

Sobre R

R es un lenguaje de programación basado en objetos de ahí que estos sean los elementos más importantes para manejar al utilizar R. Los objetos en R tienen algunos atributos como se muestra en la tabla 1

Atributo	Significado	Definición
Nombre	Nombre R interno es usado para referirse al objeto en cuestión	Definido por el usuario
Tipo	Es la clase a la pertenece el objeto	Puede ser definido por el usuario
Modo	Contiene los valores del objeto	No definido por el usuario
Dimensiones	Tamaño del contenido	Puede ser definido por el usuario

Nombre

La creación de un objeto en R es de la siguiente manera

Código
Nombre <- Contenido Para sobrescribir un objeto: Nombre <- Nuevo contenido Eliminar un objeto: rm(Nombre)

Modos

Los modos se refieren a la clase de valores que contienen los objetos. Por lo general, sólo se encontrará algunos de los siguientes modos básicos:

En R	Lenguaje cotidiano
character	Una letra, palabra u oración
Numeric	Un numero puede contener decimales
Logical	Un indicador de verdadero o falso

Dentro de R, el modo del contenido de cualquier objeto puede identificarse utilizando la función `class()` después de haber descompuesto el objeto en componentes básicos de tipo vectorial. Además, se puede emplear la función `str()` que intenta realizar automáticamente parte del subconjunto por usted y darle una visión general de los componentes de su objeto

Tipo

En R, el tipo de cualquier objeto puede identificarse mediante la función `class()` o `str()`. Tenga en cuenta que, para un objeto de tipo vector, la función `class()` devuelve el modo del contenido de los vectores.

Tipo de objeto R	Lenguaje cotidiano
Vector	Una lista de valores
Factor	Una lista de variables con valores predefinidos
Matrix	Una tabla de valores
Data Frame	Una tabla de valores
Fuction	Una expresión funcional tipo receta que detalla un proceso
List	Una lista donde cada elemento corresponde a una lista, tabla o receta

Vector

Vectores son creados usando la función `c(...,...)` (“c”, significa concatenar; “,” separado cada unidad del vector) en R:

```
Letters_vec <- c("A", "B", "C")
Letters_vec

## [1] "A" "B" "C"
class(Letters_vec)

## [1] "character"
```

También puede convertir el modo de elemento de objeto para que sea de carácter utilizando la función `as.character()`, como numérico con la función `as.numeric()` y como lógico usando `as.logical()`.

Factor

Los factores se crean utilizando la función `factor(x , niveles , ...)` ("x" representa los datos; "niveles" indica los niveles preconcebidos que deben tomar nuestros datos y suele estimarse a partir de los datos por defecto) en R.

```
Letters_fac <- factor(x = c("A", "B", "C"))
Letters_fac

## [1] A B C
## Levels: A B C
class(Letters_fac)

## [1] "factor"
```

También puede convertir los tipos de objetos en factor utilizando la función `as.factor()`.

List

Las listas se crean con la función `list(...)` ("..." representa los objetos pasados a la lista) en R

```

Vectors_ls <- list(Numbers_vec, Letters_vec)
Vectors_ls

## [[1]]
## [1] 1 2 3
##
## [[2]]
## [1] "A" "B" "C"
class(Vectors_ls)
## [1] "list"

```

Tener en cuenta que una lista puede contener todos los objetos concebibles de R dentro de cada posición de la lista (sí, incluso las listas de listas son posibles). También puede convertir los tipos de objetos a ser de lista utilizando la función `as.list()`.

Matrix

Las matrices se crean con la función `matrix(data, nrow, ncol, byrow, dimnames)` ("data" representa los datos; "nrow" y "ncol" indican el número de filas y columnas respectivamente, "byrow" es un argumento lógico que indica si se rellenan los datos en la matriz por filas o por columnas, "dimnames" permite asignar nombres a las columnas y filas) en R.

```

Combine_mat <- matrix(data = c(Numbers_vec, Letters_vec), ncol = 2)
Combine_mat

##      [,1] [,2]
## [1,] "1"  "A"
## [2,] "2"  "B"
## [3,] "3"  "C"
class(Combine_mat)
## [1] "matrix"

```

También puede convertir los tipos de objetos en matriciales utilizando la función `as.matrix()`

Data Frame

Los data frame se crean con la función `data.frame(...)` ("..." puede representar una matriz o indexar columnas individuales) en R

```

Combine_df <- data.frame(Combine_mat)
Combine_df

##   X1 X2
## 1  1  A
## 2  2  B
## 3  3  C
class(Combine_df)
## [1] "data.frame"

```

También puede editar los nombres de las columnas y filas de un objeto `data.frame` utilizando los comandos `colnames()` y `rownames()` respectivamente.

Creación de un data.frame a partir de vectores como columnas individuales:

```
Combine2_df <- data.frame(Numbers = Numbers_vec, Letters = Letters_vec)
Combine2_df

##   Numbers Letters
## 1         1      A
## 2         2      B
## 3         3      C
class(Combine_df)
## [1] "data.frame"
```

Dimensión de objetos

Las dimensiones de los objetos son el último de los atributos esenciales de los objetos que vamos a considerar. Indican cómo están dispuestos los datos. Se evalúan utilizando la función `dim(...)` en R ("`...`" significa cualquier nombre de objeto en R)

Las matrices tienen dimensiones de rowcount x column count

```
dim(Combine_mat)
## [1] 3 2
```

Los data frames también tienen dimensiones de rowcount x columncount

```
dim(Combine_df)
## [1] 3 2
```

Los vectores, los factores y las listas no tienen dimensiones. . . sólo tienen una longitud

```
c(dim(Letters_vec), dim(Letters_fac), dim(Vectors_ls))
## NULL

c(length(Letters_vec), length(Letters_fac), length(Vectors_ls))
## [1] 3 3 2
```

Renombrar

Los nombres pueden servir como etiquetas para los elementos de un objeto y se implementan/asignan utilizando la función `names()` en el nivel más básico (es decir, "para" vectores y factores"). Otras implementaciones de nombres vienen en forma de las siguientes funciones:

- `colnames()` (etiquetas de columna, data.frames)
- `rownames()` (etiquetas de filas, data.frames)
- `dimnames()` (etiquetas de dimensiones, matrices)

```
colnames(Combine2_df)
## [1] "Numbers" "Letters"
rownames(Combine2_df)
## [1] "1" "2" "3"
Combine2_df$Numbers # subsetting by column 'Numbers'
## [1] 1 2 3
```

Indexación

El sistema de indexación es básicamente la contrapartida numérica del sistema de nomenclatura y se pone en marcha mediante el uso de corchetes ([]):

- [elementnumber] para vectores y factores
- [[elementnumber]] para las listas
- [rownumber, columnnumber] para data.frames y matrices

Primer elemento de la segunda columna de nuestra matriz:

```
Combine_mat[1, 2]
## [1] "A"
```

Vectorización

R es un lenguaje vectorizado, esto nos dice que cada operación realizada en un objeto se aplica a cada elemento del objeto.

```
Numbers_vec + 1
## [1] 2 3 4
```

Paquetes en R

Los paquetes son la forma que tiene R de ofrecer al usuario una gama más amplia de funcionalidades (al igual que un mod de un juego de ordenador o las pistas adicionales de un CD).

Hay miles de paquetes para R que han sido diseñados por otros usuarios de R, probados vigorosamente, y están disponibles libremente para su uso.

Todos los paquetes están disponibles a través de la Comprehensive R Archive Network (<https://cran.r-project.org/>) y se puede obtener una visión general de los paquetes disponibles aquí: https://cran.rproject.org/web/packages/available_packages_by_date.html.

Para instalar y cargar en la consola

install.packages() se utiliza para instalar paquetes en R

```
# vegan is a common library of functionality in biostatistics
install.packages("vegan")
```

library() se utiliza para cargarlos en el entorno de trabajo actual

```
library(vegan)
```

Operadores lógicos

Los enunciados lógicos son indicadores de si algo es cierto o no. Las utilizamos con frecuencia en la vida real

Operador	
==	Igual que
!=	Diferente de
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que

Estructuras básicas

Las sentencias if() se basan en sentencias lógicas. Comprueban si algo es correcto y luego actúan en consecuencia

```
# is 1 smaller than 2?
if (Numbers_vec[1] < Numbers_vec[2]) {
  # if the statement is correct
  print("Is smaller") # print this to the console
} else {
  # if the statement is not correct
  print("Is not smaller") # print this to the console
}

## [1] "Is smaller"
```

Los bucles for() están en acción mientras un indicador está dentro de un rango de datos especificado rango de datos:

```
# loop from one to length of the letter vector in steps of 1
for (i in 1:length(Letters_vec)) {
  # print current iteration element of letters vector
  print(Letters_vec[i])
}

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

Los bucles while() funcionan de forma muy parecida a los bucles for() y están en acción mientras una declaración indicia es TRUE

```
# while our data frame has equal to or less than 3 columns
while (dim(Combine_df)[2] <= 3) {
  # bind letters factor vector to data frame as column
  Combine_df <- cbind(Combine_df, Letters_fac)
}
Combine_df # inspect the result

##   X1 X2 Letters_fac Letters_fac
## 1  1  1 A          A          A
## 2  2  2 B          B          B
## 3  3  3 C          C          C
```

Variables

Una variable presenta información más o menos valiosa sobre una posible multitud de características de un sistema de estudio.

Las variables pueden ser clasificadas en múltiples tipos, entre los mas comunes sistemas de clasificación son:

Variables categóricas

También conocidas como variables
cuantitativas

- Normal
- Ordinal

Variables continuas

También conocidas como variables
cualitativas

- Discretas
- Continuas

A su vez, existen las variables categóricas las cuales son variables que permiten clasificar o agrupar en clases, estas pueden tener números finitos, asignar una unidad a una clase comúnmente en R son de tipo factor o carácter. Ejemplos:

- Clasificaciones de biomas (por ejemplo, "Bosque boreal", "Tundra", etc.)

- Sexo (por ejemplo, "macho", "hembra")
- Posición jerárquica (por ejemplo, " α -Individuo", " β -Individuo", etc.)
- Tipo de suelo (por ejemplo, "Arena", "Barro", "Permafrost", etc.)
- Tipo de hoja (por ejemplo, "compuesta", "de una sola hoja", etc.)
- Estado de reproducción sexual (por ejemplo, "juvenil", "maduro", etc.)
- Pertenencia a la especie
- Pertenencia a un grupo familiar

Las variables continuas son aquellas que se utilizan para establecer un rango de posibles valores, toman valores finitos y son de tipo numeric. Ejemplos:

- Temperatura
- Precipitación
- Peso
- pH
- Altitud
- Tamaño del grupo
- Índices de vegetación
- Tiempo

Las variables continuas pueden convertirse en variables categóricas mediante un método llamado binning

Dado un rango de variables, se pueden establecer tantos "bins" como se quiera. Por ejemplo:

Dado un rango de temperatura de 271K - 291K, puede haber 4 bins de igual tamaño:

- Bin A: $271\text{ K} \leq 276\text{ K}$
- Bin B: $276\text{ K} \leq 281\text{ K}$
- Bin C: $281\text{ K} \leq 286\text{ K}$
- Bin D: $286\text{ K} \leq 291\text{ K}$

Otra forma de clasificar las variables son las escalas en las que se representan

Las diferentes escalas de las variables requieren diferentes procedimientos estadísticos para los análisis.

Las escalas variables incluyen:

- Nominal
- Binary
- Ordinal
- Interval
- Relation/Ratio

Algunos libros de estadística enseñan escalas de números enteros según las escalas mencionadas. Algunas personas lo discuten y afirman que estas escalas son escalas de relación.

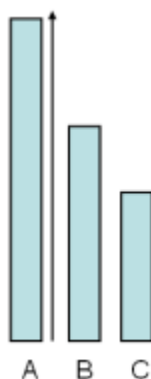
Nominal y Binary

Las escalas nominales de las variables corresponden a variables categóricas que no pueden ponerse en un orden significativo.

- Las variables en escalas nominales ponen las unidades en categorías distintas
- Estas variables pueden ser numéricas, pero no ofrecen una interpretación matemática

Ejemplos:

- Color de los pétalos (rojo, verde, azul, etc.)
- Identificaciones individuales



Las escalas binarias son un caso especial de las escalas nominales que sólo toman dos valores posibles: 0 y 1.

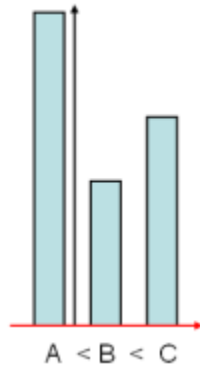
Ordinal

Las escalas ordinales de las variables corresponden a variables categóricas que pueden ser ordenarse de forma significativa.

- Las variables en escalas ordinales ponen las unidades en distintas categorías
- Estas variables pueden ser numéricas e interpretación matemática

Ejemplos:

- Tamaño (pequeño, mediano, grande, etc.)
- Variables continuas en forma de bandeja



Interval/Discrete

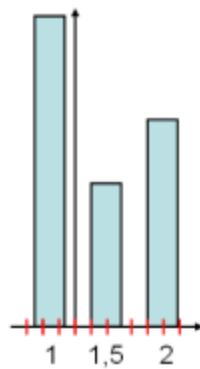
Las escalas de intervalo de las variables corresponden a una mezcla de variables continuas.

- Las variables en escalas de intervalo se miden en intervalos iguales desde un punto cero/punto de origen definido
- El punto de origen no implica una ausencia de la característica medida

Ejemplos:

Temperatura [°C]

pH



Relation/Ratio

Las escalas de relación/relación de las variables corresponden a variables continuas.

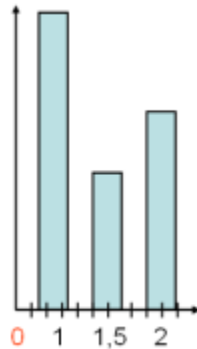
- Las variables de las escalas de relación/relación se miden en intervalos iguales a partir de un punto cero/punto de origen definido
- El punto de origen implica una ausencia de la característica medida

Ejemplos:

Temperatura [K]

Peso

Las escalas de números enteros son un caso especial de las escalas de relación que sólo permiten números enteros números integrales



Distribuciones

Una distribución de un conjunto de datos estadísticos (muestra/población) muestra todos los posibles valores/intervalos de los datos en cuestión y su frecuencia

Distribuciones de frecuencia

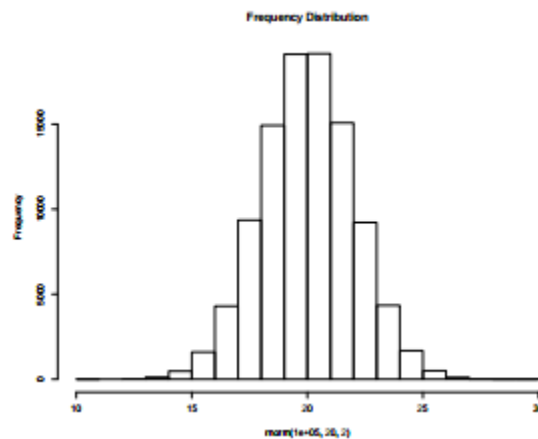
Teoría

- Representaciones simples de los datos frecuencias de valores
- Pueden establecerse para cada variable

Práctica en R

- Visualización mediante la función "hist()" función

```
hist(rnorm(100000, 20, 2),  
main = "Frequency Distribution")
```



Distribución de la densidad de la probabilidad

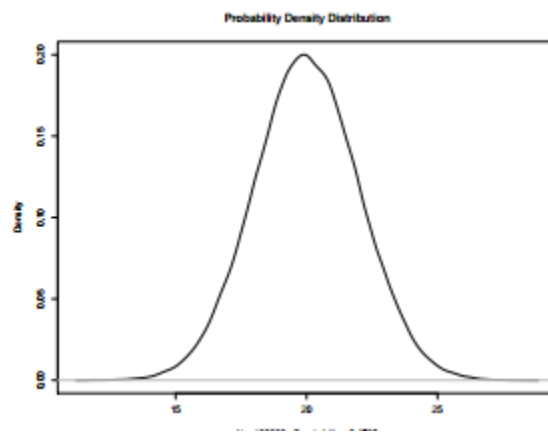
Teoría

- Representación del valor de los datos probabilidades
- Puede establecerse para variables continuas

Práctica en R

Visualización mediante la función "density()" función

```
plot(density(rnorm(100000,20,2)),  
     main = "Probability Density Distribution")
```



Las distribuciones de densidad de probabilidad tienen la mayor importancia en estadística

Algunos puntos clave sobre estas distribuciones:

- El área bajo la curva (AUC) es igual a 1
- Una probabilidad para cada valor individual dado es 0
- El AUC entre dos valores del eje X es igual a la probabilidad de muestrear aleatoriamente un valor entre estos dos puntos.

Distribución normal/gaussiana univariante

- Se utiliza para representar variables aleatorias de valor real cuyas distribuciones no se conocen
- Se aplica el teorema del límite central (se extrae un número suficiente de muestras y se obtiene la distribución normal)
- Estas distribuciones suelen conocerse también como "curvas de campana"

Comprobación de la normalidad

La comprobación de la normalidad de los datos es crucial para ciertos procedimientos estadísticos

La prueba de Shapiro-Wilks en teoría

- Supuesto de base: Los datos están normalmente distribuidos
- Si el valor $p < \text{nivel de significación elegido}$, los datos no se distribuyen normalmente
- Muy sensible al tamaño de la muestra

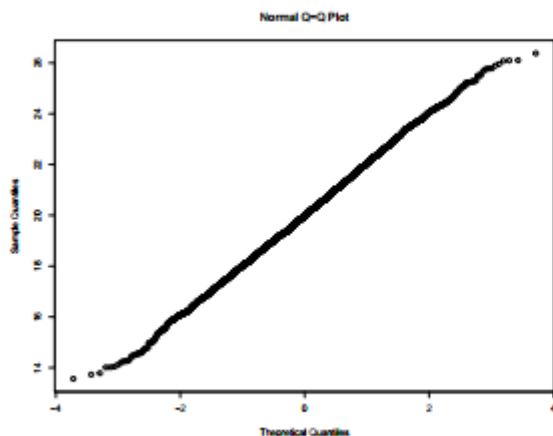
```
shapiro.test(rnorm(5000,  
  20, 2))  
  
##  
##  Shapiro-Wilk normality test  
##  
## data:  rnorm(5000, 20, 2)  
## W = 1, p-value = 0.7  
→ Clearly a normal distributed set of values
```

```
shapiro.test(seq(1, 500,  
  5))  
  
##  
##  Shapiro-Wilk normality test  
##  
## data:  seq(1, 500, 5)  
## W = 1, p-value = 0.002  
→ Clearly no normal distributed set of values
```

La grafica QQ en teoría

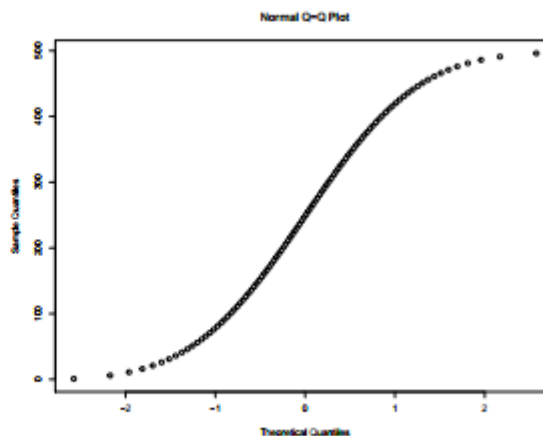
- Método para comparar dos distribuciones de probabilidad mediante el trazado de sus cuantiles entre sí
- Si las dos distribuciones que se comparan comparadas son similares, el gráfico mostrará muestra la línea $y = x$.
- Compara la distribución de los datos con la distribución normal.

```
qqnorm(rnorm(5000, 20, 2))
```



→ Clearly a normal distributed set of values

```
qqnorm(seq(1, 500, 5))
```



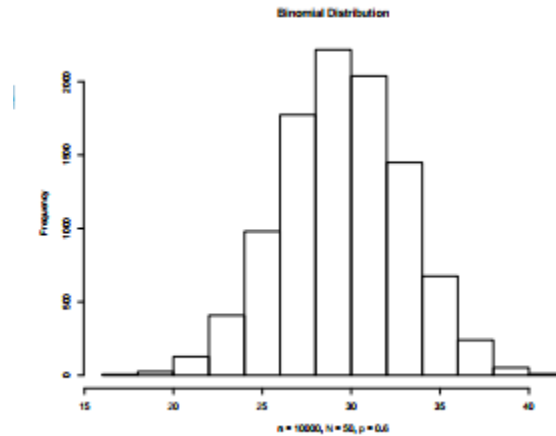
→ Clearly no normal distributed set of values

Distribución Binomial

Una de las distribuciones más importantes. Es aplicable a:

- Variables que sólo pueden tomar dos valores posibles (por ejemplo, "estados")

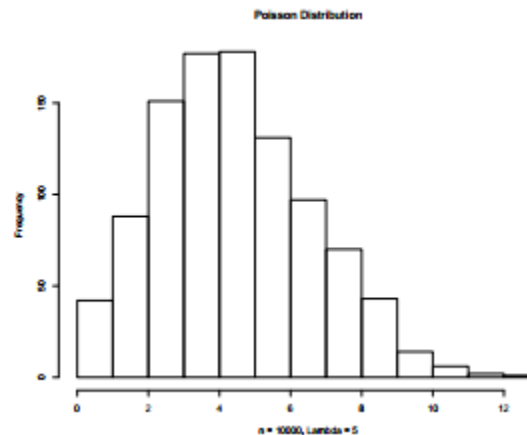
- Todos los registros de la variable tienen la misma probabilidad p de estar en uno de los dos estados
- Se compone de tres criterios:
 - p - la probabilidad de "éxito"
 - n - el tamaño de la muestra (la frecuencia de la muestra)
 - N - el "total binomial" (cuántos individuos muestreamos cada vez)



Distribución de Poisson

Otra de las distribuciones más importantes. Es aplicable a:

- Los objetos focales se colocan aleatoriamente en una o varias dimensiones
- Se coloca una "ventana de recuento" aleatoria (normalmente un considerando el tiempo) colocada por encima del muestreo esquema
- Se compone de dos criterios
 - λ - la media (= expectativa, recuento medio, intensidad) así como la varianza (es decir, varianza = media)
 - n - tamaño de la muestra



¿Cómo medir las distribuciones?

Las distribuciones pueden describirse mediante los parámetros clásicos de la estadística descriptiva

Estadísticos

- Media aritmética
- Moda
- Mediana
- Mínimo, Máximo, Rango

Varianza

- Desviación estándar
- Rango de cuantiles
- Asimetría
- Kurtosis

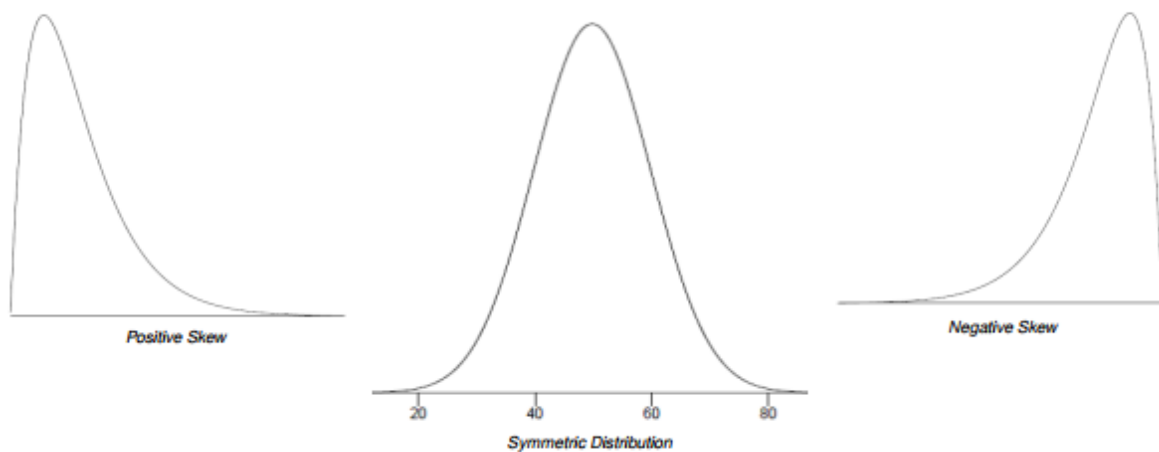
Asimetría I

Definición: Describe la simetría y la longitud relativa de la cola de las distribuciones.

Sesgo positivo: La cola de la derecha es más larga que la de la izquierda

Sesgo = 0: Distribución simétrica

Sesgo negativo: La cola izquierda es más larga que la derecha



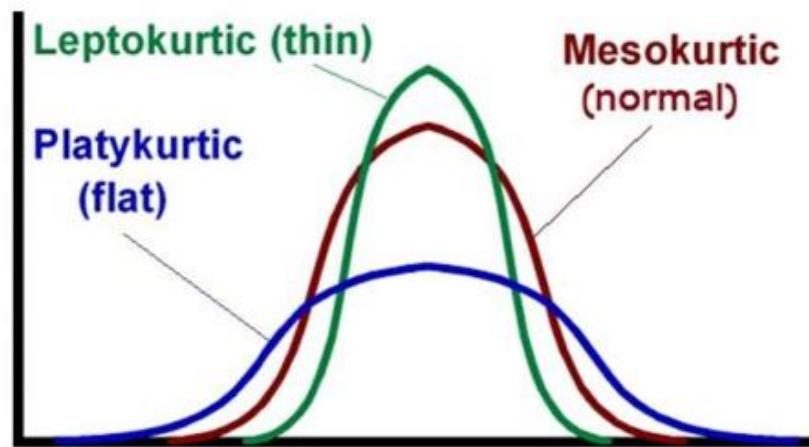
Kurtosis I

Definición: Describe la uniformidad/"cola" de las distribuciones

Positiva Kurtosis: Distribución de cola corta, también conocida como leptocúrtica

Kurtosis = 0: Representación de la base de una distribución dada, también conocida como mesocúrtica.

Negative Kurtosis: Distribución de cola larga, también conocida como platocúrtica.



Referencias

[1]. Erik Kusch AARHUS UNIVERSITY, Center for biodiversity and dynamics in a changing world, "Introduction to R", 2019.