

Introducción al Análisis de Datos

Matías Alfonso

2019-10-30

Índice general

I	Programación en R	5
1	Preliminares	7
1.1	¿Por qué R?	7
1.2	Software Libre	8
1.3	Sistema de Paquetes	8
2	Primeros pasos	9
2.1	Asignación de datos y evaluación.	9
2.2	Working directory	10
2.3	Comentarios	10
2.4	Objetos básicos en R	10
2.5	Factores	11
2.6	Cómo buscar ayuda	12
3	Estructuras de datos	13
3.1	Vectores	13
3.2	Listas	13
3.3	Matrices	14
3.4	Data frames	14
3.5	Valores faltantes	15
4	Obteniendo datos	17
4.1	Datos tabulares	17
5	Operaciones básicas	19
5.1	Subsetting. Selección de elementos.	19
5.2	Operaciones vectorizadas	22
6	Estructuras de control	23
6.1	If / else	23
6.2	while	24
6.3	for	25
7	Funciones	27

II	Introducción al análisis estadístico	29
8	Introducción	31
8.1	Datos primarios y secundarios	31
8.2	Resumen de la información	31
9	Bases de datos	33
9.1	EnPreCoSP	33
9.2	Memoria	33
9.3	Titanic	33
9.4	Base prácticos	34
9.5	CEPRAM	34
10	Distribuciones de frecuencia	35
10.1	Definiciones	35
10.2	Aplicaciones	35
10.3	Actividades	37
11	Gráficos de frecuencias	39
11.1	Histogramas	39
11.2	Gráfico de barras	42
11.3	Gráfico de tortas	43
11.4	Ojiva de Galton	45
12	Medidas de resumen	47
12.1	Medidas de posición	47
12.2	Medidas de dispersión	51
12.3	Medidas de forma	56
12.4	Estandarización	58
13	Gráficos de Resumen	61
13.1	Gráficos de barras	61
13.2	Gráficos de caja	62
14	Relaciones entre variables	65
14.1	Tablas de contingencia	65
14.2	Riesgo relativo	67
14.3	Odds ratio	68
14.4	Intensidad de la asociación	68
14.5	Coficiente Phi de Pearson-Yule (φ)	70

Parte I

Programación en R

Capítulo 1

Preliminares

R es un lenguaje de programación desarrollado inicialmente por Ross Ihaka y Robert Gentleman en el departamento de Estadística de la Universidad de Auckland en 1993. Está orientado específicamente con un enfoque al análisis estadístico.

R se desarrolla a partir de un lenguaje denominado S, desarrollado por John Chambers en 1976, disponible a partir del software comercial S-PLUS.

Es un lenguaje interactivo, permite la ejecución de instrucciones en líneas de comando en una consola.

1.1 ¿Por qué R?

R puede ser ejecutado en múltiples plataformas y en la gran mayoría de los sistemas operativos. Puede ser ejecutado en tablets, teléfonos o computadoras. La utilización de scripts permite compartir fácilmente los análisis con los colegas, así como asegurar la reproductibilidad de los resultados. Todo lo que realizamos mediante una interfaz gráfica con el mouse no deja registros de nuestro trabajo e impide que podamos repasar nuestro trabajo para corregir errores.

La versatilidad y la potencia que otorga un lenguaje de programación es mucho mayor que la que podemos obtener con softwares estadísticos de interfaz gráfica. La comunidad de usuarios y desarrolladores de R está en constante crecimiento en los últimos años. Hay una enorme cantidad de gente realizando nuevos desarrollos en R cada día, que están a la vanguardia de la ciencia computacional y estadística.

1.2 Software Libre

La mayor ventaja que tiene R con respecto a otros softwares de análisis estadístico es que es un software libre. ¿Qué quiere decir eso? Por un lado, que es gratuito. Por otro, que el código fuente con el que R fue desarrollado está abierto, se puede descargar y está disponible online. Actualmente el copyright de R lo posee la R Foundation. R forma parte del sistema GNU, desarrollado por la Free Software Foundation. De acuerdo a la Free Software Foundation, con el software libre se garantizan cuatro libertades fundamentales:

- La libertad de ejecutar el programa para cualquier propósito. (Libertad 0)
- La libertad de estudiar cómo el programa funciona y adaptarlo a tus propias necesidades. (Libertad 1)
- La libertad de redistribuir copias de manera que puedas ayudar a alguien. (Libertad 2)
- La libertad de mejorar el programa, y liberar tus mejoras al público, de manera que se beneficie toda la comunidad. (Libertad 3)

1.3 Sistema de Paquetes

El sistema de funcionalidades de R se encuentra agrupado en paquetes. La mayor parte de los paquetes se encuentran disponibles en Comprehensive R Archive Network (CRAN). Hay un conjunto de paquetes principales, de base, que incluye todos los paquetes que se instalan por defecto cuando instalamos R. Luego, tenemos un montón de paquetes con funcionalidades específicas que podemos instalar en función de nuestras necesidades.

Capítulo 2

Primeros pasos

2.1 Asignación de datos y evaluación.

R es un *lenguaje interpretado*. Esto quiere decir que le podemos ir pasando instrucciones y el programama las irá interpretando. Cuando ejecutamos el programa, nos encontramos con el prompt a la espera de intrucciones:

>

Una de las operaciones más sencillas que podemos realizar es la asignación de valores a las variables. El operador de asignación es `<-`.

```
x <- 1
print(x)
```

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

```
x
```

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

```
texto <- "hola mundo"
texto
```

```
## [1] "hola mundo"
```

Podemos imprimir el valor de una variable con la función `print()` o directamente escribiendo la variable.

Tenemos dos formas de interactuar con R:

- Tipear directamente los comandos en el prompt y ejecutarlos.
- Escribir un archivo de texto con todas las intrucciones y luego ejecutarlo. Este archivo se denomina script.

2.2 Working directory

Lo primero que debemos hacer cuando comenzamos a trabajar en R es configurar el directorio de trabajo. Una buena costumbre es crear un directorio nuevo de trabajo cuando comenzamos un proyecto nuevo. Luego configuramos esa carpeta como directorio de trabajo. Colocamos allí todos los archivos vinculados a ese proyecto. Para determinar en qué directorio estamos parados, podemos utilizar el comando `getwd()`. Para configurar el directorio de trabajo, utilizamos

```
setwd(#RUTA-A-DIRECTORIO)
```

2.3 Comentarios

Todo lo que escribamos luego de un `#` en una instrucción, no será evaluado.

```
x <- c(3, 4, 5)
## Esto no se ejecuta

x
```

```
## [1] 3 4 5
```

Ello nos permite comentar el código que escribimos, a manera de documentación.

2.4 Objetos básicos en R

Casi todo lo que encontremos en R, se denominan *objetos*. Hay 5 tipos de objetos básicos o atómicos:

- lógico
- numérico
- entero
- complejo
- caracter

Veamos algunos ejemplos:

```
## Logico
TRUE
```

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

```
FALSE
```

```
## [1] FALSE
```

```
## Numérico
c(1.509, 2.859)
```

```
## [1] 1.509 2.859
```

```
## Enteros
1:10
```

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

```
## Caracter
"casa"
```

```
## [1] "casa"
```

Existen muchos más clases de objetos en R. Para averiguar de que tipo es un objeto, podemos utilizar la función `class()`

```
x <- 1:10
class(x)
```

```
## [1] "integer"
```

```
class("TRUE")
```

```
## [1] "character"
```

Para preguntar por la clase de un objeto, podemos utilizar el comando `class()`

```
x <- 1:10
class(x)
```

```
## [1] "integer"
```

```
y <- "casa"
class(y)
```

```
## [1] "character"
```

2.5 Factores

Los factores son básicamente objetos de clase entero, pero con etiquetas. Son datos categóricos y pueden estar ordenados o no.

```
f <- factor(c("si", "si", "no", "si"))
f
```

```
## [1] si si no si
## Levels: no si
```

```
f <- factor(c("bajo", "bajo", "medio", "alto"),
           levels = c("bajo", "medio", "alto"),
```

```
ordered = TRUE)
f

## [1] bajo bajo medio alto
## Levels: bajo < medio < alto
```

2.6 Cómo buscar ayuda

R tiene un sistema de ayuda integrado. Si queremos saber para qué sirve un comando determinado o como pasarle los argumentos, podemos utilizar `?` o `help()`. Supongamos que queremos saber cómo se utiliza la función `sum()`

```
help(sum)

?vector
```

Capítulo 3

Estructuras de datos

3.1 Vectores

La forma más elemental de almacenar datos en R es en un vector. Un **vector** es una concatenación de objetos del mismo tipo. Podemos utilizar la función `c()` para crear vectores.

```
x <- c(1, 2, 3, 2.5)
x <- c(TRUE, FALSE)
x <- c(T, F)
x <- c("casa", "árbol", "patio")

## También podemos utilizar la función vector.
x <- vector(mode = "numeric", length = 10)
```

Si concatenamos elementos de diferente clase, R realizará una coerción automática de la clase de los objetos.

```
x <- c("casa", 2) ## character
x <- c(TRUE, 2) ## numeric

class(x)

## [1] "numeric"
```

3.2 Listas

Las listas también son una concatenación de elementos, pero pueden contener elementos de diferente clase. Para crear una lista, podemos utilizar `list()`

```
x <- list("peso", 2, "altura", 3, TRUE)
x

## [[1]]
## [1] "peso"
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] "altura"
##
## [[4]]
## [1] 3
##
## [[5]]
## [1] TRUE
```

3.3 Matrices

Las matrices son vectores, pero con un atributo de dimensión. La dimensión en sí es un vector de enteros de largo 2.

```
m <- matrix(1:9, nrow = 3)
m

##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
dim(m)

## [1] 3 3
```

Al igual que los vectores, contienen objetos de la misma clase. Las matrices tienen algunas propiedades matemáticas interesantes, pues se pueden realizar operaciones especiales con ellas, por ejemplo, se pueden sumar o multiplicar.

3.4 Data frames

Los data frames son datos tabulados. Son tablas, donde cada columna puede ser de una clase diferente. Es un objeto particularmente útil para el análisis estadístico.

```
data.frame(Id = c(1, 2, 3),  
           Nombre = c("Juan", "Carlos", "Ramona"),  
           Altura = c(1.76, 1.80, 1.65))
```

```
##   Id Nombre Altura  
## 1  1   Juan   1.76  
## 2  2 Carlos   1.80  
## 3  3 Ramona   1.65
```

3.5 Valores faltantes

Existen dos tipos de valores faltantes en R:

- NA
- NaN

Capítulo 4

Obteniendo datos

Existen una enorme cantidad de funciones para abrir archivos de diversos tipos.

4.1 Datos tabulares

Un formato estándar y abierto para guardar información en forma de tablas son archivos separados por comas (.csv) Para leer estos datos podemos utilizar:

- `read.tables()`
- `read.csv()`

Podemos descargar la base de datos del Titanic del siguiente link [titanic.csv](#). Descargamos y colocamos el archivo en una carpeta **data**, dentro del directorio de trabajo:

```
## Leemos los datos desde un archivo y los guardamos en la variable base
base <- read.csv("data/titanic.csv")
```

```
## Imprimimos las primeras 5 filas de la base
head(base)[, 1:3]
```

```
##   PassengerId Survived Pclass
## 1           1         0       3
## 2           2         1       1
## 3           3         1       3
## 4           4         1       1
## 5           5         0       3
## 6           6         0       3
```

También podemos leer datos directamente de la web

```
## Datos Abiertos
## Cantidad de consultas médicas y odontológicas en centros de Salud del Primer nivel :
consultas <- read.csv("http://datos.salud.gob.ar/dataset/5fcacd04-58eb-4b43-89a0-55231

head(consultas)

##   provincia_id      provincia_desc      año consultas_cantidad
## 1             2 Ciudad Autonoma de Buenos Aires año_2003      559785
## 2             6      Buenos Aires año_2003      9544395
## 3            10      Catamarca año_2003      372199
## 4            14      Cordoba año_2003      3491961
## 5            18      Corrientes año_2003      764887
## 6            22      Chaco año_2003      1476825

## Recetas de medicamentos escenciales 2003-2019
recetas <- read.csv("http://datos.salud.gob.ar/dataset/dff3bf69-3514-42a3-a2aa-0414958
```

4.1.1 Excel

Podemos leer archivos Excel importando la librería `readxl`

```
library(readxl)

## Excel
memoria <- read_xls("./data/EXP1.xls")

head(memoria)[1:3]

## # A tibble: 6 x 3
##   Sujeto  NPD1  NPD2
##   <dbl> <dbl> <dbl>
## 1     1    17    16
## 2     4     9    10
## 3     7    11    15
## 4    10    13    14
## 5    13    11    17
## 6    16    12    16
```

Capítulo 5

Operaciones básicas

5.1 Subsetting. Selección de elementos.

Podemos seleccionar elementos o subconjuntos específicos de un objeto de R.
Hay diferentes operadores: [, [[, \$

5.1.1 Vectores

```
## Creamos un vector con las primeras 10 letras del abecedario.  
letras <- letters[1:10]  
letras
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
```

```
## Por posición  
letras[2]
```

```
## [1] "b"
```

```
letras[2:3]
```

```
## [1] "b" "c"
```

Si los elementos están etiquetados, podemos recuperar los elementos del vector por sus nombres.

```
peso <- c(Juan = 70, Pedro = 85, Ramona = 65)  
peso["Juan"]
```

```
## Juan  
## 70
```

```
peso[c("Juan", "Ramona")]
```

```
##   Juan Ramona
##    70      65
```

También podemos utilizar un vector lógico para recuperar elementos de un vector.

```
condicion <- peso > 68
condicion
```

```
##   Juan  Pedro Ramona
##  TRUE   TRUE  FALSE
```

```
peso[condicion]
```

```
##   Juan Pedro
##    70     85
```

5.1.2 Matrices

```
matriz <- matrix(letters[1:9], nrow = 3)
```

```
## Indicamos el índice de fila y columna.
matriz[1, 2]
```

```
## [1] "d"
```

```
matriz[2:3, 1:2]
```

```
##      [,1] [,2]
## [1,] "b"  "e"
## [2,] "c"  "f"
```

5.1.3 Listas

```
lista <- list(Nombre = c("Juan", "Pedro", "Ramona"),
             Peso = c(70, 85, 65),
             Altura = c(1.70, 1.78, 1.65))
```

```
## Podemos recuperar los elementos por posición o por nombre
```

```
## Devuelve un objeto lista
lista[1]
```

```
## $Nombre
## [1] "Juan" "Pedro" "Ramona"
lista["Nombre"]

## $Nombre
## [1] "Juan" "Pedro" "Ramona"
## Devuelve la clase del objeto seleccionado
lista[[1]]

## [1] "Juan" "Pedro" "Ramona"
lista[["Nombre"]]

## [1] "Juan" "Pedro" "Ramona"
lista$Nombre

## [1] "Juan" "Pedro" "Ramona"
```

5.1.4 Data frames

```
df <- as.data.frame(lista)
df

##   Nombre Peso Altura
## 1   Juan   70   1.70
## 2  Pedro   85   1.78
## 3 Ramona   65   1.65
df[1,]

##   Nombre Peso Altura
## 1   Juan   70   1.7
df[,1]

## [1] Juan  Pedro  Ramona
## Levels: Juan Pedro Ramona
df$Nombre

## [1] Juan  Pedro  Ramona
## Levels: Juan Pedro Ramona
df[["Nombre"]]

## [1] Juan  Pedro  Ramona
## Levels: Juan Pedro Ramona
```

```
df[1:2, 1:2]
```

```
##      Nombre Peso
## 1     Juan    70
## 2     Pedro    85
```

5.2 Operaciones vectorizadas

Podemos realizar operaciones elemento a elemento en los vectores.

```
x <- 1:10
y <- 15:6
x + y
```

```
##      [1] 16 16 16 16 16 16 16 16 16 16
```

```
x * y
```

```
##      [1] 15 28 39 48 55 60 63 64 63 60
```

```
x / y
```

```
##      [1] 0.06666667 0.14285714 0.23076923 0.33333333 0.45454545 0.60000000
##      [7] 0.77777778 1.00000000 1.28571429 1.66666667
```

También podemos realizar operaciones lógicas.

```
x == y
```

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

```
x < y
```

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

```
x > y
```

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

Capítulo 6

Estructuras de control

Las estructuras de control son condiciones lógicas que nos permiten controlar el flujo de ejecución del código.

- if / else
- while
- for

6.1 If / else

El `if`. Es el **condicional**. Si la condición es verdadera, ejecuto el código. Si no, salteo esa porción de código. Si queremos que dada la condición de **Falso** el programa realice algo en particular podemos utilizar la sentencia `else`.

```
x <- 5

if(is.integer(x)){
  "x es entero"
} else {
  "x no es entero"
}

## [1] "x no es entero"

if(is.numeric(x)){
  "x es entero"
}

## [1] "x es entero"
```

6.2 while

Realiza un loop mientras la condición sea verdadera.

```
x <- 1

## Le sumo 1 a x. Repito condición hasta que x llegue a 10
while(x < 10){
  x <- x + 1
  print(x)
}
```

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

```
x <- 1

## Idem utilizando un while y un if
while(TRUE){
  if(x < 10){
    x <- x + 1
    print(x)
    next
  } else {
    break
  }
}
```

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


6.3 for

Realiza una operación para una secuencia de objetos

```
letras <- letters[1:10]
```

```
for(i in letras){  
  print(i)  
}
```

```
## [1] "a"  
## [1] "b"  
## [1] "c"  
## [1] "d"  
## [1] "e"  
## [1] "f"  
## [1] "g"  
## [1] "h"  
## [1] "i"  
## [1] "j"
```

```
for(i in 1:length(letras)){  
  print(letras[i])  
}
```

```
## [1] "a"  
## [1] "b"  
## [1] "c"  
## [1] "d"  
## [1] "e"  
## [1] "f"  
## [1] "g"  
## [1] "h"  
## [1] "i"  
## [1] "j"
```


Capítulo 7

Funciones

Hasta ahora hemos venido utilizando varias funciones. Las funciones nos permiten encapsular el código, facilitan la lectura y ahorran trabajo. Si tenemos que ejecutar el mismo código más de dos veces, probablemente convenga definir una función.

```
## Definimos una función que calcula la media
media <- function(x){
  sum(x)/length(x)
}
media(1:10)
```

```
## [1] 5.5
```

```
## Definimos una función que no devuelve ningún valor, pero imprime un
## mensaje en pantalla
holaMundo <- function(){
  print("Hola Mundo!")
}
holaMundo()
```

```
## [1] "Hola Mundo!"
```

```
## Definimos una función que toma dos caracteres y los une, separados por un -
unir <- function(a, b){
  paste(a, b, sep = '-')
}
unir("A", "B")
```

```
## [1] "A-B"
```

```
class(unir)
```

```
## [1] "function"
```


Parte II

Introducción al análisis estadístico

Capítulo 8

Introducción

La estadística sirve para resumir la información cuando trabajamos con muchos datos. Podemos hacer una división entre:

- Estadística descriptiva
- Estadística inferencial

En la **estadística descriptiva** abordaremos diferentes técnicas que nos permitan resumir un conjunto de datos. Cuando trabajemos con **estadística inferencial**, intentaremos estimar parámetros o medidas de variables de una **población** a partir de una **muestra**.

8.1 Datos primarios y secundarios

- **Datos primarios:** Es el registro primitivo de la información. Son tablas de doble entrada en el que cada **fila** representa una **unidad de observación** y cada **columna** una **variable**.
- **Datos secundarios:** Implican algún procesamiento de los datos primarios. Podemos incluir aquí las tablas de distribución de frecuencia y las tablas cruzadas o tablas de contingencia.

8.2 Resumen de la información

Resumiremos la información a través de:

- Tablas
- Gráficos

- Medidas de resumen (estadísticos).

Capítulo 9

Bases de datos

Para trabajar en las siguientes unidades haremos uso de las siguientes bases de datos:

9.1 EnPreCoSP

Base de Datos de consumo de sustancias psicoactivas. Indec Disponible en: “https://www.indec.gob.ar/ftp/cuadros/menusuperior/enprecosp/bases_enprecosp2011.rar”

Agregar un ‘|’ en la última columna de la primer file para poder leerlo correctamente. Datos corregidos:

Base corregida: enprecosp

```
enprecosp <- read.table("data/enprecosp_2011.txt", header = TRUE, sep = "|")
```

9.2 Memoria

Base de memoria. experimento

9.3 Titanic

Titanic

9.4 Base prácticos

Prácticos

9.5 CEPRAM

Base Libro de Códigos

Capítulo 10

Distribuciones de frecuencia

10.1 Definiciones

Frecuencias absolutas simples. Es la cantidad de casos que asumen determinado valor de variable.

Frecuencias relativas simples. Es la proporción de casos que asumen determinado valor.

Frecuencias absolutas acumuladas. Es la cantidad de casos que asumen determinado valor o valores inferiores a el.

Frecuencias relativas acumuladas. Es la proporción de casos que asumen determinado valor o valores inferiores a el.

10.2 Aplicaciones

Trabajemos con la ENPreCoSP. Veamos como es el estado de salud general subjetiva de la población y construyamos una tabla de distribución de frecuencias para la variable BISG01 (En general, ¿usted diría que su salud es...)

```
## Seleccionamos la nueva variable y la guardamos en una nueva variable
saludsub <- enprecosp$BISG01
```

```
## Vemos cuantos casos tenemos en total
length(saludsub)
```

```
## [1] 34343
```

```
## Convertimos a factor y etiquetamos los códigos de valores
saludsub <- factor(saludsub,
```

```

        labels = c("Excelente", "Muy buena", "Buena", "Regular", "Mala"),
        ordered = TRUE)

## Calculamos las frecuencias absolutas simples
f <- table(saludsub)

## Calculamos las frecuencias relativas simples
frel <- prop.table(f)

## Calculamos las frecuencias absolutas acumuladas
fcum <- cumsum(f)

## Calculamos las frecuencias relativas acumuladas
frelcum <- cumsum(frel)

## Juntamos todo y armamos una tabla de distribución de frecuencias.
cbind(f, fcum, frel, frelcum)

```

```

##           f  fcum      frel  frelcum
## Excelente 3919  3919 0.11411350 0.1141135
## Muy buena  8830 12749 0.25711208 0.3712256
## Buena      15410 28159 0.44870862 0.8199342
## Regular    5485  33644 0.15971231 0.9796465
## Mala        699  34343 0.02035349 1.0000000

```

Ahora analicemos las variables BISG02 (Ha sufrido algún accidente el último año) y BISG03 (Ha sufrido alguna enfermedad el último año)

```

## Seleccionamos BISG02 y la guardamos en una nueva variable
accidente <- enprecosp$BISG02
enfermedad <- enprecosp$BISG03

## Convertimos a factor y etiquetamos los códigos de valores
accidente <- factor(accidente,
                    labels = c("Sí", "No", "Ns/Nc"))
enfermedad <- factor(enfermedad,
                    labels = c("Sí", "No", "Ns/Nc"))

## Construimos las frecuencias para la variable accidente
f <- table(accidente)
frel <- prop.table(f)

## Juntamos todo y armamos una tabla de distribución de frecuencias.
cbind(f, frel)

```

```

##           f      frel
## Sí      2669 0.0777159829

```

```
## No      31657 0.9217890109
## Ns/Nc    17 0.0004950063

## Construimos las frecuencias para la variable enfermedad
f <- table(enfermedad)
frel <- prop.table(f)

## Juntamos todo y armamos una tabla de distribución de frecuencias.
cbind(f, frel)
```

```
##           f      frel
## Sí      8009 0.233206185
## No     26292 0.765570859
## Ns/Nc    42 0.001222957
```

Veamos como está conformada la edad de la muestra (BHCH05). Como es una variable continua, primero debemos construir los intervalos de clase.

```
## Seleccionamos la edad y la guardamos en una nueva variable
edad <- enprecosp$BHCH05
```

```
## Veamos los valores mínimos y máximos para la edad
range(edad)
```

```
## [1] 16 65
```

```
edad_reco <- cut(edad, breaks = c(16, 24, 34, 49, 65), include.lowest = TRUE)
```

```
## Realizamos la tabla de distribución de frecuencias igual que anteriormente
f <- table(edad_reco)
frel <- prop.table(f)
fcum <- cumsum(f)
frelcum <- cumsum(frel)
```

```
## Juntamos todo y armamos una tabla de distribución de frecuencias.
cbind(f, fcum, frel, frelcum)
```

```
##           f  fcum      frel  frelcum
## [16,24]  6592  6592 0.1919460 0.1919460
## (24,34]  8726 15318 0.2540838 0.4460298
## (34,49] 10326 25644 0.3006726 0.7467024
## (49,65]  8699 34343 0.2532976 1.0000000
```

10.3 Actividades

Capítulo 11

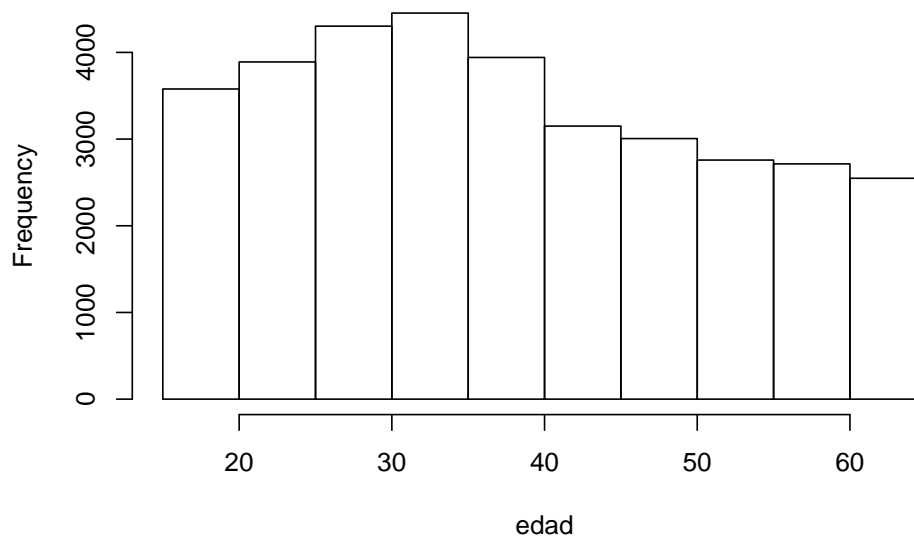
Gráficos de frecuencias

11.1 Histogramas

El histograma es una buena manera de observar la distribución de los datos en **variables continuas**. Realicemos algunos gráficos para las variables trabajadas en el Capítulo 10. Veamos como de distribuye la edad de la muestra. Para realizar un histograma, podemos utilizar la función `hist`

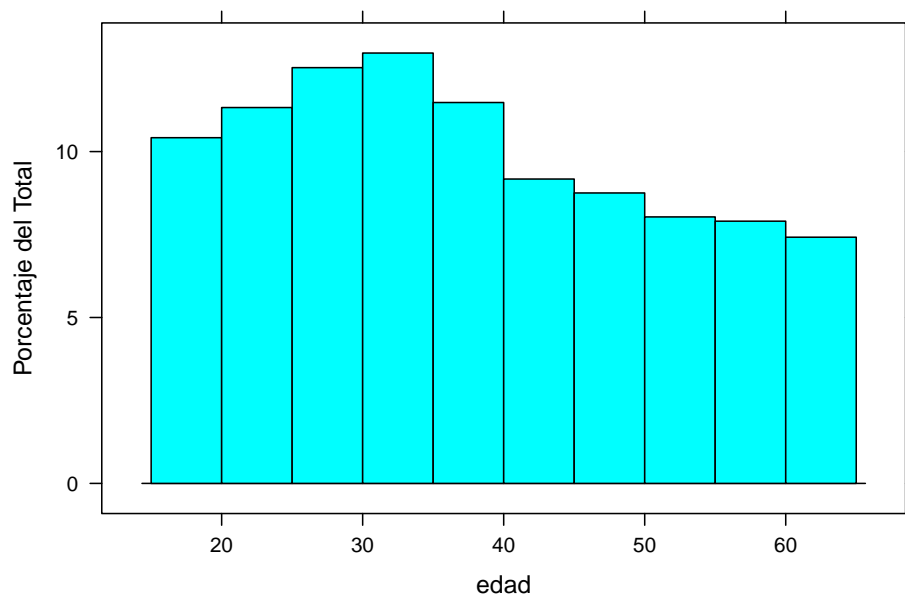
```
## Seleccionamos la edad y la guardamos en una nueva variable
edad <- enprecosp$BHCH05

## Realizamos un histograma.
hist(edad)
```

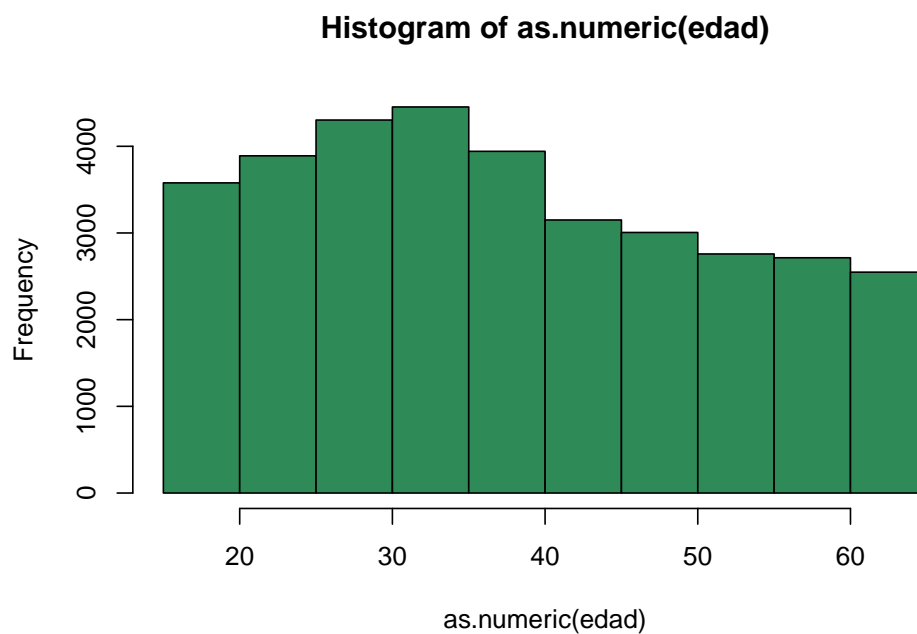
Histogram of edad

También podemos representar las frecuencias relativas o los porcentajes.

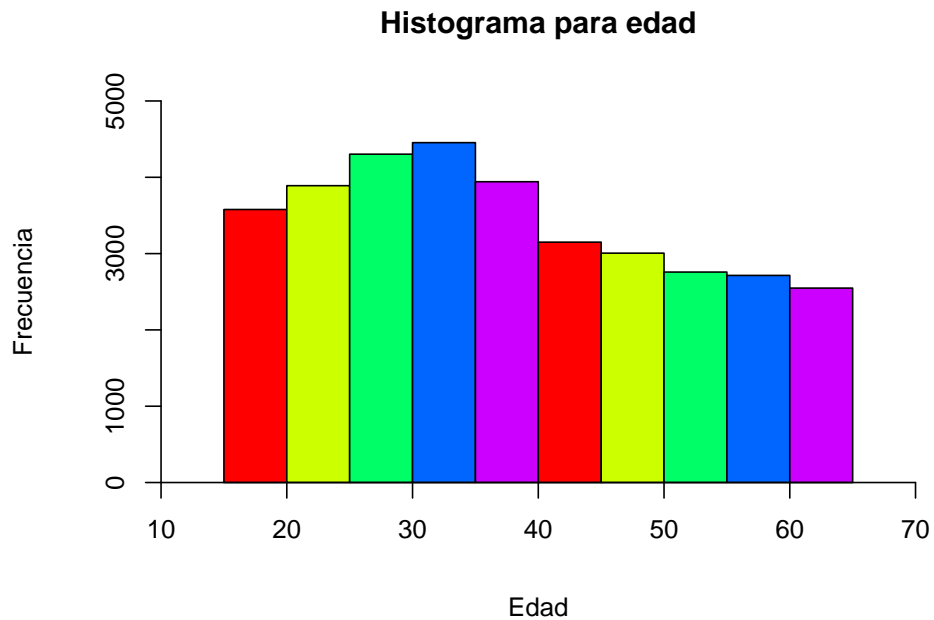
```
## Realizamos un histograma con las frecuencias relativas  
library(lattice)  
histogram(edad,  
           breaks = 10,  
           ylab = "Porcentaje del Total")
```




```
## Cambiamos el número de barras. Agregamos color
hist(as.numeric(edad),
     col = "seagreen")
```



```
## Estilamos el gráfico
hist(as.numeric(edad),
     axes = FALSE,
     main = "Histograma para edad",
     xlab = "Edad",
     ylab = "Frecuencia",
     # col = "steelblue",
     xlim = c(10,70),
     ylim = c(0, 5000),
     col = rainbow(5))
axis(1, pos = 0)
axis(2, pos = 10)
```



11.2 Gráfico de barras

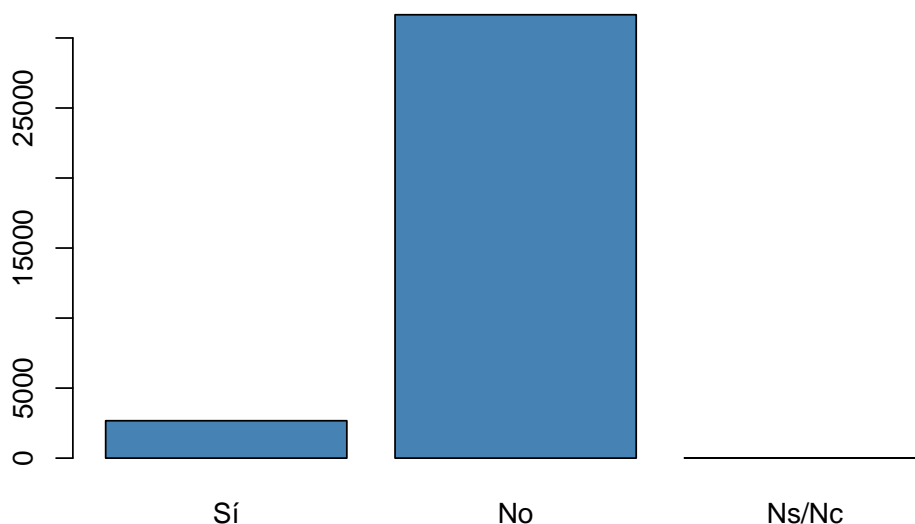
El gráfico de barras nos permite visualizar las frecuencias en **variables cualitativas**. Para realizar un gráfico de barras podemos utilizar la función `barplot`

```
## Seleccionamos BISG02 y la guardamos en una nueva variable
accidente <- enprecosp$BISG02

## Convertimos a factor y etiquetamos los códigos de valores
accidente <- factor(accidente,
  labels = c("Sí", "No", "Ns/Nc"))
## Construimos las frecuencias para la variable accidente
f <- table(accidente)
frel <- prop.table(f)

## Juntamos todo y armamos una tabla de distribución de frecuencias.
dfreq <- cbind(f, frel)

barplot(dfreq[,1],
  col = "steelblue")
```



11.3 Gráfico de tortas

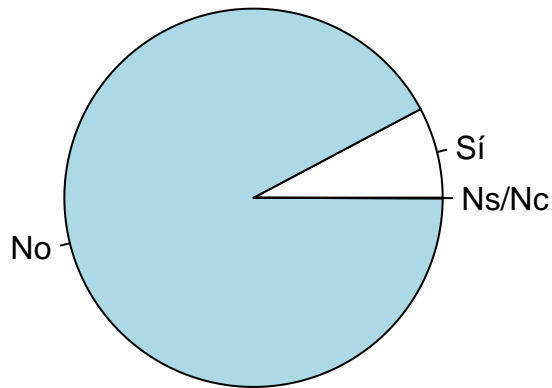
Los gráficos de tortas también nos permiten graficar frecuencias. Los gráficos de torta están actualmente desaconsejados. Se aconseja en su lugar el uso de gráfico de barras. Los gráficos de barras permiten visualizar más fácilmente las diferencias de proporciones que los gráficos de barras, particularmente cuando representamos más de dos proporciones. Para ver una revisión acerca de la discusión de gráficos de barras y de torta vea Spence (2005). Para realizar un gráfico de tortas, podemos utilizar la función `pie`

```
## Seleccionamos BISG02 y la guardamos en una nueva variable
accidente <- enprecosp$BISG02

## Convertimos a factor y etiquetamos los códigos de valores
accidente <- factor(accidente,
                    labels = c("Sí", "No", "Ns/Nc"))

## Construimos las frecuencias para la variable accidente
f <- table(accidente)
frel <- prop.table(f)

pie(frel)
```



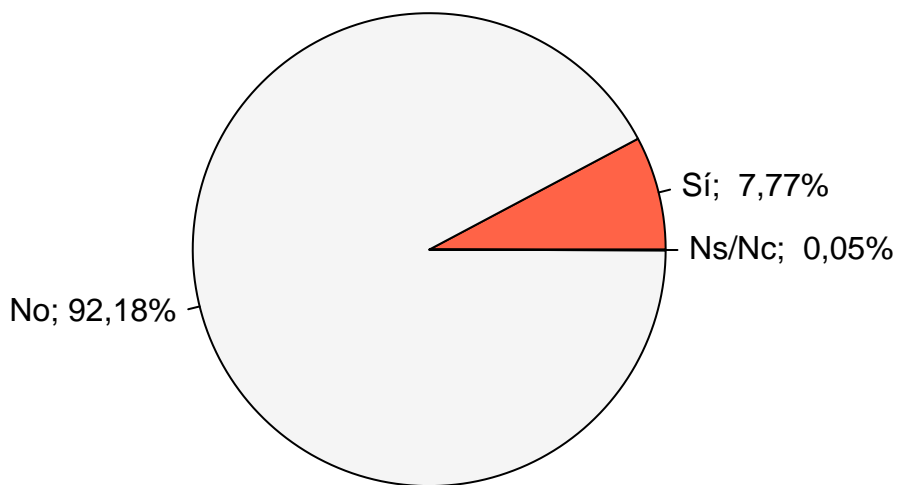
Estilamos el gráfico. Agregamos los porcentajes.

```
## Cargamos librería para formatear los porcentajes
library(formattable)

## Guardamos la tabla con porcentajes en una nueva variable
porcentaje <- percent(frel, digits = 2, dec = ",")

## Construimos las etiquetas
etiq <- paste(names(porcentaje), "; " , round(porcentaje, 4), sep = "")

pie(porcentaje,
    labels = etiq,
    radius = 1,
    col = c("tomato", "whitesmoke", "violetred")
)
```



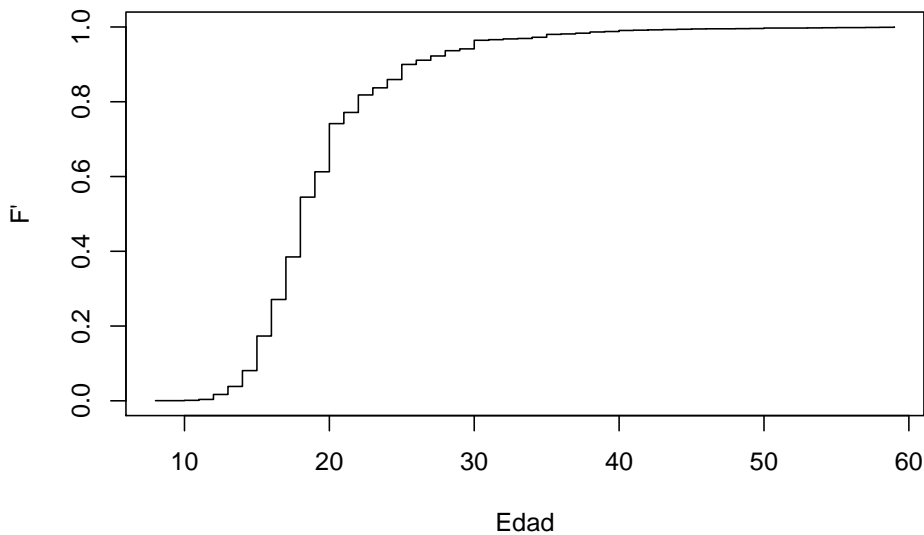
11.4 Ojiva de Galton

La ojiva de Galton nos permite graficar las **frecuencias relativas acumuladas** y buscar **percentiles** y **cuantiles empíricos**. Veamos como se distribuye la edad de inicio de consumo de marihuana en la muestra:

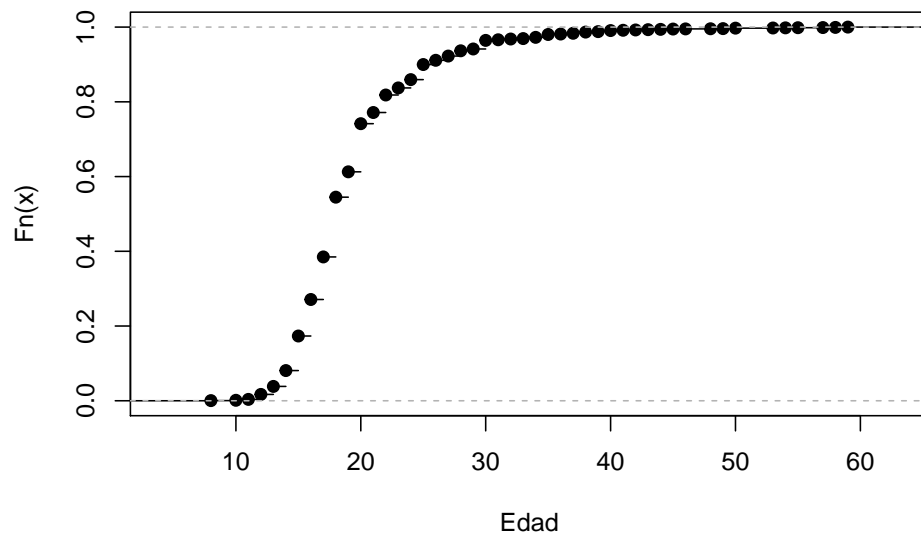
```
## Seleccionamos la edad y la guardamos en una nueva variable
## Quito los valores 99, que corresponden a NS/NC
edad <- enprecosp$BIMA03[enprecosp$BIMA03 != 99]

f <- table(edad)
frel <- prop.table(f)
frelcum <- cumsum(frel)

plot(names(frelcum), frelcum,
      type = "s",
      ylab = "F'",
      xlab = "Edad")
```



```
## Realizamos un gráfico similar, pero utilizando la función
## distribución acumulada empírica
plot(ecdf(edad),
      main = "Ojiva de Galton",
      xlab = "Edad")
```

Ojiva de Galton

Capítulo 12

Medidas de resumen

12.1 Medidas de posición

Las medidas de posición nos van a dar información acerca de diferentes localizaciones de los datos en una variables. Pueden ser **centrales**, como la *media*, la *mediana* y la *moda*. O **no centrales**, como los **cuartiles** y **percentiles**.

12.1.1 Proporción

Es una **frecuencia relativa**. Vimos varios ejemplos de proporciones cuando realizamos las tablas de distribución de frecuencia en el Capítulo 10. Calculemos la proporción de personas que alguna vez en la vida consumieron marihuana.

```
## Proporción de personas que consumieron marihuana alguna vez en la vida (PV_MA)
marihuana <- enprecosp$PV_MA
marihuana <- factor(marihuana,
                    labels = c("Sí", "No")
                    )

p <- prop.table(table(marihuana))
p

## marihuana
##          Sí          No
## 0.07326093 0.92673907
```

Entonces, el 7,33% de los encuestados consumió alguna vez marihuana en la vida.

12.1.2 Moda

Es el valor de la variable que más se repite. Es el valor de la variable que tenga la frecuencia más alta.

```
## En el periodo en que usted consumía marihuana con
## mayor frecuencia ¿cada cuánto consumía?
fconsumo <- enprecosp$BIMA04
fconsumo <- factor(fconsumo,
                    levels = c(1:6, 9),
                    labels = c("Casi todos los días",
                               "3 0 4 días a la semana",
                               "1 o 2 días a la semana",
                               "De 1 a 3 días al mes",
                               "Menos de una vez al mes",
                               "Una sola vez",
                               "Ns/Nc"))

## Calculamos las frecuencias absolutas
p <- table(fconsumo)
p
```

```
## fconsumo
##      Casi todos los días  3 0 4 días a la semana  1 o 2 días a la semana
##                      259                      119                      267
##      De 1 a 3 días al mes  Menos de una vez al mes                      Una sola vez
##                      248                      485                      1120
##                      Ns/Nc
##                      18
```

```
## Buscamos el valor con la frecuencia máxima
which.max(p)
```

```
## Una sola vez
##           6
```

12.1.3 Mediana

Es el valor de la variable que deja por debajo y por arriba, el 50% de los casos. Podemos calcularla a partir de variables de **nivel ordinal**. La podemos observar a partir de las frecuencias relativas acumuladas. Continuando con la frecuencia de consumo de marihuana

```
## En el periodo en que usted consumía marihuana con
## mayor frecuencia ¿cada cuánto consumía?
fconsumo <- enprecosp$BIMA04
```



```
## Excluimos el valor 9, Ns/Nc
fconsumo <- factor(fconsumo,
  levels = c(6:1),
  labels = c("Una sola vez",
    "Menos de una vez al mes",
    "De 1 a 3 días al mes",
    "1 o 2 días a la semana",
    "3 o 4 días a la semana",
    "Casi todos los días"
  ),
  ordered = TRUE)

## Frecuencias absolutas
frec <- table(fconsumo)
## Frecuencias relativas
frel <- prop.table(frec)
## Frecuencias relativas acumuladas
cumfrel <- cumsum(frel)
cumfrel
```

##	Una sola vez	Menos de una vez al mes	De 1 a 3 días al mes	
##	0.4483587		0.6425140	0.7417934
##	1 o 2 días a la semana	3 o 4 días a la semana	Casi todos los días	
##	0.8486789	0.8963171	1.0000000	

El primer valor que supera 0.5 es la mediana. En este caso, la mediana es *menos de una vez al mes*.

12.1.4 Cuartiles y percentiles

Los cuartiles dividen al conjunto de datos en 4. El primer cuartil (**1Q**) es el valor de la variable que deja por debajo el 25% de los casos. El tercer cuartil (**3er cuartil**) es el valor de la variable que deja por debajo el 75% de los casos. El segundo cuartil (**2Q**) es el valor que deja por debajo el 50% de los casos. Es la **mediana**.

Si dividimos a la distribución de datos en 100, obtenemos los **percentiles**. El **percentil r** es el valor de la variable que deja el r por ciento de los casos por debajo de él. Utilicemos estas medidas para comparar la edad de inicio de consumo de alcohol, tabaco y marihuana.

```
## Seleccionamos las variables de edad de inicio de consumo
## para tabaco, alcohol y marihuana
edad_tabaco <- enprecosp$BITA03[enprecosp$BITA03 != 99]
edad_alcohol <- enprecosp$BIBA03[enprecosp$BIBA03 != 99]
edad_marihuana <- enprecosp$BIMA03[enprecosp$BIMA03 != 99]
```

```
## Calculamos los cuantiles
quantile(edad_tabaco, na.rm = TRUE)

##   0%  25%  50%  75% 100%
##    3   15   16   18   64

quantile(edad_alcohol, na.rm = TRUE)

##   0%  25%  50%  75% 100%
##    1   15   17   20   64

quantile(edad_marihuana, na.rm = TRUE)

##   0%  25%  50%  75% 100%
##    8   16   18   21   59

## Calculamos los percentiles 5 y 95
quantile(edad_tabaco, c(0.05, 0.95), na.rm = TRUE)

##   5% 95%
##   12  25

quantile(edad_alcohol, c(0.05, 0.95), na.rm = TRUE)

##   5% 95%
##   13  26

quantile(edad_marihuana, c(0.05, 0.95), na.rm = TRUE)

##   5% 95%
##   14  30
```

12.1.5 Media

Es el promedio. Se obtiene sumando todos los datos y dividiendo por el número de casos.

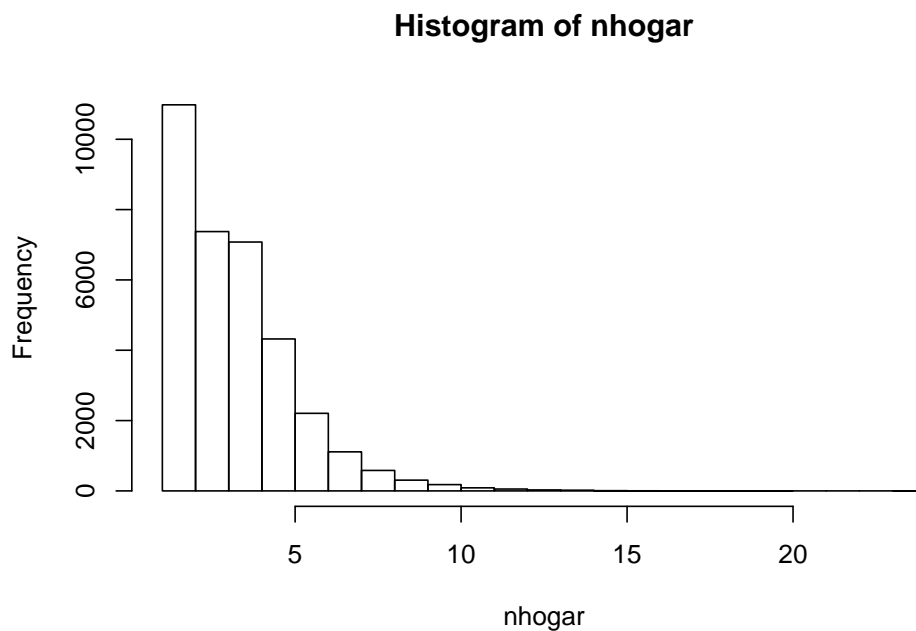
$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

La media, a diferencia de la mediana, es sensible a valores extremos. Observemos un ejemplo.

```
## Cantidad de miembros en el hogar
nhogar <- enprecosp$CNTDDCOMP
range(nhogar)

## [1] 1 24
```

```
## Graficamos  
hist(nhogar)
```



```
## Calculamos las medidas de resumen  
sum(nhogar)/length(nhogar)
```

```
## [1] 3.572751
```

```
mean(nhogar)
```

```
## [1] 3.572751
```

```
median(nhogar)
```

```
## [1] 3
```

12.2 Medidas de dispersión

Son medidas que nos indican el grado de agrupación de los datos

12.2.1 Rango o recorrido

Es la distancia entre el valor máximo y el valor mínimo.

$$R = x_n - x_1$$

Donde:

x_n es el valor máximo

x_1 es el valor mínimo

```
## Veamos cual es el rango de la variables cantidad de miembros del hogar
nhogar <- enprecosp$CNTDDCOMP
range(nhogar)
```

```
## [1] 1 24
```

```
## Rango para cantidad de habitaciones del hogar
nhabitaciones <- enprecosp$BHH002
range(nhabitaciones)
```

```
## [1] 0 20
```

También son de utilidad el **rango o recorrido intercuartilar** y el **rango o recorrido semi-intercuartilar**.

El **rango o recorrido intercuartilar** se simboliza con AIQ y es la distancia entre el tercer y el primer cuartil. El **rango o recorrido intercuartilar** se simboliza con SRIC y es la mitad del AIQ. Calulemos estas distancias para el tiempo de consumo de tabaco, alcohol y marihuana.

```
## Seleccionamos las variables de edad de inicio de consumo
## para tabaco, alcohol y marihuana
edad_tabaco <- enprecosp$BITA03[enprecosp$BITA03 != 99]
edad_alcohol <- enprecosp$BIBA03[enprecosp$BIBA03 != 99]
edad_marihuana <- enprecosp$BIMA03[enprecosp$BIMA03 != 99]
```

```
## Calculamos los cuantiles
q_tabaco <- quantile(edad_tabaco, na.rm = TRUE)
q_alcohol <- quantile(edad_alcohol, na.rm = TRUE)
q_marihuana <- quantile(edad_marihuana, na.rm = TRUE)
```

```
## Calculamos el rango intercuartilar
aiq_t <- q_tabaco[4] - q_tabaco[2]; aiq_t
```

```
## 75%
```

```
## 3
```

```
aiq_al <- q_alcohol[4] - q_alcohol[2]; aiq_al
```

```
## 75%
```

```
## 5
```

```
aiq_ma <- q_marihuana[4] - q_marihuana[2]; aiq_ma
```

```
## 75%
```

```
## 5
## Calculamos el rango semi-intercuartilar
aiq_t/2

## 75%
## 1.5
aiq_al/2

## 75%
## 2.5
aiq_ma/2

## 75%
## 2.5
```

12.2.2 Varianza y desvío estándar

Una de las medidas más utilizadas para medir la variabilidad cuando los datos son cuantitativos es la varianza (s^2) y el desvío estándar (s).

Sea n la cantidad de casos. La **varianza** es la suma de los cuadrados de los desvíos con respecto a la media, dividido por $n - 1$.

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

El **desvío estándar** es la raíz cuadrada de la varianza.

$$s = \sqrt{s^2}$$

```
## Calculemos la varianza para los edad de
## Inicio de consumo de tabaco
## Quitamos los valores faltantes
edad_tabaco <- edad_tabaco[!is.na(edad_tabaco)]

## Calculo la varianza
n <- length(edad_tabaco)
xn <- mean(edad_tabaco)
s2 <- sum((edad_tabaco - xn)^2) / (n - 1); s2

## [1] 19.81292
## Desvío estándar
sqrt(s2)

## [1] 4.45117
```

```
## Utilizando la función sd
var(edad_tabaco)

## [1] 19.81292
sd(edad_tabaco)

## [1] 4.45117
## Calculamos sd para alcohol y marihuana
sd(edad_alcohol, na.rm = TRUE)

## [1] 4.705321
sd(edad_marihuana, na.rm = TRUE)

## [1] 5.476129
```

La varianza y el desvío estandar dependen de la unidad de medida que se utilice. Por ejemplo, la varianza de la edad de inicio de consumo de alcohol se mide en años al cuadrado. Y el desvío en años. Ello hace que la medida sea diferente si utilizamos escalas diferentes. Imaginemos que medimos el peso de un grupo de personas en gramos y en kilos. La varianza, para ese grupo de personas, medida en *gramos*² será más grande que cuando la midamos en *kilos*².

A veces, incluso, nos interesa comparar magnitudes diferentes, por ejemplo, peso y altura, y comparar la varianza de ambas variables. Para poder interpretar más fácilmente el desvío estandar en términos relativos se utiliza el **coeficiente de variación** (CV)

$$CV = \frac{s}{\bar{x}} * 100$$

El coeficiente de variación es adimensional y me permite comparar el desvío estandar en términos de porcentajes.

```
sd(edad_tabaco) / mean (edad_tabaco) * 100

## [1] 26.02141
sd(edad_marihuana, na.rm = TRUE) / mean(edad_marihuana, na.rm = TRUE) * 100

## [1] 28.00788
```

12.2.3 Varibilidad en variables cualitativas

Para medir la variabilidad en variables cualitativas podemos hacer uso de coeficientes de incertidumbre. La idea es que mientras más concentrados estén los datos en una categoría, tendrán menor incertidumbre o menor variabilidad. Mientras estén repartidos más equitativamente, entonces tendremos mayor incertidumbre o mayor variabilidad.

Una medida útil en estos casos es la H de Shanon and Weaver (1949).

$$H(x) = - \sum_{i=1}^k f'_i * \log_2(f'_i)$$

Utilizando este coeficiente, comparemos la viabilidad para hombres y para mujeres de: ¿Cuán fácil o difícil le sería conseguir tranquilizantes sin indicación médica? (EnPreCoSP)

```
## Seleccionamos las variables
facil_tranq <- enprecosp$BIAC07_01
sexo <- enprecosp$BHCH04

## Convertimos a factor
facil_tranq <- factor(facil_tranq,
                      labels = c("Me sería fácil",
                                "Me sería difícil",
                                "No podría conseguir",
                                "Ns/Nc"))

sexo <- factor(sexo,
               labels = c("Varón",
                           "Mujer"))

## Calculamos las frecuencias relativas por sexo
tab <- prop.table(table(facil_tranq, sexo), 2); tab
```

```
##              sexo
## facil_tranq   Varón   Mujer
## Me sería fácil  0.2874517 0.2626644
## Me sería difícil 0.2714259 0.2753287
## No podría conseguir 0.3279914 0.3616081
## Ns/Nc          0.1131311 0.1003988
```

```
phombre <- tab[,1]
pmujer <- tab[,2]
```

```
## Calculamos el coeficiente de incertidumbre
-sum(phombre * log2(phombre))
```

```
## [1] 1.910841
```

```
-sum(pmujer * log2(pmujer))
```

```
## [1] 1.882528
```

12.3 Medidas de forma

Con la librería `e1071` podemos calcular la simetría y la curtosis de una distribución.

```
library(e1071)

## Seleccionamos la edad y la guardamos en una nueva variable
edad <- enprecosp$BHCH05
## Simetría
skewness(edad, na.rm = TRUE)
```

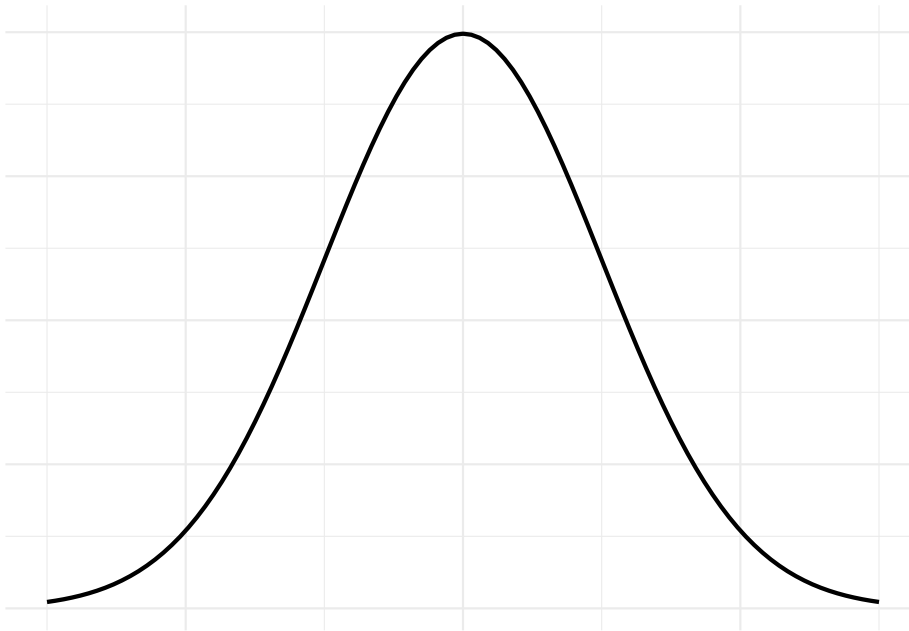
```
## [1] 0.2444952

## Curtosis
kurtosis(edad, na.rm = TRUE)
```

```
## [1] -1.048094
```

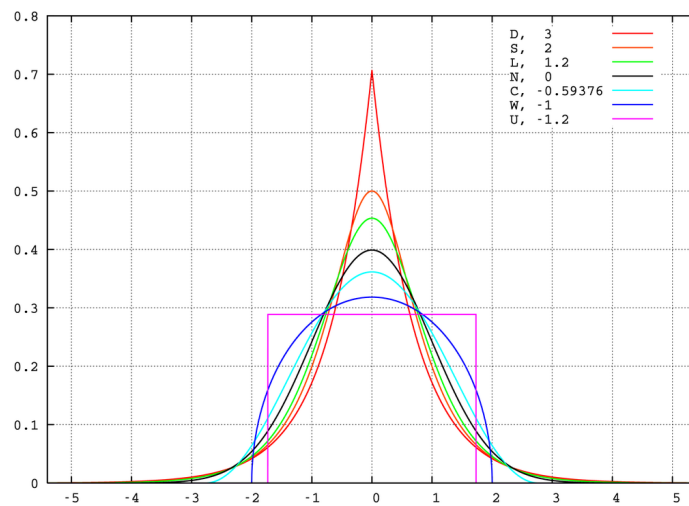
La distribución normal es un concepto teórico que nos permite aproximar el comportamiento de una gran cantidad de variables. Existe una definición matemática precisa de la distribución normal. Por ahora, nos conformaremos con saber que la distribución normal tiene una forma acampanada.

```
ggplot(data.frame(x = 0), aes(x = x)) +
  stat_function(fun = dnorm, col = "black", size = 1) +
  xlim(c(-3, 3)) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.ticks = element_blank()) +
  labs(x = NULL, y = NULL)
```

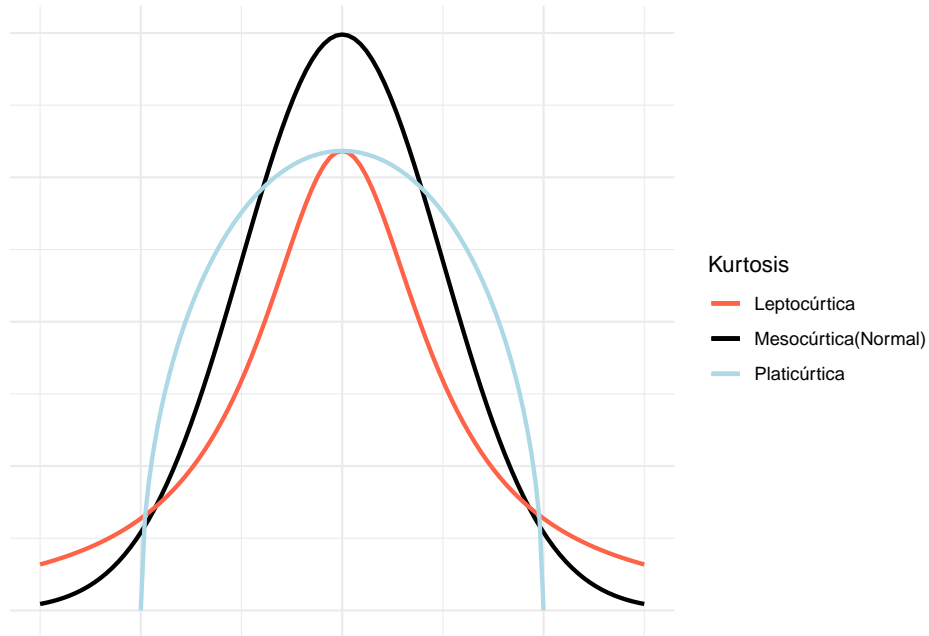
Dos medidas comunmente utilizadas son la asimetría y la kurtosis. La **kurtosis** mide que tan coludas son las distribuciones con respecto a la normal. El 0 representa una distribución normal. Valores positivos son distribuciones más coludas (con más valores extremos) y valores negativos distribuciones menos coludas.

```
knitr::include_graphics("./img/1008px-Standard_symmetric_pdfs.png")
```



- Un índice entre -0.5 y 0.5 indica que la distribución es mesocúrtica.
- Un índice mayor a 0.5 insdica que la distribución es leptocúrtica

- Un índice menor a -0.5 indica que la distribución es platicúrtica



12.4 Estandarización

La estandarización mediante los procesos de **centrado** y **escalado** de los datos. Para centrar los datos, les restamos la media. Para escalarlos, los dividimos por su desvío estándar.

```
## Seleccionamos la edad y la guardamos en una nueva variable
edad <- enprecosp$BHCH05
```

```
## Centramos
edad_centrada <- edad - mean(edad)
```

```
## Escalamos
edad_estandarizada <- edad_centrada / sd(edad)
```

```
head(edad)
```

```
## [1] 16 59 65 39 19 55
```

```
head(edad_estandarizada)
```

```
## [1] -1.61510952 1.49337797 1.92712041 0.04756984 -1.39823830 1.20421634
```

Los puntajes estandarizados, también llamados **puntajes z**, son adimensionales. Esa transformación es útil para comparar a los individuos con su grupo de referencia, y detectar, por ejemplo, valores extremos. Al ser una medida relativa, también nos sirve para comparar a un individuo en diferentes variables. Veremos posteriormente que, en las distribuciones normales, aproximadamente el 95% de los casos se encuentra entre -2 y 2 desvíos estandar. Por lo tanto, encontrar individuos con puntajes z mayores a 2 o menores a -2 nos indica que son más bien casos atípicos.

Capítulo 13

Gráficos de Resumen

13.1 Gráficos de barras

Los gráficos de barras también sirven para graficar las medias. También podemos agregar las

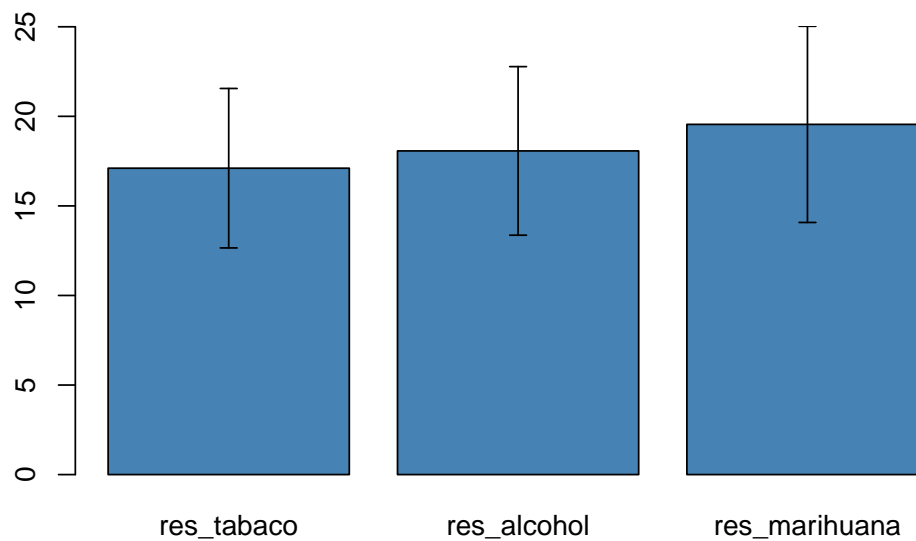
```
library(Hmisc)

## Seleccionamos las variables de edad de inicio de consumo
## para tabaco, alcohol y marihuana
edad_tabaco <- enprecosp$BITA03[enprecosp$BITA03 != 99]
edad_alcohol <- enprecosp$BIBA03[enprecosp$BIBA03 != 99]
edad_marihuana <- enprecosp$BIMA03[enprecosp$BIMA03 != 99]

## Calculamos la media y el desvío estandar
res_tabaco <- c(mean(edad_tabaco, na.rm = TRUE), sd(edad_tabaco, na.rm = TRUE))
res_alcohol <- c(mean(edad_alcohol, na.rm = TRUE), sd(edad_alcohol, na.rm = TRUE))
res_marihuana <- c(mean(edad_marihuana, na.rm = TRUE), sd(edad_marihuana, na.rm = TRUE))

## Construimos una matriz con los resultados
m <- cbind(res_tabaco, res_alcohol, res_marihuana)

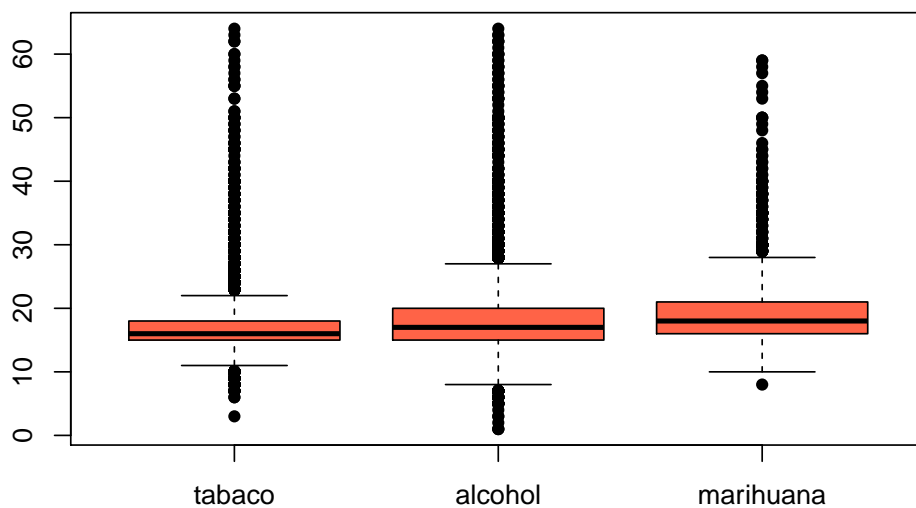
## Armamos el gráfico de barras
b <- barplot(m[1,], ylim = c(0, 25), col = "steelblue")
# errbar(colnames(m), m[1,], m[1,] + m[2,], m[1,] - m[2,])
arrows(b, m[1,] - m[2,], b, m[1,] + m[2,], length=0.05, angle=90, code=3)
```



```
# x <- factor(c("tabaco", "alcohol", "marihuana"))
# medias <- c(mean(edad_tabaco, na.rm = TRUE),
#             mean(edad_alcohol, na.rm = TRUE),
#             mean(edad_alcohol, na.rm = TRUE))
# sd <- c(sd(edad_tabaco, na.rm = TRUE),
#         sd(edad_alcohol, na.rm = TRUE),
#         sd(edad_alcohol, na.rm = TRUE))
#
# barplot(height = medias, width = 1, col = "steelblue")
```

13.2 Gráficos de caja

```
boxplot(list(edad_tabaco, edad_alcohol, edad_marihuana),
        names = c("tabaco", "alcohol", "marihuana"),
        col = "tomato",
        pch = 16)
```



Attaching package: 'dplyr'

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

group_rows

The following objects are masked from 'package:Hmisc':

src, summarize

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

Capítulo 14

Relaciones entre variables

Hasta aquí hemos venidos trabajando principalmente con medidas de resúmenes, tablas y gráficos para variables individuales. Muchas veces interesa observar como las variables se comportan conjuntamente. Para ello, también haremos uso de tablas, gráficos y medidas de resúmenes. En términos técnicos, lo que nos interese es explicar la variabilidad. Podemos explicar parcialmente la variabilidad de una variable conociendo los valores de otras variables que estén **asociadas**. Diremos que existe **asociación de variables** cuando la variación de una de ellas influya en la variación de las otras. Por ejemplo, el peso y la altura son dos variables que están asociadas. Esperaríamos que las personas que son más altas sean, en promedio, mas pesadas que aquellas que son más bajas. Diremos que dos variables son **independientes** cuando la variación de una de ellas no afecta la variación de la otra.

14.1 Tablas de contingencia

Son llamadas también **tablas de distribución conjunta**, **tablas bivariadas** o **tablas cruzadas**. Permite observar la relación entre variables cualitativas. Se representan las categorías de una variables en las filas, y las categorías de la otra en las columnas, y se pueden calcular diferentes tipos de frecuencias absolutas o relativas.

Supongamos que queremos observar la relación entre la prevalencia de consumo con el riesgo percibido para el consumo frecuente de alcohol (BIBA14_02).

```
## Prevalencia de consumo del último mes y riesgo percibido
## Alcohol
riesgo_perc <- factor(enprecosp$BIBA14_02,
                      labels = c("Ningún riesgo",
                                "Riesgo leve o moderado",
```

```

                                "Gran riesgo",
                                "No sé que riesgo corre"))

prev_mes <- factor(enprecosp$P1M_BA,
                   labels = c("Sí",
                               "No"))

## Realizamos una tabla de frecuencias absolutas conjunta
tc <- table(riesgo_perc, prev_mes)
tc

```

```

##              prev_mes
## riesgo_perc      Sí   No
## Ningún riesgo      202   61
## Riesgo leve o moderado 1701 1139
## Gran riesgo      14261 16601
## No sé que riesgo corre   163   215

```

En estas tablas también se pueden expresar las frecuencias marginales. Las **frecuencias marginales** son los totales por fila y por columna. Se corresponden con las frecuencias univariadas para cada una de las variables.

```

## Frecuencias marginales
addmargins(tc)

```

```

##              prev_mes
## riesgo_perc      Sí   No   Sum
## Ningún riesgo      202   61   263
## Riesgo leve o moderado 1701 1139 2840
## Gran riesgo      14261 16601 30862
## No sé que riesgo corre   163   215   378
## Sum              16327 18016 34343

```

También podemos calcular las frecuencias relativas

```

## Frecuencias relativas al total
tc_tot <- prop.table(tc)
tc_tot <- addmargins(tc_tot)

format(tc_tot, digits = 0, nsmall = 4)

```

```

##              prev_mes
## riesgo_perc      Sí   No   Sum
## Ningún riesgo      "0.0059" "0.0018" "0.0077"
## Riesgo leve o moderado "0.0495" "0.0332" "0.0827"
## Gran riesgo      "0.4153" "0.4834" "0.8986"
## No sé que riesgo corre "0.0047" "0.0063" "0.0110"

```

```
##      Sum                "0.4754" "0.5246" "1.0000"

## Frecuencias relativas por filas
tc_filas <- addmargins(tc, 1)
tc_filas <- prop.table(tc_filas, 1)

format(tc_filas, digits = 0, nsmall = 4)

##               prev_mes
## riesgo_perc      Sí      No
## Ningún riesgo    "0.7681" "0.2319"
## Riesgo leve o moderado "0.5989" "0.4011"
## Gran riesgo      "0.4621" "0.5379"
## No sé que riesgo corre "0.4312" "0.5688"
## Sum              "0.4754" "0.5246"

## Frecuencias relativas por columna
tc_cols <- addmargins(tc, 2)
tc_cols <- prop.table(tc_cols, 2)

format(tc_cols, digits = 0, nsmall = 4)

##               prev_mes
## riesgo_perc      Sí      No      Sum
## Ningún riesgo    "0.0124" "0.0034" "0.0077"
## Riesgo leve o moderado "0.1042" "0.0632" "0.0827"
## Gran riesgo      "0.8735" "0.9215" "0.8986"
## No sé que riesgo corre "0.0100" "0.0119" "0.0110"
```

14.2 Riesgo relativo

	Enfermo	Saludable	Total
Expuesto	D_e	H_e	N_e
No Expuesto	D_n	H_n	N_n

$$RR = \frac{D_e/N_e}{D_n/N_n}$$

14.3 Odds ratio

$$OR = \frac{D_e/H_e}{D_n/H_n}$$

14.4 Intensidad de la asociación

Las medidas de asociación nos permiten evaluar el grado o intensidad de la relación entre las variables.

14.4.1 Q de Kendall - Yule

El Q de Kendall-Yule es una medida de intensidad de asociación para variables dicotómicas. Supongamos que tenemos la siguiente tabla de doble entrada:

	Sí	No	Total
Positiva	a	b	a+b
Negativa	c	d	c+d
Total	a+c	b+d	n

El coeficiente de Q es:

$$Q = \frac{ad - bc}{ad + bc}$$

Evaluemos si el sexo está relacionado con la prevalencia mensual de consumo de alcohol.

```

sexo <- factor(enprecosp$BHCH04J,
               labels = c("Varón",
                          "Mujer"))

prev_alcohol <- factor(enprecosp$P1M_BA,
                      labels = c("Sí",
                                  "No"))

tc <- table(sexo, prev_alcohol)

tc

##           prev_alcohol
## sexo      Sí      No

```

```
## Varón 11378 11434
## Mujer 4949 6582

## Veamos el grado de asociación
a <- tc[1,1]
b <- tc[1,2]
c <- tc[2,1]
d <- tc[2,2]

## Calculamos el coeficiente Q de Kendall-Yule
Q <- (a*d-b*c)/(a*d+b*c)
Q

## [1] 0.1392118
```

14.4.2 Chi cuadrado (χ^2)

```
## Riesgo leve percibido
## En su opinión, ¿cuál cree usted que es el riesgo que
## corre una persona que toma bebidas alcohólicas de vez
## en cuando

riesgo_perc <- factor(enprecosp$BIBA14_01,
                      labels = c("Ningún riesgo",
                                "Riesgo leve o moderado",
                                "Gran riesgo",
                                "No sé qué riesgo"))

# prop.table(table(riesgo_perc))

prev_alcohol <- factor(enprecosp$P1M_BA,
                      labels = c("Sí",
                                "No"))

tc <- table(riesgo_perc, prev_alcohol)

## Calculamos las frecuencias esperadas y observadas
chi <- chisq.test(tc)
chi$observed

##               prev_alcohol
## riesgo_perc      Sí      No
## Ningún riesgo    4153 2813
## Riesgo leve o moderado 7432 7309
```

```
## Gran riesgo          4549 7622
## No sé qué riesgo     193  272
chi$expected

##               prev_alcohol
## riesgo_perc      Sí      No
## Ningún riesgo    3311.7049 3654.2951
## Riesgo leve o moderado 7008.0164 7732.9836
## Gran riesgo      5786.2131 6384.7869
## No sé qué riesgo  221.0656  243.9344
chi_val <- chi$statistic; chi_val

## X-squared
## 967.376
```

14.5 Coeficiente Phi de Pearson-Yule (φ)

$$\varphi = \sqrt{\frac{\chi^2}{n}}$$

```
sexo <- factor(enprecosp$BHCH04J,
               labels = c("Varón",
                           "Mujer"))

prev_alcohol <- factor(enprecosp$P1M_BA,
                       labels = c("Sí",
                                   "No"))

tc <- table(sexo, prev_alcohol)

chi <- chisq.test(tc)
chi_val <- chi$statistic

n <- sum(tc)
phi <- sqrt(chi_val/n)
```

14.5.1 Coeficiente de contingencia C de Pearson

Mide el grado de asociación en tablas de contingencia con más de dos categorías.

$$C = \sqrt{\frac{\chi^2}{\chi^2 + n}}$$

Va de 0 a 1 y se compara con un C máximo:

$$C_{max} = \sqrt{\frac{\min(f, c) - 1}{\min(f, c)}}$$

Donde: f es el número de filas g es el número de columnas

```
## Riesgo leve percibido
## En su opinión, ¿cuál cree usted que es el riesgo que
## corre una persona que toma bebidas alcohólicas de vez
## en cuando

riesgo_perc <- factor(enprecosp$BIBA14_01,
                      labels = c("Ningún riesgo",
                                "Riesgo leve o moderado",
                                "Gran riesgo",
                                "No sé qué riesgo"))

# prop.table(table(riesgo_perc))

prev_alcohol <- factor(enprecosp$P1M_BA,
                      labels = c("Sí",
                                "No"))

tc <- table(riesgo_perc, prev_alcohol)

## Calculamos las frecuencias esperadas y observadas
chi <- chisq.test(tc)
chi$observed

##               prev_alcohol
## riesgo_perc   Sí    No
## Ningún riesgo  4153 2813
## Riesgo leve o moderado 7432 7309
## Gran riesgo    4549 7622
## No sé qué riesgo    193  272

chi$expected

##               prev_alcohol
## riesgo_perc   Sí          No
## Ningún riesgo  3311.7049 3654.2951
## Riesgo leve o moderado 7008.0164 7732.9836
```

```
## Gran riesgo          5786.2131 6384.7869
## No sé qué riesgo     221.0656  243.9344
```

```
chi_val <- chi$statistic; chi_val
```

```
## X-squared
## 967.376
```

```
## Calculamos el C de Pearson
```

```
f <- nrow(tc)
c <- ncol(tc)
n <- sum(tc)
C <- sqrt(chi_val/(chi_val + n)); C
```

```
## X-squared
## 0.1655185
```

```
Cmax <- sqrt((min(f, c) - 1) / min(f, c)); Cmax
```

```
## [1] 0.7071068
```

14.5.2 Coeficiente V de Cramer

$$V = \sqrt{\frac{\chi^2}{n * \min(f - 1, c - 1)}}$$

```
V <- sqrt(chi_val/(n * min(ncol(tc), nrow(tc))))
```


Bibliografía

Shanon, C. and Weaver, W. (1949). The mathematical theory of information.
Urbana: University of.

Spence, I. (2005). No humble pie: The origins and usage of a statistical chart.
Journal of Educational and Behavioral Statistics, 30(4):353–368.