

# Introducción al Software Estadístico R

Gudelia Figueroa Preciado, José A. Montoya Laos

Agosto 2015



# Contenido

<b>1</b>	<b>Conceptos Básicos</b>	<b>1</b>
1.1	Instalación del software R . . . . .	1
1.2	Tipos de Objetos . . . . .	3
1.2.1	Vectores . . . . .	3
1.2.2	Matrices . . . . .	4
1.2.3	Arreglos . . . . .	5
1.2.4	Factor . . . . .	5
1.2.5	Data.frame . . . . .	6
1.3	Algunas Estructuras de Programación . . . . .	7
<b>2</b>	<b>Análisis Exploratorio de Datos</b>	<b>9</b>
2.1	Medidas Descriptivas . . . . .	9
2.2	Algunos tipos de gráficas . . . . .	10
2.2.1	Diagrama de barras y Diagrama de pastel . . . . .	10
2.2.2	Diagramas de caja y Diagramas de Tallo y Hojas . . . . .	12
2.2.3	Histogramas . . . . .	13
2.3	Diagramas de dispersión . . . . .	15
<b>3</b>	<b>VARIABLES ALEATORIAS Y DISTRIBUCIONES MUESTRALES</b>	<b>19</b>
3.1	VARIABLES ALEATORIAS . . . . .	19
3.1.1	Poisson . . . . .	19
3.1.2	Binomial . . . . .	19
3.1.3	Binomial Negativa . . . . .	20
3.1.4	Uniforme . . . . .	20
3.1.5	Exponencial . . . . .	20
3.1.6	Normal . . . . .	21
3.1.7	Distribución Ji-Cuadrada . . . . .	21
3.1.8	Distribución <i>t</i> -Student . . . . .	21
3.1.9	Distribución F . . . . .	22

---

3.1.10	Ejercicios . . . . .	22
3.2	Distribución muestral de la media . . . . .	23
3.2.1	Ejercicios . . . . .	25
<b>4</b>	<b>Inferencia Estadística</b>	<b>27</b>
4.1	Intervalos de Confianza y Pruebas de Hipótesis . . . . .	27
4.1.1	Prueba de Normalidad . . . . .	27
4.1.2	Intervalo de confianza y Prueba de hipótesis para una media	28
4.1.3	Intervalo de confianza y Prueba de hipótesis para una proporción	30
4.1.4	Prueba de hipótesis para comparar dos varianzas . . . . .	31
4.1.5	Intervalo de confianza y Prueba de hipótesis para dos medias	31
4.1.6	Análisis de Varianza . . . . .	32

# Capítulo 1

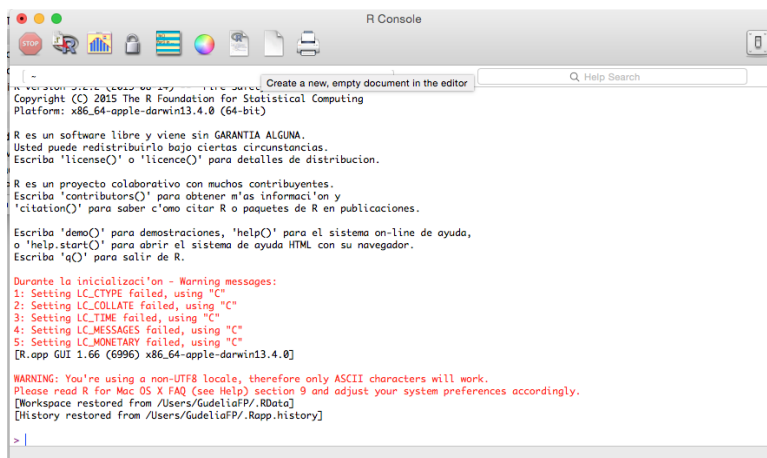
## Conceptos Básicos

### 1.1 Instalación del software R

R es un lenguaje que permite realizar análisis estadístico y gráfico y facilita al usuario el implementar funciones adicionales creando o modificando las funciones existentes. La instalación de este software puede hacerse desde la página del Proyecto R, que se encuentra en la siguiente dirección:

<https://www.r-project.org>

Para descargar el software R se selecciona un CRAN mirror. Por ejemplo, puede escogerse la opción <https://cran.itam.mx/>, del Instituto Tecnológico Autónomo de México. Este software puede instalarse en ambientes Mac, Windows, Linux y Unix. Después de seleccionar el sistema operativo a utilizar, se sigue el proceso de instalación que indica el archivo .exe descargado. Al ejecutar el programa aparecerá una pantalla como la mostrada en la Figura 1.1. Es conveniente crear una carpeta de trabajo donde R guarde y lea la información directamente.



```
[-] Create a new, empty document in the editor Q Help Search
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribución.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

Durante la inicialización - Warning messages:
1: Setting LC_CTYPE failed, using "C"
2: Setting LC_COLLATE failed, using "C"
3: Setting LC_TIME failed, using "C"
4: Setting LC_MESSAGES failed, using "C"
5: Setting LC_MONETARY failed, using "C"
[R.app GUI 1.66 (6996) x86_64-apple-darwin13.4.0]

WARNING: You're using a non-UTF8 locale, therefore only ASCII characters will work.
Please read R for Mac OS X FAQ (see Help) section 9 and adjust your system preferences accordingly.
[Workspace restored from /Users/GudeliaFP/.RData]
[History restored from /Users/GudeliaFP/.Rapp.history]

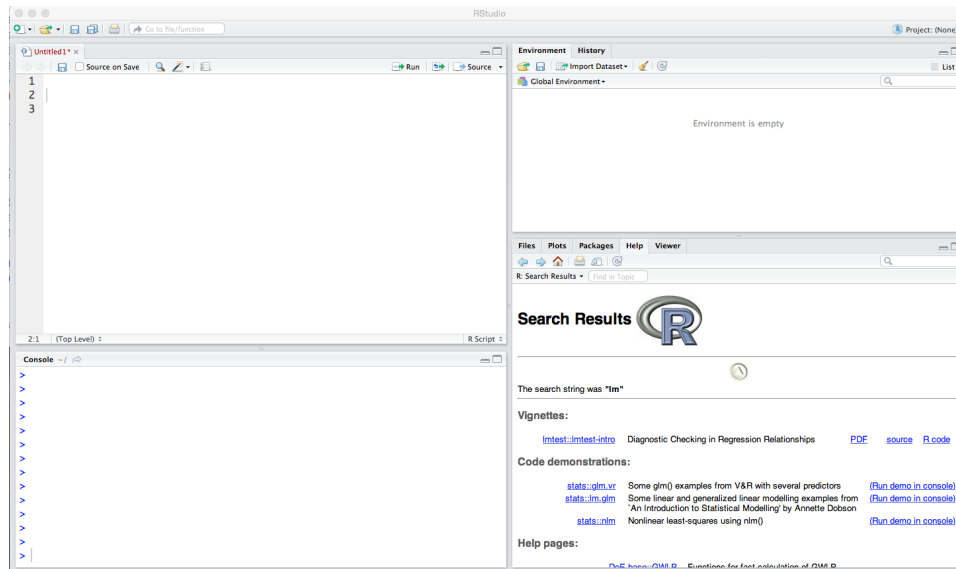
> |
```

Figura 1.1: Consola Software R

Una interfase gráfica que facilita el uso del software R y que se utilizará en el transcurso de estas notas es RStudio, que puede descargarse de la siguiente dirección

<https://www.rstudio.com>

y cuyo ambiente se presenta en la Figura 1.2. Este ambiente gráfico permite observar, al mismo tiempo, el script o programa que estemos ejecutando, los resultados que arrojan cada una de las líneas de código de nuestro programa, las gráficas construidas, el historial de comandos realizados, una ayuda general, etcétera.



**Figura 1.2:** Ambiente gráfico de RStudio

Para crear un nuevo script seleccionamos **File > New File > RScript**. Una vez creado el script se guardará con las instrucciones **File > Save as > Ejemplo1.R**; donde Ejemplo1.R es el nombre que estamos asignando al programa creado.

La ayuda en R puede solicitarse desde la consola de RStudio. Por ejemplo, con la instrucción `help(hist)` solicitamos ayuda sobre la construcción de un histograma. Esto también puede hacerse en la ventana de R que contiene las pestañas: **Files, Plots, Packages, Help** y **Viewer**.

Antes de realizar el primer programa en R es conveniente conocer que:

- El signo `#` nos permite incluir comentarios en los programas. Las líneas del programa que empiecen con este signo serán ignoradas al momento de la compilación.
- El signo `<-` se utiliza para asignar valores a objetos.
- La instrucción `ls()` lista los objetos en memoria.
- La instrucción `rm()` borra los objetos en memoria.

## 1.2 Tipos de Objetos

A continuación se presentan los distintos tipos de objetos que se utilizarán en los programas que presentaremos en el transcurso de estas notas.

### 1.2.1 Vectores

Es importante conocer como guardar, en una variable, distintos tipos de valores. A continuación se presentan las instrucciones más utilizadas en la programación en R.

- `x <- c(0,1,2,3,25,23)` le indica a R que “x” es un vector numérico.
- `y <- c(“A”, “B”, “C”, “D”)` permite asignar a “y” un vector de caracteres.
- `z <- c(T,T,F,F)` deposita en “z” un vector lógico.

A continuación se presentan algunas operaciones que pueden efectuarse con vectores:

- `rep()`: Repite un vector un número específico de veces.
- `seq()`: Crea una secuencia de valores. Requiere un valor inicial, un valor final y el tamaño del incremento.
- `cbind()`: Concatena vectores (por columnas).
- `rbind()`: Concatena vectores (por filas).
- `cumsum()`: Suma acumulada de los elementos de un vector.
- `vec1==vec2`: Compara los elementos de dos vectores. Produce TRUE si son iguales y FALSE si son diferentes.
- `vec1!=vec2`: Compara los elementos de dos vectores. Produce TRUE si son diferentes o FALSE si son iguales.
- `x[i]`: selecciona el *i-ésimo* elemento del vector x.

En el caso que “vec” sea un vector lógico, esto es, que sus entradas sean TRUE o FALSE:

- `all(vec)`: Produce TRUE si todas las entradas de vec son TRUE. En otro caso produce FALSE.
- `any(vec)`: Produce TRUE si alguna de las entradas de vec es TRUE. En otro caso produce FALSE

### 1.2.1.1 Ejercicios

Copiar el siguiente código directamente en un nuevo script y ejecutar cada una de las líneas que se incluyen, esto con el fin de familiarizarse con las operaciones con vectores.

```
rm(list=ls(all=TRUE))
vec1 <- -c(1,2,3,4,5)
vec2 <- -c(1,-2,3,-4,5)
rep(vec1,times=4) #Repite el vector cuatro veces
rep(vec2,times=5)
rep(0,times=10)
seq(from=0,to=1,by=0.01) #Genera una secuencia
cbind(vec1,vec1) #Une vectores por columnas
rbind(vec1,vec1) #Une vectores por filas
cumsum(vec1) #Suma acumulada de los elementos de un vector
```

### 1.2.2 Matrices

Es R es muy sencillo formar matrices, solamente debe tenerse en cuenta que todas las columnas en una matriz deben ser de la misma clase (numérico, caracter, etcétera), y de la misma longitud. Por ejemplo:

- `MatrizA <- matrix(x,nrow=3,ncol=2, byrow=F)`: convierte al vector “x”, definido en el punto anterior, en una matriz numérica llamada `MatrizA`, de tres filas y dos columnas. Los elementos del vector “x” son colocados verticalmente en `MatrizA`.
- `MatrizB <- matrix(y, nrow=2,ncol=2, byrow=T)`: convierte al vector “y”, previamente descrito, en una matriz de dos filas y dos columnas.
- `MatrizC <- matrix(z,nrow=2, ncol=2, byrow=T)`: convierte al vector “z”, descrito anteriormente, en una matriz lógica de dos filas y dos columnas. En este caso los elementos del vector “z” son colocados horizontalmente en la matriz llamada `MatrizC`.

Para obtener el elemento *ij*-ésimo de una matriz *M*, se utiliza la instrucción `M[i,j]`.

Ahora, supongamos que *X*, *Y* y *Z* son matrices y que *c* es un escalar. Algunas operaciones que pueden realizarse con matrices son:

- $X + Y$ : Suma de matrices.



- $X \pm c$ : Suma o resta, a cada elemento de la matriz  $X$  un escalar  $c$ .
- $X - Y$ : Resta de matrices.
- $X \% * \% Y$ : Producto de matrices (dimensión compatible).
- $X * c$ , o  $X / c$ : Multiplica o divide cada elemento de la matriz  $X$  por un escalar  $c$ .
- $X^Y$ : Potencia de matrices elemento a elemento (dimensión compatible).
- $X^c$ : Eleva cada elemento de la matriz  $X$  a la potencia  $c$ .
- $X / Y$ : División de matrices elemento a elemento (dimensión compatible).
- $t(X)$ : Transpuesta de  $X$ .
- $\text{solve}(X)$ : Inversa de  $X$ .
- $\text{det}(X)$ : Determinante de  $X$ .
- $\text{diag}(X)$ : Matriz identidad de  $k \times k$  (contiene unos en la diagonal).

### 1.2.3 Arreglos

Los arreglos son similares a las matrices, sólo que éstos pueden tener más de dos dimensiones. por ejemplo, con la siguiente instrucción construimos un arreglo que consta de tres matrices de dos por dos.

```
> Array1<-array(c(1,2,3,4,5,6,7,8,9,10,11,12),dim=c(2,2,3))
> Array1
, , 1
     [,1] [,2]
[1,]  1   3
[2,]  2   4
, , 2
     [,1] [,2]
[1,]  5   7
[2,]  6   8
, , 3
     [,1] [,2]
[1,]  9  11
[2,] 10  12
```

**Figura 1.3:** Ejemplo de un arreglo

### 1.2.4 Factor

Un objeto de tipo Factor almacena el valor de una variable nominal, esto es, una variable cualitativa, a la que no puede asociarse orden alguno. Por ejemplo, la instrucción:

```
F <- factor(c("Hombre", "Hombre", "Mujer"))
```

nos proporciona un vector F que contiene la información (Hombre, Hombre, Mujer).

### 1.2.5 Data.frame

Los objetos tipo data.frame son muy útiles en R, son más generales que una matriz ya que permiten considerar columnas que pueden tener distintas clases de objetos (numéricos, caracteres, factores, etcétera). A continuación se muestra la construcción de un data.frame que incluye varios tipos de objetos:

```
D <- data.frame(c(0,1),c("Azul", "Rojo"),c(TRUE,FALSE))
names(D) <- c("X1", "X2", "X3")
```

con lo cual obtenemos lo mostrado en la Tabla 1.1:

X1	X2	X3
0	Azul	TRUE
1	Rojo	FALSE

**Tabla 1.1:** Ejemplo de un data.frame

#### 1.2.5.1 Práctica con data.frame

Crear en R un objeto tipo data.frame como se muestra a continuación. A este data.frame se le ha asignado el nombre DT, formado por dos vectores de dimensión quince, que se nombran con NDVI y Temp.

```
DT <- data.frame(c(0.28,0.30,0.31,0.31,0.34),c(16.04,17.17,18.70,23.40,24.70))
names(DT) <- c("NDVI", "Temp")
```

El resultado que se obtiene con las instrucciones anteriores se muestra en la Figura 1.4.

```
> DT
  NDVI Temp
1 0.28 16.04
2 0.30 17.17
3 0.31 18.70
4 0.31 23.40
5 0.34 24.70
```

**Figura 1.4:** Ejemplo 2 de data.frame

## 1.3 Algunas Estructuras de Programación

R permite crear estructuras repetitivas y la ejecución condicional de sentencias a través de las instrucciones **for**, **while**, **if-else**, **repeat** y **switch**. A continuación se muestra cómo se utilizan algunas de ellas.

- **for**: Permite crear una estructura repetitiva. Para cada  $i$  en `VectorValores`, se repiten las sentencias o instrucciones programadas dentro de las llaves.

```
for(i in VectorValores)
{
  sentencias
}
```

- **while**: Permite crear una ejecución condicionada. Se ejecutan la sentencias mientras que la condición se cumpla.

```
while(condición)
{
  sentencias
}
```

- **if-else**: Permite crear una ejecución condicionada. Si se cumple la condición entonces se ejecutan las sentencias1. En caso contrario, se ejecutan las sentencias2.

```
if(condición)
{
  sentencias1
}
else
{
  sentencias2
}
```



## Capítulo 2

# Análisis Exploratorio de Datos

### 2.1 Medidas Descriptivas

En esta sección veremos cómo pedir en R ciertas medidas descriptivas que nos permiten analizar uno o varios conjuntos de datos. Entre las más utilizadas están:

- `min(y)`: Valor mínimo de  $y$ .
- `max(y)`: Máximo valor de  $y$ .
- `mean(y)`: Media muestral o promedio.
- `median(y)`: Mediana muestral.
- `var(y)`: Varianza muestral de  $y$  (utiliza denominador  $n-1$ ).
- `sd(y)`: Desviación estándar de  $y$  (utiliza denominador  $n-1$ ).
- `summary(y)`: Proporciona el mínimo, primer cuartil, mediana, media, tercer cuartil y máximo de  $y$ .

Por ejemplo, las siguientes instrucciones arrojan los resultados mostrados en la Figura 2.1.

```
rm(list=ls())  
  
x<-c(9,23,22,19,14,19,20,21,23,24)  
  
summary(x)  
  
var(x)  
  
sd(x)
```

```

> rm(list=ls())
> x<-c(9,23,22,19,14,19,20,21,23,24)
> summary(x)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  9.00  19.00   20.50   19.40   22.75   24.00
> var(x)
[1] 21.6
> sd(x)
[1] 4.64758

```

Figura 2.1: Algunas medidas descriptivas

## 2.2 Algunos tipos de gráficas

### 2.2.1 Diagrama de barras y Diagrama de pastel

Construir gráficas para variables cualitativas es muy sencillo en R. Por ejemplo, solicitar un diagrama de barras para los resultados de opinión emitidos por ejemplo, para la pregunta 12 de un test, puede hacerse de la siguiente manera.

```

rm(list=ls())

respuesta<-c("SI", "NO", "NO SE", "SI", "SI", "NO", "NO SE", "SI", "NO",
"SI")

table(respuesta)

barplot(table(respuesta),xlab="Respuesta del Encuestado", ylab="Frecuencias",
main = "Diagrama de barras", border = "blue", col = "yellow")

```

La gráfica que se obtiene se muestra en la Figura 2.2.

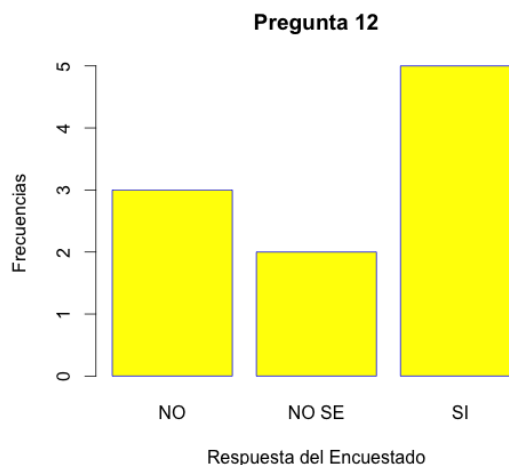
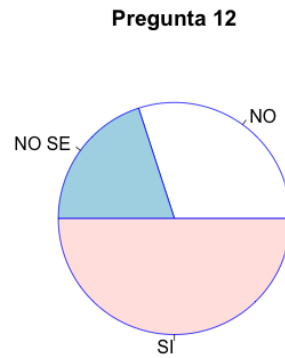


Figura 2.2: Diagrama de barras

Ahora, la construcción de un diagrama de pastel como el mostrado en la Figura 2.3, se obtiene tan sólo con la instrucción.

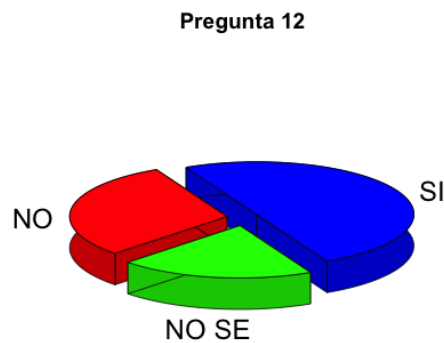
```
pie(table(respuesta),main = "Pregunta 12", border = "blue")
```



**Figura 2.3:** Diagrama de pastel

Este diagrama puede graficarse en tres dimensiones, como se muestra en la Figura 2.4, cargando previamente la librería *plotrix* y ejecutando la instrucción que se muestra enseguida.

```
require(plotrix)  
pie3D(table(respuesta), main = "Pregunta 12", start= 90, explode=0.1)
```



**Figura 2.4:** Diagrama de pastel

En ocasiones se desea un diagrama de barras para una variable cuantitativa. El procedimiento es el mismo, como se muestra a continuación, donde se capturaron los aciertos obtenidos en un test.

```

aciertos<-c(17,17,18,17,18,19,18,19,17,16,16,17,18,18,18,16,19,16,19,18,17,19,18)

table(aciertos)

barplot(table(aciertos), main="Aciertos en un Test", xlab="Aciertos",
ylab="Frecuencias", border = "red", col="blue")

```

La gráfica de barras resultante se muestra en la Figura 2.5.

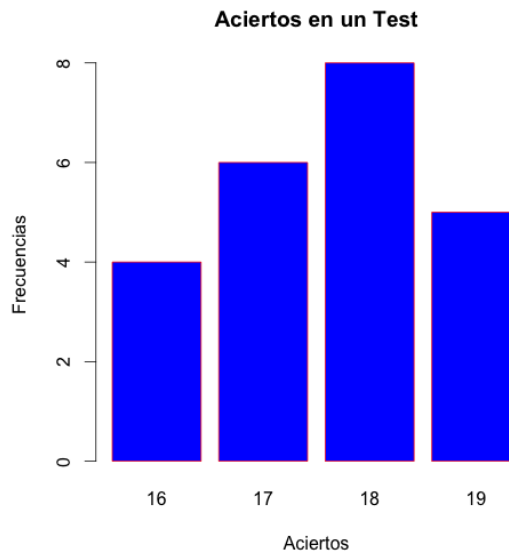


Figura 2.5: Diagrama de barras

## 2.2.2 Diagramas de caja y Diagramas de Tallo y Hojas

A continuación se muestra cómo construir un diagrama de caja, en forma horizontal, para una variable “x” donde se depositan ciertas mediciones. En este caso, se pide que el diagrama sea de color rojo en su interior y con un borde azul. La Figura 2.6 muestra la gráfica obtenida.

```

rm(list=ls())

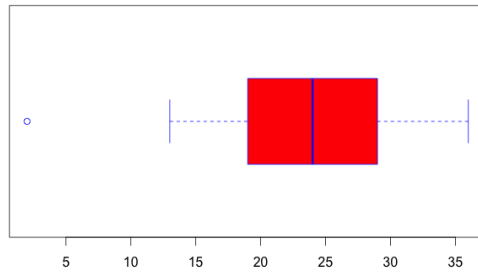
x<-c(2,13,18,19,23,22,19,19,19,20,21,23,24,24,25,27,28,28,29,29,32,32,33,35,36)

boxplot(x, horizontal=TRUE, border="blue", col="red")

```

En muchas situaciones es útil construir un diagrama de tallo y hojas, el cual se obtiene simplemente con la instrucción *stem(x)*. Para el caso de la variable “x”, este diagrama muestra en la Figura 2.7.





**Figura 2.6:** Diagrama de caja con color y borde específicos

```
> stem(x)
The decimal point is 1 digit(s) to the right of the |
0 | 2
1 | 389999
2 | 0123344578899
3 | 22356
```

**Figura 2.7:** Diagrama de tallo y hojas

Ahora depositaremos nuevas mediciones en una variable que denotamos por “y”, las cuales se muestran a continuación, y construiremos diagramas de caja para ambas variables “x” y “y”. Los diagramas se presentarán en forma horizontal y los etiquetaremos como ‘Grupo 1’ y ‘Grupo 2’. Agregaremos color, incluiremos la etiqueta ‘Edades’ en el eje X, y el título ‘Edades por Grupo’. Las instrucciones utilizadas se muestran a continuación.

```
y<-c(23,26,27,25,29,29,32,31,30,30,33,38,36,39,37,38,39,44,45,46,48)
boxplot(x,y,horizontal=TRUE,names=c(“Grupo 1”,“Grupo 2”),
border=9,col=c(“red”,“yellow”),xlab=“Edades”,main=“Edades por Grupo”)
```

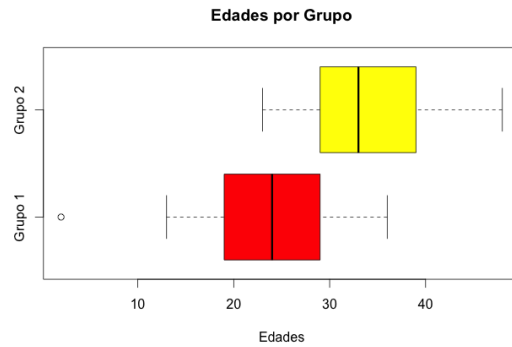
El resultado obtenido puede verse en la Figura 2.8

### 2.2.3 Histogramas

La construcción de histogramas en R es muy sencilla, basta con utilizar la instrucción

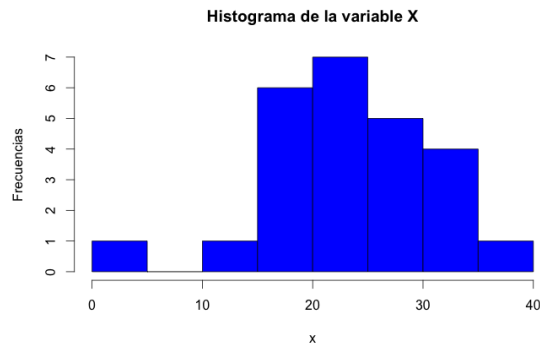
```
hist(x)
```

y obtendremos un histograma para la variable “x”, creada previamente. Podemos agregar color al histograma, un enunciado de ‘Frecuencias’ en el eje Y, así como el título ‘Histograma de la variable X’, de la siguiente manera:



**Figura 2.8:** Diagramas de caja en una sólo gráfica

```
hist(x, main="Histograma de la variable X", col="green", ylab = "Frecuencias", freq=TRUE)
```



**Figura 2.9:** Histograma para la variable X

Con la instrucción `par(mfrow=c(2,1))` podemos graficar dos histogramas en una sólo gráfica, uno abajo del otro. Ahora, si deseamos graficarlos uno enseguida del otro podemos conseguirlo de la siguiente manera:

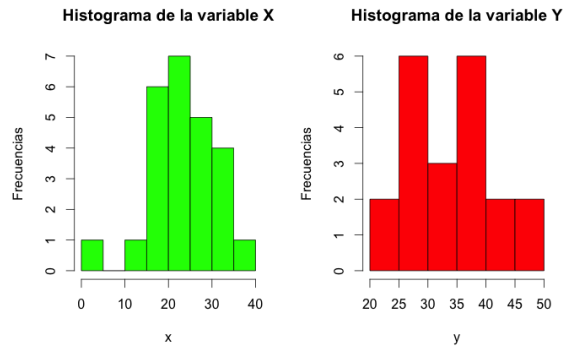
```
par(mfrow=c(1,2))
```

```
hist(x, main="Histograma de la variable X", col="green", ylab = "Frecuencias", freq=TRUE)
```

```
hist(y, main="Histograma de la variable Y", col="red", ylab = "Frecuencias", freq=TRUE)
```

Para graficar los dos histogramas y los dos diagramas de caja en una sólo gráfica, podemos correr las siguientes instrucciones:

```
par(mfrow=c(2,2))
```



**Figura 2.10:** Histogramas para las variables X e Y

```
hist(x, main="Histograma de la variable X", col="green", ylab = "Frecuen-
cias", freq=TRUE)
```

```
hist(y, main="Histograma de la variable Y", col="red", ylab = "Frecuencias",
freq=TRUE)
```

```
boxplot(x, main="Diagrama de Caja para X", col="green")
```

```
boxplot(y, main="Diagrama de Caja para Y", col="red")
```

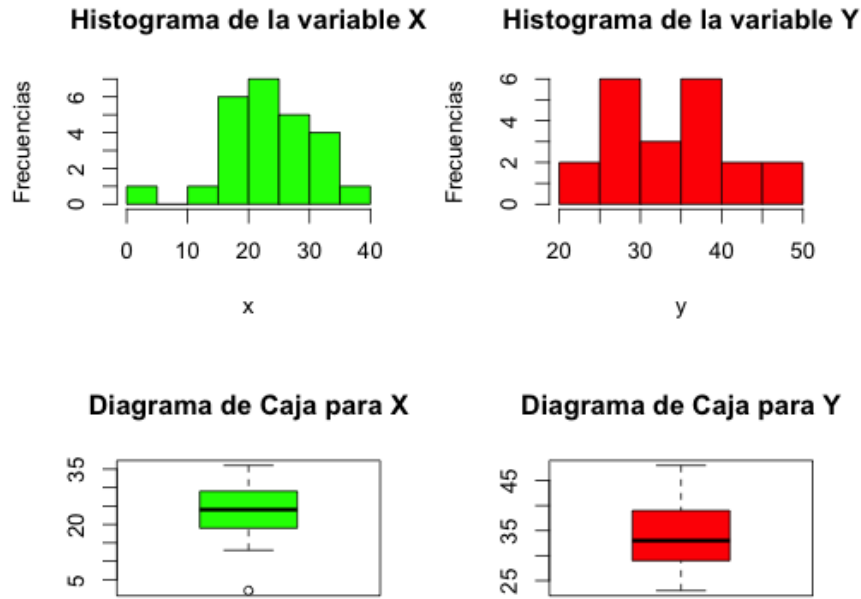
La gráfica resultante se muestra en la Figura 2.11. El presentar varias gráficas en una sola figura facilita la comparación de conjuntos de datos.

## 2.3 Diagramas de dispersión

Para ilustrar cómo construir diagramas de dispersión en R y calcular la recta de regresión lineal simple asociada a ciertos datos, utilizaremos un conjunto de datos muy recurrido al introducir el tema de regresión lineal, pues ilustra diferentes aspectos que debieran analizarse al ajustar un modelo de regresión, en este caso lineal. Este conjunto fue creado por Anscombe (1973), está contenido en R y basta llamarlo con el nombre de *anscombe* para ver cómo está conformado. El conjunto se compone de las variables que se muestran en la Figura 2.12.

Construiremos primero unos diagramas de dispersión con las instrucciones siguientes.

```
attach(anscombe)
par(mfrow=c(2,2))
plot(x1,y1)
plot(x2,y2)
plot(x3,y3)
```



**Figura 2.11:** Histogramas y diagramas de caja para las variables X e Y

```
> anscombe
  x1 x2 x3 x4  y1  y2  y3  y4
1 10 10 10  8  8.04 9.14  7.46  6.58
2  8  8  8  8  6.95 8.14  6.77  5.76
3 13 13 13  8  7.58 8.74 12.74  7.71
4  9  9  9  8  8.81 8.77  7.11  8.84
5 11 11 11  8  8.33 9.26  7.81  8.47
6 14 14 14  8  9.96 8.10  8.84  7.04
7  6  6  6  8  7.24 6.13  6.08  5.25
8  4  4  4 19  4.26 3.10  5.39 12.50
9 12 12 12  8 10.84 9.13  8.15  5.56
10 7  7  7  8  4.82 7.26  6.42  7.91
11 5  5  5  8  5.68 4.74  5.73  6.89
```

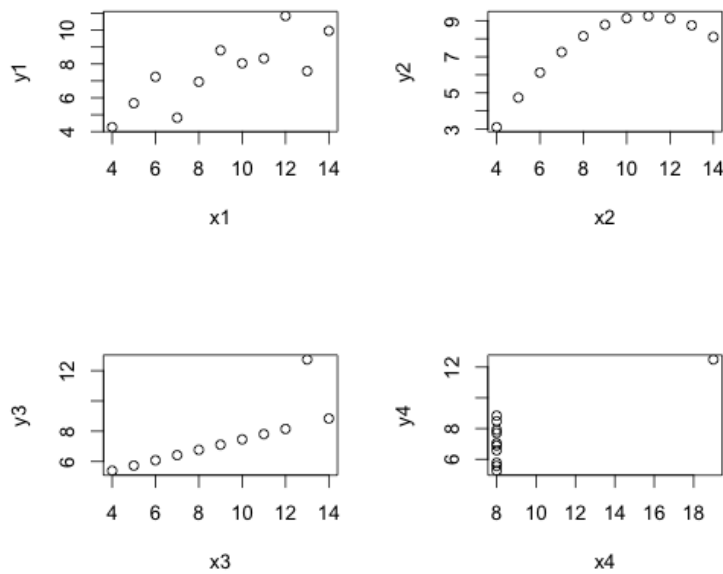
**Figura 2.12:** Conjunto de datos de Anscombe

```
plot(x4,y4)
```

La gráfica resultante podemos observarla en la Figura 2.13.

Si ahora se calcula la ecuación de regresión para cada uno de los conjuntos de parejas de datos, lo cual puede obtenerse con las instrucciones.

```
regresion1<-lm(y1 ~ x1)
```



**Figura 2.13:** Diagramas de dispersión

```
regresion2<-lm(y2 ~ x2)
```

```
regresion3<-lm(y3 ~ x3)
```

```
regresion4<-lm(y4 ~ x4),
```

El resultado obtenido es presentado en la Figura 2.14.

Analizando los resultados presentados en las Figuras 2.13 y 2.14, es comprensible el porqué este conjunto de datos es tan utilizado, ya que sus diagramas de dispersión son muy diferentes entre sí, pero conducen a la misma ecuación de regresión, y no sólo eso, el coeficiente de correlación también es el mismo, los resultados para cada uno de los elementos de la tabla de análisis de varianza de esta regresión, también son los mismos. La idea de utilizarlo es mostrar que un análisis estadístico debe combinar diversos aspectos que en su conjunto complementarán todo el análisis.

---

```
> regresion1

Call:
lm(formula = y1 ~ x1)

Coefficients:
(Intercept)      x1
    3.0001      0.5001

> regresion2

Call:
lm(formula = y2 ~ x2)

Coefficients:
(Intercept)      x2
    3.001      0.500

> regresion3

Call:
lm(formula = y3 ~ x3)

Coefficients:
(Intercept)      x3
    3.0025      0.4997

> regresion4

Call:
lm(formula = y4 ~ x4)

Coefficients:
(Intercept)      x4
    3.0017      0.4999
```

**Figura 2.14:** Ecuaciones de regresión

## Capítulo 3

# Variables Aleatorias y Distribuciones Muestrales

A continuación se muestra cómo calcular la densidad, la función de distribución, la función cuantil y generar diversas variables aleatorias. Estas funciones permitirán presentar, de manera más sencilla, el tema de distribuciones muestrales.

### 3.1 Variables Aleatorias

En R podemos calcular la función de densidad de probabilidad de una variable aleatoria ( $d$ ), la función de distribución ( $p$ ), es posible generar muestras de variables aleatorias ( $r$ ), calcular cuantiles ( $q$ ), etcétera, tanto en variables aleatorias discretas como continuas. Esto se obtiene escribiendo, por ejemplo, `dbeta()`, `pbeta()`, `qbeta()`, `rbeta()`, para una variable aleatoria que sigue una distribución Beta, o bien, `dnorm()`, `pnorm()`, `qnorm()`, `rnorm()` en el caso de una variable aleatoria con distribución normal.

A continuación veremos algunos casos de variables aleatorias, tanto discretas como continuas, que ejemplifican lo anterior.

#### 3.1.1 Poisson

Consideremos una variable aleatoria Poisson con media  $\lambda = 30$ . La Tabla 3.1 muestra las instrucciones que podemos utilizar en R, para calcular la probabilidad, función de distribución, función cuantil, así como generar muestras aleatorias en una Poisson.

#### 3.1.2 Binomial

En el caso de que  $X$  sea una variable aleatoria Binomial con parámetros  $N = 100$  y  $p = 0.3$ , las instrucciones se describen en la Tabla 3.2.

Instrucción	Descripción
dpois(25, $\lambda = 30$ )	Función de probabilidad evaluada en $X = 25$
ppois(25, $\lambda = 30$ )	Función de distribución $P(X \leq 25)$
qpois(0.95, $\lambda = 30$ )	Cuantil 0.95
rpois(50, $\lambda = 30$ )	Muestra de $n = 50$ observaciones de una v.a. Poisson

**Tabla 3.1:** Variable aleatoria Poisson

Instrucción	Descripción
dbinom(25, size=100, p=0.3)	Función de probabilidad evaluada en $X = 25$
pbinom(25, size=100, p=0.3)	Función de distribución $P(X \leq 25)$
qbinom(0.95, size=100, p=0.3)	Cuantil 0.95
rbinom(50, size=100, p=0.3)	Muestra de $n = 50$ observaciones de una v.a. binomial

**Tabla 3.2:** Variable aleatoria Binomial

### 3.1.3 Binomial Negativa

Si  $X$  es una variable aleatoria Binomial Negativa, la cual puede pensarse como el número de fallas antes de conseguir  $r$  éxitos, donde  $p$  es la probabilidad de éxito, esto es, la variable tiene parámetros  $r = 100$  y  $p = 0.3$ , las instrucciones son las descritas en la Tabla 3.3.

Instrucción	Descripción
dnbinom(25, size=100, p=0.9)	Función de probabilidad evaluada en $X = 25$
pnbinom(25, size=100, p=0.9)	Función de distribución $P(X \leq 25)$
qnbinom(0.95, size=100, p=0.9)	Cuantil 0.95
rnbinom(50, size=100, p=0.9)	$n = 50$ observaciones de una v.a. binomial negativa

**Tabla 3.3:** Variable aleatoria Binomial Negativa

A continuación mostramos cómo obtener lo anterior en el caso de algunas variables aleatorias continuas.

### 3.1.4 Uniforme

Supóngase que  $X$  es una variable aleatoria Uniforme con parámetros  $a = 0$  y  $b = 1$ , las instrucciones a seguir se muestran en la Tabla 3.4

### 3.1.5 Exponencial

Si  $X$  es ahora una variable aleatoria Exponencial con parámetro  $\alpha = E(X) = 10$ , las instrucciones serán las mostradas en la Tabla 3.5



Instrucción	Descripción
dunif(0.5, min=0, max=1)	Función de densidad evaluada en $X = 0.5$
punif(0.5, min=0, max=1)	Función de distribución $P(X \leq 0.5)$
qunif(0.95, min=0, max=1)	Cuantil 0.95
runif(50, min=0, max=1)	Muestra simulada de tamaño $n = 50$

**Tabla 3.4:** Variable aleatoria Uniforme

Instrucción	Descripción
dexp(10, rate=1/10)	Función de densidad evaluada en $X = 10$
pexp(10, rate=1/10)	Función de distribución $P(X \leq 10)$
qexp(0.95, rate=1/10)	Cuantil 0.95
rexp(50, rate=1/10)	Muestra simulada de tamaño $n = 50$

**Tabla 3.5:** Variable aleatoria Exponencial

### 3.1.6 Normal

Supóngase ahora que  $X$  es una variable aleatoria Normal con parámetros  $\mu = E(X) = 0$  y desviación estándar  $\sigma = \sqrt{Var(X)} = 1$ . Lo mostrado anteriormente podemos conseguirlo con las instrucciones que se presentan en la Tabla 3.6. Cabe mencionar que en el caso de la distribución normal estándar pueden obviarse los valores de mean=0 y de sd=1.

Instrucción	Descripción
dnorm(0, mean=0, sd=1)	Función de densidad evaluada en $X = 0$
pnorm(0, mean=0, sd=1)	Función de distribución $P(X \leq 0)$
qnorm(0.95, mean=0, sd=1)	Cuantil 0.95
rnorm(50, mean=0, sd=1)	Muestra simulada de tamaño $n = 50$

**Tabla 3.6:** Variable aleatoria Normal

### 3.1.7 Distribución Ji-Cuadrada

Consideremos ahora que  $X$  es una variable aleatoria que sigue una distribución Ji-Cuadrada con  $\nu$  grados de libertad. En esta distribución también se puede calcular lo anteriormente mostrado, como se puede observar en la Tabla 3.7

### 3.1.8 Distribución $t$ -Student

Si la variable aleatoria  $X$  se distribuye como una  $t$ -Student con  $\nu$  grados de libertad, las instrucciones a utilizar son muy similares a las mostradas anteriormente, como puede verse en la Tabla 3.8.

Instrucción	Descripción
ddhisq(5.99, df=2)	Función de densidad evaluada en $X = 5.99$
pchisq(5.99, df=2)	Función de distribución $P(X \leq 5.99)$
qchisq(0.95, df=2)	Cuantil 0.95
rchisq(50, df=2)	Muestra simulada de tamaño $n = 50$

**Tabla 3.7:** Variable aleatoria Ji-Cuadrada

Instrucción	Descripción
dt(0, df=10)	Función de densidad evaluada en $X = 0$
pt(0, df=10)	Función de distribución $P(X \leq 0)$
qt(0.95, df=10)	Cuantil 0.95
rt(50, df=10)	Muestra simulada de tamaño $n = 50$

**Tabla 3.8:** Variable aleatoria  $t$ -Student

### 3.1.9 Distribución F

Para el caso que  $X$  siga una distribución F o de Fisher, su densidad, función de distribución, función cuantil y la generación de números aleatorios es muy similar a lo mostrado previamente, como se observa en la Tabla 3.9

Instrucción	Descripción
df(2, df1=10, df2=15)	Función de densidad evaluada en $X = 2$
pf(2, df1=10, df2=15)	Función de distribución $P(X \leq 2)$
qf(0.95, df1=10, df2=15)	Cuantil 0.95
rf(50, df1=10, df2=15)	Muestra simulada de tamaño $n = 50$

**Tabla 3.9:** Variable aleatoria F

### 3.1.10 Ejercicios

1. Verificar qué se obtiene con cada una de las siguientes instrucciones. Interpretar los resultados.
  - (a) `norm <- rnorm(100, 2, 5)`
  - (b) `norm[1:10]`
  - (c) `dnorm(-1.96)`
  - (d) `dnorm(0)`
  - (e) `pnorm(1.959964)`
  - (f) `pnorm(0)`
  - (g) `pchisq(7,15)`

- (h) `rchisq(10,12)`
  - (i) `rt(15,25)`
  - (j) `pt(0,25)`
  - (k) `rf(15,5,7)`
  - (l) `df(5,5,7)`
2. Graficar una distribución normal con media de 10 y desviación estándar de 2.
  3. Graficar una distribución  $t$ -Student con  $\nu = 15$  grados de libertad.
  4. Graficar una distribución F con  $\nu_1 = 10$  y  $\nu_2 = 20$  grados de libertad. Analizar los cambios que esta distribución tiene al invertir los grados de libertad del numerador y del denominador.
  5. Graficar, en una sólo figura, distribuciones  $t$ -Student con 10, 20, 30, 50 y 100 grados de libertad. Analizar los resultados obtenidos.

## 3.2 Distribución muestral de la media

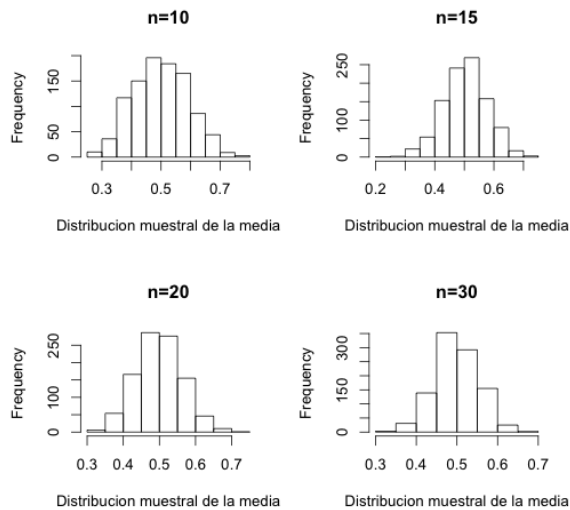
Para introducir el tema de la distribución de la media muestral, se simularán muestras de una uniforme y se analizará el comportamiento de la media de estas muestras, para diferentes tamaños. Por otra parte, para un conjunto particular de datos se tomarán muestras aleatorias de éste y se estudiará el efecto que tiene el variar el tamaño de estas muestras en la distribución de la media muestral.

A continuación se ilustra cómo generar mil muestras de tamaño  $n = 30$ , de una distribución uniforme, con parámetros  $a = 0$  y  $b = 1$ .

```
rm(list = ls(all = TRUE))
Vmedias<-numeric(1000)
for (i in 1:1000){muestra<-runif(30,0,1); Vmedias[i]<-mean(muestra)}
histogram(Vmedias, main="Distribucion muestral con n=30")
```

En la Figura 3.1 se muestra cómo cambia la forma del histograma cuando se varía el tamaño de muestra. Se trabajó con muestras de tamaños 10, 15, 20 y 30. Al calcular la media y desviación estándar del vector de medias obtenido, que hemos llamado “Vmedias”, lo cual se obtiene simplemente con las instrucciones “mean(Vmedias)” y “sd(Vmedias)”, obtenemos una media de 0.5002326 y una desviación estándar de 0.5002326, que son muy aproximadas a las que teóricamente podemos calcular.

Consideremos ahora el simular muestras de un conjunto de datos, que consideraremos como nuestra población y donde conocemos, por tanto, la media y desviación estándar verdadera. Para ello utilizaremos un conjunto de datos muy famoso, atribuido a Ronald Fisher (Fisher, 1936) y que se encuentra en R bajo el nombre de *iris*. Este conjunto de datos contiene el largo y ancho del sépalo, así



**Figura 3.1:** Distribución muestral de medias

como el largo y ancho del pétalo de tres especies de flores iris: setosa, virginica y versicolor. Trabajaremos con el ancho del pétalo de todas las flores, el cual tiene una media de 1.199333 y una desviación estándar de 0.7622377, las que consideraremos como  $\mu$  y  $\sigma$ , ya ésta será nuestra población de datos. Simularemos diez mil muestras de tamaños 10, 20, 30 y 100, para observar cómo influye el tamaño de la muestra en la distribución de la media muestral.

```
rm(list=ls())

attach(iris)

mu<-mean(iris$Petal.Width)

m<-10000

n<-10

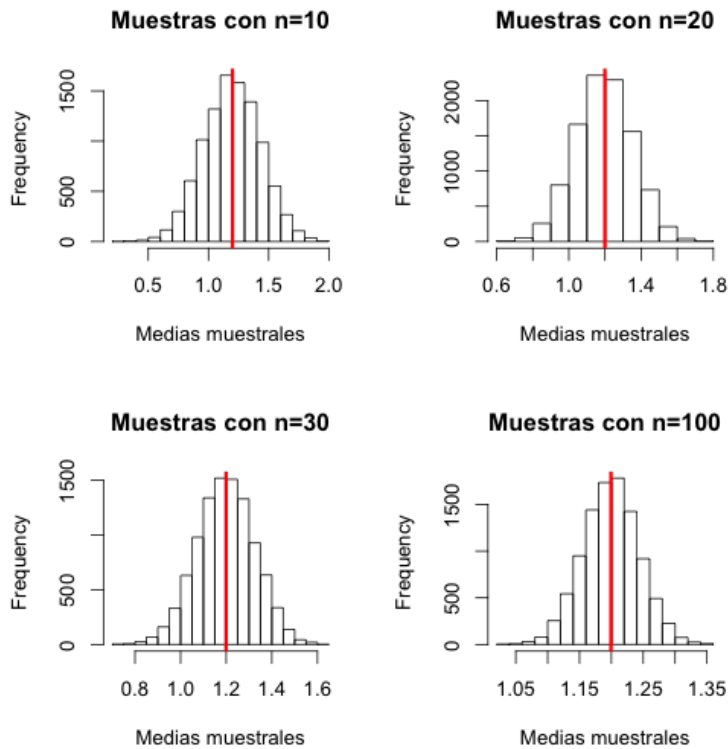
medias<-numeric(m)

for (i in 1:m) {Wpetal<-sample(iris$Petal.Width,n)
medias[i]<-mean(Wpetal)}

hist(medias)

abline(v=mu, col="red", lwd=3)
```

Los resultados de esta simulación se muestran en la Figura 3.2. En particular, la media del vector de diez mil medias de muestras de tamaño cien resulta 1.199543, que es muy cercano al valor  $\mu = 1.199333$ , que aparece en los histogramas con una línea de color rojo.



**Figura 3.2:** Distribución muestral de medias: Datos de Iris

### 3.2.1 Ejercicios

1. Utilizando el programa que se muestra al inicio de la Sección 3.2
  - (a) Ejecutar el programa simulando 5000 y 10000 muestras.
  - (b) Variar los tamaños de muestra, considerando tamaños de 5, 10, 50 y 100.
  - (c) Variar los valores de los parámetros y explicar el efecto de éstos en la distribución de las medias muestrales.
  - (d) Replicar este programa utilizando las distribuciones binomial, exponencial, Poisson y normal.
  - (e) Emitir conclusiones generales de los resultados obtenidos.
2. Analizar la distribución de la media de diez mil muestras de tamaños 5, 10, 50 y 100 de la variable largo del pétalo del conjunto de datos de las flores de iris, construyendo histogramas para cada uno de estos tamaños. Interpretar los resultados obtenidos.



## Capítulo 4

# Inferencia Estadística

### 4.1 Intervalos de Confianza y Pruebas de Hipótesis

En este capítulo se presenta cómo calcular intervalos de confianza y efectuar pruebas de hipótesis en R, para diversos parámetros. Dado que muchos de estos procedimientos requieren el supuesto de normalidad, se inicia presentando una de las pruebas de normalidad que R contiene, además de un método gráfico para verificarla.

#### 4.1.1 Prueba de Normalidad

El software R tiene algunas pruebas para verificar normalidad en los datos; una de ellas es la prueba de Shapiro-Wilk. Se mostrará el uso de ella verificando la normalidad del ancho del sépalo de las flores iris setosa, para lo cual efectuamos lo siguiente:

```
rm(list=ls())

shapiro.test(iris$Sepal.Width[iris$Species=="setosa"])

> rm(list=ls())
> shapiro.test(iris$Sepal.Width[iris$Species=="setosa"])

      Shapiro-Wilk normality test

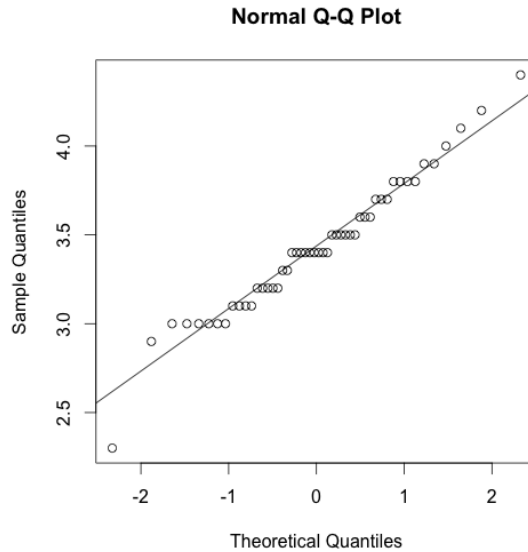
data:  iris$Sepal.Width[iris$Species == "setosa"]
W = 0.97172, p-value = 0.2715
```

**Figura 4.1:** Prueba Normalidad para ancho del sépalo en Iris Setosa

En la Figura 4.1, puede observarse que el supuesto de normalidad puede sostenerse dado que el p-valor resultante fue de 0.2715. Una gráfica que nos permite ver si los datos pueden estar distribuidos normalmente es la que se obtiene con la siguiente instrucción y cuyo resultado se muestra en la Figura 4.2, que confirma la conclusión anterior.

```
qqnorm(iris$Sepal.Width[iris$Species=="setosa"])
```

```
qqline(irisSepal.Width[irisSpecies=="setosa"])
```



**Figura 4.2:** Q-Q plot para el ancho del sépalo en Iris Setosa

#### 4.1.2 Intervalo de confianza y Prueba de hipótesis para una media

A continuación se muestra cómo calcular un intervalo de confianza para una media, utilizando los datos correspondientes al ancho del sépalo de la iris virginica. Dado que se requiere el supuesto de normalidad, se realiza una prueba de Shapiro-Wilk y al no rechazar ésta ( $p$ -valor=0.189), como puede observarse en la Figura 4.3, se calcula el intervalo del 95% de confianza para la media, partiendo de un estadístico que sigue una distribución  $t$ -Student, que supone una varianza poblacional desconocida. El intervalo resultante es (2.8823, 3.0656).

```
rm(list=ls())
shapiro.test(irisSepal.Width[irisSpecies=="virginica"])
t.test(irisSepal.Width[irisSpecies=="virginica"])
```

Para efectuar una prueba de hipótesis es necesario tener un interés específico en contrastar cierta hipótesis y verificar los supuestos que la prueba requiera. Por ejemplo, si se desea probar que la duración promedio en horas, de cierto componente electrónico es mayor que 400 horas y se cuenta con una muestra aleatoria de diez mediciones, la prueba de hipótesis puede realizarse como se indica a continuación, donde previamente se verifica la normalidad de los datos.

```
sobrevivencia<-c(207,381,411,673,534,294,697,344,418,554)
```



```

Console ~/ |
> rm(list=ls())
> shapiro.test(iris$Sepal.Width[iris$Species=="virginica"])

      Shapiro-Wilk normality test

data:  iris$Sepal.Width[iris$Species == "virginica"]
W = 0.96739, p-value = 0.1809

> t.test(iris$Sepal.Width[iris$Species=="virginica"])

      One Sample t-test

data:  iris$Sepal.Width[iris$Species == "virginica"]
t = 65.208, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 2.882347 3.065653
sample estimates:
mean of x
 2.974

> |

```

**Figura 4.3:** Intervalo para una media

```
shapiro.test(sobrevivencia)
```

```
t.test(sobrevivencia,mu=400,alternative="greater")
```

Los resultados obtenidos se muestran en la Figura 4.4. Nótese que la misma función de R,  $t.test(x)$  permite realizar pruebas de hipótesis y calcular intervalos de confianza. En el caso de especificar una hipótesis alternativa unilateral, el intervalo resultante será también unilateral, como puede verse en la Figura 4.4.

```

Console ~/ |
> sobrevivencia<-c(207,381,411,673,534,294,697,344,418,554)
> shapiro.test(sobrevivencia)

      Shapiro-Wilk normality test

data:  sobrevivencia
W = 0.95778, p-value = 0.7603

> t.test(sobrevivencia,mu=400,alternative="greater")

      One Sample t-test

data:  sobrevivencia
t = 1.0136, df = 9, p-value = 0.1686
alternative hypothesis: true mean is greater than 400
95 percent confidence interval:
 358.5268      Inf
sample estimates:
mean of x
 451.3

```

**Figura 4.4:** Prueba de hipótesis para una media

En el caso de querer efectuar una prueba de hipótesis para una media, en una población normal con varianza conocida, puede utilizarse la instrucción  $z.test(data, mu = mu_0, sigma.x=sd_0, alternative = c("two.sided", "less", "greater"))$ .

### 4.1.3 Intervalo de confianza y Prueba de hipótesis para una proporción

En R es muy sencillo construir intervalos de confianza para una proporción o bien realizar una prueba de hipótesis para este parámetro. Por ejemplo, si la hipótesis de investigación es el probar si más del 5% de las personas sufren de alguna reacción alérgica al tomar un medicamento y se cuenta con una muestra aleatoria de cien personas, donde se observó que 35 de ellas mostraron signos de alergia, la prueba de hipótesis y el intervalo de confianza bilateral, pueden obtenerse con las siguientes instrucciones.

```
x = 35
```

```
n = 100
```

```
binom.test(x, n, p = 0.05, alternative = "greater", conf.level = 0.95)
```

```
binom.test(x, n, p = 0.05, alternative = "two.sided", conf.level = 0.95)
```

```

Console ~/
> x=35
> n=100
> binom.test(x, n, p = 0.05, alternative = "greater", conf.level = 0.95)

      Exact binomial test

data:  x and n
number of successes = 35, number of trials = 100, p-value < 2.2e-16
alternative hypothesis: true probability of success is greater than 0.05
95 percent confidence interval:
 0.2707545 1.0000000
sample estimates:
probability of success
              0.35

> binom.test(x, n, p = 0.05, alternative = "two.sided", conf.level = 0.95)

      Exact binomial test

data:  x and n
number of successes = 35, number of trials = 100, p-value < 2.2e-16
alternative hypothesis: true probability of success is not equal to 0.05
95 percent confidence interval:
 0.2572938 0.4518494
sample estimates:
probability of success
              0.35

```

**Figura 4.5:** Intervalo de confianza y Prueba de hipótesis para una proporción

En la Figura 4.5 podemos ver que el p-valor obtenido ( $p = 2.2^{-16}$ ) proporciona evidencia para rechazar la hipótesis nula  $H_0 : p = 0.05$ . Por otra parte, el intervalo de confianza para la proporción  $p$ , resulta  $(0.2572, 0.4518)$ , con el cual obtenemos la misma conclusión.

#### 4.1.4 Prueba de hipótesis para comparar dos varianzas

En ocasiones se requiere comparar las varianzas de dos poblaciones, ya sea por ser el problema de interés o por ser un supuesto necesario para efectuar alguna otra prueba, como es el caso de la prueba *t*-Student para comparar dos medias. En R es muy sencillo efectuar este tipo de pruebas y una manera de hacerlo es como se muestra enseguida, donde se está comparando la variabilidad del ancho del sépalo de las flores iris setosa e iris virginica.

```
rm(list=ls())

var.test(iris$Sepal.Width[iris$Species=="setosa"],
iris$Sepal.Width[iris$Species=="virginica"],ratio=1)
```

```

      F test to compare two variances

data:  iris$Sepal.Width[iris$Species == "setosa"] and iris$Sepal.Width[iris$Species == "virginica"]
F = 1.3816, num df = 49, denom df = 49, p-value = 0.2614
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.7840128 2.4346017
sample estimates:
ratio of variances
      1.381578
```

**Figura 4.6:** Prueba F de comparación de varianzas

Los resultados de esta prueba, que se muestran en la Figura 4.6, proporcionan evidencia para no rechazar el supuesto de homogeneidad de varianzas para las mediciones del ancho del sépalo de los dos tipos de flores iris.

#### 4.1.5 Intervalo de confianza y Prueba de hipótesis para dos medias

En el caso de contar con dos muestras aleatorias independientes, seleccionadas de dos poblaciones distribuidas normalmente, en las cuales se desea comparar las medias, también podemos utilizar la prueba *t.test* que incluye R. Para ello es necesario saber si las varianzas pueden o no suponerse homogéneas, por lo cual es necesario realizar primero una prueba de hipótesis para comparar estas varianzas.

A continuación se ilustra cómo realizar esta prueba utilizando las calificaciones obtenidas en el test de inteligencia Wechsler, para dos grupos independientes de personas, grupos que se obtienen de acuerdo a la clasificación de si estas personas son o no fumadores de mariguana. El grupo de no fumadores afirma nunca haber utilizado mariguana, mientras que el grupo de fumadores declara utilizarla regularmente. Los conjuntos de datos satisfacen la normalidad, por lo que las medias pueden compararse mediante una prueba *t* de Student, como se muestra a continuación.

```
rm(list=ls())

no_fumadores= c(18,22,21,17,20,17,23,20,22,21)

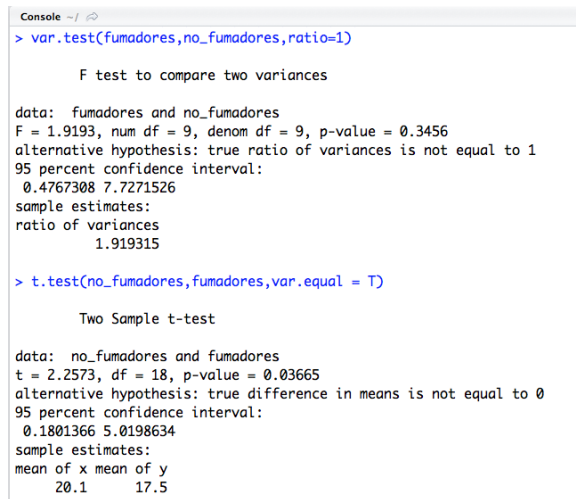
fumadores =c(16,20,14,21,20,18,13,15,17,21)
```

```
var.test(fumadores,no_fumadores,ratio=1)

t.test(no_fumadores,fumadores,var.equal = T)
```

Como puede observarse en la Figura 4.7, la prueba para comparar varianzas arroja un p-valor de 0.3456, por lo que no se rechaza la homogeneidad de varianzas. Luego, en la prueba  $t$  para comparar medias puede establecerse este supuesto, utilizando la instrucción `var.equal=T`. Al nivel de significancia de 0.05, la prueba  $t$  indica que la muestra proporciona evidencia para rechazar la igualdad de medias.

En caso de no cumplir la homogeneidad de varianzas se debe establecer `var.equal=F` y el software R efectuará la prueba de Welch, la cual es una adaptación de la prueba  $t$  con varianzas no-homogéneas.



```
Console ~/
> var.test(fumadores,no_fumadores,ratio=1)

      F test to compare two variances

data:  fumadores and no_fumadores
F = 1.9193, num df = 9, denom df = 9, p-value = 0.3456
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4767308 7.7271526
sample estimates:
ratio of variances
      1.919315

> t.test(no_fumadores,fumadores,var.equal = T)

      Two Sample t-test

data:  no_fumadores and fumadores
t = 2.2573, df = 18, p-value = 0.03665
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1801366 5.0198634
sample estimates:
mean of x mean of y
      20.1      17.5
```

**Figura 4.7:** Intervalo de confianza y Prueba de hipótesis para dos medias

#### 4.1.6 Análisis de Varianza

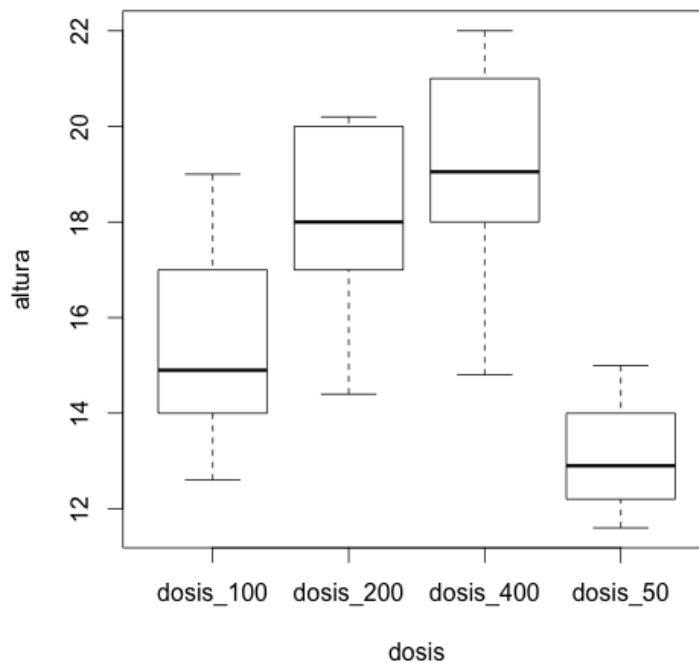
En esta sección veremos cómo efectuar un análisis de varianza en una clasificación. La prueba que utilizaremos nos permite comparar varias medias de muestras independientes, bajo el supuesto de que éstas provienen de poblaciones normales, que satisfacen la homogeneidad de varianzas.

A continuación se muestran mediciones de la altura alcanzada por cierto tipo de planta, al utilizar cuatro diferentes cantidades de nitrógeno en un diseño completamente al azar. Las dosis de nitrógeno representan, en este caso, los tratamientos. Una vez verificado el supuesto de normalidad, la comparación de medias se puede realizar mediante las siguientes instrucciones.

```
rm(list=ls())

altura<-c(12.4,12.8,12.2,13,14,14.2,11.6,15,12,13.2,16,12.6,14.8,13,14,
15,14,17,18,19,17.8,14.4,20,15.8,17,20,19.6,18,
```

```
20.2,18,21,14.8,19.1,15.8,18,20,21.1,22,19,18.2)
dosis<-c(rep("dosis_50",10),rep("dosis_100",10),rep("dosis_200",10),rep("dosis_400",10))
rendimiento=data.frame(altura,dosis)
rendimiento
plot(altura ~dosis,data=rendimiento)
bartlett.test(altura ~dosis,data=rendimiento)
aov.out<-aov(altura ~dosis,data=rendimiento)
summary(aov.out)
TukeyHSD(aov.out,conf.level=0.99)
```



**Figura 4.8:** Diagramas de caja del crecimiento de la planta

Como puede verse en la Figura 4.9, la prueba de Bartlett (*bartlett.test()*) que permite verificar la homogeneidad de varianzas, cuando se analizan varias muestras, arroja evidencia para no rechazar este supuesto de homogeneidad ( $p$ -valor=0.185). Por otra parte, el análisis de varianza resulta significativo, por lo cual se rechaza la hipótesis de igualdad de medias.

```

> bartlett.test(altura-dosis,data=rendimiento)

Bartlett test of homogeneity of variances

data: altura by dosis
Bartlett's K-squared = 4.8838, df = 3, p-value = 0.1805

> aov.out<-aov(altura~dosis,data=rendimiento)
> summary(aov.out)
      Df Sum Sq Mean Sq F value Pr(>F)
dosis   3  214.7   71.57  19.35 1.21e-07 ***
Residuals 36  133.1    3.70
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> TukeyHSD(aov.out,conf.level=0.99)
Tukey multiple comparisons of means
 99% family-wise confidence level

Fit: aov(formula = altura ~ dosis, data = rendimiento)

$dosis
      diff      lwr      upr    p adj
dosis_200-dosis_100  2.74 -0.1359887  5.6159887 0.0150760
dosis_400-dosis_100  3.56  0.6840113  6.4359887 0.0011022
dosis_50-dosis_100  -2.30 -5.1759887  0.5759887 0.0521923
dosis_400-dosis_200  0.82 -2.0559887  3.6959887 0.7762839
dosis_50-dosis_200  -5.04 -7.9159887 -2.1640113 0.0000062
dosis_50-dosis_400  -5.86 -8.7359887 -2.9840113 0.0000003

```

**Figura 4.9:** Análisis de varianza en una clasificación

Fertilizante A	50	52	59	56	55	51	56
Fertilizante B	57	58	61	59	54	68	
Fertilizante C	50	52	53	51	57		

**Tabla 4.1:** Rendimiento de trigo

Para verificar ahora cuál o cuáles de estas medias son diferentes entre sí, se utilizó la prueba de Tukey que permite comparar las medias de todas las parejas de grupos posibles. La hipótesis de igualdad entre un par de medias se rechazará cuando el p-valor asociado a esta prueba sea muy pequeño, digamos menor a 0.05, como ocurre en cuatro de los seis casos, tal cual puede apreciarse en la Figura 4.9. La prueba de Tukey es una de varias de las pruebas de comparaciones múltiples que R incluye.

#### 4.1.6.1 Ejercicios

- Se desea comparar la efectividad de tres fertilizantes A, B y C, los cuales se asignaron a 7, 6 y 5 diferentes parcelas, respectivamente, siguiendo un diseño completamente al azar. Los rendimientos de trigo, en quintales métricos por hectárea para las 18 parcelas, fueron los siguientes:
  - Comparar los rendimientos promedio de trigo con los tres diferentes fertilizantes y emitir una conclusión utilizando un nivel de significancia de 0.05. Deben verificarse todos los supuestos necesarios, realizarse un análisis descriptivo y explicar los resultados obtenidos.
- Mencionar tres pruebas de comparaciones múltiples incluidas en R y explicar cómo, y bajo qué condiciones se utilizan.

# Bibliografía

Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, 27:17–21.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 2:179–188.