Asignatura: **Bioestadística**

Fundamentos de Análisis Estadístico de Datos Experimentales (FAEDE)

Juana María Vivo

Dpto. Estadística e Investigación Operativa

Índice

1. Revisión del lenguaje de programación estadístico R

R (*https://cran.r-project.org/*) es un lenguaje de programación orientada a objetos derivado de dos lenguajes existentes, S (Becker, Chambers and Wilks 1985) and Scheme (Steel and Sussman 1975). Concretamente, R presenta una apariencia similar a S mientras que la implementación y semántica subyacente se derivan de Scheme. R es un lenguaje que permite implementar interactivamente técnicas para el análisis estadístico de datos y gráficos, El objetivo en esta primera sección es proporcionar una perspectiva general de las posibilidades de R.

Figura 1: Página principal de **The R Project for Statistical Computing**.

Entre las principales ventajas, en las que se basa el desarrollo continuo y la acelerada expansión de R, cabe señalar las siguientes:

(1) R está disponible como software libre en el proyecto General Public Licence (GNU) de Free Software Foundation (accesible desde *http://www.gnu.org/*) en forma de código fuente, y en una amplia variedad de plataformas tales como UNIX y sistemas similares (incluyendo FreeBSD y Linux), Windows y MacOS. R version 4.3.2 (octubre-2023) es la última versión dsiponible en *https://www.r-project.org/*.

(2) Entorno flexible que integra una multitud de comandos y funciones básicas en la librería denominada *base*, la cual constituye el núcleo de R, que permite incorporar nuevas técnicas de análisis de datos. Basta teclear *library(help = "base")* para visualizar el listado completo de funciones. Además, complementando el base-paquete existen, a libre disposición de los usuarios, numerosas librerías específicas con las técnicas estadísticas y gráficas disponibles y con explicación de su uso, lo que amplía su cobertura y aplicación en diversos campos.

(3) Enorme calidad del apoyo y soporte disponible. Entre otros, se recomienda: An Introduction to R, accesible desde *https://cran.r-project.org/doc/manuals/r-release/R-intro.html*, como guía para principiantes en R. Este manual editado por el *R Development Core Team*, proporciona información sobre programación,

así como, técnicas estadísticas y gráficas para la versión de R más reciente (R-release), para la versión actual parcheada (R-patched) y para la versión en desarrollo (R-devel) en Linux. Los manuales para Mac o Windows se encuentran disponibles en su correspondiente instalación. Más documentación en *https://cran.r-project.org/manuals.html* y en *https://cran.r-project.org/other-docs.html*. También, es interesante tener en cuenta la información disponible en *https://datos.gob.es/es/noticia/cursos-para-aprender-mas-sobre-r* sobre cursos y libros online de acceso gratuito.

Instalación de R

La instalación de R es prácticamente automática, si se dispone de acceso a internet. Desde la página oficial *https://www.r-project.org/* (Figura 1), basta clicar en CRAN (Comprehensive R Archive Network) bajo Download, y seleccionar del listado de mirrors el más cercano a tu localización geográfica (en nuestro caso, el servidor en Madrid: *https://cran.rediris.es/*). En Download and Install R de la nueva página que emerge, clicar en el enlace Download R for Linux, Download R for (Mac) OS X o Download R for Windows. Para esta última opción, el proceso de instalación se completaría clicando en el link install R for the first time, y posteriormente en Download R 4.0.2 for Windows, para descargar el fichero ejecutable R-4.0.2-win.exe. Para más detalles sobre la instalación de R en cada una de estas plataformas, consultar la información suministrada por el *R Development Core Team* en *http://cran.rediris.es*.

Trabajando con RStudio...

De la diversidad de interfaces de R sobre los diferentes sistemas operativos, la inclinación actual generalizada por RStudio como plataforma de interacción es significativa. RStudio se conecta con R, esto es, cuando tú escribes y ejecutas una línea de comando en RStudio, éste se la envía a R. En este sentido, se recomienda disponer de la última versión de R intalada ya que RStudio detectará y trabajará siempre con la más reciente. La versión en prueba de RStudio es gratuita y está disponible en *http://www.rstudio.org/*.

La barra de menú principal está constituida por once menús: *File*, *Edit*, *Code*, *View*, *Plots*, *Session*, *Build*, *Debug*, *Profile*, *Tools* y *Help*.

El diseño estándar de RStudio se compone de los cuatro paneles siguientes:

1. En la zona superior izquierda se encuentra el editor de código, para crear, abrir y editar ficheros de código R (con extensión *.R*), denominados *script*, así como para visualizar datos. Los scripts suelen incluir comentarios explicativos, necesariamente precedidos de una celdilla.

2. En la zona inferior izquierda se localiza la consola de R, *R-consola*, fácilmente reconocible por el promt ">", que es el espacio de trabajo interactivo.

3. La zona superior derecha generalmente dispone de dos pestañas diferenciadas:

   - *Environment*, donde aparece la lista de los objetos creados en memoria, que se puede guardar en un fichero con extensión *.RData* (o *.rda*).
   - *History*, que contiene el histórico de las líneas de código ejecutadas en R, que se puede guardar en un fichero con extensión *.Rhistory*.

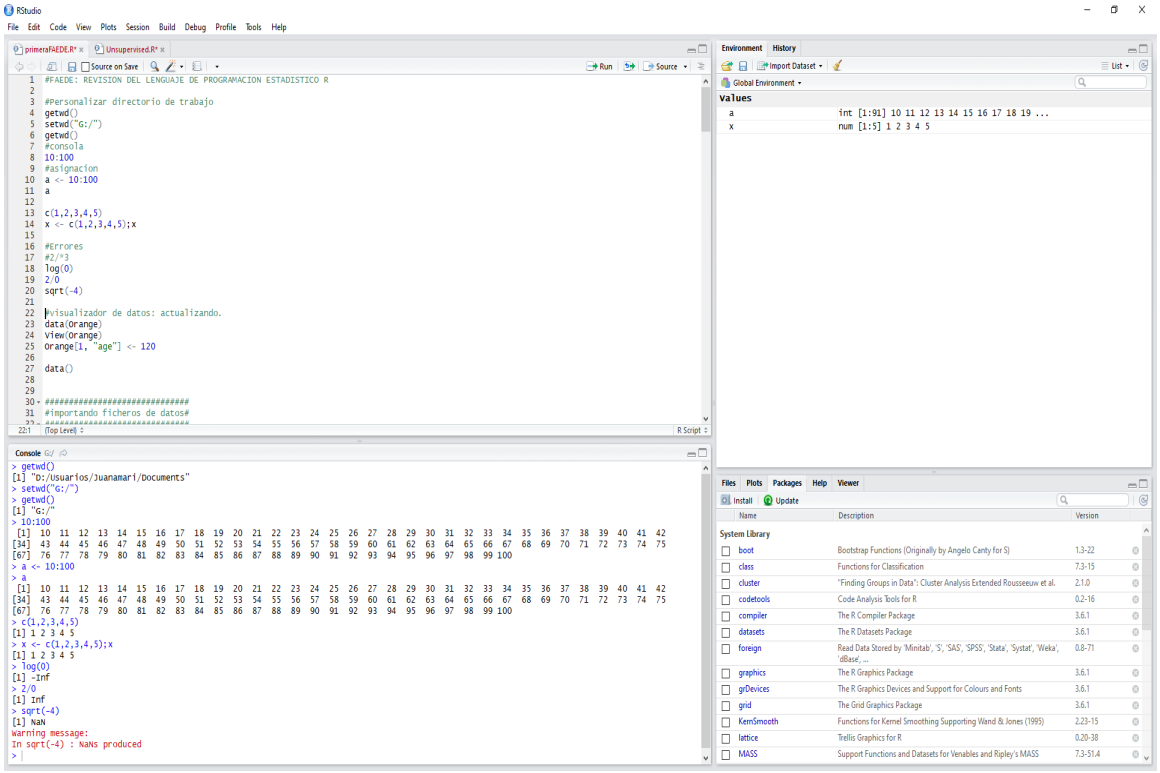


Figura 2: Sesión de trabajo en RStudio.

4. La zona inferior derecha dispone de cinco pestañas diferenciadas:

   - *Files*, que da acceso al árbol de directorios y ficheros del disco duro.
   - *Plots*, donde aparecen los gráficos creados en la consola, que se pueden guardar en diferentes formatos.
   - *Packages*, que facilita la administración de los paquetes de R instalados.
   - *Help*, en el que se abren las páginas de ayuda.
   - *Viewer*, que permite visualizar contenidos de una web local.

Se recomienda acceder a la cheat sheet (*https://github.com/rstudio/cheatsheets/blob/main/rstudio-ide.pdf*) con las funciones más útiles y lista de métodos abreviados de teclado implementados en RStudio.

Como introducción al procedimiento general para realizar un análisis de datos en R, empezaremos con un ejemplo sencillo que requiere solamente el paquete *base*, simultáneamente se repasarán conceptos básicos de estadística. Posteriormente, explicaremos algunos conceptos como fórmulas y funciones genéricas, que son útiles en programación con R, independientemente del tipo de análisis realizado. Concluiremos con un vistazo rápido a los diferentes paquetes.

En adelante, como pauta de buenas prácticas, se adoptará la siguiente rutina de trabajo al inicio de una sesión de trabajo con R.

```
#Identificación del directorio de trabajo al iniciar R.
 getwd()

    [1] "C:/Documents and Settings/umu/Mis documentos"
```

```
#Personalización del directorio de trabajo.
 setwd("C:/Master/Bioinf/PracticaR")
```

En este contexto, cabe destacar que puede resultar cómodo recuperar el área de trabajo de una sesión anterior desde el menú File, i.e., cargar scripts (*.R*), objetos y ficheros de datos *.RData* y líneas de comandos (*.Rhistory* grabados en ocasiones anteriores.

Paquetes de R

A pesar de que R dispone de una gran cantidad de funciones básicas para el análisis de datos, la gran mayoría de los métodos estadísticos disponibles en R están distribuidos en más de 10.000 paquetes, *packages*, accesibles en el CRAN. Un *paquete* o *librería* es básicamente un conjunto de funciones que se incorporan a R a través del comando *library("package")*. Algunos de estos paquetes vienen instalados junto con el paquete base, otros están dentro del grupo recommended ya que usan métodos habituales en estadística, y finalmente mucho otros paquetes están dentro del grupo contributed. Así pues, para ampliar la funcionalidad de R se deben instalar paquete adicionales, que se debe cargar para su uso.

```
#Instalación de un paquete
 install.packages("nombrepaquete")
#Para cargar una librería instalada
 library("nombrepaquete")
#Para enumerar las librerías instaladas
 installed.packages( )
#Para actualizar las librerías instaladas
 update.packages( )
#Para eliminar librerías instaladas
 remove.packages("nombrepaquete")
```

Calculando con R

Como uso más básico de la *R-consola*, se puede llevar a cabo operaciones interactivamente, utilizándola como una calculadora científica. Por ejemplo:

```
> 1+2 # Escribir 1+2 y pulsar Enter
[1] 3
```

Ahora, probar con los siguientes input en línea e identificar los operadores:

```
> 3*7
> 7/4
> 7%%4
> 7%%4
> 7^ 3
> exp(1)
> ln(1)
```

```
> sqrt(36)
> abs(-4)
```

También, los input pueden producir errores como en los siguientes casos:

```
> log(0)
[1] -Inf
> 2/0
[1] Inf
> sqrt(-4)
[1] NaN
Warning message:
In sqrt(-4) : NaNs produced
```

Para ampliar información sobre funciones matemáticas en R, teclear:

```
> help("Math")
> help("Special")
> help("Arithmetic")
> help("Trig")
```

Estructuras de datos en R

El tratamiento de datos R requiere tener en cuenta el tipo de objetos que se está manejando. Cabe señalar que para definir un objeto se utiliza el comando de R de la asignación "<-".

Vectores

Una primera estructura que se puede considerar es el vector, i.e., una colección ordenada de elementos de la misma naturaleza.

Ejemplo 1.1 *Ejemplos de vectores:*

```
> c(1,2,3,4,5) # vector numérico
> c(F,T,T,F,F) # vector lógico
> c("Juan","Pepe","Pedro","Antonio") # vector de caracteres
```

En los ejemplos anteriores, se ha tenido en cuenta la función de concatenación c, pero cabe señalar que también se pueden escribir utilizando *":"*.

Ejemplo 1.2 *Más ejemplos de vectores:*

```
> 1:5
> 5:1
> c(1:5,10:5,12)
```

Observación 1.1 *Nótese que la función c puede ser utilizada para concatenar vectores como en el ejemplo que se muestra a continuación:*

```
> x<-c(1,2,3)
> y<-c(T,F,T)
```

```
> c(x,y)
> [1] 1 2 3 1 0 1
```

Para generar vectores también es habitual el uso de función tales como `seq(from, to, by, length.out, along.with, ...)`, `rep(x, times = 1, length.out = NA, each = 1)` o `sequence(nvec)`.

Ejemplo 1.3 *Ejecutar las siguientes líneas de comando:*

```
> 1:10
> seq(from=0, to=10)
> seq(0, 10, 0.5)
> seq(from=5, by=-0.5, length.out=7)
> rep(1:4, 2)
> rep(1:4, each = 2)
> rep(1:4, c(2,2,2,2))
> rep(1:4, c(2,1,2,1))
> rep(1:4, each = 2, len = 4)
> rep(1:4, each = 2, len = 10)
> rep(1:4, each = 2, times = 3)
> sequence (c(4,3))
```

Factores

Este segundo tipo de objeto es también un vector pero de cadenas de caracteres que permite representar datos cualitativos `factor(x = character(), levels, labels = levels, exclude = NA, ordered = is.ordered(x), nmax = NA)` y se utiliza para codificar un vector como factor.

Ejemplo 1.4 *Ejecutar la siguiente línea de comando:*

```
> factor(letters[1:20, labels="letter"])
```

Listas

Una lista es una colección de elementos de distinta naturaleza. La función de uso es `list`.

Ejemplo 1.5 *Los elementos de una lista pueden ser obtenidos mediante el operador* $.

```
> milista=list(hombre="Pedro", mujer="María", casados=T, n.hijos=3, edad.hijos=c(1,2,4))
> milista
$hombre
[1] "Pedro"
$mujer
[1] "María"
$casados
[1] TRUE
$n.hijos
[1] 3
```

```
$edad.hijos
[1] 1 2 4
```

Matrices y arrays

Otras estructuras usuales de datos son las matrices y arrays. La función de uso para crear una matriz de datos es `matrix(data, nrow, ncol, byrow=F)`, cuyos argumentos corresponden al orden de la matriz, i.e., número de filas y de columnas, respectivamente. Nótese que la disposición por defecto es por columnas.

Ejemplo 1.6 *Obtener las salidas correspondientes a los siguientes comandos:*

```
> matrix(1:12)
> matrix(1:12,nrow=3)
> m <- matrix(1:12, nrow=3, byrow=T)
> colnames(m) <- c("Dato 1", "Dato 2", "Dato 3", "Dato 4")
> rownames(m) <- c("Primero", "Segundo", "Tercero")
```

Ejemplo 1.7 *Identifica la utilidad de las siguientes funciones sobre matrices, ejecutando las siguientes líneas:*

```
> dim(m)
> length(m)
> m[1,]
> m[,1:3]
> m[-1,]
> rbind(m,c(1,1,1,1))
> cbind(m,c(1,1,1))
> apply(m,1,summary)
> apply(m,2,summary)
```

La generalización de las matrices al caso multidimensional se denomina array, cuyo comando de uso es `array(datos, dimensiones)`.

Ejemplo 1.8 *Ejecutar la siguiente línea de comando:*

```
> array(1:12,c(2,3,2))
```

Data frames

Por último, consideramos los data frames que también constituyen una generalización de las matrices, en el sentido de que las columnas pueden ser unas cualitativas y otras cuantitativas, como en el siguiente ejemplo.

Ejemplo 1.9 *Ejecutar los siguientes comandos:*

```
> L3 <- LETTERS[1:3]
> fac <- sample(L3, 10, replace = TRUE)
> (d <- data.frame(x = 1, y = 1:10, fac = fac))
> is.data.frame(d)
```

Gráficos en R

La herramienta gráficas de R constituyen una de las componentes más potentes de este lenguaje, ya que incluye una amplia variedad de funciones para realizar gráficas estadísticas, lo que nos permite crear de manera automática desde gráficos sencillos hasta figuras de gran calidad para artículos y libros.

La salida de estos gráficos se muestra, por defecto en la pantalla, presentando la posibilidad de exportación de gráficas en distintos formatos ('postscript', 'pdf', 'png', 'jpeg', 'bmp',...). En este sentido, puede resultar interesante echar un vistazo a la demo de gráficos con colores *demo("graphics")*.

Algunas funciones gráficas usuales de R son las que se recogen en la siguiente tabla:

1-dim	2-dim	3-dim
barplot	plot	outer
boxplot	curve	persp
hist	lines	interp
piechart	points	image
stripchart	abline	scatterplot

Cabe destacar que las funciones gráficas de R se clasifican dentro del sistema usual en:

- *funciones gráficas de alto nivel* que permiten crear los gráficos básicos (plot, hist, boxplot, pairs,...) y
- *funciones gráficas de bajo nivel* que permiten modificar los gráficos creados (points, lines, text, axis, abline...).

Adicionalmente, disponemos de paquetes específicos, por ejemplo *lattice* o *ggplot2*, cuyas herramientas gráficas permiten describir situaciones complejas, a menudo multivariantes, con un sólo gráfico organizado en paneles (Figura 3). Podemos crear grandes cantidades de gráficos mediante scripts para el tratamiento de datos masivos (Big Data).

Ejemplo 1.10 *Ejecutar el siguiente código:*

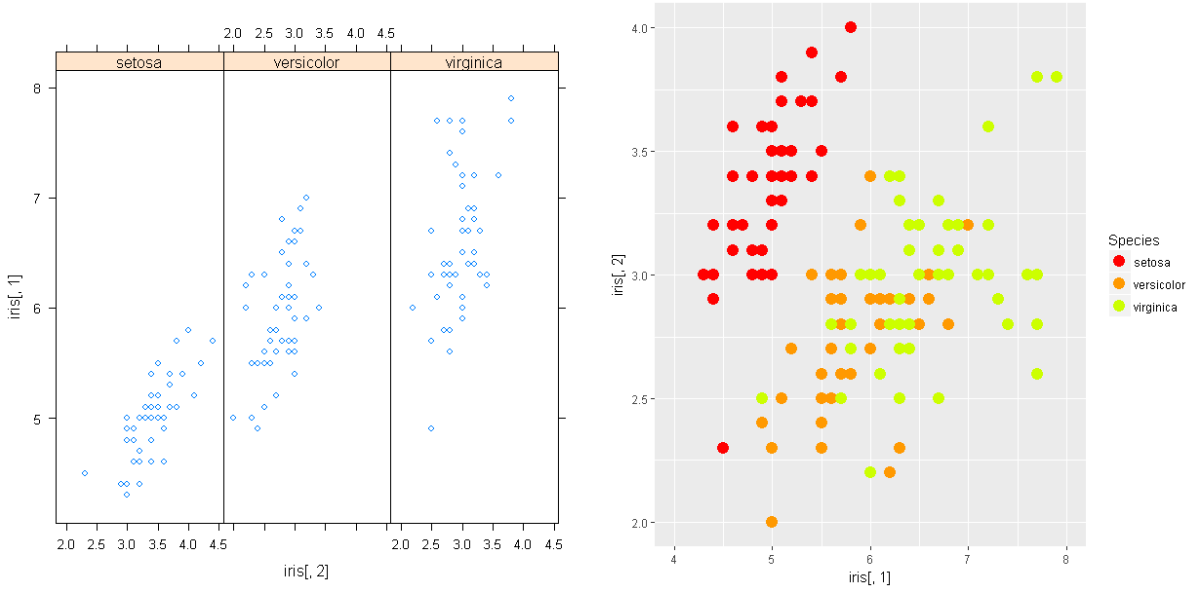
```
> pairs(iris[1:4], main = "Anderson's Iris Data - 3 species",
  + pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])
```

Ejemplo 1.11 *Ejecutar el siguiente código:*

```
> x <- rnorm(100)
> y <- rnorm(100)
> plot(x, y, xlab="Eje X", ylab="Eje Y", xlim=c(-2,2), ylim=c(-2,2), pch=22,
  + col="blue", bg="green", bty="7", main="Diagrama de Dispersión", las=2, cex=1.5)
```

Ejemplo 1.12 *Ejecutar el siguiente código:*

```
> o <- par(mfrow=c(2,2))
> plot(x=rnorm(200))
> hist(x=rnorm(200))
> boxplot(x=rnorm(200))
```



(a) Gráfico con *lattice*.

(b) Gráfico con *ggplot2*.

Figura 3: Gráficos con paquetes específicos

```
> qqnorm(rnorm(200))
> par(o)
```

Ejemplo 1.13 *Ejecutar las siguientes líneas de comandos:*

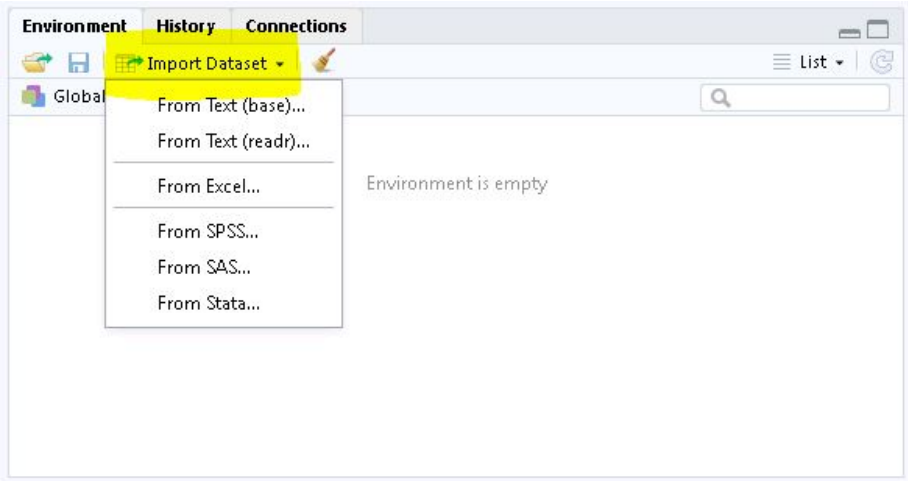
```
> x <- seq(-10,10,length=50)
> y=x
> f=function(x,y) x^ 2-y^ 2
> z=outer(x,y,f)
> o <- par(mfrow=c(1,2))
> persp(x,y,z)
> persp(x,y,z,theta=30,phi=30)
> par(o)
```

Ejemplo 1.14 *Ejecutar las siguientes líneas de comandos:*

```
> x=seq(-10,10)
> y=x^ 2
> plot(x,y)
> abline(h=40,col=3)
> abline(v=0,col=4)
> text(-5,60,"ExpErimEnto")
> axis(4)
```

Lectura e introducción de datos en R

RStudio dispone de una vía sencilla de lectura de datos desplegando Import Dataset desde la pestaña *Environment*, para ficheros de datos en formato texto (*From CSV...*), para hojas de cálculo Excel (*From Excel...*), para ficheros de datos de los paquetes estadístico como SPSS (*From SPSS...*), SAS (*From SAS...*) y Stata (*From Stata...*). La ventana emergente es válida para ficheros almacenados en disco, y también en servidores accesibles desde URL.



Para todas las opciones, la ventana emergente permite visualizar la base de datos a la vez que se indica el separador de los datos, de las cifras decimales, etc. Las líneas de comandos específicas ejecutadas se muestran en la *R-consola*, lo que conduce a una alternativa para importar directamente datos en los formatos anteriormente mencionados mediante tales sentencias. A continuación se ilustran en detalle algunos ejemplos:

```
#Importar fichero de datos ASCII de datos.
read.table (''ruta del archivo/datos1.txt'', header=TRUE, sep=, na.strings=''NA'', dec='.'')
```

El comando nos permitirá leer el fichero "datos1.txt" que se encuentra en tal ruta (separada por /), que dichos datos tienen cabecera, es decir que existe una primera la con los nombres de las variables (en caso contrario, los datos cabecera el argumento será header=FALSE). Con na.strings="NA" indicamos que los valores faltantes debe tomarlos como NA (nulos). El separador decimal de los datos que tenemos es el punto en vez de la coma. Así mismo, se puede añadir a la función otros argumentos tales como indicar el tipo de valores de los datos (lógicos, enteros, etc.), el número de columnas, etc.

```
#Importar fichero de texto.
 elemapi <- read.csv(''C:/Master/bioinf/elemapi.csv'', header=TRUE, sep='';'',
+ na.strings=''NA'', dec='','', strip.white=TRUE)
```

La sintaxis de este comando sobre el conjunto de datos "elemapi" indica que tiene que leer el fichero teniendo en cuenta la siguiente ruta: `''C:/Master/bioinf/elemapi.csv''`, teniendo en cuenta, que

- el fichero contiene los nombres de las variables en la cabecera (`header=TRUE`),
- el separador de los datos es el punto y coma (`sep='';''`),
- los valores perdidos se han codificado como NA (`na.strings=''NA''`),
- el separador de cifras decimales es la coma (`dec='',''`) y

- que en caso de leer variables de tipo carácter se eliminen los espacios anteriores y posteriores al valor registrado en dichas variables (`strip.white=TRUE`).

Unicamente se ha procedido a la lectura de los datos y aunque no se visualizan los datos, Workspace indica que el conjunto de datos activo es `elemapi` con 400 observaciones. de 23 variables.

```
#Visualización de los datos importados.
 View(elemapi)
#Otro ejemplo
 ElPulso <- read.table("C:/Master/bioinf/ElPulso.txt", header=T, quote=" \ ")
 View(ElPulso)
```

En el caso de bases de datos de la web, el cuadro de diálogo que aparece solicita la introducción la URL desde la que queremos importar los datos: `http://stat.ethz.ch/Teaching/Datasets/NDK/streamCV.dat`, por ejemplo. Este fichero consta de 4 variables STREAM (diferentes ríos); ZINC (concentración de zinc escalada en cuatro niveles); DIVERSITY (diversidad de las especies del río) y ZNGROUP (codifica los niveles de zinc en forma numérica).

```
#Importar ficheros desde Internet.
 stream <- read.table ("http://stat.ethz.ch/Teaching/Datasets/NDK/stream.dat",header=T)
#Estructura y tipo de un objeto ('data.frame':    34 obs. of 4 variables)
 str(stream)
#Visualización de variables individuales.
 stream[,"ZINC"]
#Medidas descriptivas de las columnas del fichero.
 summary(stream)
#histograma de la variable DIVERSITY
par(mfrow = c(1,2))    # Número de imágenes una debajo de la otra [1] o al lado [2]
hist(stream[,"DIVERSITY"]
#scatter-plot de DIVERSITY frente ZNGROUP
plot(stream[,"ZNGROUP"],stream[,"DIVERSITY"])
```

Importar ficheros en formato xls:

```
 install.packages("XLConnect",dependencies=T)
 require(XLConnect)
 wb<-loadworbook("ruta/archivo.xls", create=FALSE)
 datos<-readworbook(wb,sheet="hoja1")
```

Asimismo, la posibilidad de leer datos en otros formatos (Minitab, S, SAS, SPSS, Stata, Systat, dBase, etc) es perfectamente factible instalando el package "foreign".

```
#Instalando "foreign"
 install.packages("foreign")
#Cargando "foreign".
```

```
 library(''foreign'')
 ambiente<-read.spss(file=''ambiente.sav'',to.data.frame=TRUE)
#Cargando en memoria las variables del data frame.
 attach(ambiente)
#Análisis descriptivo numérico
 summary(ambiente)
 by(OZONO,OZONO,length)#Nº de lugares clasf.por ozono.
 by(SULFATO,OZONO,mean)#Media de sulfato por grupo de ozono.
 by(PH,PROVIN,summary)#Est.resumen de PH por provincia.
#Diagrama de cajas por factores
 boxplot(SULFATO~PROVIN)
 boxplot(PH~OZONO)
#Gráficos
 hist(SULFATO,main=''SULFATO'')
 boxplot(PH,main=''Diagrama de cajas del PH'')#Gráficos por grupos
 par(mfrow=c(2,2))
 hist(PH,main=''Histograma del PH'')
 by(PH,PROVIN,function(X,xlim){hist(X,xlim=xlim)},xlim=range(PH))
# mean,median,25th and 75th quartiles,min,max
 summary(ambiente)
# Tukey min,lower-hinge, median,upper-hinge,max
 fivenum(ambiente)
```

Hasta ahora, las funciones que hemos visto están localizadas en el paquete base. En la siguiente tabla se facilita un listado de las medidas descriptivas básicas junto con sus funciones:

Medidas Descriptivas	Función
Suma	**sum**(..., na.rm=FALSE)
Máximo	**max**(..., na.rm=FALSE)
Mínimo	**min**(..., na.rm=FALSE)
Posición del máximo	**which.min**(x)
Posición del mínimo	**which.max**(x)
Máximo en paralelo	**pmax**(...,na.rm=FALSE)
Mínimo en paralelo	**pmin**(...,na.rm=FALSE)
Sumas y productos acumulados	**cumsum**(x), **cumprod**(x)
max's y min's acumulados	**cummax**(x), **cummin**(x)
Media	**mean**(x, trim=0, na.rm=FALSE)
Media ponderada	**weighted.mean**(x,w,na.rm=FALSE)
Mediana	**median**(x,na.rm=FALSE)
Cuantiles	**quantile**(x,prob=(0,0.25,0.5,0.75,1),na.rm=F)
5-Tukey: min, lower-hinge, mediana, upper-hinge, máximo	**fivenum**(x, na.rm=FALSE)
min,1c,mediana,media,3c,max	**summary**(x, na.rm=FALSE)
Rango inter-cuartílico	**IQR**(x, na.rm=FALSE)
Rango	**range**(...,na.rm=FALSE, finite=FALSE)
Varianza	**var**(x, y=x, na.rm=FALSE, use)
Desviación Típica	**sd**(x, na.rm=FALSE)
Desviación mediana absoluta	**mad**(x,center,constant=1.4426, na.rm=FALSE)

```
x<-rgamma(50,1,3);summary(x);fivenum(x);mean(x);median(x);quantile(x);
+ quantile(x,c(0.35,0.9)); sd(x);var(x);range(x);IQR(x);min(x);which.min(x);x[which.min(x)];
+ pmin(x[1:5],x[6:10]);max(x); which.max(x);x[which.max(x)];pmax(x[4:8],x[2:6])
```

Como se comentó anteriormente, aunque el paquete *base* dispone de bastantes funciones para el análisis de datos, la gran mayoría de funciones estadísticas en R se encuentran en diversas librerías. Para encontrar el paquete que disponga de la función requerida se debe proceder del siguiente modo:

```
help.search("skewness")
#skewness está disponible en varios paquetes: agricolae, e1071...)
#Para instalar un paquete, por ejemplo e1071
install.packages("e1071")
#Para cargar una librería instalada, por ejemplo e1071
library("e1071")
Loading required package: class
[1] "e1071" "class" "relimp" "methods" "stats" "graphics"
[7] "utils" "datasets" "Rcmdr" "car" "tcltk" "grDevices"
[13] "base"
```

Indica que el paquete *e1071* ha sido leído correctamente

```
#Cálculo de asimetría y curtosis de PH.
 skewness(PH)
 kurtosis(PH)
```

Otros paquetes nos permiten ampliar la variedad de funciones estadísticas disponibles en el paquete base, por ejemplo:

```
#ejemplo1: ''Hmisc''
 library(''Hmisc'')
 describe(ambiente)
# n, nmiss, unique, mean, 5,10,25,50,75,90,95th percentiles
# 5 lowest and 5 highest scores
#ejemplo2: ''pastecs''
 library(''pastecs'')
stat.desc(ambiente)
# nbr.val, nbr.null, nbr.na, min max, range, sum,
# median, mean, SE.mean, CI.mean, var, std.dev, coef.var
#ejemplo3: ''psych''
 library(''psych'')
describe(ambiente)
# item name ,item number, nvalid, mean, sd,
# median, mad, min, max, skew, kurtosis, se
```

Observación 1.2 (Otras fuentes de ayuda en R) *Podemos acceder a la ayuda de todos los paquetes de R que tengamos disponibles localmente a través de cualquier explorador (Mozilla, Netscape, Internet Explorer, etc.). Para ello, a través de los comandos* help.start(), help(), apropos(), help.search(), help(package=nombre paquete), demo(nombre paquete) *entre otros, podemos acceder a la ayuda sobre las funciones en los paquetes instalados, así como a los manuales básicos de R y a otro material mediante el comando* vignette(tópico, nombre paquete,...) *para algunos paquetes, listado disponible tecleando* browseVignettes(). *Además, RStudio suministra ayuda para completar automáticamente códigos usando la tecla Tab. Por otro lado, si disponemos de internet, además de las URLs de R y RStudio, podemos escribir en la R-consola* RSiteSearch("foo"), *donde foo es la palabra sobre la que buscamos ayuda.*

Observación 1.3 *Para comprobar cómo se debe citar R en una publicación teclear:* citation() *y una para citar una librería concreta:*

> *citation(''nombrepaquete'').*

Ejercicio 1.1 *Ejecutar el comando:* > *citation(''nombrepaquete'')*

Observación 1.4 *RStudio tiene capacidad de recordar comandos anteriores con las teclas de flecha. Además, si desea consultar una lista de los comandos recientes y seleccione un comando en esta lista, puede utilizar las teclas Ctrl + Arriba para revisar la lista.*

Observación 1.5 *Los shortcuts más útiles en http://www.rstudio.com/ide/docs/using/keyboard_shortcuts:* Ctrl + L *(limpia la consola),* Esc *(Interrupción R), entre otros.*

Observación 1.6 *La notación de funciones y packages debe tenerse en cuenta de la manera que se indique, puesto que R distingue entre mayúsculas y minúsculas.*

Observación 1.7 *Para terminar la sesión con R teclear la función q(), o lo que es lo mismo desde el menú principal File<- Quit R*

Observación 1.8 *Para guardar tu sesión de trabajo. :Save workspace image?.*

Programando en R...

En este apartado, algunas estructuras básicas de programación en R, tales como funciones, expresiones condicionales y bucles, son revisadas e ilustradas con ejemplos sencillos.

Funciones La estructura básica de una función en R viene dada por:

```
nombrefuncion=function(args) expr; return(output)
```

Para empezar con un ejemplo sencillo, vamos a construir una función que calcule la raíz cuadrada de la cuasivarianza de un vector x. Un código de comando podría ser el que sigue:

```
desvt<-function(x) {  n=length(x)-1
 sqrt(sum((x-mean(x))^2)/n)
 }
x<-1:10
desvt(x)
```

Por último, comprobaremos que el resultado obtenido con la función desvt coincide con el que se obtendría con la correspondiente función básica, ejecutando la siguiente línea de comando:

```
sd(x)
```

Expresiones condicionales Las expresiones condicionales permiten evaluar una condición para cada uno de los valores de un vector, siendo el output un vector de la misma longitud obtenido aplicando expr (exprT) en las componentes del vector que satisfacen la condición *cond*, y expralternativa (exprF) en las que no se cumple.

```
if(cond) expr else expralternativa
ifelse(cond,exprT,exprF)
```

Bucles Los bucles son estructuras repetitivas (loops) que permiten la ejecución de comandos agrupados entre llaves.

```
for(var in seq) expr
while(cond) expr
repeat expr
```

`break` Fuerza la salida de un bucle.

`next` En un bucle (`next, while o repeat`) fuerza a la iteración siguiente .

Bioconductor

Como extensión de R, existen paquetes específicos incluidos en un proyecto de software libre (de código y desarrollo abierto) que utiliza el lenguaje de programación estadístico R, llamado Bioconductor (www.bioconductor.org/), que proporciona herramientas de análisis y comprensión de datos genómicos de alto rendimiento. Originalmente, Bioconductor se desarrolló para el análisis de datos de microarray pero en la actualidad cuenta con paquetes para el estudio de datos de secuenciación, ensayos celulares, ensayos de altas prestaciones, etc. y acceso a multitud de bases de datos con información genética.

Para instalar la versión actual de Bioconductor, los usuarios de R deben actualizar su instalación y a continuación obtener la última versión de Bioconductor directamente ejecutando desde la consola de R las siguientes sentencias de comandos:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
 install.packages("BiocManager")
BiocManager::install()
```

El comando BiocManager::install() instala o actualiza Bioconductor y paquetes del CRAN en una versión de Bioconductor. Es el método recomendado para instalar los paquetes en Bioconductor en vez de la sentencia install.packages de R. Para obtener más información sobre este comando en la línea de comandos ejecutar:

```
?install
```

La instalación de paquetes específicos se lleva a cabo ejecutando el comando BiocManager::install(). Por ejemplo, para instalar los paquetes "GenomicFeatures" y "AnnotationDbi" se realizaría a través de las siguiente secuencia de comandos:

```
BiocManager::install(c("GenomicFeatures", "AnnotationDbi", "ROCpAI"))
```

Para cargar Bioconductor es preciso ejecutar:

```
library(BiocManager)
```

Ejercicios Propuestos

Ejercicio 1.2 *Utiliza las funciones rep() y seq() para crear un vector que contenga:*

a) los valores: 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4

b) los valores: 4, 4, 4, 4, 3, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1, 1

c) los valores: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5

Ayuda: rep(1:4,4) o rep(1:4,len=16) o sequence(rep(4,4))

rep(4:1,rep(4,4)) ; rep(1:5,1:5)

Ejercicio 1.3 *Construir un factor de 30 elementos con tres categorías (1, 2 y 3). Etiquetar las categorías como A, B y C. Utilizar la función table() para comprobar que realmente hay 10 de cada categoría.*

Ayuda: f<-as.factor(sample(rep(1:3, times=10))); levels(f)<-c("A", "B", "C"); table(f)

Ejercicio 1.4 *En algunas ocasiones queremos discretizar una variable continua en categorías, para ello utilizaremos la función cut(). Simular 100 valores de una $\mathcal{N}(0,1)$ y dividir los valores en 5 categorías.*

Ayuda: y<-cut(rnorm(100),breaks=5); table(y).

Ejercicio 1.5 *Con los siguientes números: 7.3 6.8 0.005 9 12 2.4 18.9 .9*

a) Calcula la media.

b) Calcula la raíz cuadrada de los números.

c) Obtén los números que son mayores que su raíz cuadrada.

d) ¿Cuántos valores son mayores que 1?

e) Obtén la raíz cuadrada de los números redondeados con dos cifras decimales.

f) ¿Cuánto difieren los números redondeados de los originales?

Ayuda: dat<-c(7.3, 6.8, 0.005, 9, 12, 2.4, 18.9, .9)
mean(dat); sqrt(dat); sum(dat>1); dat[dat>sqrt(dat)];datr<-round(sqrt(dat),2); datr; sqrt(dat)-datr

Ejercicio 1.6 *Considerar las estaturas siguientes de los individuos en tres ciudades españolas:*

Madrid	*1.55*	*1.79*	*1.64*	
Barcelona	*1.81*	*1.90*	*1.50*	*1.52*
Murcia	*1.95*	*1.60*	*1.72*	*1.80*

Definir y construir dos variables, estatura y ciudad, de forma que ciudad sea un factor de clasificación. Obtener la estatura media global y por ciudades.

Solución:

```
estatura<- c(1.55,1.79,1.64,1.81,1.90,1.5,1.52,1.95,1.6,1.72,1.8)
ciudad<-as.factor( rep(c(1,2,3), c(3,4,4)) )
levels(ciudad)<-c("Madrid","Barcelona","Murcia")
mean(estatura); by(estatura, ciudad, mean)
mean(estatura[ciudad=="Murcia"]), etc.
```

Ejercicio 1.7 *Los datos 22,22,23,24,26,27,28,29,29,29,31,33,34,35,35,35,36,38,39, son las edades de los pacientes atendidos en una consulta médica esta mañana. Calcula su coeficiente de variación e interpreta el resultado obtenido.*

Ayuda

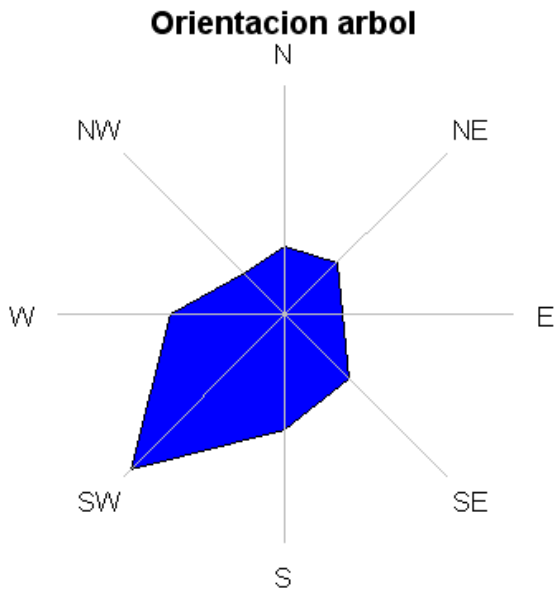
```
edad<- c(22,22,23,24,26,27,28,29,29,29,31,33,34,35,35,35,36,38,39)
sqrt(var(edad))
 mean(edad)
abs(mean(edad))
CV<-sqrt(var(edad))/abs(mean(edad))
CV
```

Como es un poco engorroso hacer esto cada vez que queremos calcular el coeficiente de variación vamos a crear una función para calcularlo.

A esta función que vamos a generar la llamaremos CVar.

```
CVar<-function(x)
{
resultado<-sqrt(var(x))/abs(mean(x))
return(resultado)
}
# Comprobamos el resultado
CVar(edad)
```

Ejercicio 1.8 *En el fichero de datos "orientaciones.dat" tenemos los datos codificados de la orientación de 1047 árboles en los que se analizó la presencia de procesionaria. Leer los datos en un vector y averiguar cuantos hay de cada tipo. Representar dichos datos, teniendo en cuenta que en realidad los valores del 1 al 8 representan los 8 puntos cardinales (E, SE, S, SW, W, NW, N, NE) tratar de buscar una mejor representación de ellos. Utilizar para ello la función radial.plot() de la librería plotrix. Guardar el gráfico resultante en un pdf.*



Ayuda: orient<-scan("orientaciones.dat"); table(orient)

```
freqs<-c(75, 119, 151, 289, 152, 76, 89, 96)
nom<-c("E", "SE", "S", "SW", "W", "NW", "N", "NE")
pdf("desc-orientaciones.pdf", paper="special", width=13, height=7)
radial.plot(freqs,labels=nom,rp.type="p",radial.lim=c(0,300),
poly.col="blue", show.grid=FALSE,
clockwise=TRUE, main="Orientacion arbol")
dev.off()
```

2. Revisión de modelos de probabilidad usuales

R dispone de las distribuciones de probabilidad más comunes implementadas en la librería base, que obviamente, como ya hemos comentado, puede ampliarse con otras librerías. Algunos de estos modelos de distribución junto con sus funciones se muestran a continuación, siendo los argumentos de las respectivas funciones sus parámetros correspondientes:

Distribución	**Función**
Binomial	**binom***(n,size,prob)*
Hypergeométrica	**hyper***(nn,m,n,k)*
Geométrica	**geom***(n,prob)*
Binomial Negativa	**nbinom***(n,size,prob)*
Poisson	**pois***(n,lambda)*
Uniforme	**unif***(n,min=0,max=1)*
Normal	**norm***(n,mean=0,sd=1)*
Gamma	**gamma***(n,shape,scale=1)*
Exponencial	**exp***(n,rate=1)*
Chi-Cuadrado χ^2	**chisq***(n,df)*
Beta	**beta***(n,shape1,shape2)*
t-Student	**t***(n,df)*
F-Snedecor	**f***(n,df1,df2)*
Weibull	**weibull***(n,shape,scale=1)*
Lognormal	**lnorm***(n,meanlog=0,sdlog=1)*

Además, si a cualquiera de estas funciones de R (**distrib**) se le agrega un prefijo '*d*' se obtiene la función de densidad o de probabilidad puntual, '*p*' para la función de distribución acumulada FDA, '*q*' para la función cuantil o percentil y '*r*' para generar (simular) variables pseudo-aleatorias (random):

generador de numeros aleatorios	**rdistrib***(n,par)*
función densidad/probabilidad	**ddistrib***(x,par)*
función distribución	**pdistrib***(x,par)*
función inversa distribución (cuantiles)	**qdistrib***(x,par)*

A continuación revisaremos modelos de distribución que se presentan en las técnicas estadísticas que estudiaremos más adelante así como en fundamentos teóricos de probabilidad, y que nos permitan abordar situaciones habituales tales como las siguientes:

(1) En un problema tengo 100 microARN, cada uno de tamaño 20 con una probabilidad de 0.7 de conseguir una purina, ¿cuál es la probabilidad de que el número promedio de purinas sea mayor de 15?

(2) ¿Cúal es la probabilidad de encontrar catorce pirimidinas en una secuencia de ADN de longitud 24?

(3) Si la expresión del gen CCND3 Cyclin D3 en pacientes con leucemia linfoblástica aguda se distribuye normalmente con media 1.9 y desviación típica 0.5, ¿cuál es la probabilidad de observar valores de expresión por encima de 2.4?

(4) Modelos de sustitución utilizados para secuencias de nucleótidos (Figura 4).

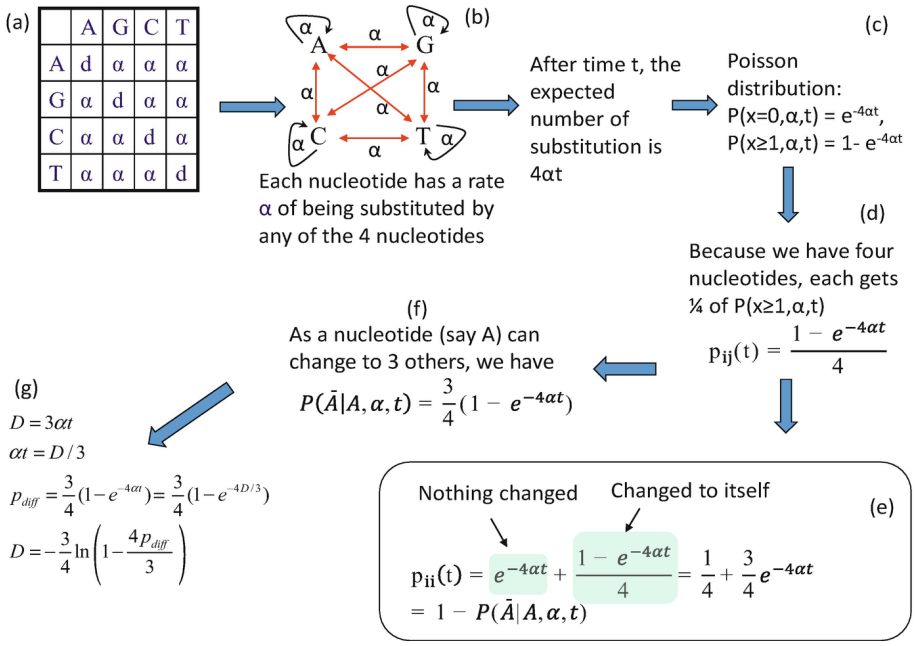


Figura 4: Modelo de sustitución nucleotídica (Xia, 2018)

Situaciones anteriores implican conteos de resultados que son identificados con variables discretas tales con las distribuciones binomial, multinomial y de poisson.

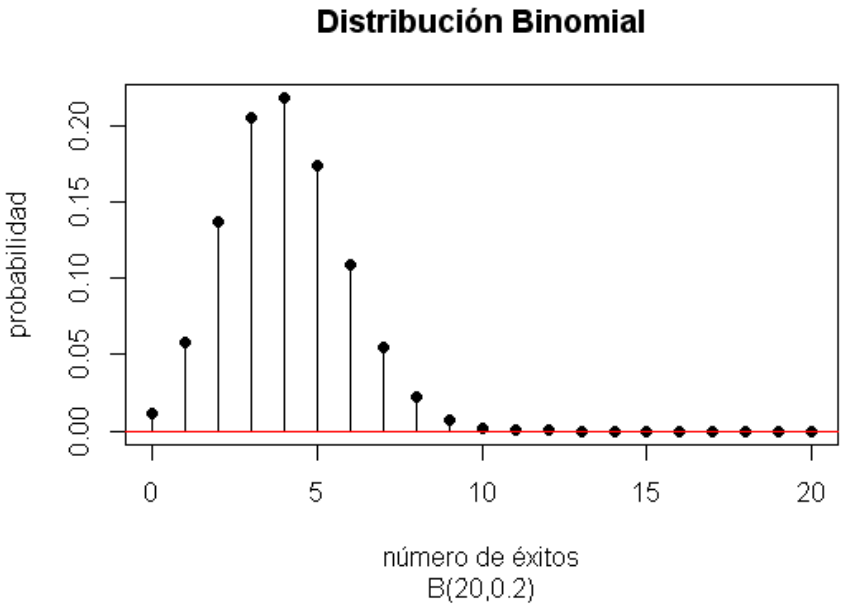
Distribución Binomial

Una v.a. X sigue una distribución binomial de parámetros $n \in \mathbb{N}$ y $p \in (0,1)$ y se denota $X \equiv \mathcal{B}(n,p)$ si describe el número de éxitos en n realizaciones independientes de un experimento que tiene probabilidad de éxito p. La variable aleatoria puede tomar los valores $\{0,1,2,\ldots,n\}$ y su función de probabilidad viene dada por:

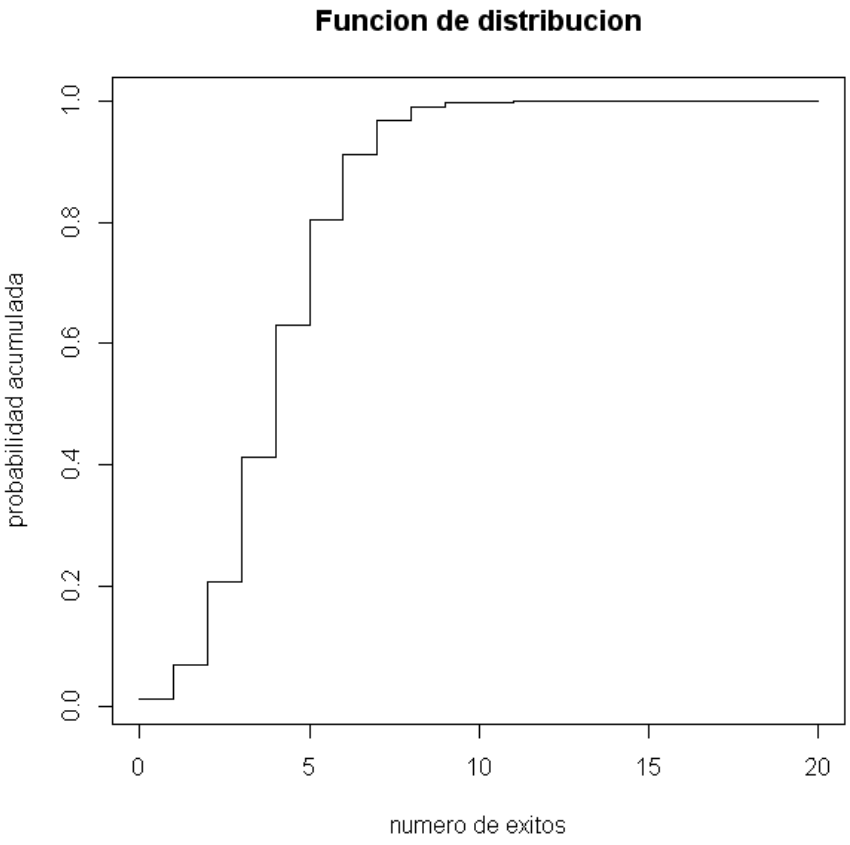
$$f(x;n,p) = \binom{n}{x} p^x (1-p)^{n-x}, \text{ para } x = 0,1,2,...,n.$$

A continuación mostramos las instrucciones en R para generar la función de probabilidad y la función de distribución de una variable aleatoria Binomial de parámetros $n = 20$ y $p = 0{,}2$.

```
#Función puntual de probabilidad de B(n=20,p=0.2)
x<-0:20
plot(x,dbinom(x,size=20,prob=0.2),xlab="numero de exitos",
+ ylab="probabilidad",
+ main="Distribución Binomial",sub="B(20,0.2)",type="h")
points(x,dbinom(x,size=20,prob=0.2),pch=16)
abline(h=0,col="red")
```




```
#Función distribución de B(n=20,p=0.2)
x<-0:20
plot(x,pbinom(x,20,0.2),type="s",xlab="numero de exitos",
+ ylab="probabilidad acumulada", main="Funcion de distribucion")
```



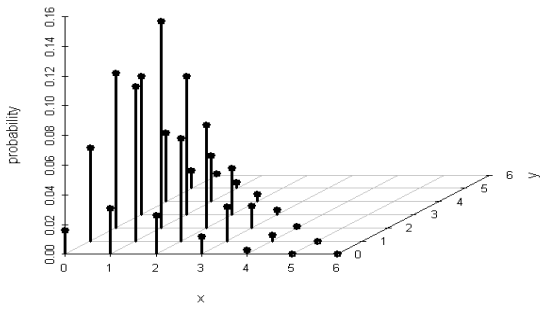
Ejemplo 2.1 *En secuencias cortas de ARN (microARN) de tamaño 20, la probabilidad de purina se distribuye según una binomial con probabilidad 0.7.*

1. *Representa la función de probabilidad y la función de distribución.*
2. *¿Cuál es la probabilidad de que haya 10 purinas?*
3. *¿Cuál es la probabilidad de que haya al menos 10 purinas?*
4. *¿Cuál es la probabilidad de que haya más de 10 purinas?*

Distribución Multinomial

La distribución multinomial es una extensión de la binomial que modeliza el conteo de n realizaciones independientes de un experimento cada uno de ellos con k resultados posibles con probabilidad $p_i (i = 1, 2, ..., k)$, $\sum_{i=1}^{k} p_i = 1$, y n_i es el número de veces que ocurre el resultado i, $i = 1, .., k$, $\sum_{i=1}^{k} n_i = n$. Se dice entonces que la v.a. k-dimensional $\mathbf{X} = (X_1, ..., X_k)$ sigue un modelo de distribución multinomial con parámetros n y $\mathbf{p} = (p_1, ...p_k)$, $\mathbf{X}$ $M(n, \mathbf{p})$ y función de probabilidad $f(\mathbf{x}; n, \mathbf{p}) = \binom{n}{n_1 \dots n_k} p_1^{n_1} \dots p_r^{n_k}$.

```
#Función distribución de M(n=20,p=(0.25,0.15,0.6)
library("scatterplot3d")
X <-t(as.matrix(expand.grid(0:6,0:6)))
X <- X[,colSums(X)<=6];X <- rbind(X,6-colSums(X))
Z <- round(apply(X,2,function(x) dmultinom(x,prob=1:3)),3)
A <- data.frame(x=X[1,],y=X[2,],probability=Z)
scatterplot3d(A,type="h",lwd=3,box=F)
```



Ejemplo 2.2 *En bioinformática, una secuencia s es una sucesión finita de caracteres generada a partir de un alfabeto definido. Representada por $s = s_1 s_2 ... s_n$, cada s_i indica la posición que ocupa ese carácter en la secuencia que si es de ADN el alfabeto es N={A,C,G,T}, si es de ARN es N_{ARN}={A,C,G,U}, y si es de una proteína, el alfabeto es A={A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,T,Y}. Esta simplificación unidimensional permite la modelización y el desarrollo de algoritmos para encontrar soluciones, aunque ignora información.*

En este marco, si se asume que los nucleótidos en una secuencia de ADN son independientes e idénticamente distribuidos, el modelo probabilístico más simple para ajustar la distribución de caracteres en la secuencia es la distribución de probabilidad multinomial, sobre el alfabeto N, de parámetros n y $\boldsymbol{p} = (p_A, p_C, p_G, p_T)$*, donde:* $P(s_i = A) = p_A, P(s_i = C) = p_C, P(s_i = G) = p_G, P(s_i = T) = p_T$*, con* $p_A + p_C + p_G + p_T = 1$.

Distribución de Poisson

Una v. a. discreta X se dice que sigue una distribución de Poisson de parámetro λ ($\lambda > 0$) y se denota $X \equiv \mathcal{P}(\lambda)$ si su función de probabilidad es de la forma

$$f(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \text{ para } x = 0, 1, 2, ...,$$

Esta distribución se utiliza para modelizar el número de sucesos independientes que se producen en una unidad de tiempo o de espacio cuando la frecuencia de esos sucesos es constante. En bioinformática, número de mutaciones o recombinaciones de una secuencia genética, distribución de errores producidos en un proceso de secuenciación, probabilidad de coincidencias de secuencias aleatorias o número de patrones de ADN diferentes.

Ejemplo 2.3 *Si el número medio de células en un cultivo de 20* μm^2 *es 5, y se distribuyen de forma estable, ¿cuántas células podríamos esperar en 16* μm^2*? Calcular la probabilidad de que no haya ninguna célula en un cultivo de 16* μm^2.

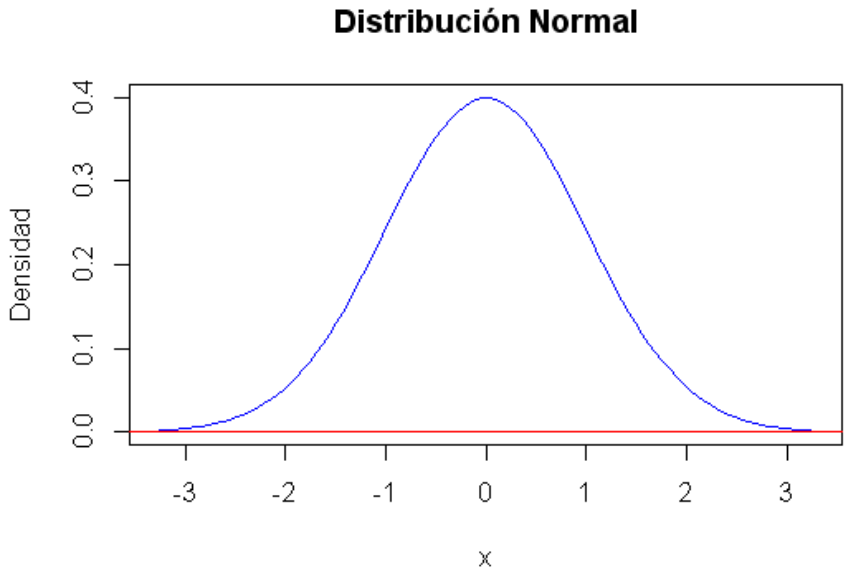
Ejemplo 2.4 *Las mutaciones del genoma del VIH ocurren al azar con una tasa* $5x10^{-4}$ *por nucleótido por ciclo de replicación. i.e., el número de mutaciones en un genoma de 10000 nucleótidos seguirá una distribución con tasa 5 después de un ciclo. Calcular la probalidad de que ocurran 3 mutaciones.*

Distribución Normal

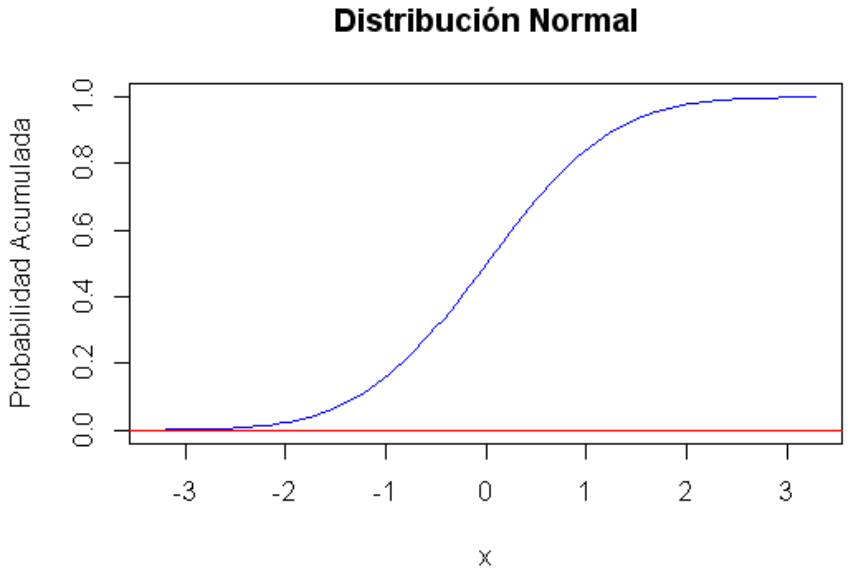
Una v. a. X, se dice que sigue una distribución normal de parámetros media μ y desviación típica σ, y se denota $X \equiv \mathcal{N}(\mu, \sigma)$, si su función de densidad es de la forma

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \text{ con } -\infty < x < \infty$$

A continuación mostramos las instrucciones en R para generar la función de densidad y la función de distribución de una variable aleatoria Normal de parámetros $\mu = 0$ y $\sigma = 1$.




```
x<- seq(-3.291, 3.291, length=100)
plot(x, dnorm(x, mean=0, sd=1),col="blue",xlab="x", ylab="Densidad",
+ main="Distribución Normal",type="l")
abline(h=0, col="red")
```

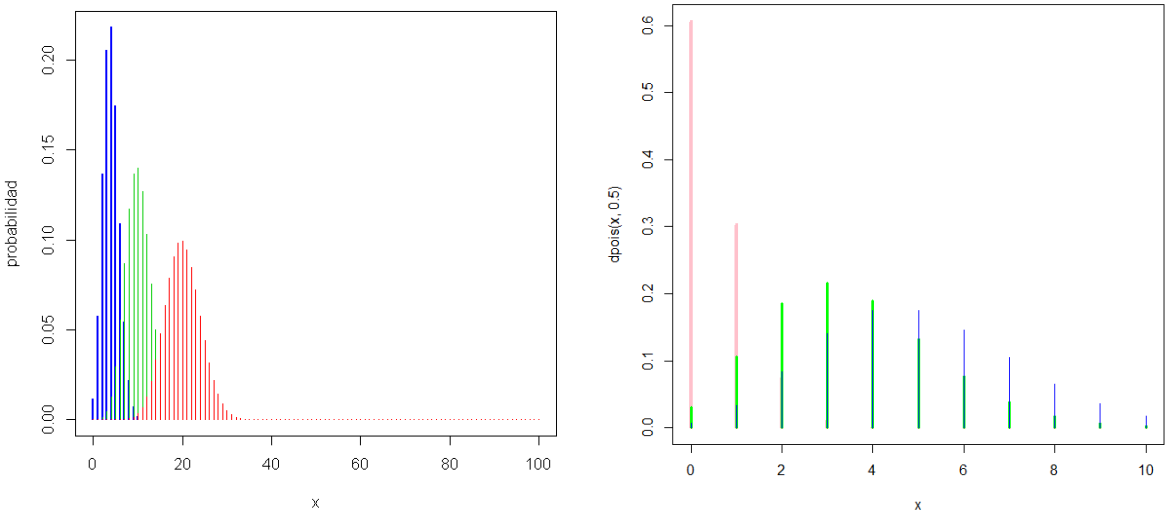



```
x<- seq(-3.291, 3.291, length=100)
plot(x, pnorm(x, mean=0, sd=1),col="blue",xlab="x", ylab="Probabilidad Acumulada",
+ main="Distribución Normal"),
+ type="l")
abline(h=0, col="red")
```

Ejemplo 2.5 *La distribución de los valores de expresión del gen Zyxin en paciente ALL se distribuye según una* $N(\mu = 1{,}5;\ \sigma = 0{,}5)$

1. *Representa la función de densidad y la función de distribución.*
2. *¿Cuál es la probabilidad de que los valores de expresión sean menores que 1.2?*
3. *¿Cuál es la probabilidad de que la media de los valores de expresión sea menor que 1.2?*
4. *¿Cuál es la probabilidad de que los valores de expresión estén entre 0.8 y 2.4?*
5. *Usando rnorm genera una muestra de tamaño 1000 de la población dada, esto es, con* $\mu = 1{,}6$ *y* $\sigma = 0{,}4$.

Aproximaciones por la Normal



(a) Modelos binomiales.

(b) Modelos de Poisson.

Figura 5: Modelos probabilísticos aproximándose a una distribución normal.

- Si X es una variable Binomial, de parámetros n y p, entonces si n es grande y ni p ni $1-p$ son próximos a cero, se puede considerar que X sigue una distribución $\mathcal{N}(\mu = np, \sigma^2 = np(1-p))$. Véase Figura 5.

$$Z = \frac{X - np}{\sqrt{np(1-p}} \equiv \mathcal{N}(0,1)$$

```
curve(dbinom(x,20,0.2),0,20,21,type="h",lwd=2,col=4,xlim=c(0,100))
curve(dbinom(x,50,0.2),0,50,51,type="h",lwd=1,col=3,add=T)
curve(dbinom(x,100,0.2),0,100,101,type="h",lwd=1,col=2,add=T)
```

- Si X es una distribución de Poisson de parámetro λ grande, $\lambda > 25$, se puede considerar que X sigue una distribución $N(\mu = \lambda, \sigma = \sqrt{\lambda})$. Véase Figura 5.

$$Z = \frac{X - \lambda}{\sqrt{\lambda}} \equiv \mathcal{N}(0,1)$$

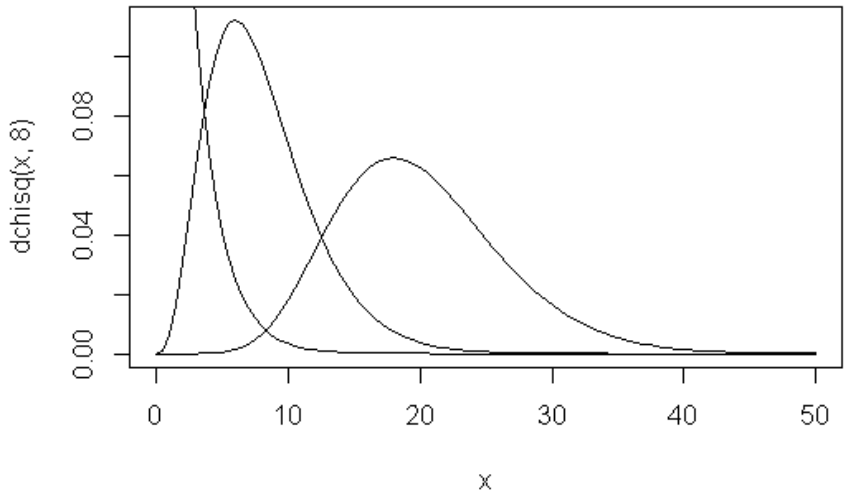
```
curve(dpois(x,0.5),0,10,11,type="h",lwd=4,col="pink",xlim=c(0,10))
curve(dpois(x,3.5),0,10,11,type="h",lwd=3,col="green",add=T)
curve(dpois(x,5),0,10,11,type="h",lwd=1,col="blue",add=T)
```

Distribución χ^2_n de Pearson

Sean $X_1, X_2, ..., X_n$, variables aleatorias con distribución $N(0,1)$ e independientes. La variable $X = X_1^2 + X_2^2 + \cdots + X_n^2$, se dice que es una χ^2_n, ji-cuadrado de Pearson con n grados de libertad.

La función de densidad de una v.a. χ^2_n es $f(x;n) = \frac{x^{\frac{n}{2}-1}e^{-\frac{x}{2}}}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})}$ si $x > 0$. Siendo Γ la función gamma definida: $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx$ para $x > 0$.

Propiedad: La suma de $\chi^2_{n_1}$, $\chi^2_{n_2}$, ... independientes, es otra χ^2_n siendo $n = n_1 + n_2 + ...$



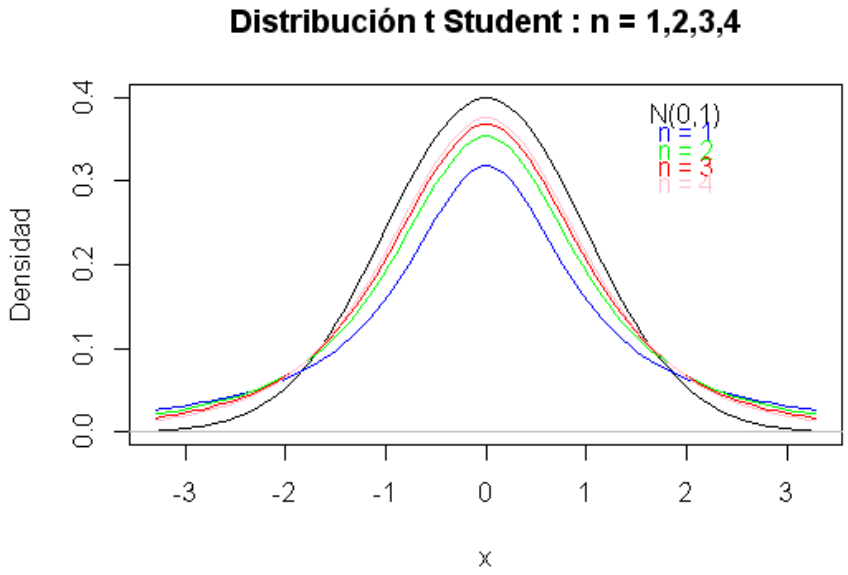
Distribución t_n de Student

Sea Z una variable aleatoria $N(0,1)$ e Y una v.a. χ^2_n ambas independientes, entonces la variable aleatoria,

$$X = \frac{Z}{\sqrt{Y/n}}$$

se denomina t_n de Student con n grados de libertad.

La función de densidad de una variable aleatoria t_n es: $f(x;n) = \frac{1}{\sqrt{\pi}\beta(\frac{1}{2},\frac{n}{2})}\left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}$ si $-\infty < x < \infty$ y $n > 0$.Siendo β la función beta definida: $\beta(\alpha_1, \alpha_2) = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)}{\Gamma(\alpha_1+\alpha_2)}$ para $\alpha_1, \alpha_2 > 0$.




```
x<- seq(-3.291, 3.291, length=100)
plot(x, dnorm(x, mean=0, sd=1), xlab="x", ylab="Densidad",
+ main="Distribución t Student : n = 1,2,3,4",, type="l")
abline(h=0, col="gray")
text(2,.38,"N(0,1)",col="black")
lines(x, dt(x, df=1),col="blue")
text(2,.36,"n = 1",col="blue")
lines(x, dt(x, df=2),col="green")
text(2,.34,"n = 2",col="green")
lines(x, dt(x, df=3),col="red")
text(2,.32,"n = 3",col="red")
lines(x, dt(x, df=4),col="pink")
text(2,.30,"n = 4",col="pink")
```

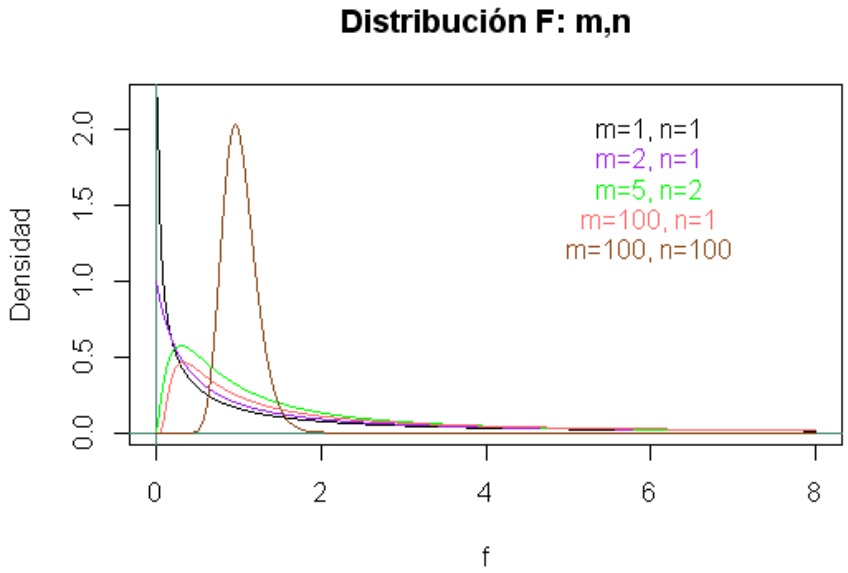
Distribución $F_{m,n}$ de Snedecor

Sean U y V dos variables independientes distribuidas según leyes ji-cuadrado de m y n grados de libertad respectivamente, la variable X,

$$X = \frac{U/m}{V/n}$$

se dice que es una variable $F_{m,n}$ de Snedecor con m y n grados de libertad, en el numerador y denominador respectivamente.

La función de densidad viene dada por la expresión, $f(x;n,m) = \frac{(m/n)^{m/2}}{\beta\left(\frac{m}{2},\frac{n}{2}\right)} \frac{x^{\frac{m}{2}}}{\left(1+\frac{m}{n}x\right)^{\frac{n+m}{2}}}$, si $x > 0$ y $m, n > 0$.




```
x<- seq(0, 8, length=400)
plot(x, df(x, df1=1, df2=1), xlab="f", ylab="Densidad",
+ main="Distribución F: m,n", type="l")
abline(h=0, col="aquamarine4")
abline(v=0, col="aquamarine4")
text(6,2,"m=1, n=1",col="black")
lines(x, df(x, df1=2, df2=1),col="purple")
text(6,1.8,"m=2, n=1",col="purple")
lines(x, df(x, df1=5, df2=2),col="green")
text(6,1.6,"m=5, n=2",col="green")
lines(x, df(x, df1=100, df2=1),col="indianred1")
text(6,1.4,"m=100, n=1",col="indianred1")
lines(x, df(x, df1=100, df2=100),col="chocolate4")
text(6,1.2,"m=100, n=100",col="chocolate4")
```

Propiedad: Si $X \equiv F_{m,n}$ entonces $\frac{1}{X} \equiv F_{n,m}$

Distribución de los estadísticos en el muestreo

Sean $X_1, X_2, ..., X_n$ una m.a.s. de tamaño n de una variable aleatoria poblacional $X \equiv N(\mu, \sigma)$

θ	*Estadístico*	*Propiedades*		*Distribución en el muestreo*
μ	$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$	$E(\overline{X}) = \mu$	$Var(\overline{X}) = \frac{\sigma^2}{n}$	$\overline{X} \equiv \mathcal{N}(\mu, \frac{\sigma}{\sqrt{n}})$
σ^2	$S^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$	$E(S^2) = \frac{n-1}{n}\sigma^2$	$Var(S^2) = \frac{2(n-1)}{n}\sigma^4$	$\frac{nS^2}{\sigma^2} \equiv \chi^2_{n-1}$
σ^2	$S_c^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$	$E(S_c^2) = \sigma^2$	$Var(S_c^2) = \frac{2}{n-1}\sigma^4$	$\frac{(n-1)S^2}{\sigma^2} \equiv \chi^2_{n-1}$
σ^2	$S_\mu^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \mu\right)^2$	$E(S_\mu^2) = \sigma^2$	$Var(S_\mu^2) = \frac{2}{n}\sigma^4$	$\frac{nS^2}{\sigma^2} \equiv \chi^2_n$

Ejercicios Propuestos

Ejercicio 2.1 *Sea* $X \equiv \mathcal{B}(p = 0,5, n = 100)$. *Hallar:*

a) $P(X \leq 45)$; $p(X \leq 52)$; $p(X < 60)$

b) Calcular las probabilidades anteriores con la aproximación binomial-normal.

Ejercicio 2.2 *Sea* $X \equiv N(\mu = 50, \sigma = 5)$. *Calcular:*

a) $P(X \leq 40)$; $P(X \leq 60)$; $P(X > 65)$

b) $P(X > 35)$; $P(40 < X < 60)$; $P(30 < X < 42)$

Ejercicio 2.3 *Genere la función de densidad y la función de distribución de una v.a. distribuida según una* χ^2_3 *de Pearson.*

Ejercicio 2.4 *Usar la función runif() (con set.seed(32078)) para generar 10 números pseudoaleatorios de*

- *una distribución uniforme (0,1)*
- *una distribución uniforme (3,7)*
- *una distribución uniforme (-2,2)*

set.seed(32078);runif(10)

set.seed(32078);runif(10, min=3, max=7)

set.seed(32078);runif(10, min=-2, max=2)

dbinom(x=4, size=6, prob=0.5)

pbinom(q=4, size=6, prob=0.5)

Ejercicio 2.5 *Calcular el valor* x *tal que* $P(X \leq x) = 0{,}89$.

qbinom(0.89,6,0.5)

Ejercicio 2.6 *Generar 10 valores pseudoaleatorios de una* $B(6, 0{,}5)$.

rbinom(10,6,0.5)

Ejercicio 2.7 *Generar 100 valores aleatorios de una distribución normal de media 3 y desviación típica 2 (utiliza la semilla 111).*

options(width=80); set.seed(111); datos<-rnorm(100,3,2)
hist(datos,freq=FALSE);curve(dnorm(x,3,2),add=TRUE)

3. Revisión de inferencia estadística básica paramétrica y no paramétrica

La Inferencia Estadística es el conjunto de métodos que permiten trasladar los resultados de una muestra a la población de manera fiable, midiendo la incertidumbre o acierto de los resultados, decisiones y sus conclusiones. Distinguimos entre inferencia estadística paramétrica, si es conocida la forma funcional del modelo de distribución que sigue la v.a. por lo que sólo tenemos que estudiar los parámetros que la determinan, y inferencia estadística no paramétrica, si el objetivo a estudiar es global, no se centra en sus parámetros.

La bondad de estas deducciones se mide en términos probabilísticos, es decir, toda inferencia se acompaña de su probabilidad de acierto.

Por ejemplo, podemos considerar que la población objeto de estudio sigue un modelo normal, de media desconocida o incluso de media y desviación típica ambas desconocidas. A partir de la muestra extraída de la población, no podemos esperar determinar exactamente los parámetros poblaciones desconocidos. Sin embargo, si dicha información es suficientemente representativa, podremos aproximarnos a través de los estadísticos muestrales convenientes a la media poblacional o/y desviación típica poblacional.

La suposición de que se satisface una determinada condición (hipótesis nula) a partir de la información recabada de la población, será el objetivo principal que deseamos comprobar, *contrastar*. En este contexto, la Estadística nos proporciona la técnica de test o contraste de hipótesis que nos permite decidir si dicha suposición inicial es significativamente errónea acotando la incertidumbre de dicha decisión, o adoptando, en caso contrario, la hipótesis nula establecida.

Habitualmente, los resultados de un contraste de hipótesis se presentan a través del $p-valor$ o nivel crítico, concluyendo a partir del mismo si la hipótesis nula es o no rechazada a un nivel de significación (α) prefijado. El $p-valor$ es el nivel de significación menor que llevaría al rechazo de la hipótesis nula H_0. Así, fijado el nivel de significación del contraste y conocido el valor del p-valor, tendremos presente la regla de decisión:

Regla de decisión		
$p-valor < \alpha$	$\Rightarrow$	**Rechazo** H_0
$p-valor \geq \alpha$	$\Rightarrow$	**No Rechazo** H_0

En esta sección vamos a poner ejemplos de algunos de estos contrastes, para ver qué tipo de problemas resuelven y tener una base para su aplicación.

Muestreando con R

Ahora bien, dado que la inferencia estadística es la metodología que nos permite inferir resultados, predicciones y generalizaciones sobre la población estadística basándose en la información contenida en las muestras representativas previamente elegidas por métodos de muestreo formales, parece lógico abordar en primer lugar las distintas técnicas de extracción de muestras de una población.

Muestreo aleatorio simple

La extracción de muestras aleatorias de tamaño predeterminado mediante la función `sample` nos permite tomar muestras de una población sin y con reposición. La función y sus argumentos: `sample(x, size, replace = FALSE, prob = NULL)` donde `x` es el vector de elementos o conjunto de enteros positivos, sobre los que se realiza la se lección; `size` es el tamaño muestral, `replace` nos permite realizar la extracción sin reposición (`FALSE`) o con reposición (`TRUE`) y `prob`: A vector of probability weights for obtaining the elements of the vector being sampled.

```
 x<- 1:15
 x
```

```
     [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

```
# Muestra aleatoria sin reposición de tamaño 15
 sample(x)
```

```
     [1] 3 4 13 5 12 8 6 14 15 10 7 1 2 11 9
```

```
# Muestra aleatoria con reposición de tamaño 15
 sample(x,replace=TRUE)
```

```
     [1] 8 10 6 5 3 6 6 5 10 5 5 10
```

```
# Muestras de tamaño 7 sin reposición y con reposición
 sample(x,7)
```

```
     [1] 12 11 5 13 2 9 10
```

```
 sample(x,7,replace=T)
```

```
     [1] 10 9 12 10 15 6 1
```

```
#Muestra aleatoria sin repetición de tamaño 60 de la base de datos iris
indices<- sample( 1:nrow( iris ), 60 )
iris.muestreado<- iris[ indices, ]

#Muestra aleatoria con repetición de tamaño 60 de la base de datos iris
indices<- sample( 1:nrow( iris ), 60, replace = TRUE )
iris.muestreado<- iris[ indices, ]

# Muestra aleatoria con reemplazamiento de 100 elementos
# de una Bernoulli
 sample(c(0,1), 100, replace = TRUE)
```

```
     [1] 0 1 1 0 1 0 1 1 0 1 1 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 1 0 1
    [38] 0 1 1 0 1 1 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 1 0 0 0 0 1 1 0
    [75] 1 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1
```

Muestreo sistemático

Para calcular un muestreo sistemático vamos a utilizar un paquete denominado *pps* y su orden "ppss" para calcular una muestra por muestreo sistemático.

```
library(pps)
x

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

ppss(x,6)

    [1] 4 6 8 10 12 2

# Elige el elemento 4 y como 12/6=2 elige de dos en dos
ppss(x,4)

    [1] 6 9 12 3
```

Muestreo estratificado

De nuevo se utiliza el paquete *pps* para llevar a cabo un muestreo sistemático y en particular la orden "ppssstrat" para obtener una muestra por muestreo estratificado.

```
library(pps)
tamanos<-c(10, 20, 30, 40, 50, 100, 90, 80, 70, 60)
# Indicamos el estrato al que pertenece cada tamaño
estratos<- c(1,1,1,2,2,3,3,3,3,3)
# n es un vector que contienen el tamaño de la muestra en cada estrato
n<- c(2,1,3)
ppssstrat(tamanos,estratos,n)

    [1] 2 3 5 6 7 9
```

Otros paquetes de R, como *sampling*, nos permiten realizar muestreo estratificados con o sin reemplazamiento.

```
library( sampling )
estratos<- strata( iris, stratanames = c("Species"), size = c(20,20,20), method = "srswor" )
iris.muestreado<- getdata( iris, estratos )
```

Para obtener un muestreo con reemplazamiento se sustituye el método srswor por el srswr.

Pruebas paramétricas y no paramétricas

Algunas técnicas estadísticas dentro de la Inferencia clásica se presentan en la siguiente tabla, junto con funciones en R.

Técnica estadística	Función
Ansari-Bradley Test	**ansari.test**
Bartlett Test for Homogeneity of Variances	**bartlett.test**
Exact Binomial Test	**binom.test**
Pearson's Chi-squared Test for Count Data	**chisq.test**
Test for Association/Correlation Between Paired Samples	**cor.test**
Fisher's Exact Test for Count Data	**fisher.test**
Fligner-Killeen Test for Homogeneity of Variances	**fligner.test**
Friedman Rank Sum Test	**friedman.test**
Kruskal-Wallis Rank Sum Test	**kruskal.test**
Kolmogorov-Smirnov Tests	**ks.test**
Cochran-Mantel-Haenszel Chi-Squared Test for Count Data	**mantelhaen.test**
McNemar's Chi-squared Test for Count Data	**mcnemar.test**
Mood Two-Sample Test of Scale	**mood.test**
Test for Equal Means in a One-Way Layout	**oneway.test**
Pairwise comparisons of proportions	**pairwise.prop.test**
Pairwise t tests	**pairwise.t.test**
Tabulate p values for pairwise comparisons	**pairwise.table**
Pairwise Wilcoxon rank sum tests	**pairwise.wilcox.test**
Power calculations for balanced one-way analysis of variance tests	**power.anova.test**
Power calculations two sample test for of proportions	**power.prop.test**
Power calculations for one and two sample t tests	**power.t.test**
Print method for power calculation object	**print.power.htest**
Test for Equal or Given Proportions	**prop.test**
Test for trend in proportions	**prop.trend.test**
Quade Test	**quade.test**
Shapiro-Wilk Normality Test	**shapiro.test**
Student's t-Test	**t.test**
F Test to Compare Two Variances	**var.test**
Wilcoxon Rank Sum and Signed Rank Tests	**wilcox.test**

Las pruebas estadística se pueden clasificar en:

a. *Pruebas paramétricas*: a partir de la información muestral sobre una población se infiere algún parámetro de ella, i.e., se estima o se toma una decisión sobre el parámetro poblacional, p.e., la media poblacional, la varianza poblacional, la proporción de un suceso...

b. *Pruebas no paramétricas*, se presenta como alternativa robusta a las anteriores cuando las distribuciones poblacionales no son conocidas, basando sus conclusiones en las propiedades de las muestras ordenadas, los rangos

de las muestras, la mediana, entre otras; o cuando se desea tomar decisiones sobre relaciones o similitudes entre variables poblacionales.

Los comandos de R que utilizaremos en los siguientes apartados nos suministrarán estimaciones puntuales, intervalos de confianza y contrastes de hipótesis. Además, los argumentos nos permitirán especificar tests unilaterales (`alternative="less" or alternative="greater"`) y tests bilaterales (`alternative="two.sided"`).

Contraste de hipótesis para el parámetro θ		
Bilateral	**Unilateral Izquierda**	**Unilateral Derecha**
$H_0 : \theta = \theta_0$	$H_0 : \theta \geq \theta_0$	$H_0 : \theta \leq \theta_0$
$H_1 : \theta \neq \theta_0$	$H_1 : \theta < \theta_0$	$H_1 : \theta > \theta_0$
`"two.sided"`	`"less"`	`"greater"`

En los restantes apartados de esta sección, analizaremos valores de expresión de genes de la base de datos *leukemia*, incluida en el paquete de R denominado *spikeslab*, del trabajo de investigación ampliamente referenciado de Golub et al. (1999). En este artículo Golub y sus colegas midieron el nivel de expresión de 3571 genes humanos en 72 pacientes con leucemia de los cuales 47 sufrían leucemia linfoblástica aguda (ALL) y los 25 restantes leucemia mieloide aguda (AML), con el objetivo de detectar un subconjunto de genes que pudiesen ser usado como tests diagnósticos, permitiendo evaluar si un nuevo paciente padece alguno de estos tipos de cancer.

Antes de ponernos a ello, etiquetaremos los códigos de la variable Y, 0 como ALL y 1 como AML y grabaremos la nueva variable *factor* en el fichero de trabajo.

```
table(leukemia$Y)
leukemia$factor<-factor(leukemia$Y, levels=0:1, labels=c("ALL","AML"))
table(leukemia$factor)
```

Pruebas paramétricas

En este apartado, hemos optado por comenzar con el test F de Snedecor, ya que el test t de Student require asumir igualdad o no de varianzas.

Contrastes de varianzas

Test F de Snedecor para dos muestras independientes Se trata de un test para contrastar las igualdades de varianzas en dos poblaciones independientes y con distribuciones normales $\mathcal{N}(\mu_1, \sigma_1)$ y $\mathcal{N}(\mu_2, \sigma_2)$, respectivamente. Obsérvese que el procedimiento se basa en el ratio de las varianzas.

Contraste de Hipótesis	*Estadístico del contraste*
$H_0 : \sigma_1^2/\sigma_2^2 = 1$ $H_1 : \sigma_1^2/\sigma_2^2 \neq 1$	$T = \frac{S_1^2}{S_2^2} \equiv_{H_0} F_{n_1-1,n_2-1}$

Para realizar el test F de Snedecor para dos muestras independientes utilizamos la función "`var.test`":

```
## Default S3 method:
```

```
    var.test(x, y, ratio = 1, alternative = c("two.sided", "less", "greater"), conf.level = 0.95,
...)
   ## S3 method for class 'formula'
    var.test(formula, data, subset, na.action, ...)
```

Ejemplo 3.1 *Para ilustrar este contraste, consideraremos el gen Gdf5 del artículo de Golub et al. (1999), que tras una búsqueda rápida en NCBI parece probable que no esté directamente relacionado con la leucemia. Los valores de expresión correspondientes se encuentran en la columna 3396 (denotada por x.3395)*

```
var.test(leukemia$x.3395 leukemia$factor)
F test to compare two variances
data: leukemia$x.3395 by leukemia$factor
F = 0.71067, num df = 46, denom df = 24, p-value = 0.3151
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.3351383 1.3911130
sample estimates:
ratio of variances
0.7106712
```

A la vista del p-valor, no se rechaza la hipótesis nula de igualdad de varianzas.

Ejemplo 3.2 *Análogamente, para el el gen CCND3 Cyclin D3, situado en la columna 1041, la hipótesis nula de igualdad de varianzas entre los pacientes ALL y AML puede ser testada.*

```
var.test(leukemia$x.1040 leukemia$factor)
F test to compare two variances
data: leukemia$x.1040 by leukemia$factor
F = 0.90082, num df = 46, denom df = 24, p-value = 0.7413
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.4248098 1.7633269
sample estimates:
ratio of variances
1.891529
```

A la vista del p-valor, no se rechaza la hipótesis nula de igualdad de varianzas.

Ejemplo 3.3 *Para el fichero ElPulso, contrastar la igualdad de varianzas de la altura en cm entre hombre y mujeres.*

```
# F para igualdad de varianzas
 var.test(ElPulso$Altura.cm ~ ElPulso$Sexo)
```

```
   F test to compare two variances
data: ElPulso$Altura.cm by ElPulso$Sexo
F = 1.0158, num df = 56, denom df = 34, p-value = 0.9796
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.538742 1.829150
sample estimates:
ratio of variances
1.015811
```

Como el p-valor es 0.9796, mayor que el nivel de significación $\alpha = 0{,}05$, no rechazamos la hipótesis nula de igualdad de varianzas poblacionales.

Contrastes de medias El comando que utilizaremos para los contrastes de medias:

```
## Default S3 method
 t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu = 0, paired = FALSE,
var.equal = FALSE,
+ conf.level = 0.95, ...)
## S3 method for class 'formula'
 t.test(formula, data, subset, na.action, ...)
```

Test t de Student para una muestra: Prueba de conformidad de la media Este procedimiento se utiliza cuando queremos contrastar si una muestra proviene de un población normal de media la constante μ_0 especificada y σ^2 desconocida. Nótese que en la mayoría de las situaciones que se presentan en las investigaciones relacionadas con los valores de expresión génica, la desviación típica de la población σ es desconocida.

Contraste de Hipótesis	*Estadístico del contraste*
$H_0 : \mu = \mu_0$ $H_1 : \mu \neq \mu_0$	$T = \frac{\overline{X}-\mu_0}{\sqrt{S^2/n}} \equiv_{H_0} t_{n-1}$

donde $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{X})^2$ es la covarianza.

Ejemplo 3.4 *Se utilizaron microarrays para medir el nivel de expresión de todos los genes de la levadura en dos medios de cultivo diferentes:*

i. medio mínimo, medido en el canal verde del microarray

ii. medio mínimo + metionina, medido en el canal rojo del microarray

Se realizaron tres repeticiones del experimento y se calcularon los log-ratios, $log_{10}(rojo/verde)$, para cada uno de los microarrays. Para un gen determinado se tiene los siguientes valores de log-ratio: 2, 3.1 y 0.3. Y nos planteamos las siguientes cuestiones ¿se activa este gen de manera significativa por la metionina? ¿cuántos falsos positivos esperaríamos con este nivel de significación, si se aplica la prueba de 6200 genes?

```
x <- c(2.0, 3.1, 0.3)
print(x)
n <- length(x)
print(n)
sample.mean <- mean(x)
sample.var <- mean((x - sample.mean)^2)
sample.sd <- sqrt(sample.var)
print(sample.sd)
sd.est <- sd(x)
print(sd.est)
print(standard.error <- sd.est/sqrt(n))
ref.mean <- 0
t.obs <- (sample.mean - ref.mean )/standard.error
print(t.obs)
y <- seq(from=-5,to=5,by=0.1)
plot(y, dnorm(y), tpye="l", col="darkblue", type="l",lwd=2, panel.first=grid(col="black"))
i <- 0
for (d in c(1,2,3,4,5,10,100,1000)) {
i <- i+1
lines(y, dt(y,df=i),type="l",col=i)
}
p.value <- pt(t.obs,df=n-1,lower.tail=F)
print(p.value)
g <- 6200
e.value <- p.value*g
print(e.value)
t.test(x,alternative="greater")
```

- El e-value representa el número esperado de los genes falsos positivos. En bioinformática, las pruebas múltiples son muy frecuentes, por ejemplo en la evaluación de la significación de cada gen en un chip representa miles de pruebas simultáneas. Una primera aproximación para corregir múltiples pruebas es aplicar la regla de Bonferroni que significa adaptar el p-valor umbral al número de pruebas simultáneas, i.e., $\alpha \leq 1/N$.

- Si $p = P(X > 0) = 0{,}01$ y la base de datos contiene N = 100.000 entradas, esperamos obtener $N * p =$ 1000 falsos positivos. El e-valor (valor esperado) también permite tomar en cuenta este efecto: e-value = N * p-value. Así, en lugar de establecer un umbral en el p-valor, hay que establecer un umbral en el e-valor. De manera que si queremos evitar falsos positivos, este umbral debería ser siempre menor que 1 lo que equivale a la corrección de Bonferroni, que consiste en la adaptación del umbral en el valor de p.

- Otra corrección consiste en la estimación de la tasa de error de la familia Wise (FWER), que es la probabilidad de observar al menos un falso positivo en el conjunto de las pruebas y se calcula como sigue: $FWER = 1 - (1 - p - valor)^N$.
- Otro enfoque es considerar, para un determinado p-valor, es la tasa de falso descubrimiento (FDR), es decir, la proporción de predicciones falsas dentro de un conjunto de pruebas significativas y se calcula como sigue: $FDR = FP/(FP+TP)$, donde FP es el número de falsos positivos y TP el número de verdaderos positivos.

Ejemplo 3.5 *De nuevo trabajaremos con el conjunto de datos leukemia de Datamaster.RData y contrastaremos si la media de expresión génica del gen Gdf5 de ALL es* $\mu = 0$.

```
t.test(leukemia$x.3395[leukemia$Y==0])
One Sample t-test
data: leukemia$x.3395[leukemia$Y == 0]
t = 0.0011339, df = 46, p-value = 0.9991
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
-0.1344803 0.1346319
sample estimates:
mean of x
7.579787e-05
```

Obsérvese que la salida de resultados incluye el t_{obs}, el p-valor y el intervalo de confianza del 95 % para μ, este último dado por (-0.1344803 0.1346319) que contiene el cero. A partir del p-valor no rechazamos H_0.

En el ejemplo anterior, el contraste es bilateral ya que H_1 es verdadera si $\mu < 0$ o $\mu > 0$, y considerar un contraste unilateral hace que el procedimiento sea ligeramente distinto como se ilustra en el ejemplo siguiente.

Ejemplo 3.6 *Asumiendo que los valores de expresión de CCND3 Cyclin D3 para pacientes ALL, contrastaremos si es* $\mu_{ALL,x,155} \geq 0$.

```
t.test(leukemia$x.1040[leukemia$factor==”ALL”], alternative="greater")
One Sample t-test
data: leukemia$x.1040[leukemia$factor == ”ALL”]
t = 41.733, df = 46, p-value <2.2e-16
alternative hypothesis: true mean is greater than 0
95 percent confidence interval:
1.815445 Inf
sample estimates:
mean of x
1.891529
```

El valor de t indica que, en relación con su error estándar, la media difiere de cero, lo que concuerda con un p-valor muy cercano a cero, que nos lleva a rechazar H_0.

Ejemplo 3.7 *Contrastar si la altura media en cm de esta población es* $\mu = 0$.

```
t.test(ElPulso$Altura.cm, alternative="two.sided", conf.level=.95)# Ho: μ = 0
   One Sample t-test
data: ElPulso$Altura.cm
t = 180.12, df = 91, p-value<2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 172.6173 176.4670
 sample estimates:
 mean of x
 174.5422
```

La salida arroja un p-valor significativamente inferior al nivel de significación fijado en $\alpha = 0{,}05$, luego rechazamos la hipótesis nula.

Ejemplo 3.8 *Contrastar si la altura media en cm de los hombres de esta población es* $\mu = 170$.

```
t.test(ElPulso$Altura.cm[ElPulso$Sexo=="Hombre"], alternative="two.sided",
+ mu=170, conf.level=.95)# Ho: μ_hombres = 170
   One Sample t-test
data: ElPulso$Altura.cm[ElPulso$Sexo=="Hombre"]]
t = 11.182, df = 56, p-value = 6.841e-16
alternative hypothesis: true mean is not equal to 170
95 percent confidence interval:
 177.9755 181.4568
sample estimates:
mean of x
 179.7161
```

Se observa que el p-valor obtenido es inferior al nivel de significación prefijado $\alpha = 0{,}05$, luego rechazamos la hipótesis.

Ejemplo 3.9 *Ahora te toca a ti. Contrastar si el pulso medio antes de la actividad realizada para hombres fumadores es 75.*

```
t.test(ElPulso$Pulse1[ElPulso$Sexo=="Hombre"& ElPulso$Fumar=="Fuma"],
+ alternative="two.sided", mu=75, conf.level=.95)
```

Test t de Student para dos muestras independientes Frecuentemente, dentro del análisis estadístico se plantea la comparación de las medias de dos grupos; en general, un grupo "experimental" y un grupo "de control" o "de contraste". Este contraste de hipótesis permite comparar las medias para dos muestras distintas.

Seguramente, observaste que entre los argumentos de `t.test` por defecto se presupone `var.equal = FALSE` y por otro lado, trabajamos con igualdad o no de varianzas en la comparación de dos poblaciones. Por este motivo, parece lógico que la realización del contraste de igualdad de varianzas sea previo al contraste de medias para dos muestras independientes para que a la vista de los resultados obtenidos, señalemos en los argumentos de `t.test, var.equal=T` si las varianzas de las dos poblaciones son iguales o mantengamos la opción por defecto `var.equal=F` si las varianzas de ambas poblaciones no se suponen iguales.

Contraste de Hipótesis	*Estadístico del contraste*
$H_0 : \mu_1 = \mu_2$ $H_1 : \mu_1 \neq \mu_2$	$T = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\left(\frac{n_1-1}{n_1+n_2-2}S_1^2 + \frac{n_2-1}{n_1+n_2-2}S_2^2\right)\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}} \equiv_{H_0} t_{n_1+n_2-2}$

El valor predeterminado asume desigualdad de varianzas. En este contexto, es importante tener en cuenta que el estadístico es una variable distribuida según el modelo de probabilidad t de Student, que cuando se asumen varianzas iguales el grado de libertad del estadístico del contraste es igual a $n_1 + n_2 - 2$, y cuando las varianzas no son iguales, R considera la aproximación de Welch (1938), dada por:

$$\upsilon = \frac{\left(\frac{s^2_{n_1-1}}{n_1} + \frac{s^2_{n_2-1}}{n_2}\right)^2}{\frac{\left(\frac{s^2_{n_1-1}}{n_1}\right)^2}{n_1-1} + \frac{\left(\frac{s^2_{n_2-1}}{n_2}\right)^2}{n_2-1}}$$

En esta segunda situación, la estimación del estadístico será menor que la primera porque se pierde precisión con la desigualdad de las varianzas.

Ejemplo 3.10 *Considerando el objetivo de Golub et al. (1999) de seleccionar aquellos genes que muestren diferencias estadísticamente significativas en expresión entre pacientes AML y ALL, los autores mantienen que el gen CCND3 Cyclin D3 juega un importante rol con respecto a la discriminación de los paciente con ALL de pacientes con AML. Asumiendo que las medias no son iguales, la hipótesis nula de igualdad de medias debe ser contrastada usando la función* t.test*, bajo la especificación de* var.equal=T *como vimos en el subapartado anterior.*

```
t.test(leukemia$x.1040 ~ leukemia$Y,var.equal=T)
Two Sample t-test
data: leukemia$x.1040 by leukemia$Y
t = 3.118, df = 70, p-value = 0.002642
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.0880387 0.4005939
sample estimates:
mean in group 0 mean in group 1
1.891529 1.647213
```

A partir del p-valor, rechazamos la hipótesis nula de medias poblacionales iguales.

Ejemplo 3.11 *Repetir el ejemplo anterior, pero especificando var.equal = F. Comparar los resultados obtenidos con los del ejemplo anterior.*

Ejemplo 3.12 *Contrastar la igualdad de pulsos medios después de la actividad entre hombres y mujeres, i.e.,* $H_0 : \mu_{pulso2,hombre} = \mu_{pulso2,mujer}$.

El procedimiento requiere que primero llevemos a cabo el correspondiente contraste de igualdad de varianzas, i.e., $H_0 : \sigma^2_{pulso2,hombre} = \sigma^2_{pulso2,mujer}$.

```
var.test(ElPulso$Pulso2 ~ ElPulso$Sexo)
F test to compare two variances
data: ElPulso$Pulso2 by ElPulso$Sexo
F = 0.40384, num df = 56, denom df = 34, p-value = 0.002546
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.2141786 0.7271842
sample estimates:
ratio of variances
0.4038387
```

Y dado que rechazamos la hipótesis de igualdad de varianzas, el argumento $var.equal = F$ del comando $t.test$ se mantiene. A continuación, se muestran los resultados de la ejecución de la sentencia

```
t.test(ElPulso$Altura.cm ~ ElPulso$Sexo) # Sexo es un factor binario

        Welch Two Sample t-test
data: ElPulso$Altura.cm by ElPulso$Sexo
t = 9.7007, df = 72.514, p-value = 9.778e-15
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
10.80570 16.39458
sample estimates:
mean in group Hombre mean in group Mujer
179.7161 166.1160
```

Se observa que el $p - valor = 9.778\text{e-}15$ es significativamente menor que $\alpha = 0{,}05$; por tanto, rechazamos la hipótesis nula de igualdad de medias. En otras palabras, existen diferencias estadísticamente significativas entre el pulso medio después de la actividad de los hombres y de las mujeres. Obsérvese que la salida de los resultados nos informa de la aplicación de `Welch Two Sample t-test`.

```
# test t para dos muestras independientes: alternativa.
t.test(y1,y2) # donde y1 e y2 son numéricos
```

Test t de Student para dos muestras dependientes Este contraste de hipótesis permite comparar las medias para dos muestras dependientes (también llamadas muestras relacionadas o apareadas), i.e., se consideraba que los valores de ambas muestras están correlacionados y también lo llevaremos a cabo mediante la función `t.test`, pero modificando la variable lógica del argumento `paired = FALSE` por defecto que presupone independencia de las poblaciones de las que proceden las correspondientes muestras por la siguiente `paired=TRUE`.

Así, si planteamos una situación en la que un mismo objeto recibe dos tratamientos diferentes, tal como pruebas con dos tipos de fármacos, puede resultar de interés comparar el efecto del primer tratamiento de la muestra de individuos $fármaco_1$ y $fármaco_2$. En este caso, se dice que los datos están emparejados y se considera la variable $D = X_1 - X_2$, a la que aplicamos el test t para una muestra.

Contraste de Hipótesis	*Estadístico del contraste*
$H_0 : \mu_1 - \mu_2 = 0$ $H_1 : \mu_1 - \mu_2 \neq 0$	$T = \frac{\overline{D}}{\sqrt{S_D^2}} \equiv_{H_0} t_{n-1}$

```
# Contrastamos si existen diferencias de medias entre pulso antes y pulso después, con α = 0,99
 t.test (Pulso1,Pulso2, paired=T, conf.level=0.99)
```

```
    t.test( ElPulso$Pulso1,ElPulso$Pulso2, paired=T, conf.level=0.99)

        Paired t-test

     data: ElPulso$Pulso1 and ElPulso$Pulso2
     t = -5.0769, df = 91, p-value = 2.023e-06
     alternative hypothesis: true difference in means is not equal to 0
     99 percent confidence interval:
     -10.825552 -3.435317
     sample estimates:
     mean of the differences
     -7.130435
```

Se observa que el $p-valor$ es significativamente menor que $\alpha = 0{,}01$; por tanto, rechazamos la hipótesis nula de diferencias de medias igual a cero. En otras palabras, existen diferencias estadísticamente significativas entre el pulso antes y después. Intervalos de confianza para la media obtiene los límites inferior y superior del intervalo de confianza que valora la precisión de la estimación que estamos realizando para la diferencia de medias.

Contrastes de proporciones

Contraste para una proporción: Prueba de conformidad En poblaciones dicotómicas con una proporción de éxitos p, el estimador puntual de este parámetro poblacional es la proporción muestral de éxitos, $\widehat{p}$, que coincide con la media de la muestra cuando se codifica como 1, éxito y como 0, fracaso. Teniendo en cuenta, además, que a partir de un tamaño muestral suficientemente grande, el estadístico p tiene una distribución asintóticamente normal. El intervalo de confianza para la proporción poblacional está centrado en la proporción muestral; siendo sus límites superior e inferior:

$$\widehat{p} \pm z_{1-\alpha/2}\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}}$$

En este marco, considerando la proporción como la media poblacional de una variable dicotómica, el contraste de hipótesis para una proporción vendría dado por:

Contraste de Hipótesis	*Estadístico del contraste*
$H_0 : p = p_0$ $H_1 : p \neq p_0$	$T = \frac{\widehat{p}-p_0}{\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}}} \equiv_{H_0} \mathcal{N}(0,1)$

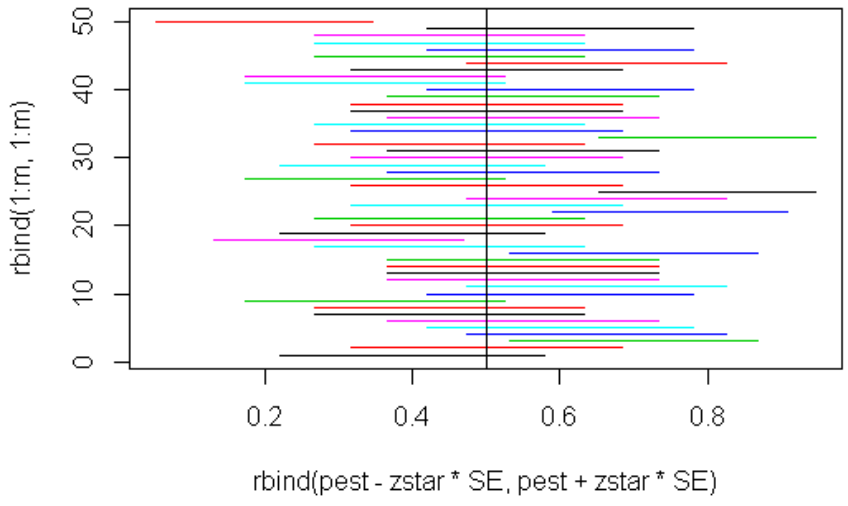
`prop.test(x, n, p=0.5, alternative="two.sided", conf.level=0.95, correct=FALSE)`

Argumentos de esta función son: el número de éxitos (`x`), el número de pruebas (`n`), la hipótesis nula (`p = p0`), la hipótesis alternativa ("`two.sided`" o "`less`" o "`greater`"), el nivel de confianza $(1-\alpha)$ e indicar si se aplica o no la corrección por continuidad de Yates.

Idénticamente a las funciones de pruebas paramétricas y no paramétricas vistas hasta ahora, por defecto se tiene la opción `alternative = "two.sided"`(contraste bilateral) pero se puede elegir `alternative = "less"`(contraste unilateral a la izquierda) o `alternative = "greater"`(contraste unilateral a la derecha).

En primer lugar, analizaremos el significado del intervalo de confianza, visualizando con cierta seguridad de la presencia del parámetro de la población dentro de un intervalo construido a partir de la muestra.

```
#Lanzamos la moneda 20 veces, y estimamos p con la proporción p̂=pest de caras obtenidas
 n=20; p = .5;pest = rbinom(1,n,p)/n
#Realizamos el lanzamiento de las 20 monedas m = 50 veces
 n=20; p = .5;m=50;pest = rbinom(m,n,p)/n
#Fijamos el nivel de confianza 1 - α = 0.90 y calculamos los intervalo: p̂ ± z_{1-α/2}√(p̂(1-p̂)/n)
 alpha = 0.10;zstar = qnorm(1-alpha/2);SE = sqrt(pest*(1-pest)/n)
#Representamos los m = 50 intervalos
 matplot(rbind(pest - zstar*SE, pest + zstar*SE),rbind(1:m,1:m),type="l",lty=1)
#Marcamos la línea para p = 0.5.
 abline(v=p)
```




```
caras = rbinom(1, size=100, pr = .5)
caras

    [1] 47

prop.test(caras,100)
```

```
    1-sample proportions test with continuity correction
data: caras out of 100, null probability 0.5
X-squared = 0.25, df = 1, p-value = 0.6171
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.3703535 0.5719775
sample estimates:
 p
0.47
```

A la vista está que el p-valor es bastante alto, 0.6171, no rechazaremos la hipótesis nula de que p=0.5.

Ejemplo 3.13 *Supongamos que para un cierto microARN se quiere contrastar la hipótesis de que la probabilidad de una purina es igual al un cierto valor p_0. Sin embargo, existen razones para creer que la probabilidad es mayor. Por lo que nuestro contraste sería: $H_0 : p \leq p_0$ vs. $H_1 : p > p_0$. Supongamos que la secuenciación revela que el microARN tiene k purinas de un total de n. Suponiendo que la distribución binomial se mantiene, la hipótesis nula puede ser testada calculando el p-valor de $P(X \geq k)$. En particular, si el microARN de longitud 22 contiene 18 purinas $H_0 : p \leq 0{,}7$ vs. $H_1 : p > 0{,}7$*

$$P(X \geq 18) = 1 - pbinom(17, 22, 0{,}7) = 0{,}1645 \geq 0{,}05 = \alpha$$

por lo que no rechazamos la hipótesis nula. También, se puede llevar a cabo a través de la función binom.test.

```
binom.test(18, 22, p = 0.7, alternative = "greater",conf.level = 0.95)
 Exact binomial test
data: 18 and 22
number of successes = 18, number of trials = 22, p-value = 0.1645
alternative hypothesis: true probability of success is greater than 0.7
95 percent confidence interval:
 0.6309089 1.0000000
sample estimates:
probability of success
 0.8181818
```

Contraste para dos proporciones Supongamos que al probar dos nuevos fármacos, fármaco 1 y fármaco 2, en 150 y 125 unidades experimentales respectivamente, se tiene que 14 unidades no han reaccionado bien con el fármaco1 y 15 unidades no han reaccionado bien con el fármaco 2. ¿Hay una evidencia estadística que nos permita asegurar que el porcentaje de reacciones adversas con ambos fármacos es distinto?

Denotando, $\widehat{p}_1$ la proporción de reacciones adversas con el fármaco 1 y 1 y $\widehat{p}_2$ con el fármaco 2, en un contexto general:

Contraste de Hipótesis	*Estadístico del contraste*
$H_0 : p_1 = p_2$ $H_1 : p_1 \neq p_2$	$T = \frac{\widehat{p}_1 - \widehat{p}_2 - (p_1 - p_2)}{\sqrt{\frac{\widehat{p}_1(1-\widehat{p}_1)}{n_1} + \frac{\widehat{p}_2(1-\widehat{p}_2)}{n_2}}} \equiv_{H_0} \mathcal{N}(0,1)$

```
prop<-c(14,15)
n<-c(150,125)
prop.test(prop,n,alt="two.sided")

        2-sample test for equality of proportions with continuity correction
data: prop out of n
X-squared = 0.2702, df = 1, p-value = 0.6032
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.10756913 0.05423579
sample estimates:
 prop 1     prop 2
0.09333333 0.12000000
```

Por tanto, a partir del nivel de significación fijado y a la vista de $p - value = 0.6032$, no tenemos evidencia estadística suficiente para rechazar la hipótesis nula.

Ejemplo 3.14 *Ejemplo basado en el estudio de las propiedades codificantes de los exones alternativos: http://bioinformatica.upf.edu/2005/projectes05/3.7.1/*

- *Hipótesis nula (H_0): No hay diferencias significativas en el uso de codones en los exones constitutivos y alternativos la posibilidad de que la observación llevada a cabo en la muestra sea únicamente fruto del azar.*
- *Hipótesis alternativa H_1: : Existen diferencias significativas en el uso de codones en los exones constitutivos y alternativos. la posibilidad de que la observación llevada a cabo en la muestra es el reflejo de la situación real en la población.*

```
Alanina <- as.table(cbind(c(9413, 12012, 4143, 10283), c(3598, 4389, 1067, 3861)))
dimnames(Alanina) <- list(codon = c("GCA", "GCC", "GCG", "GCT"), exon = c("CONST","ALTERNA"))
(XsqAla <- chisq.test(Alanina))
attributes(XsqAla)
XsqAla$observed
XsqAla$expected
XsqAla$residuals
XsqAla$stdres
Glutamina <- as.table(rbind(c(14272,5527), c(15087,6066)))
dimnames(Glutamina) <- list(codon = c("GAA", "GAG"), exon = c("CONST","ALTERNA"))
(XsqGlu <- chisq.test(Glutamina))
XsqGlu$observed
```

```
XsqGlu$expected
XsqGlu$residuals
XsqGlu$stdres
```

Ejemplo 3.15 *Datos extraídos de Almoguera (2011) accesible en https://repositorio.uam.es/handle/10486/6784 sobre el estudio del alelo ATA como factor de susceptibilidad a la esquizofrenia. Se agruparon los genotipos según el número de alelos ATA (2, 1 ó 0) y las distribuciones del recuento de mujeres esquizofrénicas (casos) y mujeres de la muestra objeto de estudio (muestra), portadoras de 2, 1 o ningún alelo ATA, respectivamente, son: muestra* $= c(20, 197, 156)$ *y casos* $= c(13, 27, 48)$.

```
> names(muestra) =c("2","1","0")
> muestra=c(20,197,156)
> muestra
[1] 20 197 156
> casos=c(13,27,48)
> prop.test(casos,muestra)
 3-sample test for equality of proportions without continuity correction
data: casos out of muestra
X-squared = 34.163, df = 2, p-value = 3.816e-08
alternative hypothesis: two.sided
sample estimates:
 prop 1 prop 2 prop 3
0.6500000 0.1370558 0.3076923
Warning message:
In prop.test(casos, muestra) : Chi-squared approximation may be incorrect
```

Es interesante controlar este aviso, a través del cálculo del estadístico chi-cuadrado como sigue:

```
> m <- matrix(c(casos, muestra-casos), ncol=2)
> chisq.test(m)
 Pearson's Chi-squared test
data: m
X-squared = 34.163, df = 2, p-value = 3.816e-08
Warning message:
In chisq.test(m) : Chi-squared approximation may be incorrect
> chisq.test(m)$expected
 [,1] [,2]
[1,] 4.718499 15.2815
[2,] 46.477212 150.5228
[3,] 36.804290 119.1957
Warning message:
In chisq.test(m) : Chi-squared approximation may be incorrect
```

Este problema aparece cuando trabajamos con muestra de pequeño tamaño, como puede apreciarse cuando manejamos frecuencias mayores.

```
> names(muestra) =c("2","1","0")
> muestra=c(200,1970,1560)
> muestra
[1] 200 1970 1560
> casos=c(130,270,480)
> prop.test(casos,muestra)
 3-sample test for equality of proportions without continuity correction
data: casos out of muestra
X-squared = 341.63, df = 2, p-value <2.2e-16
alternative hypothesis: two.sided
sample estimates:
 prop 1 prop 2 prop 3
0.6500000 0.1370558 0.3076923
 muestra=c(20,197,156)
```

Una alternativa es aplicar el test exacto de Fisher que se describe en el apartado de inferencia no paramétrica, pero que adelantamos el resultado:

```
> fisher.test(m)
 Fisher's Exact Test for Count Data
data: m
p-value = 8.493e-08
alternative hypothesis: two.sided
```

Por tanto, las proporciones de mujeres esquizofrénicas portadoras de 2, 1 o 0 alelos no coinciden, y las diferencias observadas no son debidas al azar, sino que son estructurales.

```
Data from Fleiss (1981), p. 139.
H0: The null hypothesis is that the four populations from which the patients were drawn have the same true proportion of smokers.
HA: The alternative is that this proportion is different in at least one of the populations.
smokers<- c( 83, 90, 129, 70 )
patients<- c( 86, 93, 136, 82 )
prop.test(smokers, patients)

        4-sample test for equality of proportions without continuity
        correction
data: smokers out of patients
X-squared = 12.6004, df = 3, p-value = 0.005585
alternative hypothesis: two.sided
sample estimates:
```

```
 prop 1 prop 2 prop 3 prop 4
0.9651163 0.9677419 0.9485294 0.8536585
```

```
table(Smokes)
```

```
Smokes
 1 2
28 64
```

```
prop.test(c(28),c(64), alternative="two.sided", p=.5, conf.level=.95,correct=FALSE)
```

```
    1-sample proportions test without continuity correction
data: c(28) out of c(64), null probability 0.5
X-squared = 1, df = 1, p-value = 0.3173
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.3229401 0.5591379
sample estimates:
 p
0.4375
```

Resulta de interés señalar que la fórmula implementada en R para el cálculo del intervalo de confianza de una proporción viene dada por el intervalo de Wilson:

$$\frac{p + \frac{1}{2n}z^2_{\alpha/2} \pm z_{\alpha/2}\sqrt{\frac{p(1-p)}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{n}{2}z^2_{\alpha/2}}$$

La diferencia en los resultados puedes comprobarla tú mismo calculando el intervalo de confianza para la proporción p de individuos que fuman con una confianza del 95 % considerando:

$$p \pm z_{\alpha/2}\sqrt{\frac{p(1-p)}{n}}$$

```
n<-92; pest<-28/64
alpha<- 0.05
z<- qnorm(1-alpha/2)
e<- z*sqrt(pest*(1-pest)/n)
n<-92; pest<-28/64
alpha<- 0.05
z<- qnorm(1-alpha/2)
e<- z*sqrt(pest*(1-pest)/n)
pest-e;pest+e
```

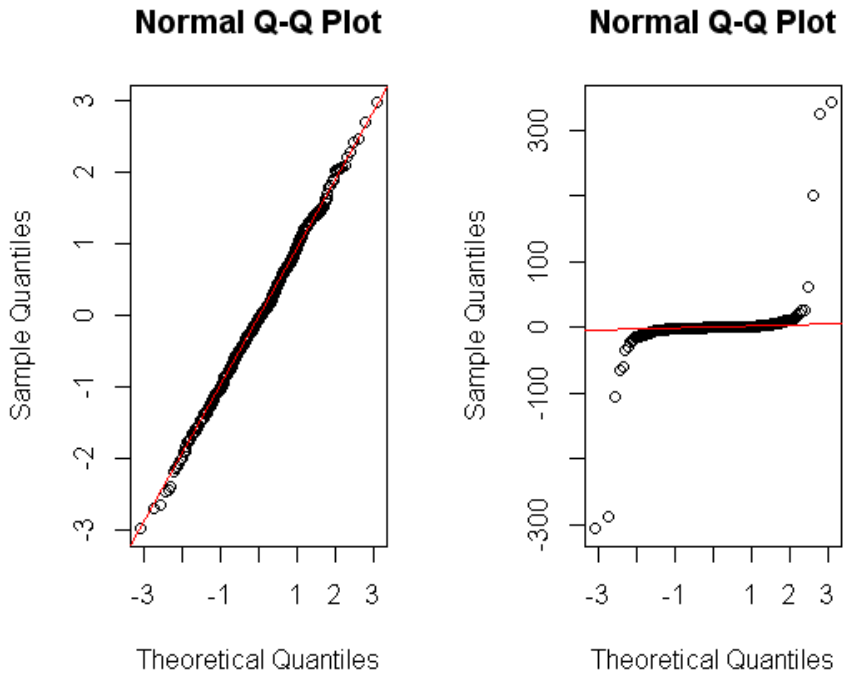
```
[1] 0.3361312
[1] 0.5388688
```

Pruebas no paramétricas

Contraste de normalidad Habitualmente, en inferencia paramétrica se requiere que las distribuciones de las poblaciones bajo estudio sean normales. En este sentido, aunque el Teorema Central del Límite permite el uso de pruebas paramétricas en poblaciones no normales si el tamaño muestral suficientemente grande, en caso de no especificar la distribución, es posible contrastar la hipótesis de normalidad de las poblaciones del estudio mediante los *contrastes de normalidad*, tras descartar situaciones claras mediante un procedimiento gráfico. Por ejemplo, podemos rechazar la normalidad de los datos usando el gráfico q-q (cuantil-cuantil) normal que representa los datos de la variable frente a los datos esperados si la distribución fuese normal. En el paquete base de R, la función *qqnorm* proporciona un gráfico q-q normal de los valores de la variable y la función *qqline* facilita el procedimiento incluyendo una línea q-q normal "teórica"que pasa por el primer y tercer cuartil, rechazando la normalidad de los datos si los puntos se encuentran alejados de la línea.

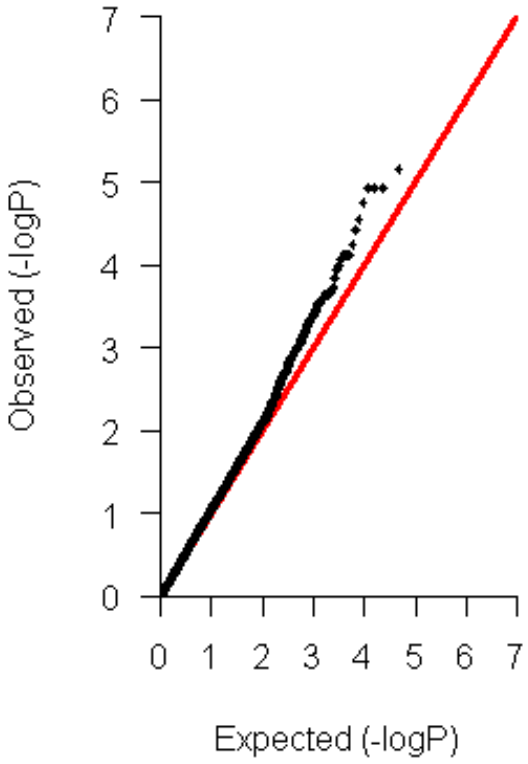
Ejemplo 3.16 *En el siguiente ejemplo, se representan los gráficos cuantil-cuantil con datos generados a partir de una distribución normal y, análogamente, para datos generados desde una distribución de Cauchy (distribución simétrica, pero con colas más pesadas que el modelo normal).*

```
set.seed(42)
n <- 500
x <- rnorm(n)
y <- rcauchy(n)
op <- par(mfrow=c(1,2))
qqnorm(x,xlab = ”Theoretical Quantiles”, ylab = ”Sample Quantiles”);qqline(x,col=2)
qqnorm(y,xlab = ”Theoretical Quantiles”, ylab = ”Sample Quantiles”);qqline(y,col=2)
par(op)
```



Ejemplo 3.17 *En el estudio del genoma completo, genome-wide association study 'GWAS' o Whole genome association study 'WGA', de la diabetes tipo 2, el siguiente script genera el gráfico q-q de los resultados de asociación dentro de su contexto genómico, anotaciones de genes y patrones de desequilibrio de vinculación local.*

```
pvals <- read.table("c:/Bioinformatica/DGI_chr3_pvals.txt", header=T)
observed <- sort(pvals$PVAL)
lobs <- -(log10(observed))
expected <- c(1:length(observed))
lexp <- -(log10(expected / (length(expected)+1)))
plot(c(0,7), c(0,7), col =" red", lwd = 3, type =" l", xlab =" Expected(−logP)",
 + ylab="Observed (-logP)", xlim=c(0,7), ylim=c(0,7), las=1,
 + xaxs ="i", yaxs ="i", bty ="l")
points(lexp, lobs, pch=23, cex=.4, bg="black")
```



Aunque la distribución observada de los p-valores coincide estrechamente con la distribución esperada en la mayor parte de su rango, se observa un exceso de p-valores < 0.001 en la cola. En detalle, en *Saxena et al. (2007). Whole-genome association analysis identifies novel loci for type 2 diabetes and triglyceride levels, Science 316(5829):1331-6.*

Sobre los tests de normalidad univariados, cabe mencionar que incluyen una gran variedad de pruebas entre las que se recomienda el uso del *test de Shapiro-Wilk* para muestras pequeñas $n \leq 50$ y del *test de Kolmogorov-*

Smirnov y su adaptación, el *test de Lilliefors* si las muestras son grandes, salvo que los datos vengan dados en una distribución de frecuencias en cuyo caso emplearemos la distribución χ^2. En los tres tests, las hipótesis nula y alternativa son las mismas, aunque la teoría detrás de cada uno de ellos es diferente:

H_0: Las observaciones provienen de una población distribuida normalmente (es decir, los datos se distribuyen normalmente)

H_1: Las observaciones provienen de una población no distribuida normalmente (es decir, los datos no se distribuyen normalmente)

Test de Shapiro-Wilk El test de Shapiro-Wilk test está basado en el grado de linealidad en un q-q plot.

Ejemplo 3.18 *Para contrastar la hipótesis de que los valores de expresión del gen Gdf5 para pacientes ALL están normalmente , este test puede ser usado del siguiente modo.*

```
shapiro.test(leukemia$x.3395[leukemia$Y==0])
Shapiro-Wilk normality test
data: leukemia$x.3395[leukemia$Y == 0]
W = 0.981, p-value = 0.6348
```

Como el p-valor es mayor que 0.05, la hipótesis nula de que los valores de expresión del gen Gdf5 sigue una distribución normal no es rechazada.

Ejemplo 3.19 *Análogamente, contrastar la hipótesis de normalidad de los valores de expresión del gen CCND3 Cyclin D3 para pacientes ALL.*

```
shapiro.test(leukemia$x.1040[leukemia$Y==0])
Shapiro-Wilk normality test
data: leukemia$x.1040[leukemia$Y ==0]
W = 0.9552, p-value = 0.06937
```

Dado que el p-valor está por encima de $\alpha = 0{,}05$, no rechazamos la hipótesis nula.

Ejemplo 3.20 *Contrastar la normalidad de la variable Peso.kg para no corredores,* $n = 35$.

```
shapiro.test(ElPulso$Peso.kg[ElPulso$Correr == "No"])

        Shapiro-Wilk normality test
data: ElPulso$Peso.kg[ElPulso$Correr == "No"]
W = 0.97425, p-value = 0.5699
```

Para un nivel de significación $\alpha = 0,05$ prefijado, como el $p - value = 0{,}5699$ no rechazamos que *Peso.kg* sea normal.

Ejemplo 3.21 *Para hacer vosotros, desde el fichero de datos "calorie", 20 observaciones y 4 variables, realizar un contraste de la normalidad de la variable wgt usando el test de Shapiro-Wilk, puesto que* $n = 20$.

Test de Kolmogorov-Smirnov Contrastar la hipótesis de normalidad a través del test de Kolmogorov-Smirnov requiere que la media y varianza poblacionales sean conocidas.

```
 mean(leukemia$x.3395[leukemia$Y==0]);sd(leukemia$x.3395[leukemia$Y==0])
[1] 7.579787e-05
[1] 0.4582803
 mean(leukemia$x.1040[leukemia$Y==0]);sd(leukemia$x.1040[leukemia$Y==0])
[1] 1.891529
[1] 0.31073
 ks.test(leukemia$x.3395[leukemia$Y==0],pnorm,7.579787e-05, 0.4582803)
 One-sample Kolmogorov-Smirnov test
data: leukemia$x.3395[leukemia$Y == 0]
D = 0.093, p-value = 0.7766
alternative hypothesis: two-sided
 ks.test(leukemia$x.140[leukemia$Y==0],1.891529, 0.31073)
 Two-sample Kolmogorov-Smirnov test
data: leukemia$x.1040[leukemia$Y ==0]
D = 0.5106, p-value = 1
alternative hypothesis: two-sided
```

Ejemplo 3.22 *Vamos a realizar el contraste de normalidad de la variable Peso.kg para corredores del fichero de datos "ElPulso" utilizando el test de Kolmogorov-Smirnov.dado que* $n = 57$.

```
#Estimaciones de la media y desviación típica poblaciones
ElPulso$Peso.kg[ElPulso$Correr=="Si"])
[1] 64.06975
sd(ElPulso$Peso.kg[ElPulso$Correr=="Si"])
[1] 10.75766
```

I.e., $\overline{x} = 64{,}06975$ y $s = 10{,}75766$.

`#Contraste de las diferencias entre la función de distr. empírica y la distribución teórica` $N(\mu = 64{,}06975; \sigma = 10{,}75766)$.

```
 ks.test(ElPulso$Peso.kg[ElPulso$Correr=="Si"]),pnorm,64.06975,10.75766)

     One-sample Kolmogorov-Smirnov test
data: ElPulso$Peso.kg[ElPulso$Correr=="Si"]
D = 0.12116, p-value = 0.3727
alternative hypothesis: two-sided
```

Fijado un nivel de significación $\alpha = 0{,}05$, como el p-valor es menor que el nivel de significación no rechazamos la hipótesis nula de que la distribución de los pesos en kg para los corredores siga una distribución normal.

Ejemplo 3.23 *Contraste de la normalidad de una v.a.generada por simulación.*

```
x<-exp(rnorm(25))
ks.test(x, "pnorm", mean = mean(x), sd = sd(x))

    One-sample Kolmogorov-Smirnov test
    data: x
    D = 0.383, p-value = 0.0008265
    alternative hypothesis: two-sided
```

Por lo tanto, si $\alpha = 0,05$, como el p-valor es menor que el nivel de significación rechazamos la hipótesis nula de que la distribución es normal.

Test de Lilliefors A diferencia de la función *ks.test*, *lillie.test* no requiere que los parámetros de las distribución sean conocidos. El test de Lilliefors se lleva a cabo fácilmente una vez instalado el paquete *nortest*.

```
install.packages("nortest")
library(nortest)
lillie.test(x)
```

Ejemplo 3.24 *Ejecutando el test sobre los valores de expresión del gen Gdf5 y del gen CCND3 Cyclin D3, respectivamente, los resultados son los siguientes.*

```
library(nortest)
lillie.test(leukemia$x.3395[leukemia$Y==0])
 Lilliefors (Kolmogorov-Smirnov) normality test
data: leukemia$x.3395[leukemia$Y == 0]
D = 0.093, p-value = 0.3919
 lillie.test(leukemia$x.1040[leukemia$Y==0])
 Lilliefors (Kolmogorov-Smirnov) normality test
data: leukemia$x.1040[leukemia$Y == 0]
D = 0.0725, p-value = 0.7757
```

La conclusión es la misma que para el test de Shapiro-Wilk.

Por último, los correspondientes gráficos q-q se pueden generar con la siguiente secuencia de comandos.

```
op <- par(mfrow=c(1,2))
qqnorm(leukemia$x.3395[leukemia$Y==0]);qqline(leukemia$x.3395[leukemia$Y==0],col=2)
qqnorm(leukemia$x.1040[leukemia$Y==0]);qqline(leukemia$x.1040[leukemia$Y==0],col=2)
par(op)
```

Contraste Chi-cuadrado de Pearson .

La distancia χ^2 entre la distribución de frecuencias observada en la muestra y la distribución de probabilidad especificada por la hipótesis nula se define viene dada por:

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i} = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

donde $n_1, n_2, ..., n_k$ son las frecuencias absolutas de los k posibles resultados y $p_1, p_2, ..., p_k$ son las probabilidades de dichos resultados si es cierta la hipótesis nula.

Además, el test χ^2 de Pearson se utiliza para probar la independencia de dos variables entre sí, mediante la presentación de los datos en tablas de contingencia.Trata de encontrar relación o asociación entre dos variables de carácter cualitativo que se presentan unicamente según dos modalidades (dicotómicas o binarias). Cuanto mayor sea de 2 el valor de χ^2, menos verosímil es que la hipótesis sea correcta. De la misma forma, cuanto más se aproxima a cero el valor de chi-cuadrado, más ajustadas están ambas distribuciones.

```
#Tabla de contingencia de 2 variables cuenta nº de coincidencias según las categorías.
 x<- matrix(c(12, 5, 7, 7), ncol = 2)
 x

     [,1] [,2]
     [1,] 12 7
     [2,] 5 7

#Contraste de independencia o no de las variables.
 chisq.test(x)

     Pearson's Chi-squared test with Yates' continuity correction
     data: x
     X-squared = 0.6411, df = 1, p-value = 0.4233
```

Dado p-value=0.4233, entonces no rechazamos la hipótesis nula si el nivel de significación es 0.05.

```
 chisq.test(x)$p.value

     [1] 0.4233054
```

Supongamos que tenemos una muestra de personas en las que hemos observado dos variables: el color de ojos, que puede ser "claro" y "oscuro", y el color de pelo, que puede ser "rubio" ó "moreno". Entonces hacemos una tabla de contingencia para cruzar dichas características, y aplicamos el test para ver si el "color de ojos" y el "color de pelo" son variables independientes:

```
color.ojos = c("claro","oscuro","oscuro","oscuro","claro","claro","claro","oscuro",
+ "oscuro","claro","claro","claro","claro","oscuro","oscuro")
color.pelo = c("rubio","moreno","moreno","rubio","moreno","rubio","rubio","moreno",
+ "rubio","moreno","rubio","rubio","rubio","moreno","moreno")
table(color.ojos,color.pelo)

      color.pelo
     color.ojos moreno rubio
      claro       2     6
      oscuro      5     2

chisq.test(table(color.ojos,color.pelo))
```

```
        Pearson's Chi-squared test with Yates' continuity correction
data: table(color.ojos, color.pelo)
X-squared = 1.637, df = 1, p-value = 0.2007
Mensajes de aviso perdidos
In chisq.test(table(color.ojos, color.pelo)) :
 Chi-squared approximation may be incorrect
```

En este caso, el p-valor es 0.2007, superior al nivel de significación. Por tanto, no rechazaremos la hipótesis de que las variables color de ojos y color de pelo son independientes. El mensaje de aviso que aparece en la salida de resultados se debe a que el número de casos utilizado es muy pequeño.

Test de Wilcoxon para una muestra En ausencia de normalidad e insuficiencia de datos en la muestra, el test de Wilcoxon para una muestra se presenta como una alternativa robusta al test t para una muestra. Concretamente, se trata de un contraste de centralidad de una población de distribución simétrica. El contraste:

$$H_0 : Me = M_0$$
$$H_1 : Me \neq M_0$$

```
wilcox.test(x,mu=5)
 wilcox.test(leukemia$x.3395[leukemia$Y==0])
 Wilcoxon signed rank test
data: leukemia$x.3395[leukemia$Y == 0]
V = 550, p-value = 0.8875
alternative hypothesis: true location is not equal to 0
```

Como el p-valor está por encima de 0.05, no rechazamos la hipótesis nula.

Test U de Mann-Witney para dos muestras independientes Esta prueba no paramétrica es la alternativa de test t de Student para dos muestras independientes, comparando si dos poblaciones tienen la misma mediana.

$$H_0 : Me_1 = Me_2$$
$$H_1 : Me_1 \neq Me_2$$

```
 # Test U de Mann-Witney para dos muestras independientes.
 wilcox.test(y~A) # donde y es numérico & A es un factor binario.
 # Test U de Mann-Witney para dos muestras independientes.
 wilcox.test(y,x) # donde y & x son numéricos.

x<-c(0.80,0.83,1.89,1.04,1.45,1.38,1.91,1.64,0.73,1.46)
y<-c(1.15,0.88,0.90,0.74,1.21)
wilcox.test(x,y)#opcional,alternative="greater"
wilcox.test(leukemia$x.1040~leukemia$Y)
```

```
 Wilcoxon rank sum test
data: leukemia$x.1040 by leukemia$Y
W = 840, p-value = 0.002477
alternative hypothesis: true location shift is not equal to 0
```

A la vista del p-valor, nuestra conclusión es que rechazamos H_0.

Test de Wilcoxon para 2 muestras apareadas El contrate que se realiza en esta opción es el alternativo al contraste de hipótesis de medias para muestras relacionadas y viene expresado del siguiente modo:

$$H_0 : Me_D = 0$$
$$H_1 : Me_D \neq 0$$

```
# dependent 2-group Wilcoxon Signed Rank Test
wilcox.test(y1,y2,paired=TRUE) # where y1 and y2 are numeric
```

```
> wilcox.test(x,y,paired=TRUE,alternative="greater")
```

For the wilcox.test you can use the alternative="less" or alternative="greater" option to specify a one tailed test.

Más pruebas no paramétricas Este procedimiento sería la alternativa no paramétrica a un análisis de la varianza con un factor.

```
#Prueba no paramétrica para k muestras independientes.
# Kruskal Wallis Test One Way Anova by Ranks
 kruskal.test(y~A) #donde y1 es una variable numérica y A es un factor
```

Este procedimiento sería la alternativa no paramétrica a un análisis de la varianza con dos factores.

```
# Randomized Block Design - Friedman Test
 friedman.test(y~A|B) # donde y es una variable numérica, A es un factor de agrupamiento
# y B es un factor de bloques
```

Ejercicios Propuestos

Ejercicio 3.1 *Obtén una muestra de tamaño 5 del un vector 1:20, con probabilidades proporcionales al valor del vector.*

Ayuda: x<- sample(1:20,5,prob=1:20)

Ejercicio 3.2 *Genera un vector que represente la sucesión de tiradas; suponiendo n = 500 tiradas.*

Solución

```
dadoplot<- data.frame(caras=sample(c(0, 1), 500,replace=TRUE))
dadoplot$FA<- with(dadoplot, cumsum(caras))
plot (FR, type='l')
abline (0.5, 0, col='red')
```

Ejercicio 3.3 *Simular 120 lanzamientos de un dado en cuyo interior se han introducido asimétricamente bolas de acero, de forma que P(1) = 0,5; P(2) = 0,25; P(3) = 0,15; P(4) = 0,04 y P(5) = P(6) = 0,03. Almacenar los resultados de los lanzamientos en la variable dado7.*

sample(1:6,120,replace=TRUE,c(0.5,0.25,0.15,0.04,0.03,0.03))

Ejercicio 3.4 *Considerando el fichero de datos ElPulso, se pide:*

a) Calcular el intervalo de confianza para el peso medio de todos los individuos con $\alpha = 0.05$.

b) Calcular el intervalo de confianza para el peso medio de las mujeres con $\alpha = 0.05$.

c) Estudios recientes afirman que la altura media de las mujeres de esta población es $\mu = 167$ cm. A la vista de estos datos, ¿podemos aceptar dicha hipótesis?

d) Calcular el intervalo de confianza para el pulso1 medio de las mujeres que no fuman.

e) Calcular el intervalo de confianza para la media del incremento del pulso ($Pulso2 - Pulso1$) para los individuos que corrieron.

f) Calcular el estimador puntual de la proporción p de individuos que fuman.

g) Calcular el intervalo de confianza para la proporción p_F de individuos que fuman con $\alpha = 0.05$.

h) Calcular el intervalo de confianza para la proporción $p_{F|M}$ de mujeres fumadoras con $\alpha = 0.05$.

i) Calcular el intervalo de confianza para la proporción p de individuos con altura superior a 180 y peso superior a 85 kg con $\alpha = 0.05$.

j) Determinar si hay diferencia significativa entre la proporción de hombres y mujeres que fuman con un nivel de significación $\alpha = 0.05$.

k) Calcular un intervalo de confianza para la diferencia de medias del pulso1 y del pulso2 para la población de los que corrieron.

l) Calcular un intervalo de confianza para la diferencia de medias del pulso1 entre hombres que fuman y no fuman.

m) Contrastar si hay diferencia significativa en el incremento del pulso (increpulso) para hombres y mujeres que se sometieron a la prueba de correr.

n) Contrastar si la variable aleatoria peso de los hombres $Peso_{\mathrm{H}}$ se ajusta a una distribución normal.

o) Contrastar si la variable aleatoria peso de las mujeres $Peso_{\mathrm{M}}$ se ajusta a una distribución normal.

p) Contrastar si la variable aleatoria altura de los hombres $Altura_{\mathrm{H}}$ se ajusta a una distribución normal.

q) Contrastar si la variable aleatoria altura de las mujeres $Altura_{\mathrm{M}}$ se ajusta a una distribución normal.

Ejercicio 3.5 *Un estudio sobre un nuevo fármaco tenemos dos grupos de pacientes, 400 tratados y 650 controles. De estos, 250 de los tratados presentan mejoría tras cierto tiempo de tratamiento, mientras que sólo 150 de los controles presentan mejoría. Se desea contrastar si la proporción de mejora en los pacientes tratados es igual a la de los controles.*

Ayuda: prop.test(c(250,150), c(400,650)).

Ejercicio 3.6 *Normalmente las hojas de la Mimosa púdica son horizontales. Si se toca ligeramente una de ellas, las hojas se pliegan. Se afirma que el tiempo medio desde el contacto hasta el cierre completo es de 2.5 segundos.*

En un experimento para comprobar dicha hipótesis se han obtenido las siguientes observaciones: 3.0, 2.9, 2.8, 2.7, 2.6, 2.4, 2.5, 2.4, 2.6 y 2.7:

a) Contrastar la normalidad de estos datos.

b) Plantear y resolver un contraste de hipótesis adecuado para comprobar si el valor 2.5 es en verdad el tiempo medio hasta el cierre o si por el contrario es diferente.

c) Extraer las conclusiones.

Ayuda: ks.test(mimosa); t.test(mimosa, mu=2.5).

Ejercicio 3.7 *Se está poniendo a prueba un proceso que en fotobiología se denomina abscisión, con la esperanza de aumentar la cosecha de fruta en los naranjos. El proceso implica exponer los árboles a luz coloreada durante quince minutos cada noche. Se recolectó fruta de 10 árboles experimentales, bajo condiciones normales primero y luego después del nuevo tratamiento. Se obtuvieron las siguientes observaciones:*

Conds. normales	*100*	*50*	*98*	*26*	*65*	*95*	*86*	*100*	*108*	*60*	*175*
Nuevo tratamiento	*129*	*60*	*141*	*56*	*150*	*100*	*102*	*126*	*111*	*59*	*179*

El promotor del nuevo proceso pretende comprobar que este incrementa la recolección. Realizar el análisis estadístico adecuado para averiguar si estos datos respaldan la pretensión del promotor.

Ayuda: tenemos un problema de comparación de la media de dos muestras emparejadas unilateral (porque nos piden que un grupo tenga la media mayor que el otro).

nor<-c(100,50,98,26,65,95,86,100,108,60,175)

trat<-c(129,60,141,56,150,100,102,126,111,59,179)

#Como son muestras emparejadas comprobamos si la variable diferencia es normal:

dif<-nor-trat

shapiro.test(dif); ks.test(dif,"pnorm",mean(dif),sd(dif))

#Estudio más detallado se puede observar los gráficos qq:

qqnorm(dif); qqline(dif)

#Utilizamos pues una prueba no paramétrica ya que tenemos pocos datos:

wilcox.test(nor, trat, paired = TRUE, alternative = "less")

Como el p-valor 0.0009766 es menor que 0.05 se rechaza la hipótesis de que la mediana de la recolección en condiciones normales es mayor que con el nuevo tratamiento y por lo tanto pensamos que el nuevo método incrementa la recolección. Si pensáramos que si que hay normalidad entonces haríamos un test t:

t.test(nor,trat,paired=T,alternative="less")

En ese caso el p-valor 0.006411 también es menor que 0.05 y la conclusión es similar salvo que podemos decir que el incremento medio es mayor en el nuevo método.

Ejercicio 3.8 *Las manadas de lobos son territoriales, con territorios de al menos 130 km2. Se piensa que los aullidos de los lobos, que comunican información tanto de la situación como de la composición de la manada, están relacionados con la territorialidad. Se obtuvieron los siguientes valores de la variable X, duración en minutos de*

una sesión de aullidos de una determinada manada sometida a estudio: 1.0, 1.8, 1.6, 1.5, 2.0, 1.8, 1.2, 1.9, 1.7, 1.6, 1.6, 1.7, 1.5, 1.4, 1.4 y 1.4. ¿Confirmarían estos datos que la duración media es superior a 1.5 minutos?

Solución: tenemos un problema de una muestra en el que queremos analizar si la media es superior a un valor. Contraste unilateral. Tras comprobar que los datos son normales hacemos el test t habitual.

x<-c(1.0,1.8,1.6,1.5,2.0,1.8,1.2,1.9,1.7,1.6,1.6,1.7,1.5,1.4,1.4,1.4)

shapiro.test(x); ks.test(x,"pnorm",mean(x),sd(x))

t.test(x, mu=1.5, alternative = "greater")

Como el p-valor 0.15 es mayor que 0.05, no tenemos evidencia estadística suficiente para rechazar la hipótesis nula.

Ejercicio 3.9 *La resistencia a la rotura de un componente eléctrico constituye una característica importante de un cierto proceso. Un fabricante utiliza un material nuevo de fabricación frente al material clásico. Se recoge una muestra de 10 elementos usando el primer componente y otra de 10 elementos usando el segundo componente. Se pueden considerar a los dos procesos como dos tratamientos o dos niveles diferentes de un factor dado.*

CNuevo	*16,85*	*16.40*	*13.21*	*16.35*	*16.52*	*17.04*	*16.96*	*17.15*	*16.59*	*16.57*
CAntiguo	*17.50*	*17.63*	*18.25*	*18.00*	*17.86*	*17.75*	*18.22*	*17.90*	*17.96*	*18.15*

Se pretende averiguar si existen diferencias significativas entre ambos tratamientos a nivel de resistencia.

Ayuda

CNuevo<- c(16.85, 16.40, 13.21, 16.35, 16.52, 17.04, 16.96, 17.15,16.59, 16.57)

CAntiguo<- c(17.50, 17.63, 18.25, 18.00, 17.86, 17.75, 18.22,17.90, 17.96, 18.15)

Se comprueba la igualdad entre las varianzas de ambas muestras:

var.test(CNuevo, CAntiguo)

F test to compare two variances

data: CNuevo and CAntiguo

F = 21.2113, num df = 9, denom df = 9, p-value = 0.0001013

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

5.268586 85.396550

sample estimates:

ratio of variances

21.21130

t.test(CNuevo, CAntiguo, paired=F, var.equal=F) #Al ser distintas para cada grupo se toma la opción correspondiente del comando siguiente.

Welch Two Sample t-test

data: CNuevo and CAntiguo

t = -4.2167, df = 9.847, p-value = 0.001843

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-2.3829934 -0.7330066

sample estimates:

mean of x mean of y

16.364 17.922

4. Referencias bibliográficas y recursos de Internet

(1) Ayala, G. (2019). *Bioinformática Estadística. Análisis estadístico de datos ómicos.* Universidad de Valencia. Accesible: https://www.uv.es/ayala/docencia/tami/tami13.pdf.

(2) Becker, R.A., Chambers, J.M., and Wilks, A.R. (1988), The New Language, Pacific Grove, CA: Wadsworth.

(3) Everitt, B.S., Hothorn, T. (2010). A Handbook of Statistical Analysis Using R. Chapman Hall.

(4) Fernández, F.R. y Mayor, J.A. (1995). Muestreo en poblaciones finitas : curso básico. EUB, Barcelona.

(5) García-Pérez, A. (2008). Estadística aplicada con R. UNED Varia.

(6) Golub et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science, Vol. 286:531-537. Accesible en http://www-genome.wi.mit.edu/MPR/.

(7) Holmes, S. and Huber, W. (2018). Modern Statistics for Modern Biology. Cambridge University Press. Accesible en http://web.stanford.edu/class/bios221/book/

(8) González-Ortiz, F.J. (2007). Prácticas de Estadística con R (Parte I y Parte II). Universidad de Cantabria.

(9) Moore, D.S. (2005). Estadística aplicada básica, 2ª ed. Antoni Bosch editor.

(10) R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

(11) Ross, S.M. (2000). Introduction to probability and statistics for engineers and scientists. Academic Press.

(12) Steel, G.L., and Sussman, G.J. (1975). Scheme: An Interpreter for the Extended Lambda Calculus, Memo 349, MIT Artificial Intelligence Laboratory.

(13) Xia, X. (2018). Nucleotide substitution models and evolutionary distances. In Bioinformatics and the Cell (pp. 269-314). Springer, Cham.

- An Introduction to R: http://cran.r-project.org/doc/manuals/R-intro.pdf
- Comunidad R hispano: http://www.r-es.org/
- Curso introducción R: http://www.uv.es/conesa/CursoR/cursoR.html
- icebreakeR: http://www.ms.unimelb.edu.au/~andrewpr/r-users/icebreakeR.pdf
- Introduction to Data Technologies: http://www.stat.auckland.ac.nz/~paul/ItDT/itdt-2010-11-01.pdf