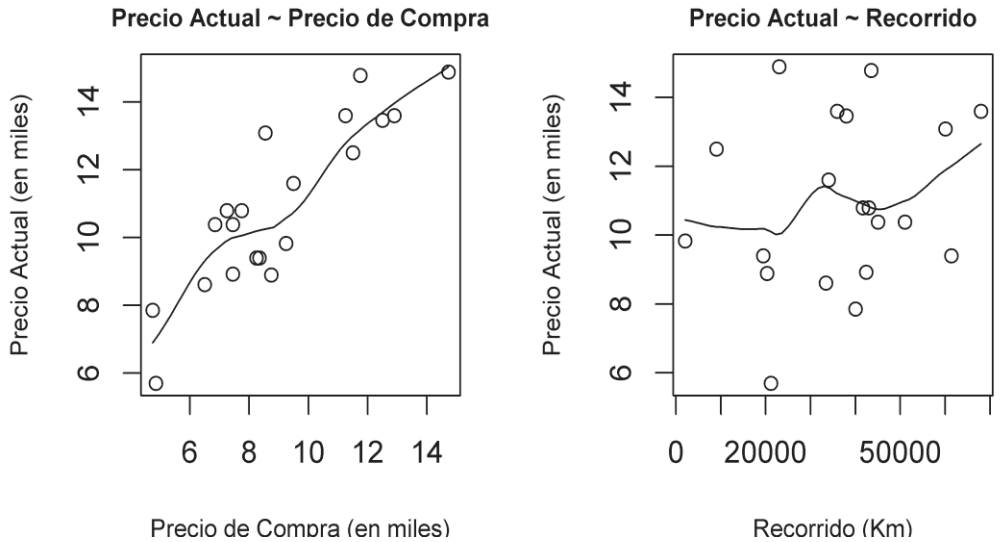


Figura 7.1 *Gráficos de dispersión: variables predictoras vs variable de respuesta*

El gráfico de dispersión junto con la curva de ajuste suavizado de la figura 7.1 sugiere una relación linealmente creciente entre la variable predictora "precioCompra" y la variable de respuesta "precioActual". Esto indica que probablemente la relación entre la respuesta (precioActual) y la variable predictora (precioCompra) es lineal y aditiva. Por otra parte, la relación entre la variable predictora "Recorrido" y la variable de respuesta "precioActual" no muestra de forma clara una relación lineal.

Gráfico de caja: Este gráfico es útil para para detectar los valores atípicos en las variables predictoras. Los valores atípicos podrían afectar drásticamente las predicciones por su influencia en la pendiente de la línea de mejor ajuste. En general, cualquier punto de datos que se encuentre fuera de 1.5 * IQR se considera un valor atípico, donde, IQR se calcula como la distancia entre los valores de los cuartiles Q1 y Q3 para esa variable.

La función **boxplo**t() tratada en el capítulo 2 nos facilita la visualización de los valores atípicos. Se recomienda configurar el parámetro *range* con el valor de 0.8 para asegurar que se visualicen los valores atípicos.

```
par(mfrow=c(1,2), mar=c(1.6,4,1,1))
boxplot(vehiculos$pCompra           # Variable predictora
                , main="Valores atípicos"
                , cex.main=0.8
                , xlab=""
                , ylab="Dólares (en miles)"
```

```
                , range=0.8          # Rango
                , boxwex = .5        # Ancho de la caja
        )
mtext(side=1, line=0.8, "Precio de compra", cex = 1)
boxplot(vehiculos$recorrido          # Variable predictora
                , main="Valores atípicos"
                , cex.main=0.8
                , xlab=""
                , ylab="Km (en miles)"
                , range=0.8          # Rango
                , boxwex = .5        # Ancho de la caja
        )
mtext(side=1, line=0.8, "Recorrido", cex = 1)
```

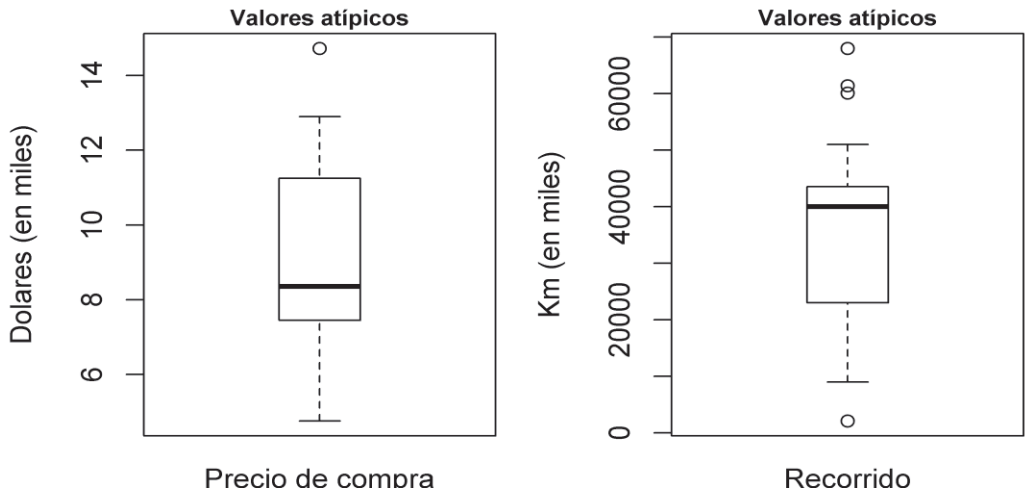


Figura 7.2 *Caja y bigote para visualizar valores atípicos*

La Figura 7.2 muestra que la variable predictora recorrido posee una mayor cantidad de valores atípicos que precio de compra. Esto no es una buena señal porque el modelo puede verse afectado seriamente con estos valores atípicos.

Gráfico de densidad: Teóricamente, se prefiere una distribución cercana a la distribución normal, sin sesgo hacia la izquierda o hacia la derecha, para que el modelo sea útil. El gráfico de la función densidad de la distribución contrastada con la curva normal es una buena medida visual de la normalidad de dicha distribución. Recordemos que para crear el gráfico de densidad con la curva normal se utiliza las funciones **density**(), **plot**(), **curve**() y **dnorm**() tratados en el capítulo 2.

Otra gráfica útil es la conocida como cuantil-cuantil que permite observar que tan cerca está la distribución de un conjunto de datos de alguna distribución conocida, en nuestro caso a la distribución normal. Si la distribución del conjunto de datos se aproxima a una distribución normal es necesario que los puntos que se grafican estén muy próximos o sobre la recta que aparece en el gráfico qq. R tiene las funciones **qqnorm()** y **qqline()** para implementar estos gráficos. Una combinación de estos gráficos se presenta en la figura 7.3.

```r
par(mfrow=c(2,2), mar=c(3,4,2,2))
# GRÁFICO DE DENSIDAD DE PRECIO-COMPRA
densidad_pc<-density(vehiculos$pCompra) # Densidad de la
                                        # variable predictora
plot(densidad_pc                        # Gráfico de densidad
     , main = "Precio Compra"
     , ylab="Densidad")
x<-vehiculos$pCompra                    # Variable predictora
curve(dnorm(x, mean(x), sd(x))          # Curva normal para
      , col="red"                       # la variable predictora
      , add=TRUE)
# GRÁFICO DE DENSIDAD DE RECORRIDO
densidad_r<-density(vehiculos$recorrido)# Densidad de la
                                        # variable predictora
plot(densidad_r                         # Gráfico de densidad
     , main = "Recorrido"
     , ylab="Densidad")
x<-vehiculos$recorrido                  # Variable predictora
curve(dnorm(x, mean(x), sd(x))          # Curva normal para
      , col="red"                       # la variable predictora
       , add=TRUE)
# GRÁFICO QQ PARA PRECIO-COMPRA
qqnorm(vehiculos$pCompra                # Puntos de la
       , main = "QQ - Precio Compra")   # distribución pCompra
qqline(vehiculos$pCompra, col = 2)      # Línea cuantil-cuantil
# GRÁFICO QQ PARA RECORRIDO
qqnorm(vehiculos$recorrido              # Puntos de la
       , main = "QQ - Recorrido")       # distribución recorrido
qqline(vehiculos$recorrido, col = 2)    # Línea cuantil-cuantil
```

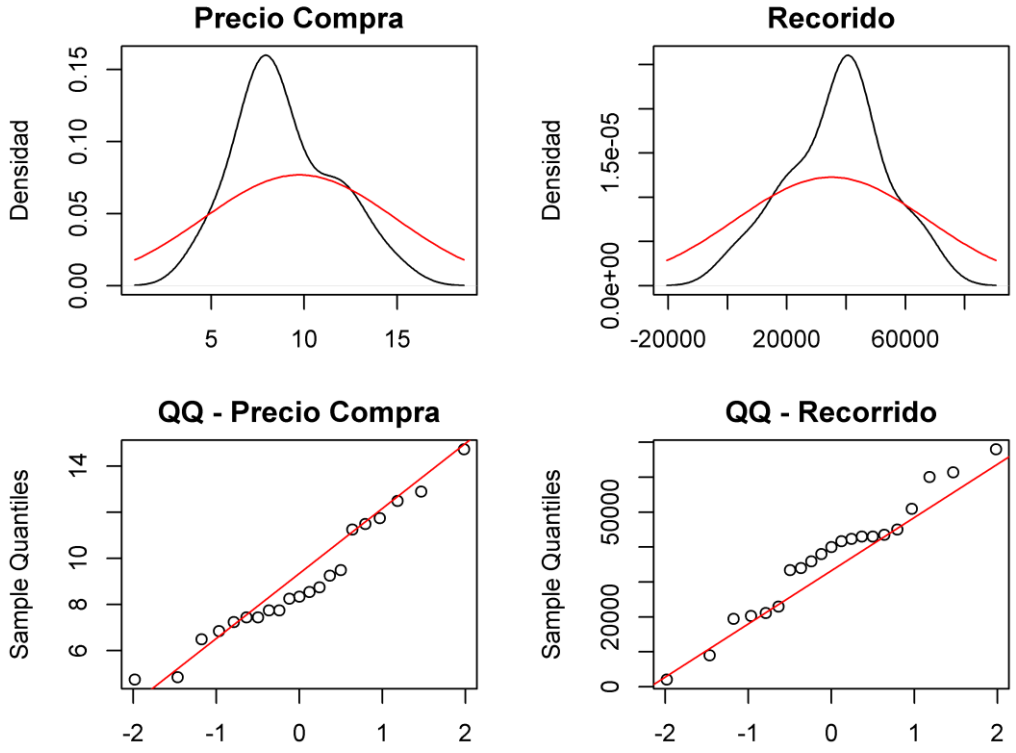


Figura 7.3 *Densidad vs curva normal apoyados con gráficos qq*

La figura 7.3 nos muestra la potencia de la combinación de los gráficos de densidad versus la curva normal y los gráficos cuantil cuantil (qq). Se observa de forma clara que la variable predictora precio compra tiene mejor condición de normalidad que la variable predictora recorrido, esto se nota sobre todo en el gráfico QQ - Precio Compra donde se observa una mejor alineación de los puntos sobre la recta que el gráfico QQ - Recorrido. Además, la forma de la distribución, vista en los gráficos de densidad versus la curva normal, señala que se trata de distribuciones leptocúrticas (es decir puntiagudas con respecto a la normal), con sesgo a la derecha para la variable predictora precio compra y con sesgo hacia la izquierda para la variable predictora recorrido. En ambos casos, hay que tomar en cuenta este detalle porque podría no ser saludable para el modelo esta situación.

La tabla 7.1 da un resumen de los resultados de las pruebas gráficas. Se ha utilizado el signo (+) como indicador de satisfacción de la prueba. El máximo valor es (+++++) y el valor mínimo (+).

Tabla 7.1 *Resultados de las pruebas gráficas para elegir la variable predictora*

Prueba	Precio Compra	Precio Actual
Dispersión	++++	+
Valores atípicos	+++	+
Normalidad	++	+

De acuerdo a la tabla 7.1, la variable precio compra tiene mejores condiciones para ser elegida como la variable predictora del modelo. A continuación haremos el análisis de correlación que es crucial para tomar una decisión sobre la selección de la variable predictora.

Análisis de correlación entre las variables

Cuando estamos interesados en la distribución conjunta de dos variables aleatorias, es útil tener un resumen de cuánto dependen las dos variables aleatorias entre sí. La covarianza y la correlación son intentos de medir esa dependencia, pero solo capturan un tipo particular de dependencia, a saber, la dependencia lineal [65]. Entonces, el coeficiente de correlación lineal mide qué tan cerca se distribuyen los puntos en un diagrama de dispersión alrededor de la línea de regresión, el valor del coeficiente de correlación siempre se encuentra entre -1 a 1 [66].

Si el coeficiente de correlación toma un valor de 1, tenemos una correlación lineal positiva perfecta y en el diagrama de dispersión se encuentran en una línea recta que se inclina entre 0 y 90 grados. Si el coeficiente de correlación toma un valor de -1, este corresponde a una correlación lineal negativa perfecta y en el diagrama de dispersión se encuentran en una línea recta que se inclina entre 90 y 180 grados. Un valor de correlación cercano a 1 manifiesta que es posible que exista una buena dependencia lineal (relación fuerte) entre las variables mientras que si es cercano a 0

niega la existencia de una dependencia lineal (relación débil). A continuación se presenta una escala de valore para interpretar la correlación entre dos variables (Ver tabla 7.2).

Tabla 7.2 *Escala de interpretación de la correlación*

Relación	Positiva	Negativa
Perfecta	R = 1	R = -1
Excelente	0.9 < = R < 1	-0.9 < = R < -1
Buena	0.8 < = R < 0.9	-0.8 < = R < -0.9
Regular	0.5 < = R < 0.8	-0.5 < = R < -0.8
Mala	R < 0.5	R < -0.5

Coeficiente de correlación lineal simple, denotado por r, mide la fuerza de la relación lineal entre dos variables para una muestra. La ecuación 7.1 muestra como se realiza el cálculo de la correlación de Pearson.

$$r = \frac{SS_{xy}}{\sqrt{SS_{xx}SS_{yy}}} \qquad [7.1]$$

De donde SS_{xy} es la la suma de las desviaciones cruzadas de y y x, SS_{xx} es la suma de los cuadrados de x y SS_{yy} es la suma de los cuadrados de y. Las ecuaciones 7.2 son usadas para encontrar su valor numérico.

$$\begin{aligned} SS_{xy} &= \sum xy - \frac{\sum x \sum y}{n} \\ SS_{xx} &= \sum x^2 - \frac{(\sum x)^2}{n} \\ SS_{yy} &= \sum y^2 - \frac{(\sum y)^2}{n} \end{aligned} \qquad [7.2]$$

En R, es suficiente usar la función **cor()** y ajustar los dos parámetros más importantes que son: *use* que configura el uso de datos perdido y cuyas opciones pueden ser all.obs (asume que no faltan datos - los datos faltantes producirán un error), complete.obs (eliminación de lista) y pair.complete.obs (eliminación por pares); y, el parámetro *method* que tiene las opciones de pearson, kendall, spearman. El código de abajo calcula los valores de correlación y los presenta como una matriz de correlaciones.

```
# COLUMNAS DE LAS VARIABLES PREDICTORAS CANDIDATAS
# Y LA VARIABLE DE RESPUESTA
columnas<-c("pCompra", "recorrido", "pActual")
correlacion<-cor(vehiculos[,columnas]
                 , use="complete.obs"   # Uso de datos
                 , method="kendall")    # Método de cálculo
kable(correlacion, caption = "Tabla 7.3: Matriz de correlaciones")
```

La tabla 7.3 nos da los resultados de la correlación entre las variables predictoras precio de compra (pCompra) y recorrido con la variable de respuesta precio actual (pActual). El valor que más se

aproxima a 1 es 0.64566 que corresponde a la variable predictora precio de compra y la variable de respuesta precio actual. De acuerdo a la tabla 7.2 este valor de correlación puede ser calificado como regular.

***Tabla 7.3**: Matriz de correlaciones*

	pCompra	recorrido	pActual
pCompra	1.0000000	-0.0671465	0.6456615
recorrido	-0.0671465	1.0000000	0.1259172
pActual	0.6456615	0.1259172	1.0000000

De acuerdo a los resultados obtenidos en las tablas 7.1 y 7.3 es posible indicar que la variable predictora a seleccionar es precio compra. Corresponde notar que aunque el modelo este bien construido podría no ser útil.

Construcción del modelo lineal

Obtenido los resultados gráficos de la dispersión, valores atípicos, la normalidad y calculada la correlación se ha procedido a realizar el análisis de las variables predictoras y se ha concluido que la variable predictora precio compra tiene las mejores condiciones - entre todas las variables predictoras candidatas - para crear el modelo lineal.

La función utilizada en R para construir modelos lineales es **lm()**. La función **lm**() toma dos argumentos principales: *formula* y *data*. En *formula* se ubica la variable de resultado y las variables predictoras que serán parte del modelo, mientras que *data* se pasa el data.frame que contiene dichas variables.

```
# VARIABLE RESULTADO ~ VARIABLE PREDICTORA
modeloLineal <- lm(pActual ~ pCompra, data=vehiculos)  # Dataframe
print(modeloLineal)
##
## Call:
## lm(formula = pActual ~ pCompra, data = vehiculos)
##
## Coefficients:
## (Intercept)      pCompra
##      3.7393       0.8023
```

R construye el modelo lineal basado en las variables predictoras y de respuesta dándonos el intercepto con el eje de la variable de respuesta (eje y) y el coeficiente (inclinación o pendiente) de la variable predictora. El modelo generado por R se plantea en la ecuación 7.3.

$$\begin{aligned} variableRespuesta &= Intercepto \pm coeficiente * variablePredictora \\ precioActual &= 3.7393 + 0.8023 * precioCompra \end{aligned} \quad [7.3]$$

Con la función **scatter.smooth**(x, y,…) x toma el valor de la variable predictora mientras se ubicala variable de respuesta en y. Esto nos da el gráfico de dispersión para el modelo lineal.

Para graficar la recta del modelo lineal obtenido, R posee la función **abline**() que toma el modelo lineal y lo gráfica junto al gráfico de dispersión. Analice el código que a continuación se presenta y observe su resultado en la figura 7.4.

```
par(mar=c(4,4,1,1))
scatter.smooth(x=vehiculos$pCompra      # Variable predictora
               , y=vehiculos$pActual    # Variable de respuesta
               , main="Precio Actual ~ Precio de Compra"
               , cex.main=0.8
               , xlab="Precio de Compra (en miles)"
               , ylab="Precio Actual (en miles)"
               , cex.lab=0.8)
# GRÁFICO DE LA RECTA DE REGRESIÓN LINEAL SIMPLE.
abline(modeloLineal     # Modelo lineal
       , lwd = 2
       , col = "red")
```

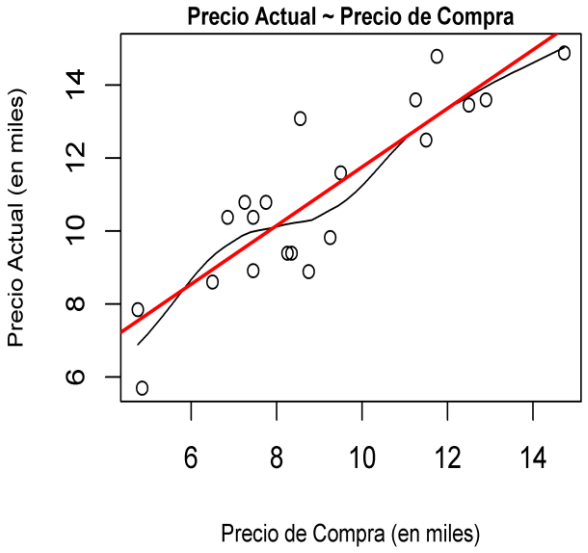


Figura 7.4 *Modelo Lineal*

Finalmente, una forma sencilla de predecir un valor de la variable de respuesta es codificar una sencilla función con el modelo lineal establecido. El siguiente código muestra cómo hacerlo.

```
B0<-3.7393
B1<-0.8023
resultado<-function(predictora)
     {B0+B1*predictora}
precioActual<-resultado(14)
precioActual
## [1] 14.9715
```

Hasta el momento hemos concretado la obtención de un modelo lineal a partir de la variable predictora, Hemos realizado pruebas para seleccionar la mejor variable predictora posible; sin embargo, aún debemos realizar otras pruebas que permitan validar el modelo lineal obtenido. Como validar el modelo lineal es lo que veremos en el siguiente apartado.

Diagnóstico para de regresión lineal simple

El modelo lineal obtenido en la ecuación 7.0 está listo para ser usado; sin embargo, no hemos realizado ninguna comprobación de la validez del modelo y esto podría conducirnos a obtener valores de respuesta erróneos. Por lo tanto, antes de usar un modelo de regresión, se debe asegurar que el modelo lineal sea estadísticamente significativo.

Comprobación analítica de la significación estadística

Para determinar que un modelo lineal simple es estadísticamente significativo, vamos a revisar el resumen del modelo lineal generado utilizando la función **summary()**

```
summary(modeloLineal)
## Call:
## lm(formula = pActual ~ pCompra, data = vehiculos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9304 -0.7964 -0.3079  0.8329  2.4911
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7393     0.9465   3.951 0.000857 ***
## pCompra       0.8023     0.1017   7.889 2.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.194 on 19 degrees of freedom
## Multiple R-squared:  0.7661, Adjusted R-squared:  0.7538
## F-statistic: 62.23 on 1 and 19 DF,  p-value: 2.062e-07
```

El **sumario del modelo lineal** nos da información vital para determinar la validez del modelo. La tabla 7.4 indica qué representa dicha información sumaria.

Tabla 7.4 *Descripción del resumen del modelo lineal*

Resultado	Descripción
Call	La fórmula del modelo
Residuals	Es la diferencia entre la respuesta real y la respuesta predicha del modelo y permite. hacer una verificación rápida de la distribución de residuos; donde idealmente, la mediana es cercana a 0, max y min, y 1Q y 3Q, son aproximadamente equivalentes
Coefficients	Dan información para encontrar el mejor modelo que se adapte adecuadamente a los puntos del gráfico de dispersión

Estimate	Representan la intersección y la pendiente
Std. error	Es la estimación del error que podemos obtener al calcular la diferencia entre el valor real y el valor predicho de la variable de respuesta. A su vez, esto informa sobre la confianza para relacionar las variables de entrada y salida. Es el error estándar de β1
Valor t	Estadístico de prueba para H0: β1=0.
Pr (> t OJO)	Valor p de la prueba de hipótesis (2 colas)
Signif. code	Indican significación estadística. A más asteriscos mejor.
Error estándar residual	σ
Degrees of freedom	Número de observaciones - número de parámetros estimados
Multiple R-squared	Medida del ajuste del modelo (0,1)
Adjusted R-squared	Medida del ajuste del modelo ajustado por el número de parámetros (0,1)
F-statistic	Estadístico de prueba para la hipótesis de que todos los coeficientes (que no sean intercepto) simultáneamente son iguales a cero
P-value	Cuanto más cerca esté de cero, más fácil podremos rechazar *la hipótesis nula*.

A continuación se presenta un resumen de cómo interpretar el los ítems del sumario que se obtiene cuando se aplica función **lm**().

p-value: Para que un modelo sea estadísticamente significativo, es necesario que cumpla con algunas condiciones. Una de esas condiciones es que los coeficientes de las variables predictoras sean diferentes de cero. El p-value mide la posibilidad de que esta condición se cumpla. En consecuencia, se plantea las siguientes hipótesis:

H0: Los coeficientes asociados con las variables = 0

H1: Los coeficientes asociados con las variables ≠ 0

Se debe cumplir que: p-value < nivel de significancia; el nivel de significancia, por lo general, se asume del 5% (0.05). Esto se interpreta visualmente por los asteriscos de importancia (*) al final

de la fila de los coeficientes. Mientras más asteriscos estén al lado del valor p de la variable, más significativa será la variable

Valor t: El estadístico t es el coeficiente dividido por su error estándar (Coeficiente / Error Estándar). El valor t del coeficiente es una medida de cuántas desviaciones estándar nuestro coeficiente estimado está muy lejos de 0. Es deseable esté muy lejos de cero ya que esto indicaría que podríamos rechazar la hipótesis nula del p-value.

Pr (> | t |) o valor p: Es la probabilidad de que obtenga un valor t tan alto o más alto que el valor observado cuando la hipótesis nula (el coeficiente β1 es igual a cero o que no hay relación) es verdadera . Si Pr (> | t |) es bajo, los coeficientes son significativamente diferentes de cero. Si Pr (> | t |) es alto, los coeficientes no son significativos.

Residual standard error: Mide la bondad de ajuste que se puede utilizar para analizar qué tan bien se ajusta un conjunto de puntos de datos con el modelo real. Cuanto más pequeña es la desviación estándar residual en comparación con la desviación estándar de la muestra, más predictivo o útil es el modelo.

Multiple R squared: Representa la proporción de la varianza para una variable dependiente que se explica por medio de una variable o varias variables independientes en un modelo de regresión. Mientras que la correlación explica la fuerza de la relación entre una variable independiente y una dependiente, R-cuadrado explica en qué medida la varianza de una variable explica la varianza de la segunda variable. Se tiene entonces que:

- 0% indica que el modelo no explica la variabilidad de los datos de respuesta en torno a su media.
- 100% indica que el modelo explica toda la variabilidad de los datos de respuesta en torno a su media.

Entonces, si el R2 de un modelo es 0.50, entonces aproximadamente la mitad de la variación observada puede explicarse por las entradas del modelo.

Ajusted R squared: El R cuadrado ajustado compara el poder explicativo de los modelos de regresión que contienen diferentes números de predictores. El R cuadrado ajustado se ajusta por el número de términos en el modelo. Es importante destacar que su valor aumenta solo cuando el nuevo término mejora el ajuste del modelo más de lo esperado solo por casualidad. El valor R cuadrado ajustado en realidad disminuye cuando el término no mejora el ajuste del modelo en una cantidad suficiente. Se debe evitarperseguir un valor alto de R cuadrado ajustado porque puede empujarnos a incluir demasiados predictores en un intento de explicar lo inexplicable.

F-statistic: El valor F siempre debe usarse junto con el valor p para decidir si sus resultados son lo suficientemente significativos como para rechazar la hipótesis nula. El valor F en la regresión es el resultado de una prueba donde la hipótesis nula es que todos los coeficientes de regresión son iguales a cero. En otras palabras, el modelo no tiene capacidad predictiva. La prueba F compara su modelo con variables predictoras cero (el modelo de solo intercepción) y decide si sus coeficientes agregados mejoraron el modelo. Si obtiene un resultado significativo, los coeficientes que haya incluido en su modelo mejoraron el ajuste del modelo.

Muchos autores recomiendan ignorar los valores de P para los coeficientes de regresión individuales si la relación general de F no es estadísticamente significativa. Esto se debe al problema de las pruebas múltiples. En otras palabras, su valor p y su valor f deben ser estadísticamente significativos para interpretar correctamente los resultados. La tabla 7.5 nos da un F = 62.23 con 1 grados de libertad de regresión (df1) y 19 grados de libertad residuales (df2). Se necesita una estadístico F de al menos 2.99 para rechazar la hipótesis nula en un nivel alfa de 0.05. En este nivel, tienes una probabilidad del 5% de estar equivocado.

La tabla 7.5 muestra el análisis u conclusión del modelo lineal desarrollado como ejemplo.

Tabla 7.5 *Validación del modelo lineal*

Coeficiente	Valor	Análisis	Conclusión
t-value	7.889	7.88 es un valor alto	OK
p-value	2.062e-07	el p-value es menor que 0.05	OK
Model F-statistic	62.23 on 1 and 19 DF	El valor resultante es alto. Se rechaza H0	OK
Model p-value	2.062e-07	el p-value es menor que 0.05	OK

Comprobación gráfica de la significación estadística

Las gráficas de diagnóstico muestran los residuos de cuatro maneras diferentes: Residuales vs equipado (residuals vs fitted), normal Q-Q, ubicación de escala o ubicación de propagación (scale-location / spread-location), residuos vs apalancamiento (residuals vs leverage). A continuacion se explican cada una de ellas. Ver figura 7.5.

Residuals vs Fitted: Se utiliza para verificar los supuestos de relación lineal. Una línea horizontal, sin patrones distintos es una indicación de una relación lineal, lo que es bueno.

Normal Q-Q: Útil para examinar si los residuos se distribuyen normalmente. Es bueno si los puntos residuales siguen la línea discontinua recta.

Scale-Location / Spread-Location: Apropiado para verificar la homogeneidad de la varianza de los residuos (homocedasticidad). La línea horizontal con puntos igualmente extendidos es una buena indicación de homocedasticidad.

Residuals vs Leverage: Eficaz para identificar casos influyentes, es decir, valores extremos que pueden influir en los resultados de la regresión. Aquí necesitamos verificar los puntos que están fuera de la línea discontinua. Un punto fuera de la línea discontinua será un punto influyente y su eliminación afectará los coeficientes de regresión.

```
layout(matrix(c(1, 2,
                3, 4),
                        nrow = 2,
                        ncol = 2)
)
plot(modeloLineal)
```

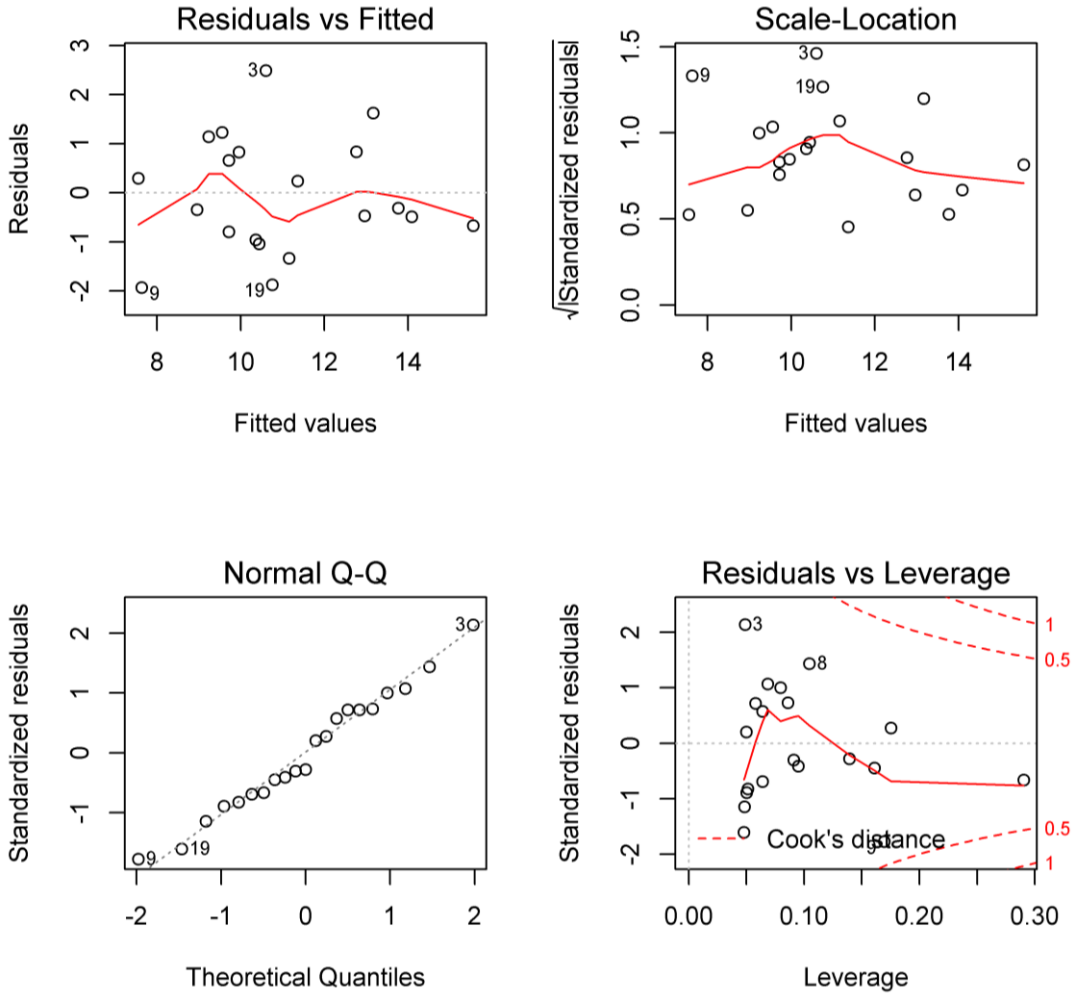


Figura 7.5 *Graficos de significancia para la regresión lineal*

Con todos los criterios expuestos es imprescindible realizar el análisis de la variable predictora del modelo de regresión que se construye para determinar el comportamiento del precio de venta.

Tabla 7.6 *Resultados del análisis gráfico de la variable predictora precioCompra*

Coeficiente	Criterio	Análisis	Conclusión
Residuals vs Fitted	Una línea horizontal, sin patrones distintos es una indicación de una relación lineal	No presenta una lineal horizontal. Sin embargo, se aproxima bien	OK
Normal Q-Q	Es bueno si los puntos residuales siguen la línea discontinua recta	No están exactamente sobre la línea. Sin embargo, su aproximación es muy buena	OK
Scale-Location	La línea horizontal con puntos igualmente extendidos es una buena indicación de homocedasticidad	Están medianamente extendidos	OK
Residuals vs Leverage	Un punto fuera de la línea discontinua será un punto influyente y su eliminación afectará los coeficientes de regresión	No existen puntos fuera de la línea discontinua	OK

No es la intención del texto que usted se tranforme en un experto en regresión lineal sino presentar conceptos básicos relacionado a los resultados que se obtienen mediante el uso de las funciones de R como **scatter.smooth**() y en especial **lm**().

AUTOEVALUACIÓN

Autoevaluación 7-1

Se cree que existe una relación entre el número de parques de diversión de algunas ciudades y el número de niños felices. Grafique el modelo lineal de la relación con la siguiente información:

Parques:	8, 7, 9, 10, 5, 12, 9, 11, 18, 6
Niños felices:	120, 90, 78, 69, 94, 100, 40, 70, 88, 59

Autoevaluación 7-2

El dueño de una concepcionera menciona que hay relación entre la antigüedad de un automóvil y su precio de venta, realice un diagnóstico de la regresión lineal simple a partir de los siguientes datos.

Antigüedad (Años):	9, 11, 7, 9, 15, 18, 5, 9, 10, 13
Precio de Venta (Miles de dólares)	8.1 ,3.6, 5, 7, 4, 10, 12.9, 11.0, 13.2, 5.6

Autoevaluación 7-3

El departamento de producción de una planta procesadora de alimentos sabe que hay una relación entre el número de empleados y la cantidad de unidad producida muestre de manera gráfica el diagnóstico estadístico de las variables. El conjunto completo de observaciones pareadas se muestra a continuación:

Número de empleados	2, 6, 9, 4, 3, 7, 6, 4, 5, 10
Unidades producidas (1hora)	11, 9, 25, 24, 16, 19, 22, 8, 15, 22

EJERCICIOS DEL CAPÍTULO

Se establece cierta relación lineal entre las exportaciones y producción de rosas de manera anual y quinquenal los datos se dan a continuación en miles.

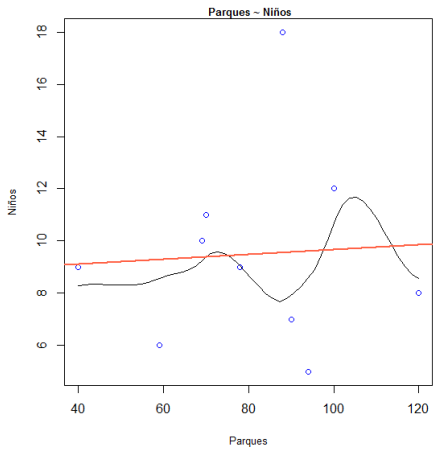
Producción	50.126, 53.190, 62.345, 72.120, 73.010, 50.000, 61.023, 49.235, 55.120, 70.506
Exportación	67.234, 68.345, 74.202, 81.201, 92.432, 61.093, 57.293, 44.287, 62.639, 59.831

1. Realice el gráfico de dispersión de las variables producción y exportación, el color de la curva de ajuste suavizada es orangered1 y los puntos de color darkgreen.
2. Realice el gráfico de densidad sobre la relación entre la producción y la exportación de rosas, pinte la línea de curva normal de color orangered1 y de la densidad color darkgreen.
3. Realice el modelo lineal de la producción y exportación
4. Obtenga el resumen del modelo lineal para la producción y exportación de rosas
5. Muestre de forma gráfica los residuos de las variables de cuatro maneras diferentes.

Capítulo 7. Respuestas de las autoevaluaciones

Autoevaluación 7-1

```
library("knitr")
parques<-c(8,7,9,10,5,12,9,11,18,6)
niños<-c(120,90,78,69,94,100,40,70,88,59)
diversion<-data.frame(parques, niños)
modeloLineal <- lm(parques ~ niños
                   , data= diversion)
par(mar=c(4,4,1,1))
scatter.smooth(x= diversion$niños
               , y= diversion$parques
               , main="Parques ~ Niños"
               , cex.main=0.8
               , xlab="Parques"
               , ylab="Niños"
               , col="blue"
```



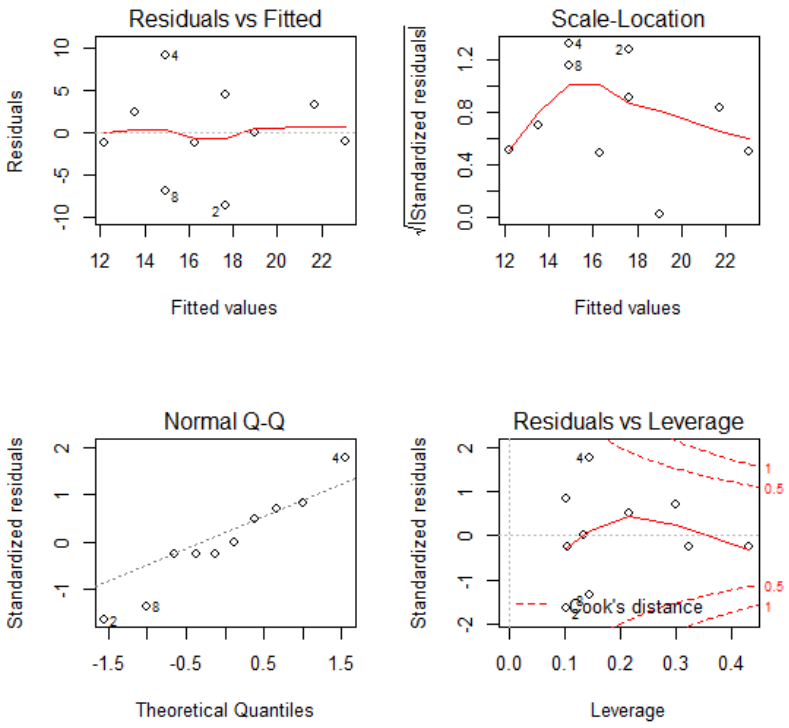
```
                , cex.lab=0.8)
# GRÁFICO DE LA RECTA DE REGRESIÓN LINEAL
abline(modeloLineal, lwd=2, col ="tomato1")
```

Autoevaluación 7-2

```
antiguedad<-c(9,11,7,9,15,18,5,9,10,13)
precio<-c(8.1, 3.6, 5, 7, 4, 10, 12.9, 11.0, 13.2, 5.6)
automoviles<-data.frame(antiguedad,precio)
modeloLineal <- lm(precio ~ antiguedad, data= automoviles)
summary(modeloLineal)
SALIDA EN CONSOLA
Call:
lm(formula = precio ~ antiguedad, data = automoviles)
Residuals:
    Min      1Q  Median      3Q     Max
-4.3390 -2.6550 -0.8941  3.2232  5.0085
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.7172     3.5573   3.013   0.0167 *
antiguedad   -0.2526     0.3174  -0.796   0.4492
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 3.652 on 8 degrees of freedom
Multiple R-squared:  0.07334,   Adjusted R-squared:  -0.04249
F-statistic: 0.6331 on 1 and 8 DF,  p-value: 0.4492
```

Autoevaluación 7-3

```
empleados<-c(2,6,9,4,3,7,6,4,5,10)
unidades<-c(11,9,25,24,16,19,22,8,15,22)
produccion<-data.frame(empleados,unidades)
modeloLineal <- lm(unidades ~ empleados, data= produccion)
layout(matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2))
plot(modeloLineal)
```

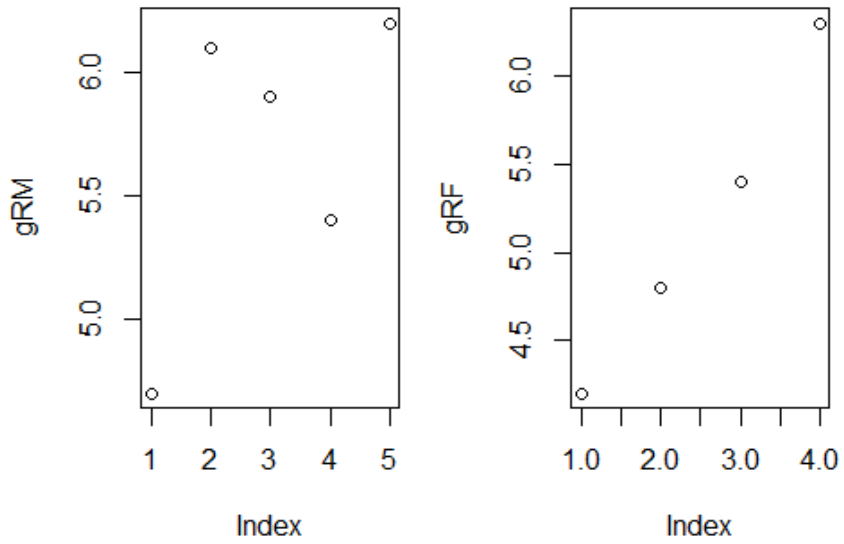


RESPUESTA A LOS EJERCICIOS DE CADA CAPÍTULO

CAPÍTULO I

Ejercicio 1

```
# DIVIDIR LA SALIDA GRÁFICA
par(mfrow=c(1,2), mar=c(5,4,3,1))
gRM<-c(4.7, 6.1, 5.9, 5.4, 6.2)
# VISUALIZAR LOS DATOS DEL VECTOR
gRM
## [1] 4.7 6.1 5.9 5.4 6.2
plot(gRM)
gRF<-c(4.2, 4.8, 5.4, 6.3)
# VISUALIZAR LOS DATOS DEL VECTOR
gRF
## [1] 4.2 4.8 5.4 6.3
plot(gRF)
```



Ejercicio 2

```
analisisSangre<-matrix(c(4.1, 6.2, 104, 11.0, 78, 4.9, 6.4, 202, 15.7, 72, 5.
6, 7.7, 287, 14.2, 89, 5.9, 7.8, 314, 13.5, 57, 4.6, 6.6, 215, 13.9, 70,7.0,8
.1,340,15.7,60, 6.1, 9.9, 302, 13.5, 71), nrow=7, ncol=5, byrow=T)
colnames(analisisSangre)<-c("gR","gB","plq","hgl","hto")
rownames(analisisSangre)<-c("Rosa","Rocío","Ruperta","Rolando","Rubén","Romar
io","Ruber")
MAXIMOS<-apply(analisisSangre, 2, max)
MINIMOS<-apply(analisisSangre, 2, min)
MEDIAS<-apply(analisisSangre, 2, mean)
analisisSangre<-rbind(analisisSangre,MAXIMOS)
analisisSangre<-rbind(analisisSangre,MEDIAS)
analisisSangre<-rbind(analisisSangre,MINIMOS)
analisisSangre
##               gR       gB plq      hgl hto
## Rosa    4.100000 6.200000 104 11.00000  78
## Rocío   4.900000 6.400000 202 15.70000  72
## Ruperta 5.600000 7.700000 287 14.20000  89
## Rolando 5.900000 7.800000 314 13.50000  57
```

```
## Rubén   4.600000 6.600000 215 13.90000  70
## Romario 7.000000 8.100000 340 15.70000  60
## Ruber   6.100000 9.900000 302 13.50000  71
## MAXIMOS 7.000000 9.900000 340 15.70000  89
## MEDIAS  5.457143 7.528571 252 13.92857  71
## MINIMOS 4.100000 6.200000 104 11.00000  57
```

Ejercicio 3

```
analisisSangre<-matrix(c(4.1, 6.2, 104, 11.0, 78, 4.9, 6.4, 202, 15.7, 72, 5.
6, 7.7, 287, 14.2, 89 ,5.9, 7.8, 314, 13.5, 57, 4.6, 6.6, 215, 13.9, 70, 7.0,
8.1, 340, 15.7, 60, 6.1, 9.9, 302, 13.5, 71),nrow=7,ncol=5,byrow=T)
colnames(analisisSangre)<-c("gR","gB","plq","hgl","hto")
rownames(analisisSangre)<-c("Rosa","Rocío","Ruperta","Rolando","Rubén","Romar
io","Ruber")
filas<-c("Rosa","Romario","Ruber")
analisisSangre[filas,]
##          gR  gB plq  hgl hto
## Rosa    4.1 6.2 104 11.0  78
## Romario 7.0 8.1 340 15.7  60
## Ruber   6.1 9.9 302 13.5  71
analisisSangre
##          gR  gB plq  hgl hto
## Rosa    4.1 6.2 104 11.0  78
## Rocío   4.9 6.4 202 15.7  72
## Ruperta 5.6 7.7 287 14.2  89
## Rolando 5.9 7.8 314 13.5  57
## Rubén   4.6 6.6 215 13.9  70
## Romario 7.0 8.1 340 15.7  60
## Ruber   6.1 9.9 302 13.5  71
```

Ejercicio 4

```
sexo<-c("F", "M", "F", "M", "F", "M", "M")
gR<-c(4.1, 4.7, 5.2, 4.9, 4.6, 6.1, 5.9)
gB<-c(5.8, 4.9, 7.3, 7.8, 6.6, 8.1, 9.9)
plq<-c("102", "200", "206", "314", "215", "340", "302")
hgl<-c(13.0, 16.7, 15.2, 14.5, 13.8, 15.7, 13.9)
hto<-c(80, 70, 100, 50, 70, 60, 70)
examenes<-data.frame(sexo, gR, gB, plq, hgl, hto)
str(examenes)
## 'data.frame':    7 obs. of  6 variables:
##  $ sexo: chr  "F" "M" "F" "M" ...
##  $ gR  : num  4.1 4.7 5.2 4.9 4.6 6.1 5.9
##  $ gB  : num  5.8 4.9 7.3 7.8 6.6 8.1 9.9
##  $ plq : chr  "102" "200" "206" "314" ...
##  $ hgl : num  13 16.7 15.2 14.5 13.8 15.7 13.9
##  $ hto : num  80 70 100 50 70 60 70
```

Ejercicio 5

```
library(XLConnect)
setwd("C:/Users/Ariosto/Desktop")
libroExcel=loadWorkbook("analisis.xlsx")
analisisXls=readWorksheet(libroExcel, sheet="Datos")
head(analisisXls, 7)
str(examenesSangre)
examenSangreXls$gR<-as.numeric(examenSangreXls$gR)
```

```
examenSangreXls$gB<-as.numeric(examenSangreXls$gB)
examenSangreXls$hgl<-as.numeric(examenSangreXls$hgl)
```

CAPÍTULO II

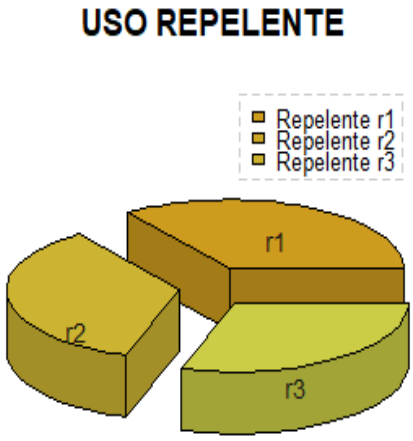
Ejercicio 1

```
# DIVIDIR LA SALIDA GRÁFICA
par(mfrow=c(1,2), mar=c(5,4,3,0))
# a
servicio1<-c(8, 8, 3, 1, 9, 7 , 5, 5, 12, 7, 11, 9)
servicio2<-c(12, 11, 10, 6, 8, 9, 9, 10, 11, 9, 8, 10)
servicios<-data.frame(servicio1, servicio2)
barplot(table(servicios$servicio1,servicios$servicio2)
        , main = "Servicios"
        , xlab="Servicio"
        , ylab="Frecuencias Absolutas"
        , col=c("orange","royalblue")
        ,legend.text = c("Servicio 1", "Servicio 2")
        , ylim = c(0,10)
        , beside=TRUE)
# b
barplot(table(servicios$servicio1,servicios$servicio2)
        , main = "Servicios"
        , xlab="Frecuencias Absolutas"
        , ylab="Servicio"
        , col=c("yellow","magenta")
        , legend.text = c("Servicio 1", "Servicio 2")
        , xlim=c(0,10)
        , horiz =TRUE)
```

Ejercicio 2

```
library(plotrix)
# Salida gráfica
par(mar=c(0,1,3,0))
repelentes<-c("r1","r2","r3","r1","r1","r3","r2","r2","r3","r1","r2","r1","r1
","r3","r2","r2","r3","r1","r2","r3")
fa.c<-table(repelentes)
```

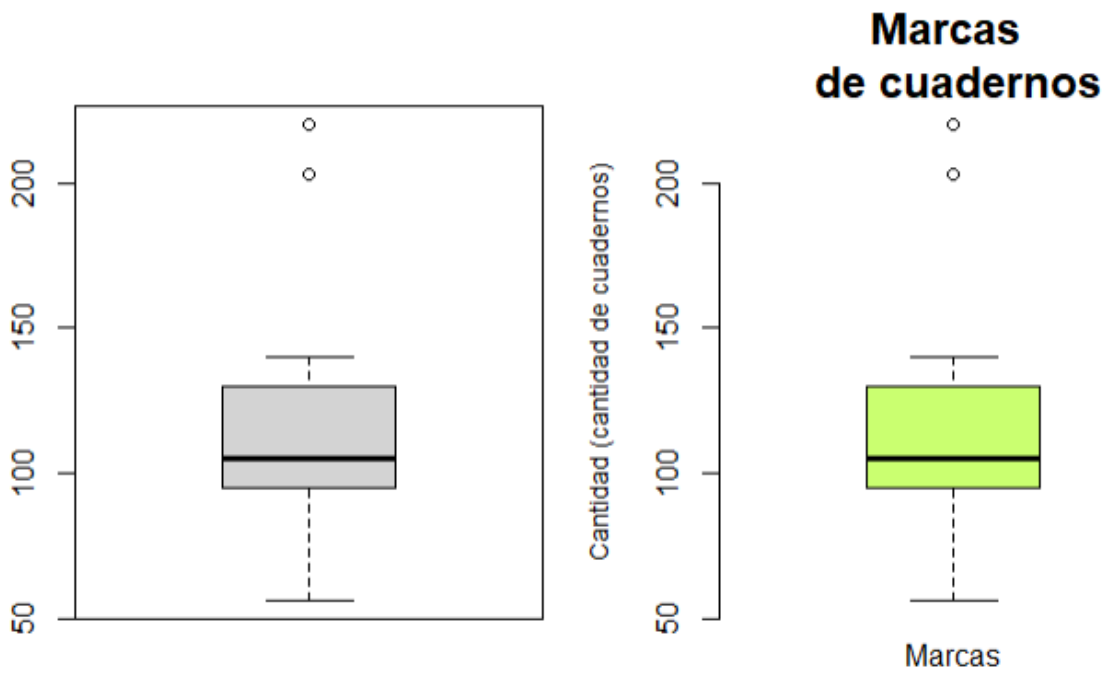
```
fr.c<-prop.table(fa.c)
escalaGrises<-colorRampPalette(c("goldenrod3", "darkolivegreen1 "))
etq<-paste(names(fr.c))
paste(round(fr.c*100, 2), "%", sep=" ")
## [1] "35 %" "35 %" "30 %"
gp<-pie3D(fr.c
          , main = "USO REPELENTE"
          , radius = 1.0
          , height=0.15
          , explode = 0.2
          , col= escalaGrises(5) )
# b
pie3D.labels(gp
             , labels=etq
             , labelcex=0.9
             , labelrad=0.8
             , labelcol="gray8")
legend("topright"
       , legend=c("Repelente r1", "Repelente r2", "Repelente r3")
       , col=escalaGrises(10)
       , fill=escalaGrises(10)
       , cex=0.8
       , box.lty=1250
       , box.lwd=1
       , box.col="gray80")
```



Ejercicio 3

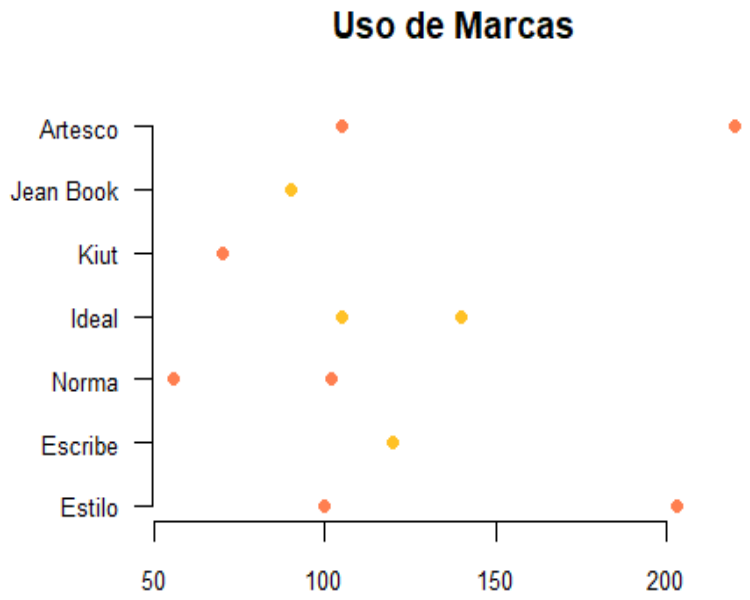
```
# DIVIDIR LA SALIDA GRÁFICA
par(mfrow=c(1,2), mar=c(5,4,3,1))
# a
marca<-c("Estilo","Escribe","Norma","Ideal","Kiut","JeanBook","Artesco","Esti
lo","Norma","Ideal","Artesco")
cantidad<-c(56, 120, 220, 140, 90, 70, 100, 102, 105, 105, 203)
usomarca<-data.frame(marca,cantidad)
boxplot(cantidad, data=usomarca)
# b
boxplot(cantidad
```

```
        , data=usomarca
        , main= "Marcas \n de cuadernos"
        , cex.main=1.5
        , ylab="Cantidad (cantidad de cuadernos)"
        , cex.lab=0.9
        , col="darkolivegreen1"
        , frame.plot=FALSE)
mtext(side=1, line=0.5, "Marcas")
```



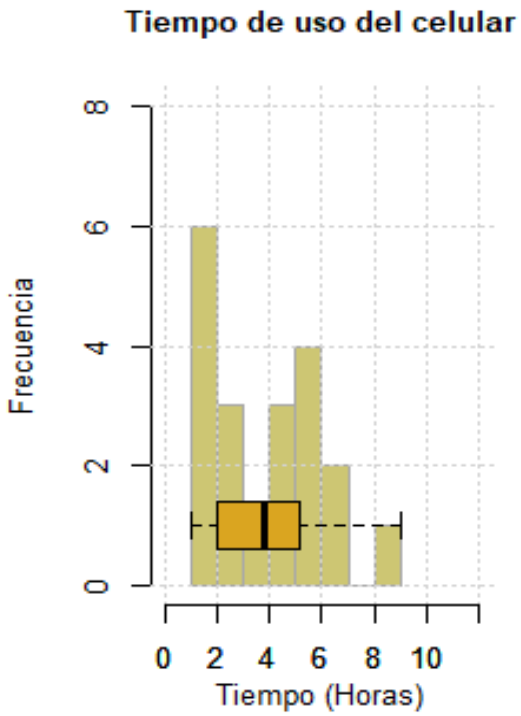
Ejercicio 4

```
par(mar=c(5,6,3,1))
marca<-c("Estilo", "Escribe", "Norma", "Ideal"
         , "Kiut", "JeanBook", "Artesco"
         , "Estilo", "Norma", "Ideal", "Artesco")
cantidad<-c(56,120,220,140,90,70,100,102,105,105,203)
usomarca<-data.frame(marca,cantidad)
stripchart( cantidad ~ marca
            , data = usomarca
            , main="Uso de Marcas"
            , xlab=""
            , cex.axis=0.8
            , group.names=c("Estilo","Escribe","Norma","Ideal"
                            ,"Kiut","Jean Book","Artesco")
            , las=1
            , frame.plot = FALSE
            , vertical=FALSE
            , method = "stack"
            , offset=0.5
            , pch=16
            , col=c("coral", "goldenrod1")
)
```



Ejercicio 5

```
# DIVIDIR LA SALIDA GRÁFICA
par(mfrow=c(1,1), mar=c(5,5,4,1))
tiempo<-c(1, 1.5, 2.3, 5.2, 6.7, 2.3,
    5.2, 6, 9, 1.6, 4.4, 2, 3, 1.2,
    4.3,   5, 3.3, 7, 5.2, 2)
layout(matrix(c(1,2,1,3), 2, 2
     , byrow = TRUE)
     , widths=c(2,2)
     , heights=c(3.5,2.2))
# HISTOGRAMA
hist(tiempo
     , main="Tiempo de uso del      c
elular"
     , cex.main=1
     , xlab=""
     , ylab = "Frecuencia"
     , col=" khaki3"
     , xlim=c(0,12)
     , ylim=c(0,8)
     , border = "darkgrey")
grid()
# BOXPLOT DENTRO DEL HISTOGRAMA
boxplot(tiempo
        , horizontal=TRUE
        , frame=F
        , col = " goldenrod"
        , width = 10
        , add=TRUE)
# TITULO DEL EJE X
mtext(side=1
      , line=2.1
      , "Tiempo (Horas)"
      , cex = 0.8)
```



Capítulo III

Ejercicio 1

```
# DIVIDIR LA SALIDA GRÁFICA
par(mfrow=c(1,2), mar=c(5,4,3,1))
curson1<-c(50,27,38,44,20,35,25,19,55,29,44,19,20,25,30,38)
curson2<-c(34,32,50,32,29,27,26,19,29,55,20,44,42,38,32,49)
cursoestadistica<-data.frame(curson1,curson2)
CE1<-round(mean(cursoestadistica$curson1), 2) # Calculo de la media
CE2<-round(mean(cursoestadistica$curson2), 2) # Calculo de la media
# AGRUPACIÓN DE RESULTADOS
M<-data.frame(curson1=CE1, curson2=CE2)
rownames(M)=c("Media")
#-----------------------------
# Mediana median()
#-----------------------------
Me1<-round(median(cursoestadistica$curson1), 2) # Calculo de la mediana
Me2<-round(median(cursoestadistica$curson2), 2) # Calculo de la mediana
# AGRUPACIÓN DE RESULTADOS
Me<-data.frame(curson1=Me1, curson2=Me2)
rownames(Me)=c("Mediana")
#-----------------------------
# Moda
#-----------------------------
# FUNCIÓN PARA CALCULAR LA MODA.
moda <- function(dataframe) {
uvalor <- unique(dataframe)
uvalor[which.max(tabulate(match(dataframe, uvalor)))]
}
Mo1<-round(moda(cursoestadistica$curson1), 2) # Calculo de la moda
Mo2<-round(moda(cursoestadistica$curson2), 2) # Calculo de la moda
# AGRUPACIÓN DE RESULTADOS
Mo<-data.frame(curson1=Mo1, curson2=Mo2)
rownames(Mo)=c("Moda")
# RESUMEN DE LAS MEDIDAS DE CENTRALIDAD
centralidad<-rbind(M, Me, Mo)
head(centralidad)
##         curson1 curson2
## Media     32.38   34.88
## Mediana   29.50   32.00
## Moda      38.00   32.00
```

Ejercicio 2

```
curson1<-c(50, 27,38,44,20,35,25,19,55,29,44,19,20,25,30,38)
curson2<-c(34,32,50,32,29,27,26,19,29,55,20,44,42,38,32,49)
r1=round(max(cursoestadistica$curson1)-min(cursoestadistica$curson1), 2)
r2=round(max(cursoestadistica$curson2)-min(cursoestadistica$curson2), 2)
Rangos<-cbind(curson1=r1, curson2=r2)
rownames(Rangos)=c("Rango")
#-----------------------------
# Varianza var()254
#-----------------------------
var1=round(var(cursoestadistica$curson1), 2)
var2=round(var(cursoestadistica$curson2), 2)
```

```
Varianzas<-cbind(curson1=var1, curson2=var2)
rownames(Varianzas)=c("Varianza")
#-----------------------------
# Desviación Estándar sd()
#-----------------------------
sd1=round(sd(cursoestadistica$curson1), 2)
sd2=round(sd(cursoestadistica$curson2), 2)
desv.Stand<-cbind(curson1=sd1, curson2=sd2)
rownames(desv.Stand)=c("SD")
#-----------------------------
# Coeficiente de variación
#-----------------------------
cv<-function(ds){
media<-mean(ds)
desv.stand<-sd(ds)
cv<-round(desv.stand/media,2)
return(cv)
}
cv1=round(cv(cursoestadistica$curson1), 2)
cv2=round(cv(cursoestadistica$curson2), 2)
Coef.variacion<-cbind(curson1=cv1, curson2=cv2)
rownames(Coef.variacion)=c("Coef.variacion")
#-----------------------------
# Medidas de dispersión
#-----------------------------
dispersion<-rbind(Rangos, Varianzas, desv.Stand, Coef.variacion)
head(dispersion)
##                curson1 curson2
## Rango            36.00   36.00
## Varianza        132.12  112.38
## SD               11.49   10.60
## Coef.variacion    0.36    0.30
```

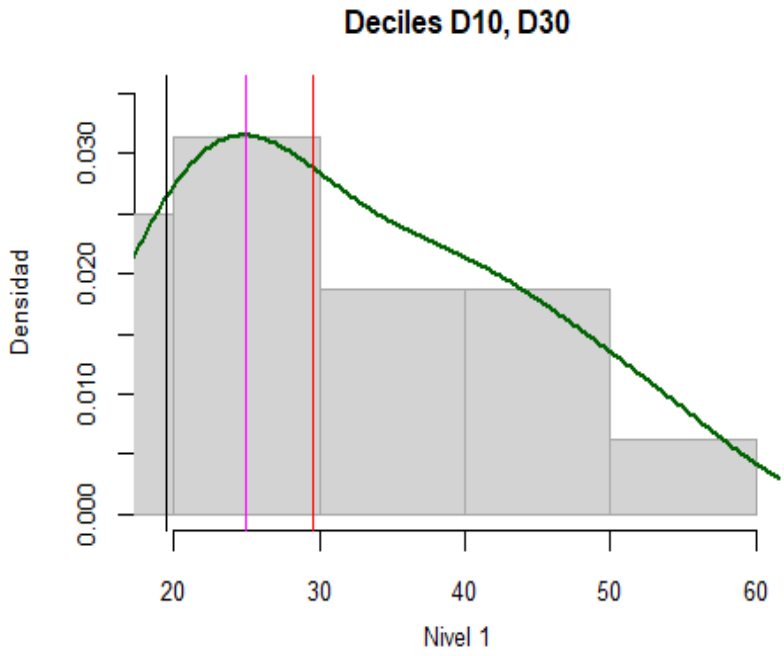
Ejercicio 3

```
curson1<-c(50, 27,38,44,20,35,25,19,55,29,44,19,20,25,30,38)
curson2<-c(34,32,50,32,29,27,26,19,29,55,20,44,42,38,32,49)
qt1<-quantile (cursoestadistica$curson2, prob = c(0.25, 0.50, 0.75))
print("CUARTILES")
## [1] "CUARTILES"
qt1
##  25%  50%  75%
## 28.5 32.0 42.5
dc1<-quantile (cursoestadistica$curson2, prob = c(0.10, 0.20, 0.30))
print("DECILES")
## [1] "DECILES"
dc1
## 10% 20% 30%
##  23  27  29
pc3<-quantile (cursoestadistica$curson2, prob = c(0.01, 0.02, 0.03))
print("PERCENTILES")
## [1] "PERCENTILES"
pc3
##    1%    2%    3%
## 19.15 19.30 19.45
print("RANGO INTERCUARTILICO")
## [1] "RANGO INTERCUARTILICO"
```

```
IQR(cursoestadistica$curson2)
## [1] 14
```

Ejercicio 4

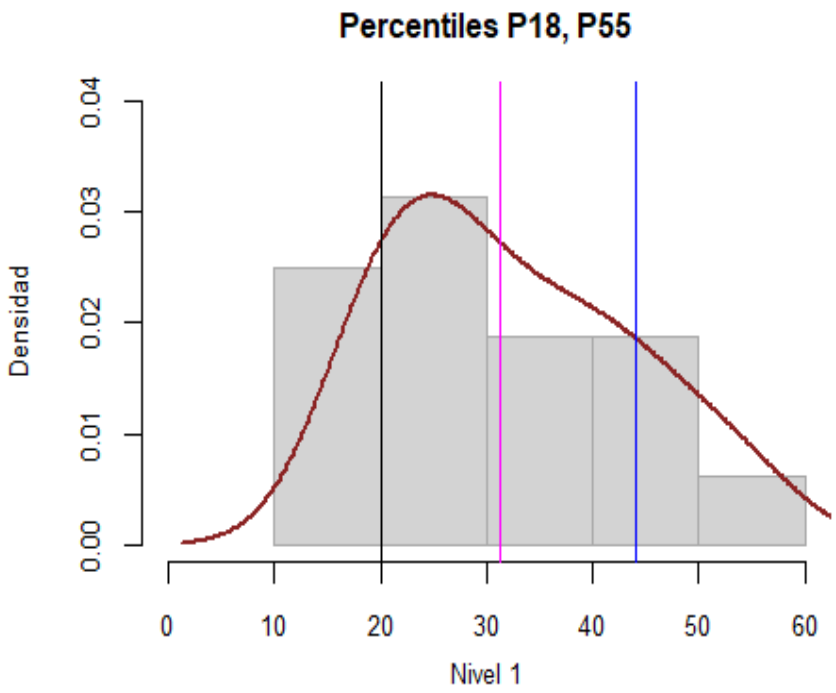
```
par(mfrow=c(1,1), mar=c(5,5,3,1))
curson1<-c(50, 27,38,44,20,35,25,19,55,29,44,19,20,25,30,38)
hist(cursoestadistica$curson1
     , probability = TRUE
     , main = "Deciles D10, D30"
     , cex.main=1, xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
     , xlim = c(19,60)
     , ylim = c(0.000, 0.035)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Nivel 1", cex=0.8)
lines(density(cursoestadistica$curson1), col="darkgreen", lwd=2)
d<-quantile(cursoestadistica$curson1, probs = c(0.10, 0.30, 0.50))
abline(v=d[1], col="black", lty=1, lwd=1)
abline(v=d[2], col="magenta", lty=1, lwd=1)
abline(v=d[3], col="red", lty=1, lwd=1)
```



Ejercicio 5

```
par(mfrow=c(1,1), mar=c(5,5,3,1))
curson1<-c(50, 27,38,44,20,35,25,19,55,29,44,19,20,25,30,38)
hist(cursoestadistica$curson1
     , probability = TRUE
     , main = "Percentiles P18, P55"
     , cex.main=1, xlab = ""
     , ylab= "Densidad"
     , cex.lab=0.8
     , cex.axis=0.8
```

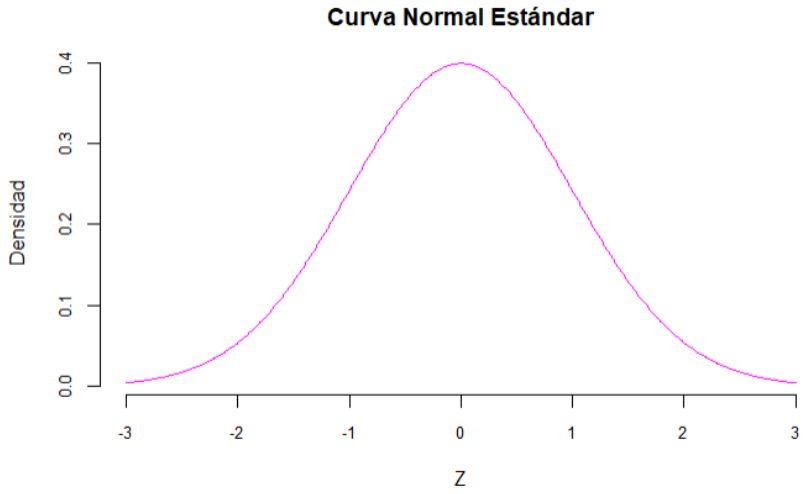
```
     , xlim = c(0,60)
     , ylim = c(0.000, 0.040)
     , border = "darkgrey"
     , plot = TRUE)
mtext(side=1, line=2.2, "Nivel 1", cex=0.8)
# FUNCIÓN DENSIDAD
lines(density(cursoestadistica$curson1), col=" brown4 ", lwd=2)
# CÁLCULO DE PERCENTILES
p<-quantile(cursoestadistica$curson1, probs = c(0.18, 0.55, 0.84))
abline(v=p[1], col="black", lty=1, lwd=1)
abline(v=p[2], col="magenta", lty=1, lwd=1)
abline(v=p[3], col="blue", lty=1, lwd=1)
```



Capítulo IV

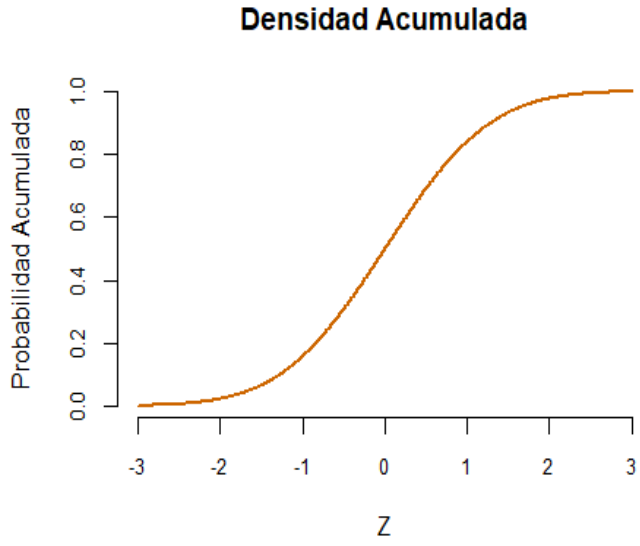
Ejercicio 1

```
par(mfrow=c(1,1), mar=c(5,5,3,1))
datos<-seq(-3,3,.01)
# SE CREA LOS VALORES DE DENSIDAD PARA LA VARIABLE ALEATORIA
Normal.densidad<-dnorm(datos, 0,1)
# GRÁFICO DE LA FUNCIÓN NORMAL
plot(datos
     , Normal.densidad
     , main="Curva Normal Estándar"
     , cex.main=0.8
     , cex=2
     , col="magenta"
     , xlab="Z"
     , ylab="Densidad"
     , cex.axis=.8
     , frame.plot = FALSE
     , type="l"
     , lwd=1)
```



Ejercicio 2

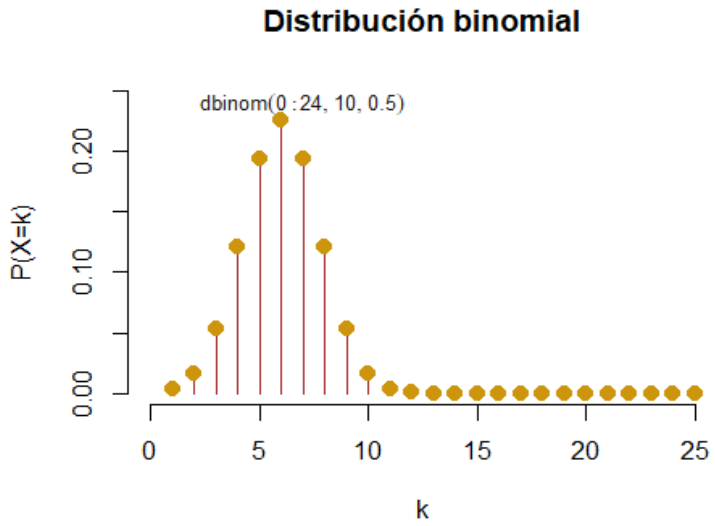
```
par(mfrow=c(1,1), mar=c(5,5,4,1))
datos<-seq(-3,3,.01)
# SE CREA LOS VALORES DE DENSIDAD PARA LA VARIABLE ALEATORIA
Normal.acumulada<-pnorm(datos, 0, 1)
# GRÁFICO DE LA FUNCIÓN NORMAL ACUMULADA
plot(datos
     , Normal.acumulada
     , main="Densidad Acumulada"
     , cex.main=1.2
     , col=" darkorange3"
     , xlab="Z"
     , ylab="Probabilidad Acumulada"
     , cex=2, cex.axis=.8
     , frame.plot = FALSE
     , type="l"
     , lwd=2)
```



Ejercicio 3

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
x<-seq(0:24)
```

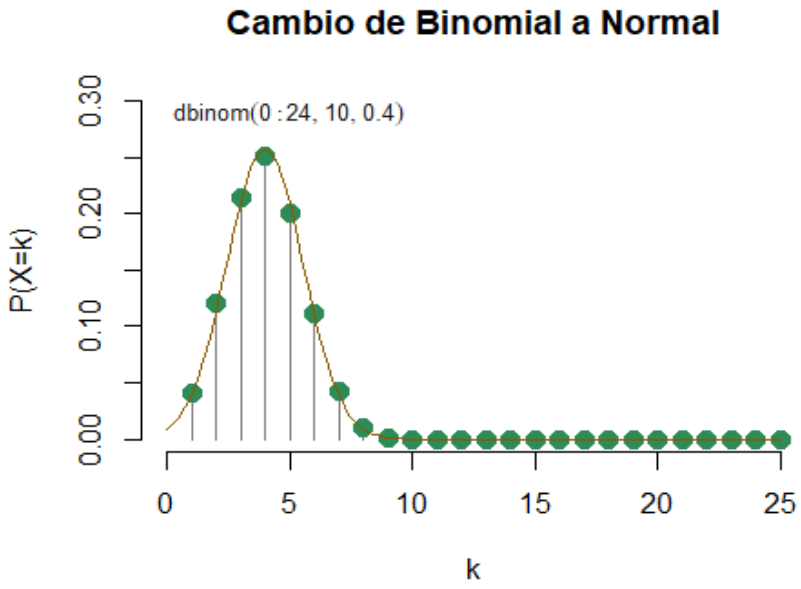
```
# CREAR LA DISTRIBUCIÓN BINOMIAL
y<-dbinom(x,12,0.5)
#VISUALIZAR LA DISTRIBUCIÓN BINOMIAL
plot(x,y
     , main="Distribución binomial"
     , xlab="k"
     , ylab="P(X=k)"
     , xlim=c(0, 25)
     , ylim=c(0,0.25)
     , type = "h"
     , lwd=1
     , lty=1
     , col="brown"
     , frame=F)
points(x,y, cex=1.4, col="darkgoldenrod3", pch=19)
text(7,0.24,expression(dbinom(x=0:24, size=10, prob = 0.5)), col="gray10", ce
x=0.8)
```



Ejercicio 4

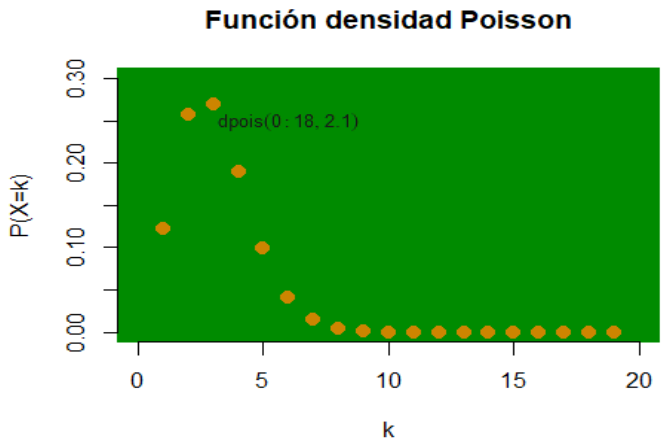
```
par(mfrow=c(1,1), mar=c(5,5,4,1))
x<-seq(0:24)
# CREAR LA DISTRIBUCIÓN BINOMIAL
y<-dbinom(x,10,0.4)
# VISUALIZAR LA DISTRIBUCIÓN BINOMIAL
plot(x,y
     , main="Cambio de Binomial a Normal"
     , xlab="k"
     , ylab="P(X=k)"
     , xlim=c(0, 25)
     , ylim=c(0,0.30)
     , type = "h"
     , lwd=1
     , lty=1
     , col="gray50"
     , frame=F)
points(x,y, cex=1.5, col="seagreen4", pch=19)
media = 10*0.4
desviacionEstandar =sqrt(10*0.4*0.6)
curve(dnorm(x,media,desviacionEstandar), lwd=1, lty=1, col="orange4", add=T)
```

```
text(5,0.29,expression(dbinom(x=0:24, size=10, prob = 0.4)), col="gray10", ce
x=0.8)
```



Ejercicio 5

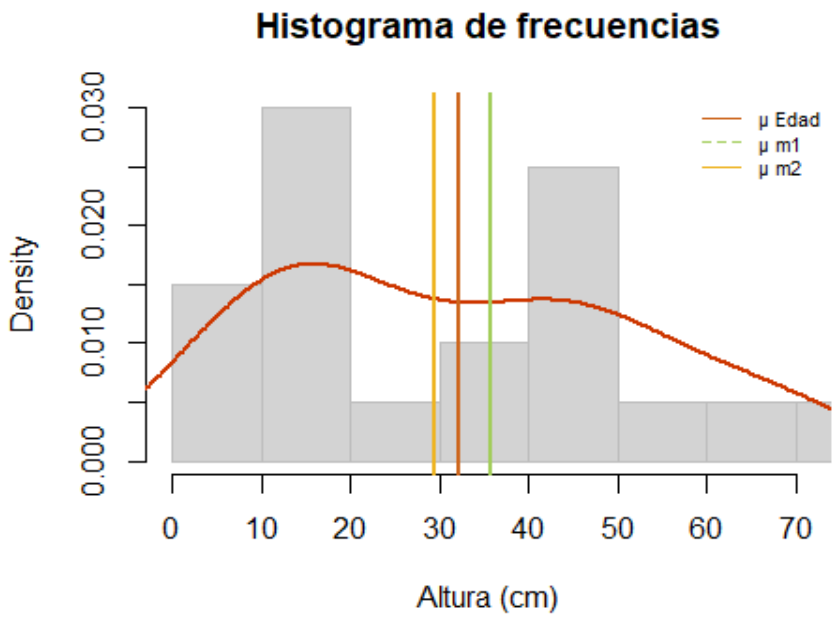
```
par(mfrow=c(1,1), mar=c(5,5,4,1))
x<-seq(0:18)
# CREAR LA DISTRIBUCIÓN POISSON
y<-dpois(0:18, 2.1)
# VISUALIZAR LA DISTRIBUCIÓN POISSON
plot(x,y
     , main="Función densidad Poisson"
     , xlab="k"
     , ylab="P(X=k)"
     , xlim=c(0, 20)
     , ylim=c(0,0.30)
     , type = "h"
     , lwd=1261
     , lty=1
     , col=" green4"
     , frame=F)
points(x,y, cex=1.4, col=" orange3", pch=19)
text(6,0.25,expression(dpois(0:18, 2.1)), col="gray10", cex=0.8)
```



Capítulo V

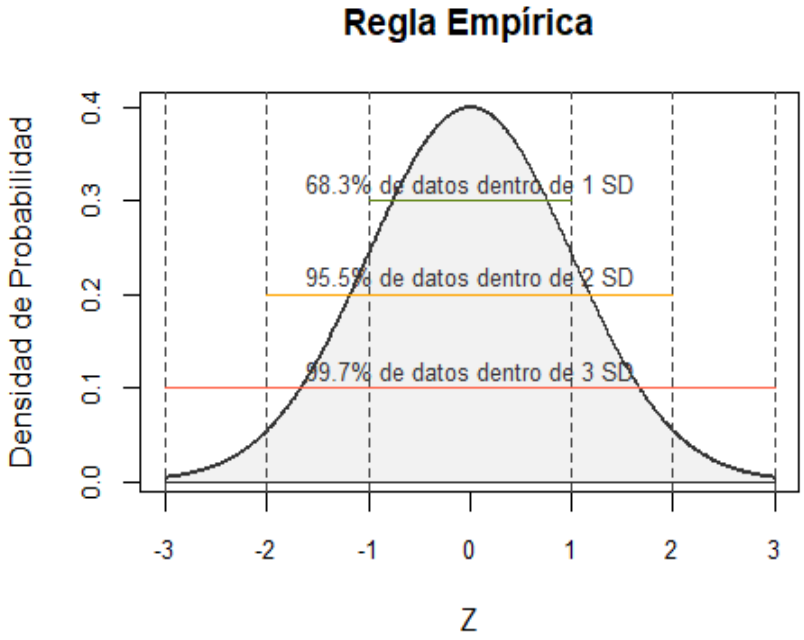
Ejercicio 1

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
Edad<-c(12,45,50,60,11,7,3,9,42,28,19,17,32,39,50,18,43,66,71,20)
# MUESTRAS TOMADAS EN EL PARALELO
m1<-c(32,50,71,50,3,9)
m2<-c(12,42, 17,66,20,19)
# MEDIA POBLACIONAL
xEdad<-mean(Edad)
# MEDIAS MUESTALES
Xm1<-mean(m1)
Xm2<-mean(m2)
medias<-data.frame(Mp= xEdad, Mm1=Xm1, Mm2=Xm2)
head(medias)
##     Mp      Mm1      Mm2
## 1 32.1 35.83333 29.33333
hist(Edad
     , probability = TRUE
     , main="Histograma de frecuencias"
     , xlab = "Altura (cm)"
     , xlim = c(0, 71)
     , border = "gray")
lines(density(Edad), col=" orangered3", lwd=2)
abline(v=mean(Edad), col=" chocolate3", lwd=2)
abline(v=mean(m1), col=" darkolivegreen3 ", lwd=2)
abline(v=mean(m2), col=" goldenrod2 ", lwd=2)
legend("topright"
       , legend=c("μ Edad","μ m1", "μ m2")
       , col=c("chocolate3"," darkolivegreen3 ", " goldenrod2 ")
       , lty=1:2
       , cex=0.7
       , bty = "n")
```



Ejercicio 2

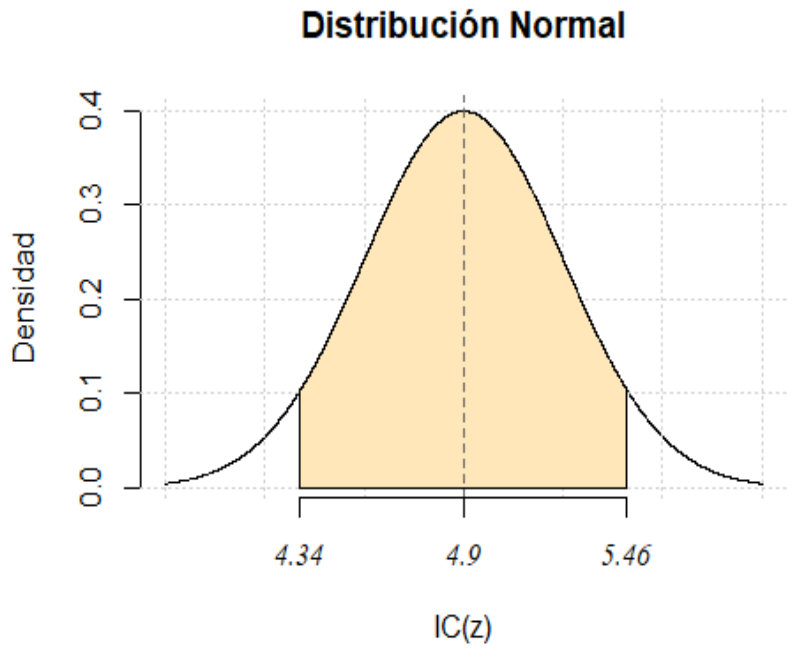
```
par(mfrow=c(1,1), mar=c(5,5,4,1))
# DISTRIBUCIÓN NORMAL
set.seed(3345)
variable.aleatoria.x<-seq(-3,3,0.01)
Normal.densidad<-dnorm(variable.aleatoria.x, 0, 1)
plot(variable.aleatoria.x
     , Normal.densidad
     , col="gray20"
     , xlab="Z"
     , ylab="Densidad de Probabilidad"
     , type="l"
     , lwd=2
     , cex=2
     , main="Regla Empírica"
     , cex.axis=.8
     , xlim = c(-3,3))
# COLOREAR EL ÁREA DE LA CURVA NORMAL
polygon(c(-3,variable.aleatoria.x,3)
        , c(0,Normal.densidad,0)
        , col="gray95"
        , border = "gray20")
# CALCULAR LA MEDIA POBLACIONAL
media<-mean(variable.aleatoria.x)
# REGLA EMPIRICA 1SD / 68.3%263
abline(v=media-1, col="gray30", lty=2)
abline(v=media+1, col="gray30", lty=2)
x<-c(media-1, media+1)
y<-c(0.3, 0.3)
lines(x,y, lwd=1, col="olivedrab4")
text(x = 0, y = 0.32
     , "68.3% de datos dentro de 1 SD"
     , col = "gray20"
     , cex = 0.8)
# REGLA EMPIRICA 2SD / 95.5%
abline(v=media-2, col="gray30", lty=2)
abline(v=media+2, col="gray30", lty=2)
x<-c(media-2, media+2)
y<-c(0.2, 0.2)
lines(x,y, lwd=1, col="orange")
text(x = 0, y = 0.22
     , "95.5% de datos dentro de 2 SD"
     , col = "gray20"
     , cex = 0.8)
# REGLA EMPIRICA 3SD / 99.7%
abline(v=media-3, col="gray30", lty=2)
abline(v=media+3, col="gray30", lty=2)
x<-c(media-3, media+3)
y<-c(0.1, 0.1)
lines(x,y, lwd=1, col="tomato")
text(x = 0, y = 0.12
     , "99.7% de datos dentro de 3 SD"
     , col = "gray20"
     , cex = 0.8)
```



Ejercicio 3

```
# a
# DATOS
Xm=4.9
SDp=2.01
n=35
CC=0.90
# CALCULAR EL NIVEL DE SIGNIFICANCIA
alfa=1-CC
# CALCULAR Z
Z<-qnorm(alfa/2, mean = 0, sd = 1, lower.tail = FALSE)
# CALCULAR SD
SD<-SDp/sqrt(n)
# CALCULAR EL LIMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Z*SD, 2)
# CALCULAR EL LIMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Z*SD,2)
# PRENSENTACION DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
head(IC)
##     LI   LS
## 1 4.34 5.46
# b
# GRÁFICO DE LA CURVA NORMAL
par(mfrow=c(1,1), mar=c(5,5,4,1))
x <- seq(-3, 3, by=0.001)
y <- dnorm(x)
plot(x, y
     , xaxt = "n"
     , type = "l"
     , main = "Distribución Normal"
     , xlab= "IC(z)"
     , ylab="Densidad"
     , frame=F)
```

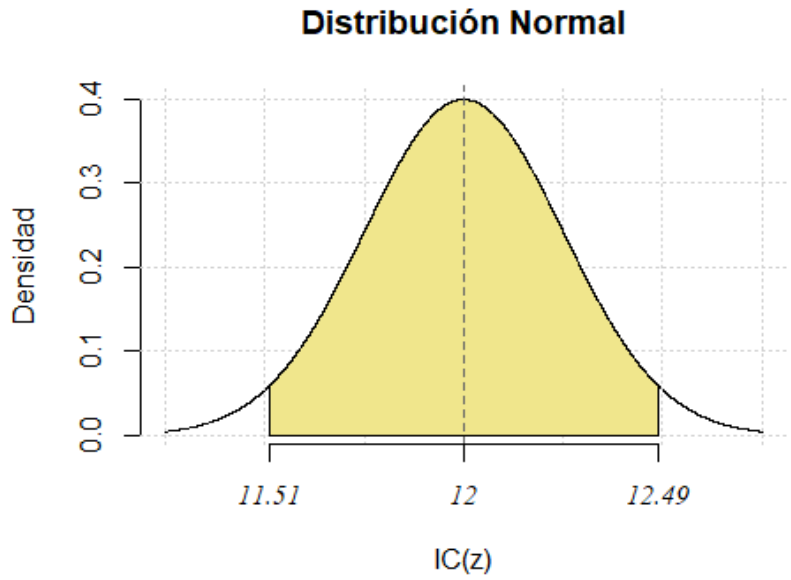
```
grid()
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-Z, Z, by=0.001)
y <- dnorm(x)
# GRÁFICO DE LA REGIÓN SOMBREADA
polygon(c(-Z,x,Z), c(0,y,0), col=" wheat1")
# LÍNEA CENTRAL
abline(v=0, lty=2, col="gray40")
# VALORES DEL INTERVALO
axis(1
     , at = c(-1*Z, 0, 1*Z)
     , font = 8
     , cex=0.8
     , labels = c(li.IC,Xm, ls.IC))
```



Ejercicio 4

```
# a
# DATOS
Xm=12
SDp=6
n=576
CC=0.95
# CALCULAR EL NIVEL DE SIGNIFICANCIA
alfa=1-CC
# CALCULAR Z
Z<-qnorm(alfa/2, mean = 0, sd = 1, lower.tail = FALSE)
# CALCULAR SD
SD<-SDp/sqrt(n)
# CALCULAR EL LIMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Z*SD, 2)
# CALCULAR EL LIMITE INFERIOR DE CONFIANZA
li.IC=round(Xm-Z*SD,2)
# PRENSENTACION DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
head(IC)
```

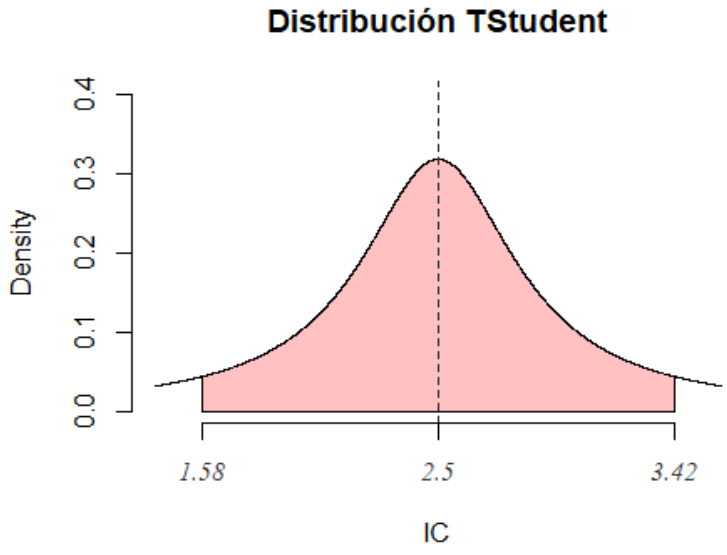
```
##      LI    LS
## 1 11.51 12.49
# b
par(mfrow=c(1,1), mar=c(5,5,4,1))
# GRÁFICO DE LA CURVA NORMAL
x <- seq(-3, 3, by=0.001)
y <- dnorm(x)
plot(x, y
     , xaxt = "n"
     , type = "l"
     , main = "Distribución Normal"
     , xlab= "IC(z)"
     , ylab="Densidad"
     , frame=F)
grid()
# GRÁFICO DEL INTERVALO DE CONFIANZA
x <- seq(-Z, Z, by=0.001)
y <- dnorm(x)
# GRÁFICO DE LA REGIÓN SOMBREADA
polygon(c(-Z,x,Z), c(0,y,0), col="khaki ")
# LÍNEA CENTRAL
abline(v=0, lty=2, col="gray40")
# VALORES DEL INTERVALO
axis(1
     , at = c(-1*Z, 0, 1*Z)
     , font = 8
     , cex=0.8
     , labels = c(li.IC,Xm, ls.IC))
```



Ejercicio 5

```
# DATOS
n=24
Xm=2.5
SD=1.8
CC=98/100
GL=n-1
# CALCULAR EL NIVEL DE SIGNIFICANCIA ALFA
```

```r
alfa=1-CC
alfaMedio=alfa/2
# CALCULAR Z
Tstudent<-qt(1-alfaMedio, GL)
# CALCULAR EL LIMITE SUPERIOR DE CONFIANZA
ls.IC=round(Xm+Tstudent*SD/sqrt(n), 2)
# CALCULAR EL LIMITE INFERIOR DE CONFIANZA267
li.IC=round(Xm-Tstudent*SD/sqrt(n),2)
# PRESENTACIÓN DEL INTERVALO DE CONFIANZA
IC<-data.frame(LI=li.IC, LS=ls.IC)
head(IC)
##     LI   LS
## 1 1.58 3.42
# b
par(mfrow=c(1,1), mar=c(5,5,4,1))
x <- seq(-3,3, by=0.001)
y <- dt(x, 1)
plot(x,y
     , xaxt = "n"
     , main="Distribución TStudent"
     , type = "l"
     , lty = 1
     , xlab = "IC"
     , ylab = "Density"
     , ylim = c(0, 0.4)
     , frame=F)
# CAMPANA DE GAUUS SOMBREADA
ICt <- seq(-Tstudent, Tstudent, by = 0.001)
ICd <- dt(ICt, 1)
polygon(c(-Tstudent,ICt,Tstudent), c(0,ICd,0), col="rosybrown1")
# LÍNEA CENTRAL
abline(v=0, lty=2, col="gray20")
# VALORES DEL INTERVALO
axis(1
     , at = c(-1*Tstudent, 0, 1*Tstudent)
     , font = 8
     , col.axis = "gray20"
     , labels = c(li.IC, Xm, ls.IC))
```



Capítulo VI

Ejercicio 1

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu = 20 \\ H_1: & \mu \neq 20 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=20

Desviación estándar poblacional=12,1

Media de la muestra=15,20

Muestra=45

Coeficiente de Confianza=0.99

Nivel de confianza=0.01

Dos colas, α/2=0.005

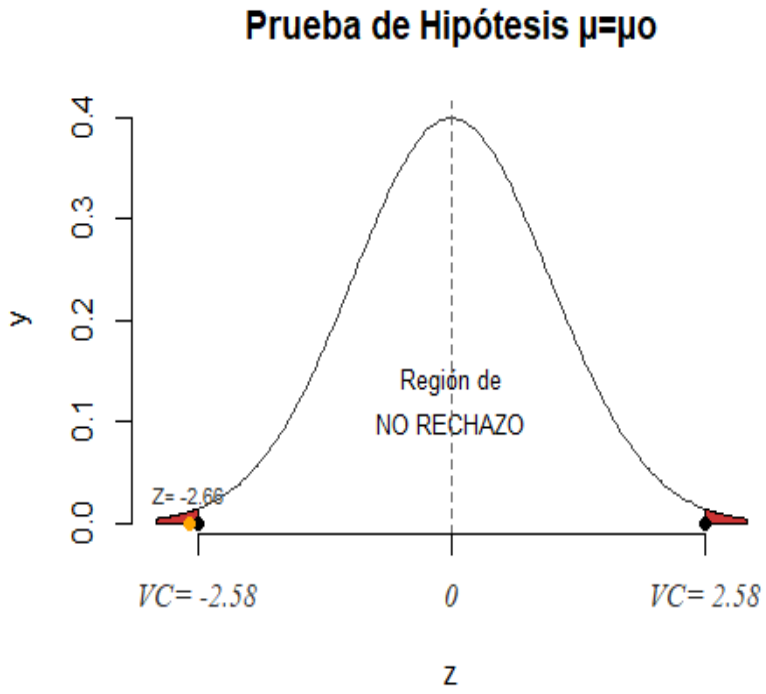
Selección del estadístico

Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{\sigma}$$
$$\sigma = \frac{\sigma_{\overline{x}}}{\sqrt{n}}$$

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
# DATOS
xp<-20
SDp<-12.1
xm<-15.2
n=45
alfa=0.01
# GENERAR LOS VALORES DE Z
z=seq(-3,3,length=200)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm<-((xm-xp)/(SDp/sqrt(n)))
vc=round(qnorm(alfa/2,0,1) , digits=2)
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
plot(z,y
     , xaxt = "n"
```

```
     , main="Prueba de Hipótesis μ=μo"
     , type="l"
     , lwd=1
     , col="gray 20"
     , frame=F)
# PINTAR LA ZONA DE NO RECHAZO, HIPÓTESIS μ>=μo
x=seq(-3,-abs(vc),by=0.01)
y=dnorm(x)
polygon(c(-3,x,-abs(vc)), c(0,y,0), col="brown3")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="brown3")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0, lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1, at = c(-vc, 0, vc), font = 8, col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0"
                  , paste("VC=", as.character(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
points(vc,0, cex=1, pch=19, col="black")
# PUNTOS DEL VALOR DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="orange")
text(Zm,0.03,paste("Z=",as.character(round(Zm,2))),cex=0.7,col="gray20")
```



Interpretar resultados: El valor z de la muestra cae en la región de rechazo. Esto significa que la evidencia estadística es suficiente para rechazar Ho.

Ejercicio 2

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \leq 120 \\ H_1: & \mu > 120 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=120

Desviación estándar poblacional=50

Media de la muestra=24

Muestra=60

Coeficiente de Confianza=0.98

Nivel de confianza=0.02

Selección del estadístico

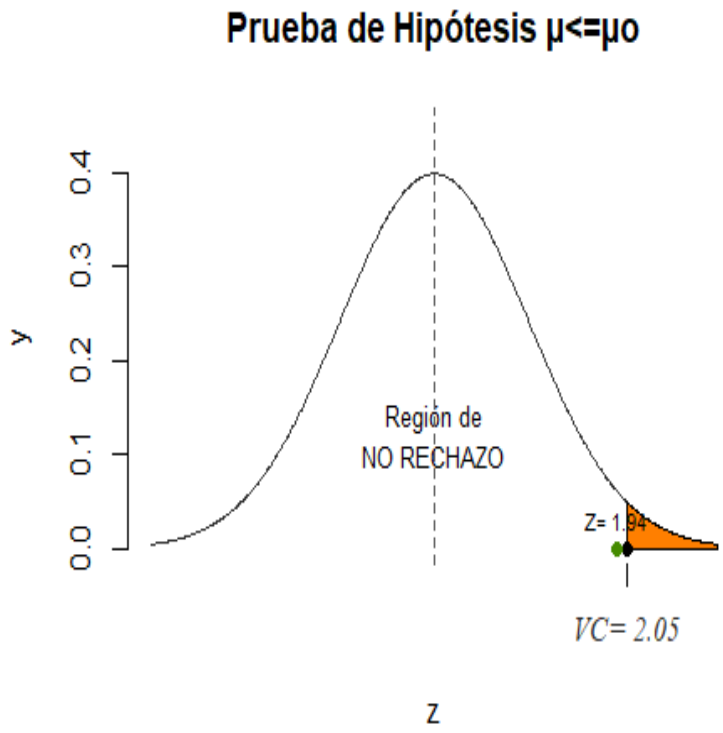
Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{\sigma}$$

$$\sigma = \frac{\sigma_{\overline{x}}}{\sqrt{n}}$$

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
# DATOS
xp<-150
SDp<-120
xm<-180
n=60
alfa=0.02
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO Y Z DE LA MUESTRA
Zm<-((xm-xp)/(SDp/sqrt(n)))
vc<-round(qnorm(alfa,0,1, lower.tail = FALSE) , digits=2)
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
plot(z,y
     , xaxt = "n"
     , main="Prueba de Hipótesis μ<=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ<=μo, HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="darkorange1")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
```

```
abline(v=0, lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1, at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)) )
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTOS DEL Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col=" chartreuse4")
text(Zm, 0.03, paste("Z=", as.character(round(Zm,2))) , cex=0.7, col="black")
```



Interpretar resultados: El valor estadístico z de la muestra se encuentra en la región de no rechazo. Por tanto, no es posible estadísticamente rechazar Ho

Ejercicio 3

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \leq 5.5 \\ H_1: & \mu > 5.5 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=5.5

Media de la muestra=6.2

Tamaño de la muestra=16

Desviación estándar=3,4

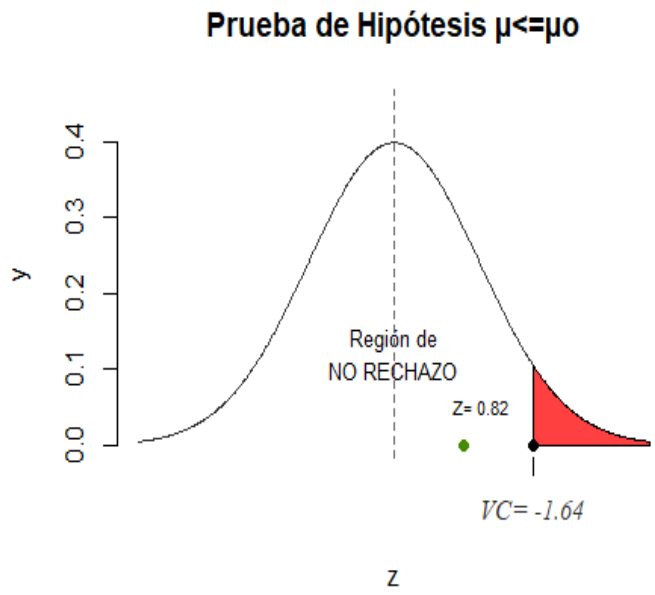
Coeficiente de confianza =0.95

Nivel de significancia=0.05

Selección del estadístico Vamos a utilizar en este caso el estadístico Z

$$Z = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
# DATOS
xp<-5.5
xm<-6.2
n=16
SD<-3.4
alfa=0.05
# GENERAR LOS VALORES DE Z
z=seq(-3,3,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA Z
y=dnorm(z)
# VALOR CRÍTICO
Zm<-((xm-xp)/(SD/sqrt(n)))
vc<-round(qnorm(alfa, 0, 1, lower.tail = FALSE), digits=2) #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
plot(z,y
     , xaxt = "n"
     , main="Prueba de Hipótesis μ<=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo, HIPÓTESIS μ<=μo
x=seq(abs(vc),3,by=0.01)
y=dnorm(x)
polygon(c(abs(vc),x,3),c(0,y,0),col="brown1")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0, lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(-vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(Zm,0, cex=1, pch=19, col="chartreuse4")
text(Zm+0.2, 0.05, paste("Z=", as.character(round(Zm, 2))), cex=0.7, col="bla
ck")
```



Interpretar resultados: El valor estadístico z de la muestra se encuentra en la región de no rechazo. Por tanto, no es posible estadísticamente rechazar Ho.

Ejercicio 4

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu = 9{,}0 \\ H_1: & \mu \neq 9{,}0 \end{cases}$$

Formular un plan de análisis Datos

Muestra=8,8.2,8.4,8.8,8.9,9,9.1,9.2,9.5

Media poblacional=9.0

Media de la muestra=xm<-mean(Muestra)

Tamaño de la muestra=n=9

Desviación standard=sd(Muestra)

Nivel de significancia=alfa=0.05

Grados de libertad=GL=n-1

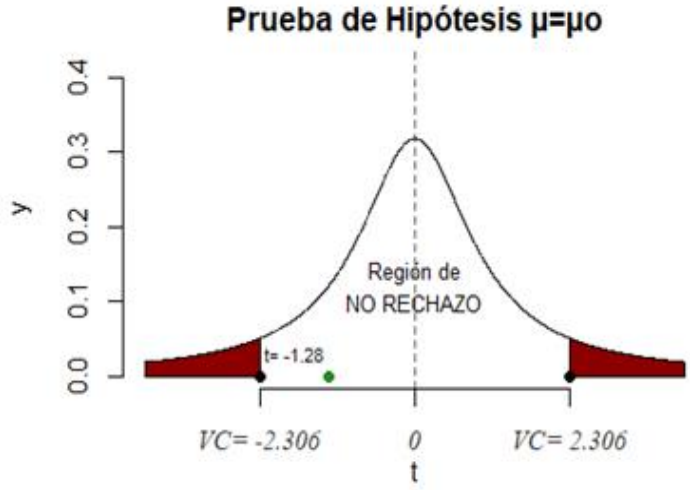
Selección del estadístico

Como n<30 se recomienda utilizar el estadístico t

$$t = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
# DATOS
```

```
xp<-9.0
muestra<-c(8,8.2,8.4,8.8,8.9,9,9.1,9.2,9.5)
xm<-mean(muestra)
n=9
SD<-sd(muestra)
alfa=0.05
GL=n-1
# GENERAR LOS VALORES DE t
t=seq(-4,4,by=0.001)
# APLICAR LA FUNCIÓN DENSIDAD PARA t
y=dt(t,1)
# VALOR CRÍTICO
tm=round((xm-xp)/(SD/sqrt(n)), 3)                #t de la muestra
vc<-round(qt(1-alfa/2, GL, lower.tail = TRUE),3) #Valor crítico
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
plot(t,y
     , xaxt = "n"
     , main="Prueba de Hipótesis μ=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo, HIPÓTESIS μ>=μo
x=seq(-4,-abs(vc),by=0.001)
y=dt(x,1)
polygon(c(-4,x,-abs(vc)), c(0,y,0), col="darkred")
# HIPÓTESIS μ<=μo
x=seq(abs(vc),4,by=0.001)
y=dt(x,1)
polygon(c(abs(vc),x,4),c(0,y,0),col="darkred")
text(0, 0.14
      , "Región de"
      , cex=0.8)
text(0, 0.1
      , "NO RECHAZO"
      , cex=0.8)
# LINEA DE Z=0
abline(v=0, lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1, at = c(-vc, 0, vc)
     , font = 8
     , col.axis = "gray20"
     , labels = c(paste("VC=", as.character(-vc)),"0", paste("VC=", as.charact
er(vc))))
# PUNTOS DEL VALOR CRÍTICO
points(-vc,0, cex=1, pch=19, col="black")
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(tm,0, cex=1, pch=19, col="forestgreen")
text(tm-0.5, 0.03
      , paste("t=", as.character(round(tm, 2)))
      , cex=0.7
      , col="black")
```



Interpretar resultados: El valor t de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza.

Ejercicio 5

Proponer la hipótesis

$$Hipótesis = \begin{cases} H_0: & \mu \leq 1{,}2 \\ H_1: & \mu > 1{,}2 \end{cases}$$

Formular un plan de análisis

Datos

Media poblacional=1.2

muestra=1.2,1.5,1.7,1.3,1.4,1.8,1.1,1.6

Media de la muestra=mean(muestra)

Tamaño de la muestra=8

Desviación standard=sd(muestra)

Nivel de significancia=0.01

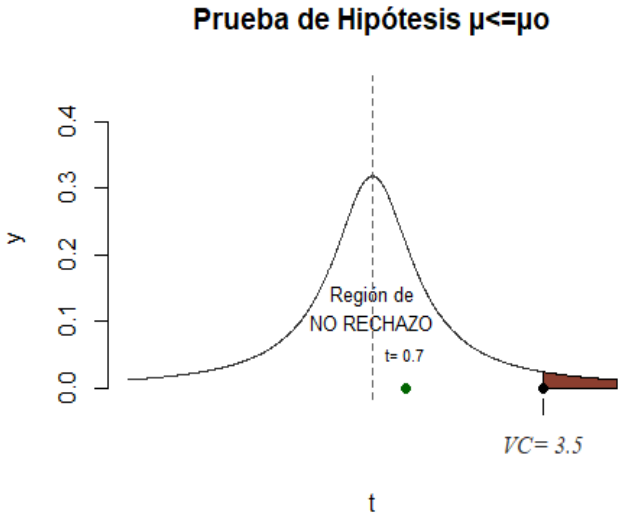
Grados de libertad=n-1

Selección del estadístico Vamos a utilizar en este caso el estadístico t

$$t = \frac{\overline{x} - \mu}{S}; S = \frac{S}{\sqrt{n}}$$

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
# DATOS
xp<-1.2
muestra<-c(.2,1.5,1.7,1.3,1.4,1.8,1.1,1.6)
xm<-mean(muestra)
```

```
n=8
SDm<-sd(muestra)
alfa=0.01
GL=n-1
# GENERAR LOS VALORES DE t
t=seq(-5,5,by=0.01)
# APLICAR LA FUNCIÓN DENSIDAD PARA t
y=dt(t,1)
# VALOR CRÍTICO Y T DE LA MUESTRA
tm<-round((xm-xp)/(SDm/sqrt(n)), 3)
vc<-round(qt(alfa/2, GL, lower.tail = FALSE),2)
# GRÁFICO DE LA DISTRIBUCIÓN NORMAL ESTÁNDAR
plot(t,y
     , xaxt = "n"
     , main="Prueba de Hipótesis μ<=μo"
     , ylim = c(0,0.45)
     , type="l"
     , lwd=1
     , col="gray20"
     , frame=F)
# PINTAR LA ZONA DE RECHAZO CON μ=μo, HIPÓTESIS μ<=μo
t=seq(abs(vc),5,by=0.01)
y=dt(t,1)
polygon(c(abs(vc),t,5),c(0,y,0),col="coral4")
text(0, 0.14, "Región de", cex=0.8)
text(0, 0.1, "NO RECHAZO", cex=0.8)
# LINEA DE Z=0
abline(v=0, lty=2, col="gray40")
# ESCRIBIR EL VALOR CRÍTICO EN EL EJE
axis(1
     , at = c(vc)
     , font = 8
     , col.axis = "gray20"
     , labels = paste("VC=", as.character(vc)))
# PUNTOS DEL VALOR CRÍTICO
points(vc,0, cex=1, pch=19, col="black")
# PUNTO DEL VALOR DE Z DE LA MUESTRA
points(tm,0, cex=1, pch=19, col="darkgreen")
text(tm, 0.05, paste("t=", as.character(round(tm, 2))), cex=0.7, col="black")
```

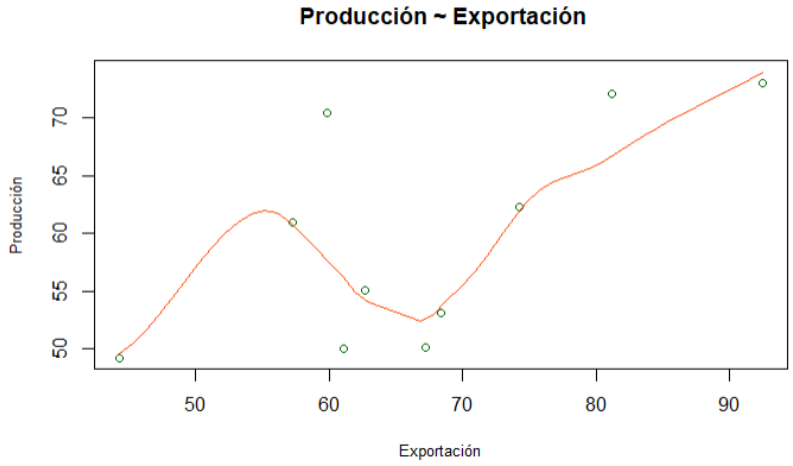


Interpretar resultados: El valor t de la muestra está ubicado en la región de no rechazo. Por lo tanto, Ho no se rechaza

Capítulo VII

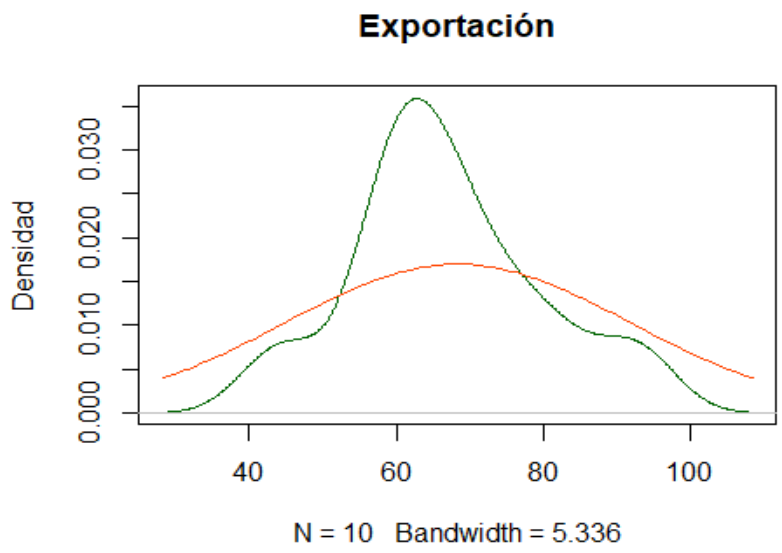
Ejercicio 1

```
produccion<-c(50.126,53.190,62.345,72.120,73.010,50.000,61.023,49.235,55.120,
70.506)
exportacion<-c(67.234,68.345,74.202,81.201,92.432,61.093,57.293,44.287,62.639
,59.831)
datosrosa<-data.frame(produccion,exportacion)
# Salida gráfica
par(mfrow=c(1,1), mar=c(5,5,4,1))
# Variable predictora ~ Variable de respuesta
scatter.smooth(x=datosrosa$exportacion, y= datosrosa$produccion
               , main=" Producción ~ Exportación"
               , cex.main=0.8
               , xlab="Exportación"
               , ylab="Producción"
               , col = "darkgreen"
               , lpars = list(col = "orangered1", lwd = 1, lty = 1)
               , cex.lab=0.8)
```



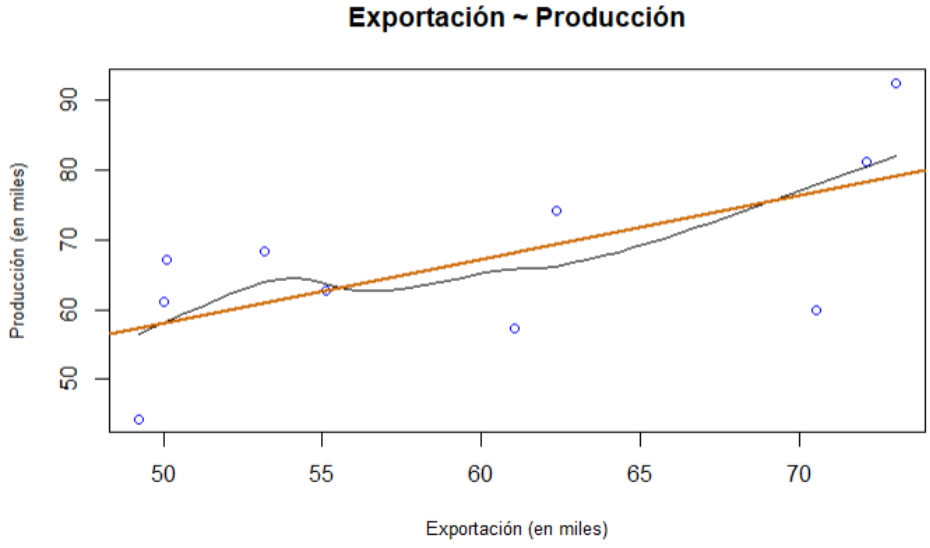
Ejercicio 2

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
produccion<-c(50.126,53.190,62.345,72.120,73.010,50.000,61.023,49.235,55.120,
70.506)
exportacion<-c(67.234,68.345,74.202,81.201,92.432,61.093,57.293,44.287,62.639
,59.831)
datosrosa<-data.frame(produccion,exportacion)
densidad_pc<-density(datosrosa$exportacion)
# DENSIDAD DE LA VARIABLE PREDICTORA
plot(densidad_pc
     , main = "Exportación"
     , ylab="Densidad"
     , col="darkgreen")
x<- datosrosa$exportacion
curve(dnorm(x, mean(x), sd(x)), col="orangered1", add=TRUE)
```



Ejercicio 3

```
par(mfrow=c(1,1), mar=c(5,5,4,1))
produccion<-c(50.126,53.190,62.345,72.120,73.010,50.000,61.023,49.235,55.120,
70.506)
exportacion<-c(67.234,68.345,74.202,81.201,92.432,61.093,57.293,44.287,62.639
,59.831)
datosrosa<-data.frame(produccion,exportacion)
# VARIABLE RESULTADO ~ VARIABLE PREDICTORA
modeloLineal <- lm(exportacion ~ produccion, data= datosrosa)
scatter.smooth(x= datosrosa$produccion, y= datosrosa$exportacion
               , main=" Exportación ~ Producción "
               , cex.main=0.8
               , xlab="Exportación (en miles)"
               , ylab="Producción (en miles)"
               , col="blue"
               , cex.lab=0.8)
# GRÁFICO DE LA RECTA DE REGRESIÓN LINEAL SIMPLE.
abline(modeloLineal, lwd = 2, col = "darkorange3")
```

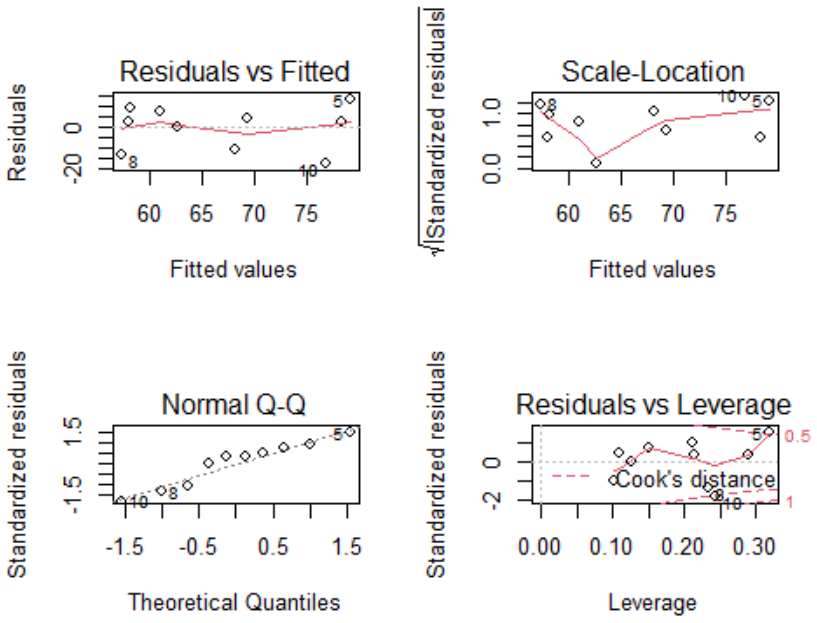


Ejercicio 4

```
produccion<-c(50.126, 53.190, 62.345, 72.120,73.010,50.000,61.023,49.235,55.1
20,70.506)
exportacion<-c(67.234, 68.345, 74.202, 81.201,92.432,61.093,57.293,44.287,62.
639,59.831)
datosrosa<-data.frame(produccion,exportacion)
# VARIABLE RESULTADO ~ VARIABLE PREDICTORA
modeloLineal <- lm(exportacion ~ produccion, data= datosrosa)
summary(modeloLineal)
## Call:
## lm(formula = exportacion ~ produccion, data = datosrosa)
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.959  -8.116   3.015   6.793  13.346
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.1638    22.8108   0.533   0.6084
## produccion    0.9166     0.3780   2.425   0.0415 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 10.79 on 8 degrees of freedom
## Multiple R-squared:  0.4236, Adjusted R-squared:  0.3516
## F-statistic:  5.88 on 1 and 8 DF,  p-value: 0.04153
```

Ejercicio 5

```
produccion<-c(50.126,53.190,62.345,72.120,73.010,50.000,61.023,49.235,55.120,
70.506)
exportacion<-c(67.234,68.345,74.202,81.201,92.432,61.093,57.293,44.287,62.639
,59.831)
datosrosa<-data.frame(produccion,exportacion)
# VARIABLE RESULTADO ~ VARIABLE PREDICTORA
modeloLineal <- lm(exportacion ~ produccion, data= datosrosa)
layout(matrix(c(1, 2,3, 4),nrow = 2,ncol = 2))
plot(modeloLineal)
```



REFERENCIAS

[1] P. C. F. David F. Groebner Patrick W. Shannon, *Businness statistics: A decision making approach*, Eighth Edition. Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458: Prentice Hall, 2011, p. 7.

[2] R. F. H. Jessica M. Utts, *Mind on statistics*, Fifth Edition. 200 First Stamford Place, 4th Floor. Stamford, CT 06902. USA: Cengage Learning, 2015, p. 15.

[3] R. Lyman and M. Longnecker, *An introduction to statistical methods and data analysis*, Sixth. 10 Davis Drive. Belmont, CA 94002-3098. USA: Brooks/Cole, 2010.

[4] M. Gardener, *Beginning r: The statistical programming language*, First Edition. 10475 Crosspoint Boulevard. Indianapolis, IN 46256: John Wiley & Sons, 2012, p. 65.

[5] L. E. S. David P. Doane, *Applied statistics in businesss and economics*, Fifth Edition. 2 Penn Plaza. New York. NY 10121. USA: McGraw-Hill Education, 2016, p. 291.

[6] E. G. M. Hui, *Learn r for applied statistics*, First Edition. 233 Spring Street, 6th Floor, New York, NY 10013: Apress, 2019, p. 16.

[7] P. Delgaard, *Introductory statistics with r*, Second Edition. 233 Spring Street, 6th Floor, New York, NY 10013: Springer, 2008, p. 50.

[8] R. I. Kabacoff, *R in action data analysis and graphics with r*, Second Edition. 20 Baldwin Road, PO Box 761, Shelter Island, NY 11964: Manning Publications Co, 2015, p. 25.

[9] R. I. Kabacoff, *R in action data analysis and graphics with r*, Second Edition. 20 Baldwin Road, PO Box 761, Shelter Island, NY 11964: Manning Publications Co, 2015, p. 32.

[10] J. Y. C. Yosef Cohen, *Statistics and data with r*, Second Edition. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, USA: John Wiley & Sons Ltd, 2008, p. 63.

[11] G. Keller, *Statistic for management and economics*, Eleventh. 20 Channel Center Street. Boston. MA 02210. USA: CENGAGE Learning, 2018, pp. 22, 23.

[12] J. F. Healey, *The essentials of statistics*, 4e ed. 20 Channel Center Street. Boston. MA 02210. USA: CENGAGE Learning, 2016, pp. 22, 23.

[13] J. L. Devore, *Probabilidad y estadistica para ingenieria y ciencias*, Septima. Av. Santa Fe, num. 505, piso 12. Mexico DF: CENGAGE Learning, 2008, p. 35.

[14] L. C. A. David S Moore George P. McCabe, *The practice of estatistics for businnes and economics*, Fourth Edition. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: W. H. Freeman; Company, 2012, pp. 29, 30.

[15] P. F. V. Norean R. Sharpe Richard D. Veaux, *Business statistics*, 3rd Edition. Edinburgh Gate. Harlow. Essex CM20 2JE. England: Pearson Education, 2015, p. 91.

[16] K. B. Carlos Cortinhas, *Statistics for business ans economics*, First European Edition. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: John Wiley & Sons, Ltd., 2012, p. 28.

[17] J. D. Roxy Peck Chris Olsen, *Introduction to statistics & data analysis*, Fourth Edition. 20 Channel Center Street. Boston, MA 02210. USA: CENGAGE Learning, 2012, p. 16.

[18] A. D. Mata, *Estadistica aplicada a la administracion y economia*, Primera. Prolongacion Paseo de la Reforma 1015, Torre A. Mexico DF: McGraw Hill Educacion, 2013, p. 11.

[19] D. S. y N. T. David Anderson, *Estadistica para negocios y economia*, Decima primera. Av. Santa Fe, num. 505, piso 12. Mexico DF: CENGAGE Learning, 2012, p. 30.

[20] R. L. y David Rubin, *Estadistica para administracion y economia*, Septima. Atlacomulco 500-5o. piso. Ciudad Juarez: PEARSON Education, 2010, p. 30.

[21] P. C. F. David F. Groebner Patrick W. Shannon, *Businness statistics: A decision making approach*, Eighth Edition. Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458: Prentice Hall, 2011, pp. 62, 63.

[22] R. L. y David Rubin, *Estadistica para administracion y economia*, Septima. Atlacomulco 500-5o. piso. Ciudad Juarez: PEARSON Education, 2010, p. 31.

[23] A. G. Bluman, *Elementary statistics, a step by step approach*, Ninth Edition. 2 Penn Plaza, New York, NY 10121: McGraw-Hill Education, 2014, p. 58.

[24] P. S. Mann, *Introductory statistics*, Ninth Edition. 222 Rosewood Drive, Danvers, MA 01923. USA: Wily, 2016, p. 58.

[25] D. J. S. y T. A. W. David R. Anderson, *Estadistica para administracion y economia*, Decima primera. Av. Santa Fe, num. 505, piso 12. Mexico: CENGAGE Learning, 2012, p. 44.

[26] J. D. Roxy Peck Chris Olsen, *Introduction to statistics & data analysis*, Fourth Edition. 20 Channel Center Street. Boston, MA 02210. USA: CENGAGE Learning, 2012, p. 125.

[27] G. Keller, *Statistic for management and economics*, Eleventh. 20 Channel Center Street. Boston. MA 02210. USA: CENGAGE Learning, 2018, pp. 244, 245.

[28] P. S. Mann, *Introductory statistics*, Ninth Edition. 222 Rosewood Drive, Danvers, MA 01923. USA: Wily, 2016, p. 233.

[29] M. Barrow, *Statistics for economics, accounting and business studies*, Fourth Edition. 90 Tottenham Court Road, London W1T 4LP: Prentice Hall, 2006, p. 24.

[30] A. D. Mata, *Estadistica aplicada a la administracion y economia*, Primera. Prolongacion Paseo de la Reforma 1015, Torre A. Mexico DF: McGraw Hill Educacion, 2013, pp. 45, 55.

[31] D. J. S. y T. A. W. David R. Anderson, *Estadistica para administracion y economia*, Decima primera. Av. Santa Fe, num. 505, piso 12. Mexico: CENGAGE Learning, 2012, p. 87.

[32] B. K. Alan Agresti Christine Franklin, *Stadistics: The art and science of learning from data*, Fourth Edition. Edinburgh Gate. Harlow. Essex CM20 2JE.England: Pearson Education, 2018, p. 77.

[33] R. B. y B. D. William Mendenhall, *Introduccion a la probabilidad y estadistica*, Decima Tercera. Av. Santa Fe, num. 505, piso 12. Mexico DF: CENGAGE Learning, 2010, p. 30.

[34] T. S. James McClave, *Stadistics*, Thirteenth Edition. KAO Two. KAO Park. Harlow. CM17 9NA. United Kingdom: Pearson Education, 2018, p. 94.

[35] J. D. C. David R. Anderson Dennis J. Sweeney y Thomas A. Williams, *Statistics for business and economics*, Decima primera. South-Western. 5191 Natorp Boulevard. Mason, OH 45040. USA: CENGAGE Learning, 2014, p. 118.

[36] D. J. S. y T. A. W. David R. Anderson, *Estadistica para administracion y economia*, Decima primera. Av. Santa Fe, num. 505, piso 12. Mexico: CENGAGE Learning, 2012, p. 95.

[37] G. Keller, *Statistic for management and economics*, Eleventh. 20 Channel Center Street. Boston. MA 02210. USA: CENGAGE Learning, 2018, p. 107.

[38] L. E. S. David P. Doane, *Applied statistics in business and economics*, Fifth Edition. 2 Penn Plaza. New York. NY 10121. USA: McGraw-Hill Education, 2016, p. 256.

[39] A. G. Bluman, *Elementary statistics, a step by step approach*, Ninth Edition. 2 Penn Plaza, New York, NY 10121: McGraw-Hill Education, 2014, p. 314.

[40] L. C. A. David S Moore George P. McCabe, *The practice of estatistics for businnes and economics*, Fourth Edition. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: W. H. Freeman; Company, 2012, p. 38.

[41] T. A. W. David R. Anderson Dennis J. Sweeney and J. J. Cochran, *Statistics for business and economics*, 12e ed. South-Western. 5191 Natorp Boulevard. Mason, OH 45040. USA: CENGAGE Learning, 2014, p. 272.

[42] A. G. Bluman, *Elementary statistics, a step by step approach*, Ninth Edition. 2 Penn Plaza, New York, NY 10121: McGraw-Hill Education, 2014, p. 315.

[43] L. C. A. David S Moore George P. McCabe, *The practice of estatistics for businnes and economics*, Fourth Edition. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: W. H. Freeman; Company, 2012, p. 45.

[44] L. C. A. David S Moore George P. McCabe, *The practice of estatistics for businnes and economics*, Fourth Edition. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: W. H. Freeman; Company, 2012, p. 46.

[45] J. L. Devore, *Probabilidad y estadistica para ingenieria y ciencias*, Septima. Av. Santa Fe, num. 505, piso 12. Mexico DF: CENGAGE Learning, 2008, p. 108.

[46] T. A. W. David R. Anderson Dennis J. Sweeney and J. J. Cochran, *Statistics for business and economics*, 12e ed. South-Western. 5191 Natorp Boulevard. Mason, OH 45040. USA: CENGAGE Learning, 2014, p. 240.

[47] M. L. R. Lyman Ott, *Introduction to statistical methods and data analysis*, Sixth Edition. Brooks/Cole. 10 Davis Drive. Belmont, CA 94002-3098. USA: CENGAGE Learning, 2010, p. 192.

[48] M. L. R. Lyman Ott, *Introductory statistics*, 9th Edition. 501 Boylston Street. Suite 900. Boston. MA 02116. USA: Addison-Wesley, 2012, p. 289.

[49] S. M. Ronald Walpole Raymond Myers, *Probabilidad y estadistica para ingenieria y ciencias*, Novena. Atlacomulco 500-5to. piso. 53519 Naucalpan de Juarez. Estado de Mexico: Pearson Educacion, 2012, p. 162.

[50] L. C. A. David S Moore George P. McCabe, *The practice of estatistics for businnes and economics*, Fourth Edition. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: W. H. Freeman; Company, 2012, p. 270.

[51] S. M. Ronald Walpole Raymond Myers, *Probabilidad y estadistica para ingenieria y ciencias*, Novena. Atlacomulco 500-5to. piso. 53519 Naucalpan de Juarez. Estado de Mexico: Pearson Educacion, 2012, p. 203.

[52] S. M. Ross, *Introductory statistics*, Third Edition. 30 Corporate Drive. Suite 400. Burlington. MA 01803. USA: Elsevier, 2010, p. 298.

[53] P. F. V. Norean R. Sharpe Richard D. Veaux, *Business statistics*, 3rd Edition. Edinburgh Gate. Harlow. Essex CM20 2JE. England: Pearson Education, 2015, p. 363.

[54] W. M. y S. W. Douglas Lind, *Estadistica aplicada a los negocios y la economia*, Decimo Quinta. Prolongacion Paseo de la Reforma 1015, Torre A, Piso 17. C.P. 01376. Mexico DF: McGrawHill Educacion, 2012, p. 85.

[55] R. B. y B. D. William Mendenhall, *Introduccion a la probabilidad y estadistica*, Decima Tercera. Av. Santa Fe, num. 505, piso 12. Mexico DF: CENGAGE Learning, 2010, p. 298.

[56] G. Keller, *Statistic for management and economics*, Eleventh. 20 Channel Center Street. Boston. MA 02210. USA: CENGAGE Learning, 2018, pp. 311, 312.

[57] T. A. W. David R. Anderson Dennis J. Sweeney and J. J. Cochran, *Statistics for business and economics*, 12e ed. South-Western. 5191 Natorp Boulevard. Mason, OH 45040. USA: CENGAGE Learning, 2014, p. 344.

[58] W. M. y S. W. Douglas Lind, *Estadistica aplicada a los negocios y la economia*, Decimo Quinta. Prolongacion Paseo de la Reforma 1015, Torre A, Piso 17. C.P. 01376. Mexico DF: McGrawHill Educacion, 2012, p. 299.

[59] P. S. Mann, *Introductory statistics*, Ninth Edition. 222 Rosewood Drive, Danvers, MA 01923. USA: Wily, 2016, p. 313.

[60] W. M. y S. W. Douglas Lind, *Estadistica aplicada a los negocios y la economia*, Decimo Quinta. Prolongacion Paseo de la Reforma 1015, Torre A, Piso 17. C.P. 01376. Mexico DF: McGrawHill Educacion, 2012, p. 334.

[61] G. Keller, *Statistic for management and economics*, Eleventh. 20 Channel Center Street. Boston. MA 02210. USA: CENGAGE Learning, 2018, p. 633.

[62] A. G. Bluman, *Elementary statistics, a step by step approach*, Ninth Edition. 2 Penn Plaza, New York, NY 10121: McGraw-Hill Education, 2014, p. 550.

[63] P. S. Mann, *Introductory statistics*, Ninth Edition. 222 Rosewood Drive, Danvers, MA 01923. USA: Wily, 2016, p. 503.

[64] J. D. Roxy Peck Chris Olsen, *Introduction to statistics & data analysis*, Fourth Edition. 20 Channel Center Street. Boston, MA 02210. USA: CENGAGE Learning, 2012, p. 748.

[65] M. J. S. Morris H. DeGroot, *Probability and statistics*, Fourth Edition. 75 Arlington Street. Suite 300. Boston. MA 02116. USA: Pearson Education, 2012, p. 248.

[66] P. S. Mann, *Introductory statistics*, Ninth Edition. 222 Rosewood Drive, Danvers, MA 01923. USA: Wily, 2016, p. 529.

SOBRE LOS AUTORES

Msc. Ariosto Vicuña Pino, es profesor de la carrera de Ingeniería en Sistemas e Ingeniería de software de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo. Tiene una Ingeniería en Computación con especialización en Sistemas Tecnológicos en la Escuela Superior Politécnica del Litoral, un Diploma en Control de la Calidad en la Universidad Técnica Particular de Loja, una maestría en Marketing en la Universidad Técnica Estatal de Quevedo y una maestría en Bibliotecología y Sistemas de Información en la Universidad de La Habana (Cuba). Ha trabajado como profesor de programación de computadoras, modelamiento de datos, análisis y diseño de sistemas orientados a objetos, entre otras materias relacionadas por cerca de catorce años. Desde el año 2018 dicta la cátedra de Paquetes Estadísticos y Estadísticas. El profesor Ariosto Vicuña Pino es autor y coautor de varios artículos científicos que han sido publicadas en revistas como Revista Ibérica de Sistemas e Tecnologias de Informação (RISTI), Revista Conrado, Revista Publicando, entre otras.

Jéssica Alexandra Ponce Ordóñez, es profesora de la Facultad de Ciencias Empresariales donde desarrolla su actividad docente en la carrera de licenciatura en Marketing perteneciente a la Universidad Técnica Estatal de Quevedo. Tiene pregrado en ingeniería en sistemas por la Universidad Técnica Estatal de Quevedo (Ecuador) y es magister en seguridad informática por la Universidad Internacional de la Rioja (España). Ha dictado las asignaturas de Matemática, Cálculo Diferencial e Integral, Estadística, Software para la Investigación, Sistemas de Información de Auditoría, TIC (Tecnología de la información y la comunicación) aplicado a la mercadotecnia. Ha participado como autora y coautora de varios artículos científicos publicados en reconocidas revistas como Revista Ibérica de Sistemas e Tecnologias de Informação (RISTI), Revista de Investigación Operacional, Revista San Gregorio, entre otras.

Dr. EDUARDO DÍAZ OCAMPO, Ph.D.
RECTOR

Ing. YENNY GUISELLI TORRES NAVARRETE, Ph.D.
VICERRECTORA ACADÉMICA

Ing. BOLÍVAR ROBERTO PICO SALTOS, M.Sc.
VICERRECTOR ADMINISTRATIVO

Econ. CARLOS EDISON ZAMBRANO, Ph.D.
DIRECTOR DE INVESTIGACIÓN - DICYT

UTEQ
UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
2022

www.uteq.edu.ec

ISBN: 978-9978-371-30-5