CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Curso DM

“Análisis Exploratorio de Datos”

(con ejemplos en R)

Primavera 2023

Basado en las slides de Felipe Bravo

Análisis Exploratorio de Datos (o EDA)

- Engloba un conjunto de técnicas para poder **comprender/resumir** las características más importantes de un conjunto de datos o **dataset**.

- Ayuda a seleccionar la herramienta adecuada para el preprocesamiento/análisis.

- Esta metodología fue impulsada por el estadístico John Tukey.

- Se basa principalmente en dos criterios: Las **estadísticas de resumen** y la **visualización de datos**.

- En esta clase se verán ambos tipos de técnicas, además de su aplicación en **R** para datasets conocidos.

R y Studio

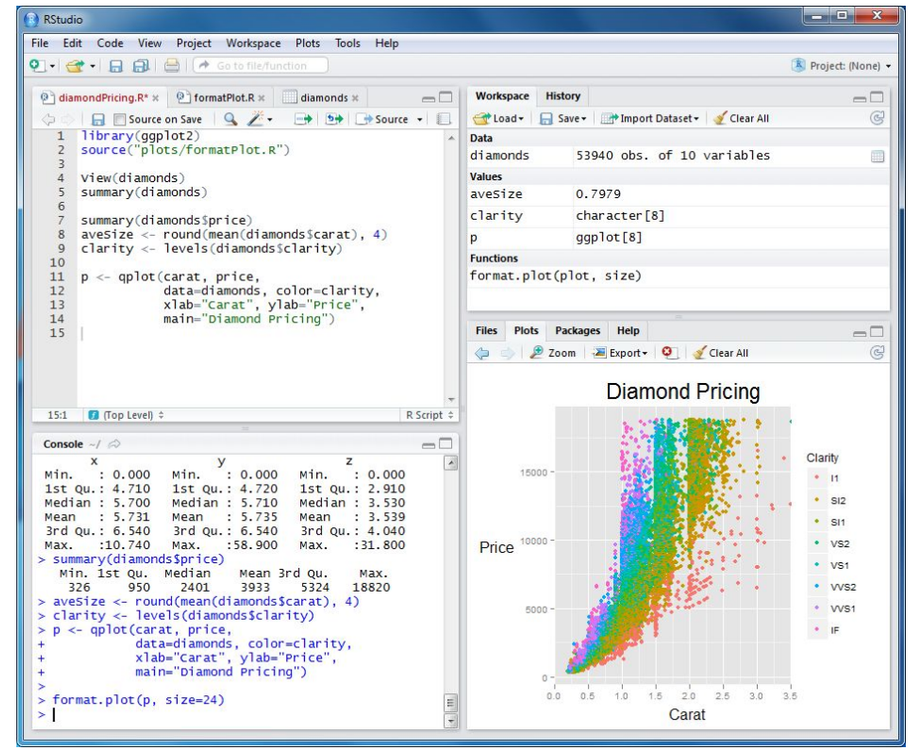
- R es un ambiente de programación estadístico totalmente gratuito: http://www.r-project.org/

- Permite manipular y almacenar datos de manera efectiva.

- Es un lenguaje de programación completo: variables, loop, condiciones, funciones.

- Provee muchas librerías para realizar distintos tipos de análisis sobre colecciones de datos, ej: visualización de datos, análisis de series temporales, análisis de grafos, análisis de texto.

- Rstudio es un IDE que ofrece un entorno amigable para trabajar con R.



El dataset Iris

Trabajaremos con un dataset muy conocido en análisis de datos llamado **Iris**.

- El dataset se compone de 150 observaciones de flores de la planta iris.
- Existen tres tipos de clases de flores iris: **virginica, setosa** y **versicolor**.
- Hay 50 observaciones de cada una.
- Las variables o atributos que se miden de cada flor son:
  - El **tipo de flor** como variable categórica.
  - El **largo y el ancho del pétalo** en cm como variables numéricas.
  - El **largo y el ancho del sépalo** en cm como variables numéricas.

El dataset Iris

Virginica

Setosa

Versicolor

- El dataset se encuentra disponible en R:

```
> data(iris) #carga el dataset al workspace

> str(iris) #ver los atributos de un dataset

'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

> attach(iris)#pasa las variables del dataframe al ambiente
```

Estadísticas de Resumen

- Las estadísticas de resumen son valores que explican propiedades de los datos.
- Algunas de estas propiedades incluyen: frecuencias, medidas de tendencia central y dispersión.
- Ejemplos:
  - **Tendencia central:** media, mediana, moda.
  - **Dispersión:** miden la variabilidad de los datos, como la desviación estándar, el rango, etc..
- La mayor parte de las estadísticas de resumen se pueden calcular haciendo una sola pasada por los datos.
- Su uso está sujeto al tipo de atributo. Por ejemplo, cuando son categóricos, generalmente hacemos estadísticas basadas en conteo.

Frecuencia

- La frecuencia de un valor de atributo es el porcentaje de veces que éste es observado.
- En R podemos contar las frecuencias de aparición de cada valor distinto de un vector usando el comando *table*:

```
> table(iris$Species)
    setosa versicolor  virginica
        50         50         50
```

- **Ejercicio:** Calcular las frecuencias porcentuales de iris$Species.

```
> table(iris$Species) / length(iris$Species)
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333
```

Medidas de Tendencia Central

Estas medidas tratan de resumir los valores observados en único valor asociado al valor localizado en el centro.

Media

- La media es la medida más común de tendencia central para una variable numérica.
- Si tenemos *m* observaciones se calcula como la media aritmética o promedio.

$$\text{mean}(x) = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

Medidas de Tendencia Central

- Es una medida muy sensible a *outliers* o valores atípicos.

```
#crea vector con distribución normal, rnorm(n, mean, sd)
> vec <- rnorm(10, 20, 10)
> mean(vec)
[1] 16.80036
```

```
> vec.ruid <- c(vec, rnorm(1,300,100))
> mean(vec.ruid)
[1] 35.36422
```

Medidas de Tendencia Central

- Podemos robustecer la media eliminando una fracción de los valores extremos usando la **media truncada** o **trimmed mean**.
- En R podemos darle un segundo parámetro a la función *mean* llamado *trim* que define la fracción de elementos extremos a descartar.

Ejemplo: Descartamos el 10% de los valores extremos en el ejemplo anterior:

```
> mean(vec, trim=0.1)
[1] 17.78799


> mean(vec.ruid,trim=0.1)
[1] 19.51609 # Mucho más robusto
```

Medidas de Tendencia Central

Mediana

- La mediana representa la posición central de la variable que separa la mitad inferior y la mitad superior de las observaciones.
- Intuitivamente, consiste en el valor donde para una mitad de las observaciones todos los valores son mayores que ésta, y para la otra mitad todos son menores.

$$\text{median}(x) = \begin{cases} x_{r+1} & \text{Si } m \text{ es impar con } m = 2r+1 \\ \frac{1}{2}(x_r + x_{r+1}) & \text{Si } m \text{ es par con } m = 2r \end{cases}$$

- Para el ejemplo anterior, vemos que la mediana es más robusta al ruido que la media:

```
> median(vec)
[1] 17.64805
> median(vec.ruid)
[1] 17.64839
```

Medidas de Tendencia Central

Moda

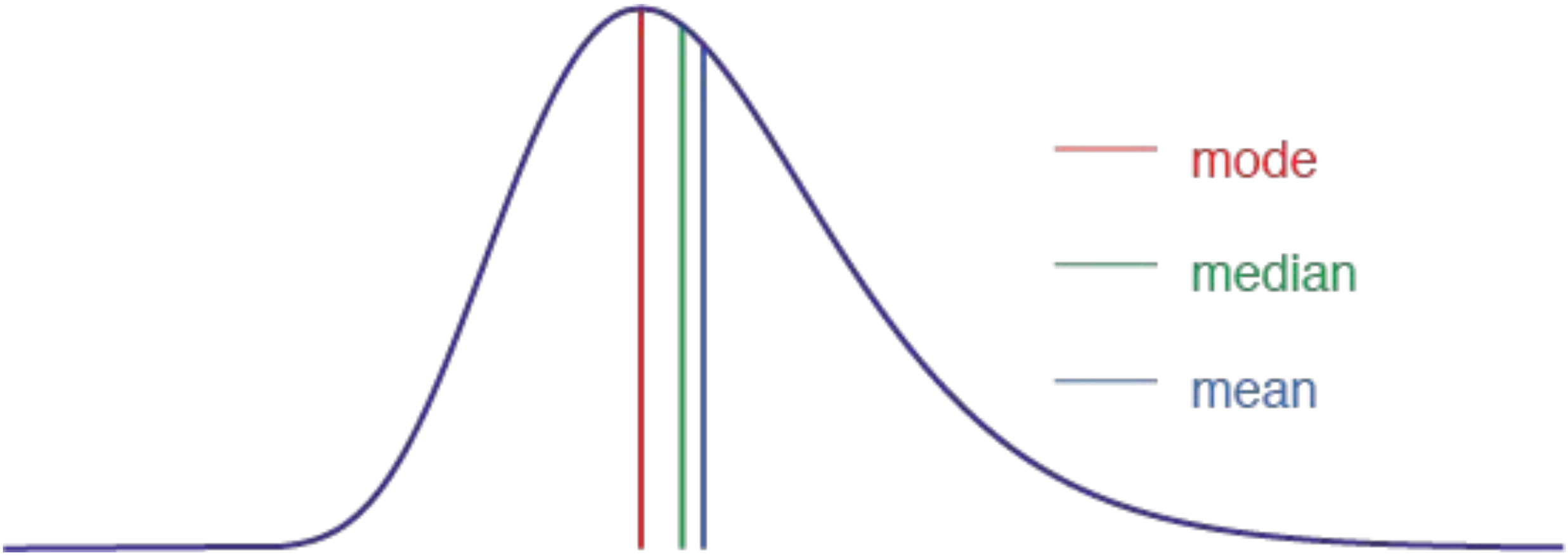
- La moda de un atributo es el valor más frecuente observado.
- No existe la función moda directamente en R, pero es fácil de calcular usando `table` y `max`:

```
my_mode <- function(var){
            frec.var <- table(var)
            valor <- which(frec.var==max(frec.var))
            names(valor)
          }
```

```
> my_mode(iris$Sepal.Length)
[1] "5"
```

- Generalmente usamos la frecuencia y la moda para estudiar variables categóricas.

Comparación entre la moda, la mediana y la media



Percentiles o Cuantiles

- El k-ésimo percentil de una variable numérica es un valor tal que el k% de las observaciones se encuentran debajo del percentil y el (100 - k) % se encuentran sobre este valor.

- En estadística se usan generalmente los cuantiles que son equivalentes a los percentiles expresados en fracciones en vez de porcentajes.

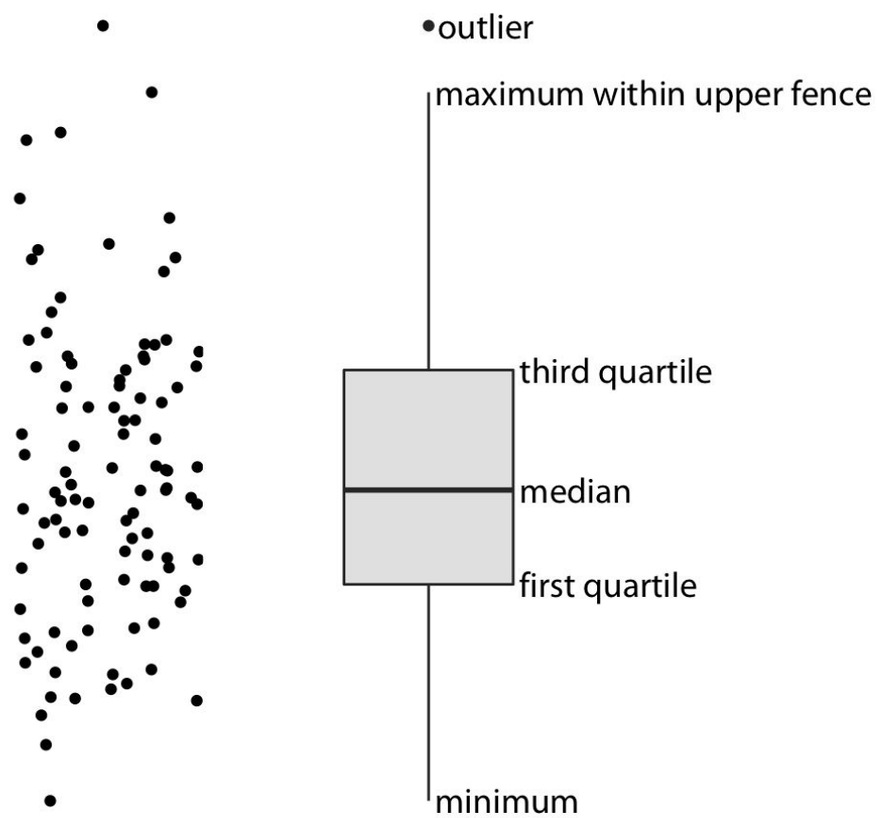
En R se calculan con el comando quantile:

```
# Todos los percentiles
quantile(Sepal.Length,seq(0, 1, 0.01))
```

Percentiles o Cuantiles

Además es muy común hablar de los **cuartiles** que son tres percentiles específicos:

- El **primer cuartil Q1** (lower quartile) es el percentil con **k = 25**.
- El **segundo cuartil Q2** es con **k = 50** que equivale a la mediana.
- El **tercer cuartil Q3** (upper quartile) es con **k = 75**.

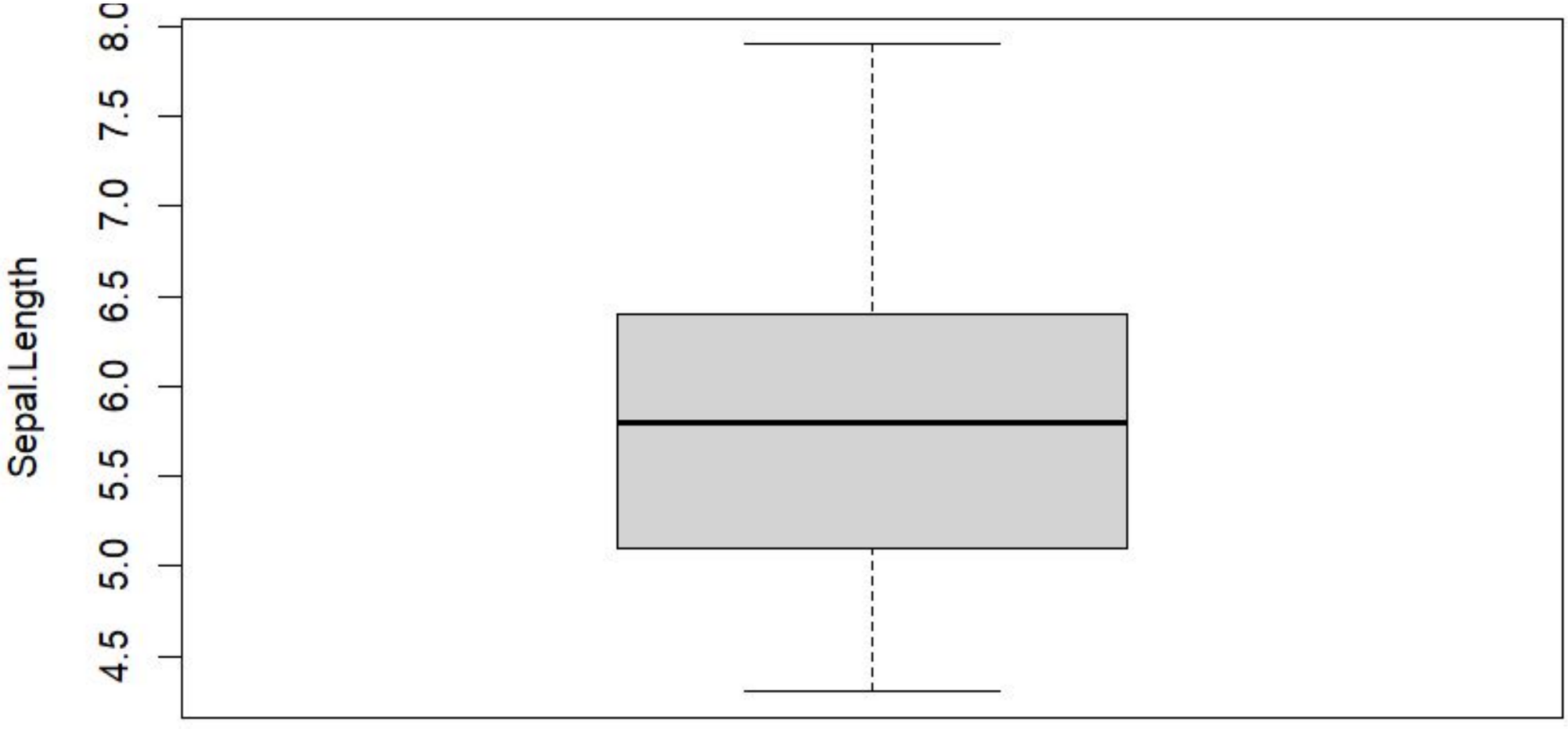


https://clauswilke.com/dataviz/

Gráficos de cajas (boxplots): La caja muestra los cuartiles del conjunto de datos, mientras que los bigotes se extienden para mostrar el resto de la distribución.

Percentiles o Cuantiles

```
# El mínimo, los tres cuartiles y el máximo
> quantile(Sepal.Length,seq(0, 1, 0.25))
  0%  25%  50%  75% 100%
 4.3  5.1  5.8  6.4  7.9
```



Resumiendo un Data Frame

- En R podemos obtener varias estadísticas de resumen de una variable o de un *data.frame* (tabla en R) usando el comando *summary*.
- Para las **variables numéricas** nos entrega el mínimo, los cuartiles, la media y el máximo.
- Para las **variables categóricas** nos entrega la tabla de frecuencias.

```
> summary(iris) #estadísticas básicas del dataframe
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

       Species
 setosa    :50
 versicolor:50
 virginica :50
```

Ejercicio

- Usando el comando tapply analice la media, la mediana y los cuartiles para las tres especies de Iris para las cuatro variables.

- ¿Nota alguna diferencia en las distintas especies?

Respuesta:

```
tapply(iris$Petal.Length,iris$Species,summary)
tapply(iris$Petal.Width,iris$Species,summary)
tapply(iris$Sepal.Length,iris$Species,summary)
tapply(iris$Sepal.Width,iris$Species,summary)
```

Medidas de Dispersión

- Estas medidas nos dicen qué tan distintas o similares tienden a ser las observaciones respecto a un valor particular.
- Generalmente este valor particular se refiere a alguna medida de tendencia central.
- El **rango** es la diferencia entre el valor máximo y el mínimo:

```
> max(Sepal.Length)- min(Sepal.Length)
[1] 3.6
```

Medidas de Dispersión

- La **desviación estándar** es la raíz cuadrada de la varianza que mide las diferencias cuadráticas promedio de las observaciones con respecto a la media.

$$\text{var}(x) = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$$

$$\text{sd}(x) = \sqrt{\text{var}(x)}$$

```
> var(Sepal.Length)
[1] 0.6856935

> sd(Sepal.Length)
[1] 0.8280661
```

Medidas de Dispersión

- Al igual que la media, la desviación estándar es sensible a outliers.
- Las medidas más robustas se basan generalmente en la mediana.
- Sea `m(x)` una medida de tendencia central de `x` (usualmente la mediana), se define la **desviación absoluta promedio** o **average absolute deviation** (AAD) como:

$$\mathrm{AAD}(x) = \frac{1}{m}\sum_{i=1}^{m} |x_i - m(x)|$$

Ejercicio: Función AAD, recibe un vector `x` y una función de media central `fun`.

```
aad <- function(x,fun=median)
        {mean(abs(x-fun(x)))}

> aad(Sepal.Length)
[1] 0.6846667
> aad(Sepal.Length,mean)
[1] 0.6875556
```

Medidas de Dispersión

- Sea *b* una constante de escala se define la **desviación media absoluta** o **median absolute deviation** como:

$$\text{MAD}(x) = b \times \text{median}(|x_i - \text{m}(x)|)$$

- En R se calcula con el comando `mad` con los parámetros `center` como una función que mide la tendencia central de la variable y `constant` como la constante b. Por defecto se usa la mediana y el valor 1,482.

```
> mad(Sepal.Length)
[1] 1.03782
```

- Finalmente, se define el **rango intercuartil (IQR)** como la diferencia entre el tercer y el primer cuartil (Q3 – Q1).

```
IQR(Sepal.Length)
[1] 1.3
```

Estadísticas de Resumen Multivariadas

- Para comparar cómo varía una variable respecto a otra, usamos medidas multivariadas.
- La covarianza `cov(x, y)` mide el grado de variación lineal conjunta de un par de variables *x*, *y* :

$$cov(x, y) = \frac{1}{m-1} \sum_{i=1}^{n} (x - \overline{x})(y - \overline{y})$$

- Donde `cov(x, x) = var(x)`
- En R se calcula con el comando *cov*:

```
> cov(Sepal.Length, Sepal.Width)
[1] -0.042434
```

Estadísticas de Resumen Multivariadas

- Para datasets de varias columnas numéricas uno puede calcular una matriz de covarianza.
- Cada celda i,j de esta matriz contiene la covarianza entre los atributos i y j.
- Esta matriz es simétrica.
- En R esta matriz se obtiene al aplicar el comando `cov` a una matriz o un data.frame de variables numéricas:

```
> cov(iris[,1:4])
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

Estadísticas de Resumen Multivariadas

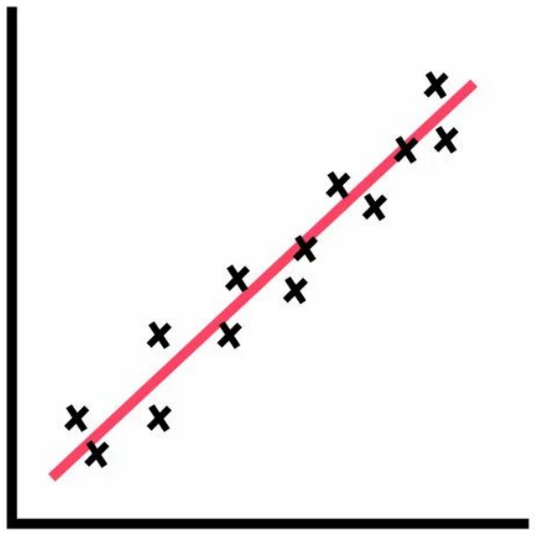
- Si dos variables son independientes entre sí, su covarianza es cero.
- Para tener una medida de relación que no dependa de la escala de cada variable, usamos la **correlación lineal**.
- Se define a la correlación lineal o coeficiente de correlación de **Pearson** r(x, y) como:

$$r(x,y) = \frac{cov(x,y)}{sd(x)sd(y)}$$

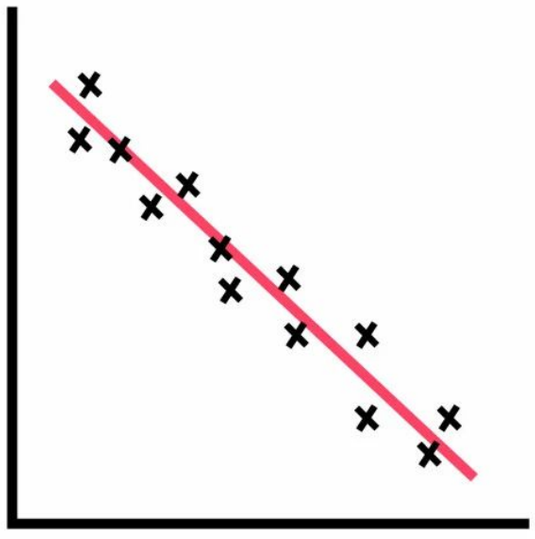
La correlación lineal varía entre –1 a 1.

- Un valor cercano a 1 indica que mientras una variable crece la otra también lo hace en una proporción lineal.
- Un valor cercano a -1 indica una relación inversa (una crece la otra decrece).

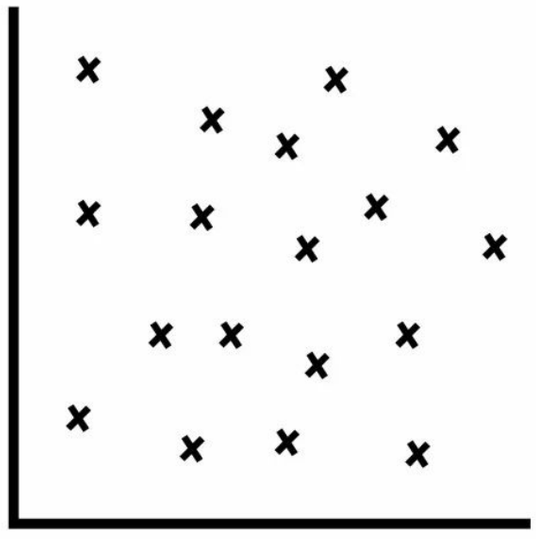
Estadísticas de Resumen Multivariadas



Positive Correlation



Negative Correlation



No Correlation

https://www.simplypsychology.org/correlation.html

Estadísticas de Resumen Multivariadas

- Una correlación cercana a 0 no implica que no pueda haber una relación no-lineal entre las variables.
- La correlación lineal se calcula en R con el comando `cor`.
- Podemos usarla de forma análoga a la covarianza para obtener una matriz de correlaciones.

```
> cor(iris[,1:4])
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.0000000	-0.1175698	0.8717538	0.8179411
Sepal.Width	-0.1175698	1.0000000	-0.4284401	-0.3661259
Petal.Length	0.8717538	-0.4284401	1.0000000	0.9628654
Petal.Width	0.8179411	-0.3661259	0.9628654	1.0000000

¡Hay que interpretar con cuidado las correlaciones!

Tablas de Contingencia

- Para analizar la relación entre variables de naturaleza categórica usamos **tablas de contingencia**.
- La tabla se llena con las frecuencias de co-ocurrencia de todos los pares de valores entre dos variables categóricas.
- En R se crean usando el comando `table` que usábamos para frecuencias, pero aplicado sobre dos vectores:

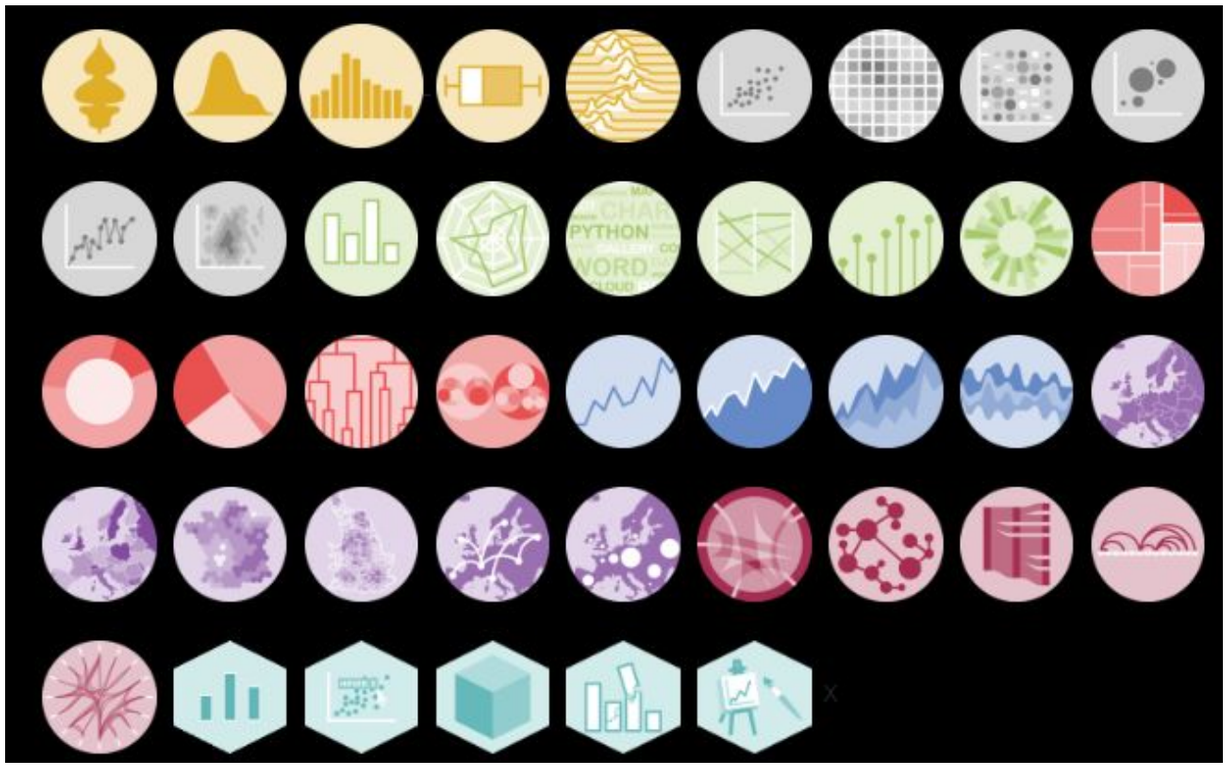
```
sexo<-c("Hombre","Hombre","Mujer","Hombre","Mujer","Mujer")
estudios<-c("universitario","secundario","secundario",
            "postgrado","secundario","universitario")

> table(sexo,estudios)
        estudios
sexo     postgrado secundario universitario
  Hombre         1          1             1
  Mujer          0          2             1
```

Visualización de Datos

- La visualización de datos es la transformación de un dataset a un formato visual que permita a las personas identificar las características y las relaciones entre sus elementos (columnas y filas).

- La visualización permite que las personas reconozcan patrones o tendencias en base a su criterio o conocimiento en el dominio de aplicación.



https://r-graph-gallery.com/

Representación

- Se entiende por representación como el mapeo que se hace a partir de los datos hacia un formato visual.

- Se traducen los datos, sus atributos y relaciones a elementos gráficos como puntos, líneas, formas y colores.

- Los objetos son usualmente representados como puntos.

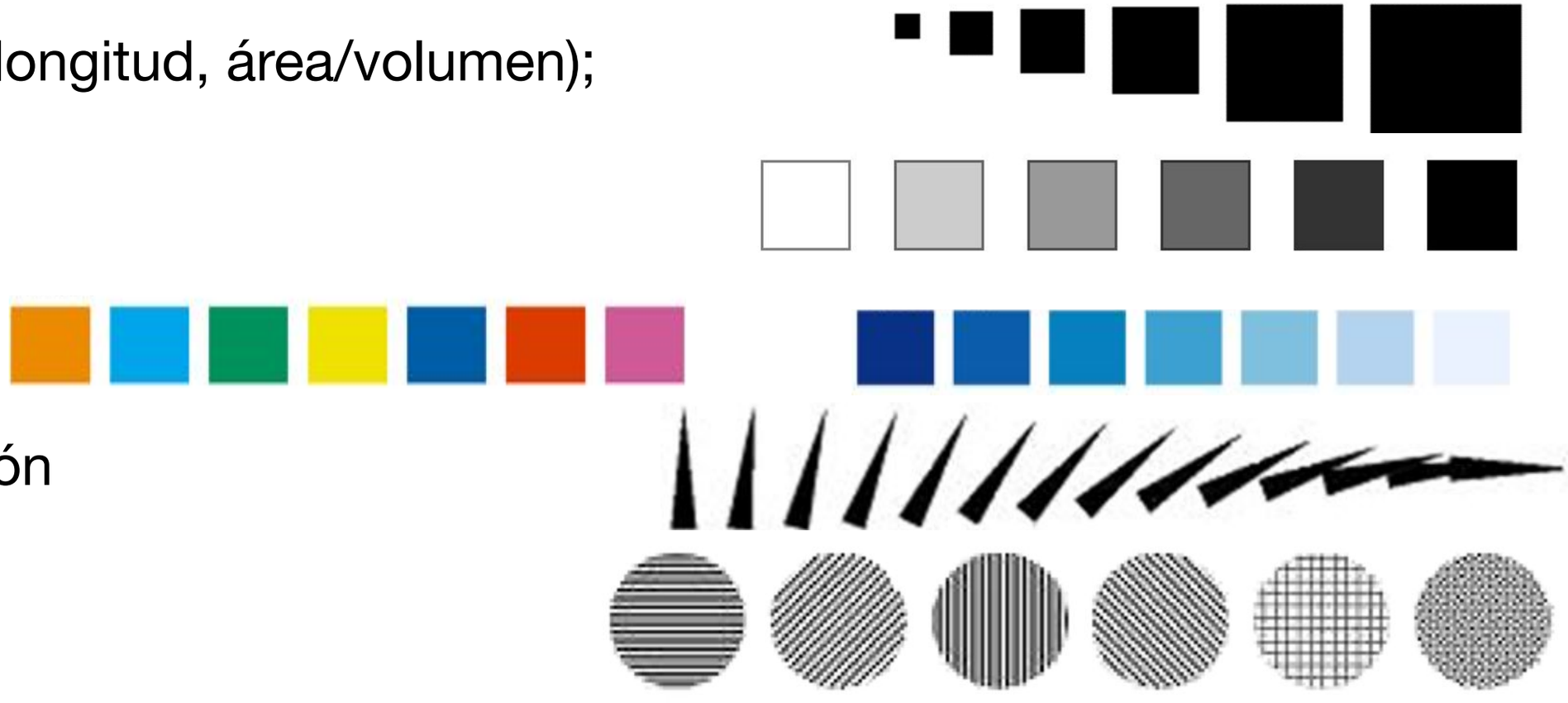
- Los valores de atributos se representan como la posición de los puntos o las características de los puntos, ej: color, tamaño y forma.

- Cuando se usa la posición para representar los valores es simple detectar si es que se forman grupos de objetos o la presencia de objetos atípicos.

Variables visuales

Permiten codificar información. [Ward, Grinstein, & Keim (2010)] destacan ocho variables visuales:

1. **Posición**
2. **Marca**
3. Tamaño (longitud, área/volumen);
4. Brillo
5. Color
6. Orientación
7. Textura
8. Movimiento



Graficando en R

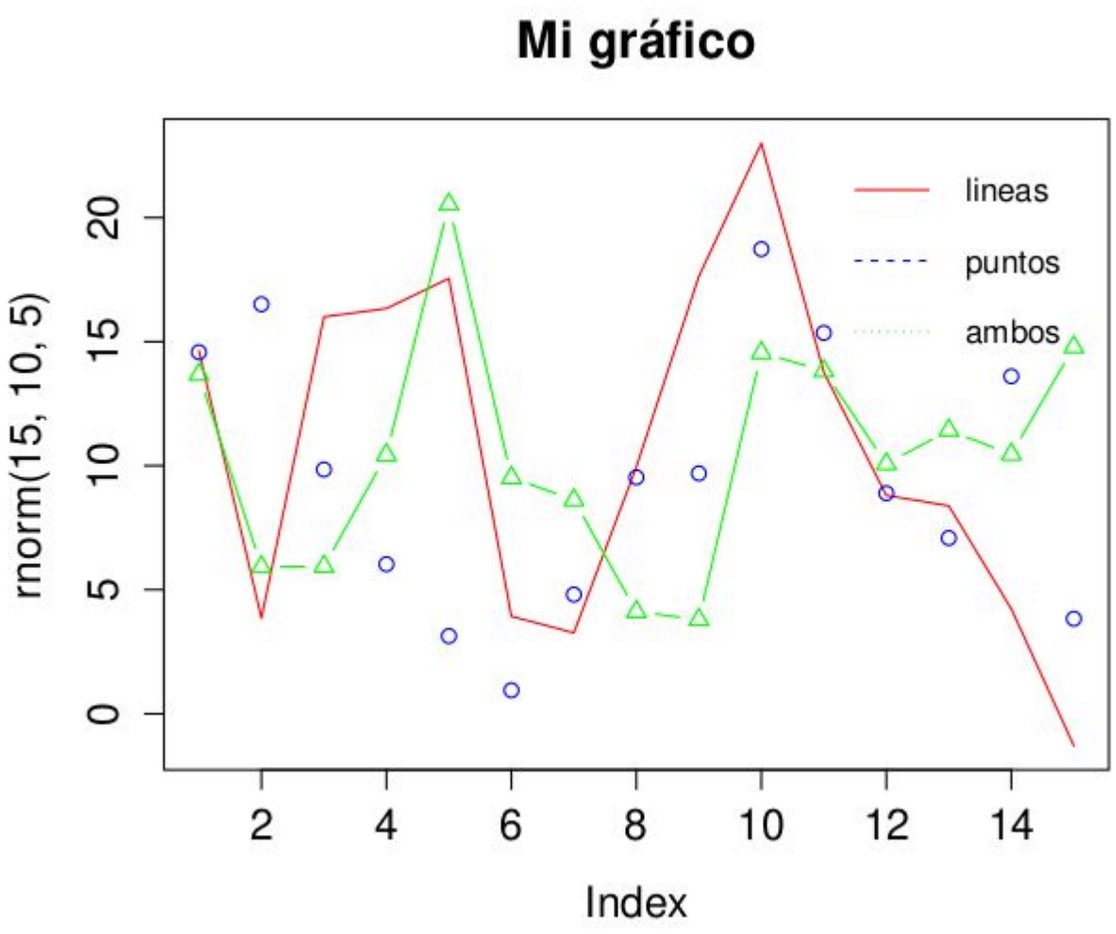
- En R la función de visualización más frecuente es `plot`. Es una función genérica cuyo resultado depende de la naturaleza de las variables usadas.
- A los gráficos les podemos agregar parámetros adicionales como: `main` para el título, `xlab` e `ylab` para el nombre de los ejes.
- Otras propiedades son `col` para definir el color, `type` para definir el tipo de gráfico: (p) para puntos o (l) para líneas.
- Además podemos agregar nuevas capas a un gráfico con el comando `lines`.
- Para grabar una imagen en un archivo podemos usar el botón **export** de Rstudio.
- Para hacerlo de la línea de comandos en R:

```
png("imagen.png")
plot(1:10)
dev.off()
```

Graficando en R

Ejemplo:

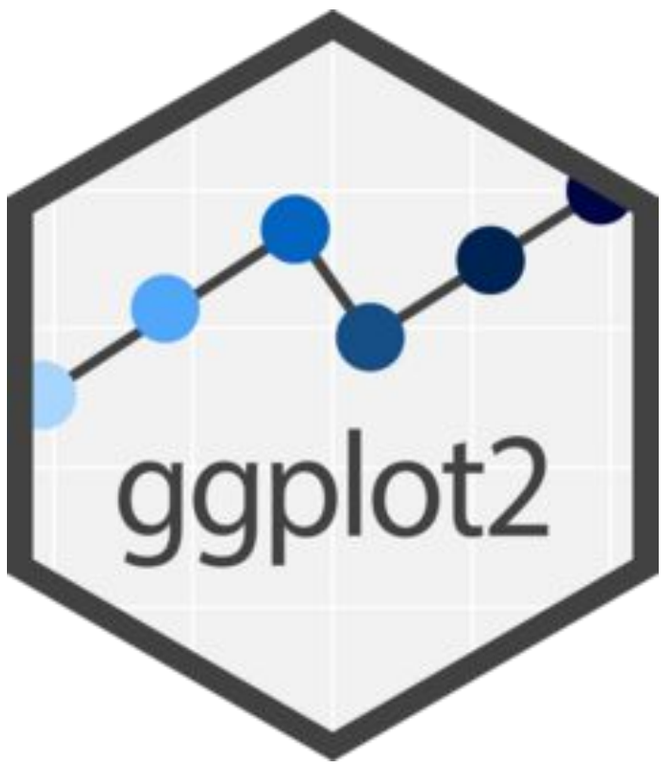
```
plot(rnorm(15,10,5),col="red",type="l")
lines(rnorm(15,10,5),col="blue",type="p",pch=1)
lines(rnorm(15,10,5),col="green",type="b",pch=2)
title(main="Mi gráfico")
legend('topright', c("lineas","puntos","ambos"),
lty=1:3, col=c("red", "blue","green"), bty='n', cex=.75)
```



Graficando en R

Una librería muy popular para hacer visualizaciones en R es **ggplot2**. Se basa en la idea de descomponer el gráfico en componentes semánticos como escalas y capas.

```
> install.packages("ggplot2")
> library(ggplot2)
```



https://ggplot2.tidyverse.org/

Visualizar cantidades

Gráficos de barras

Muestra la relación entre una variable numérica y una categórica. Cada entidad de la variable categórica se representa como una barra. El tamaño de la barra representa su valor numérico. Es por eso que las barras deben empezar en cero, de modo que la longitud de la barra sea proporcional a la cantidad mostrada.

Consideraciones:

- Si las barras representan categorías ordinales, respetar el orden.
- Si las barras representan categorías nominales, ordenarlas (ascendente/descendentemente).

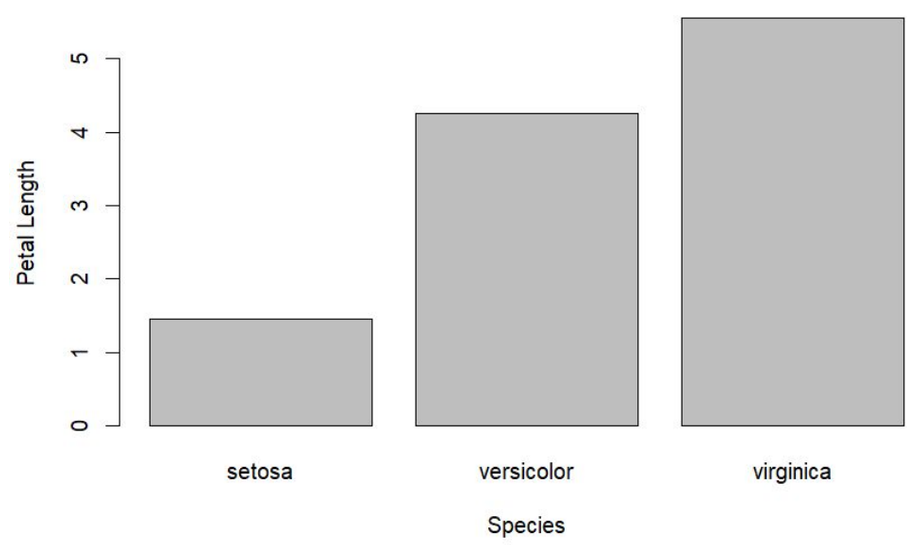
Visualizar cantidades

Gráficos de barras

```
# Calculamos el promedio por especie
df <- aggregate(iris[,1:4], by = list(iris$Species), FUN = mean)
```

```
##      Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa        5.006       3.428        1.462       0.246
## 2 versicolor        5.936       2.770        4.260       1.326
## 3  virginica        6.588       2.974        5.552       2.026
```

```
barplot(Petal.Length~Group.1, data=df, xlab=c('Species'), ylab=c('Petal Length'))
```

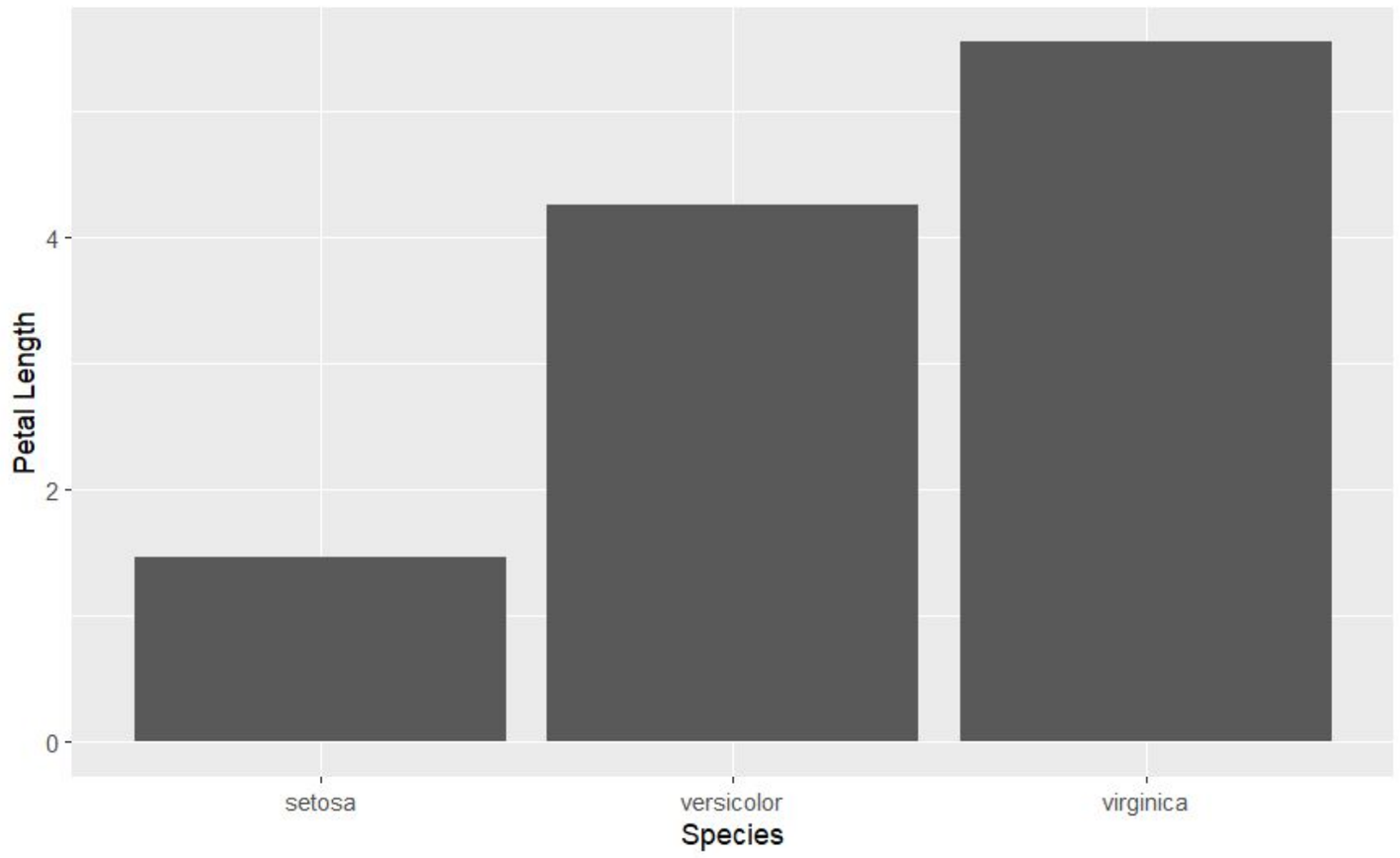


Visualizar cantidades

Gráficos de barras

Lo mismo usando ggplot2:

```
ggplot(df, aes(x = Group.1, y = Petal.Length)) +
  geom_bar(stat="identity") +
  xlab('Species') + ylab('Petal Length')
```



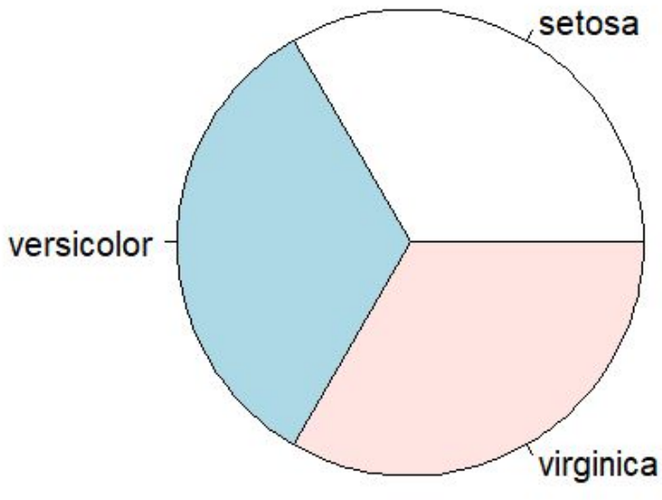
Visualizar proporciones

Tenemos opciones como los **Gráficos de barras, Treemaps, Gráficos de Torta o Pie Charts,** entre otros.

Gráficos de Torta

- Los gráficos de torta, gráficos circulares o pie charts representan la frecuencia de los elementos en un círculo.
- Cada elemento tiene una participación proporcional a su frecuencia relativa.
- Se usan generalmente para variables categóricas.
- No se recomienda mucho su uso, pues pueden entregar información engañosa. Preferible usar gráfico de barras.

```
> pie(table(iris$Species))
```



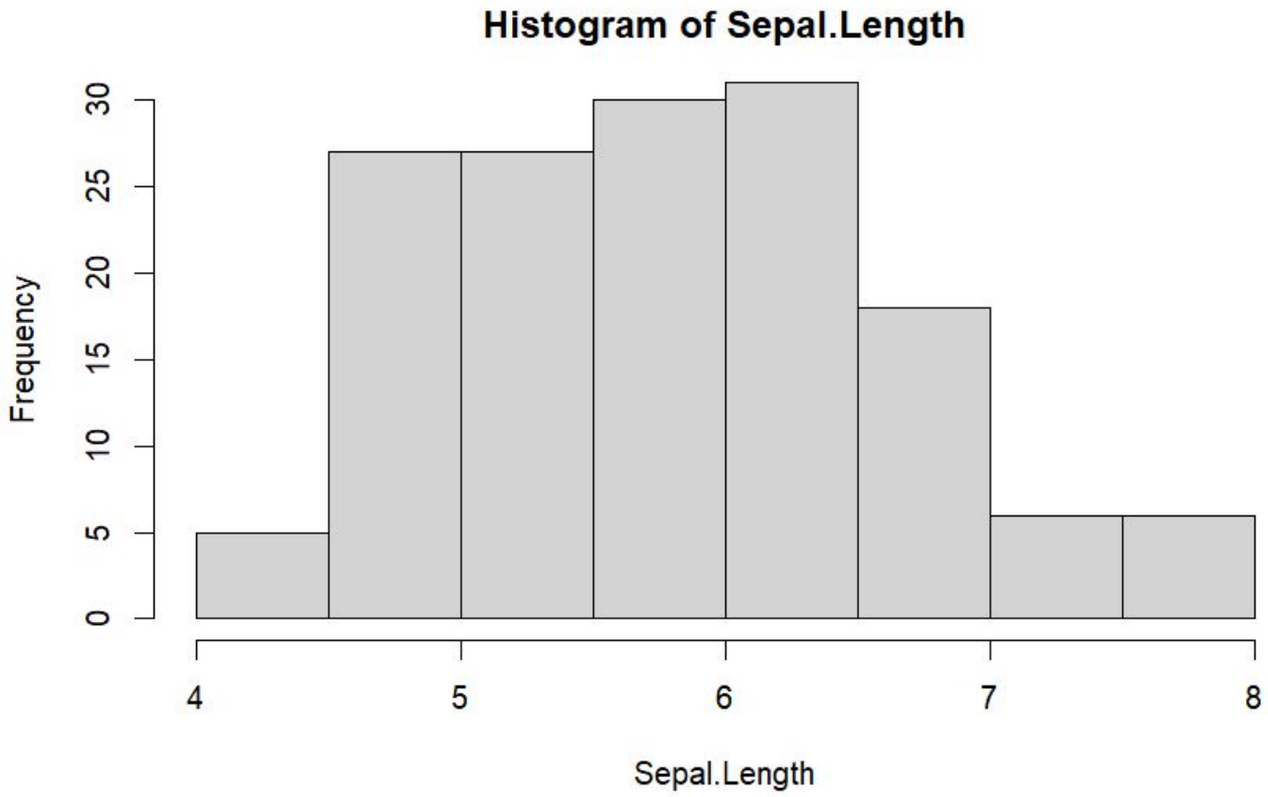
Visualizar distribuciones

Histogramas

Representa la distribución de una o más variables mediante el recuento del número de observaciones que caen dentro de intervalos distintos (contenedores o bins).

- La altura de cada barra indica el número de elementos o frecuencia del contenedor.

```
> hist(Sepal.Length)
```

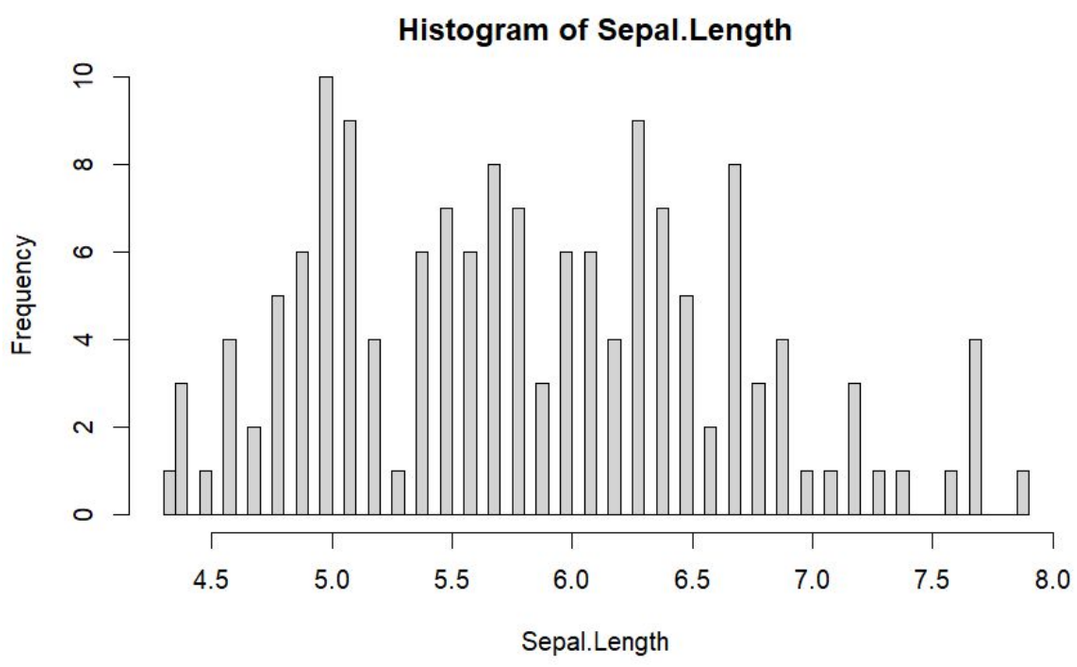


Visualizar distribuciones

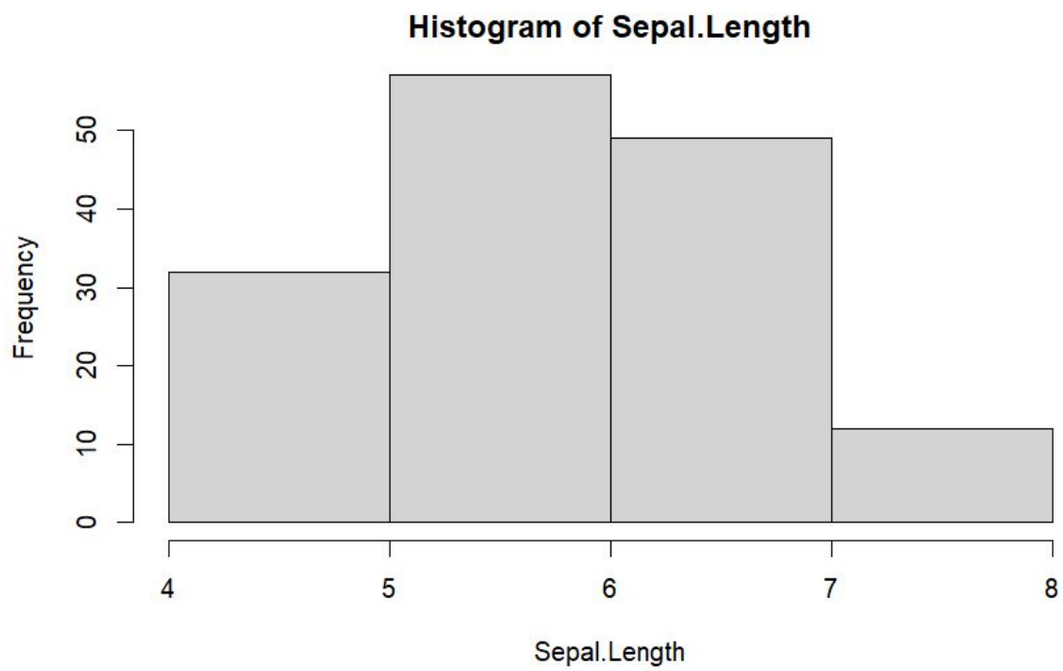
Histogramas

- Es fundamental probar diferentes anchos de intervalo. En R se puede definir esa cantidad con el parámetro `nclass`.
  - Intervalos demasiado grandes, pueden borrar características importantes.
  - Intervalos demasiado pequeños pueden estar dominados por la variabilidad aleatoria, ocultando la forma de la verdadera distribución.

```
> hist(Sepal.Length,nclass=100)
```




```
> hist(Sepal.Length,nclass=4)
```



Visualizar distribuciones

Densidad

Es una versión suavizada del histograma y nos permite determinar más claramente si los datos observados se comportan como una densidad conocida (ej: normal).

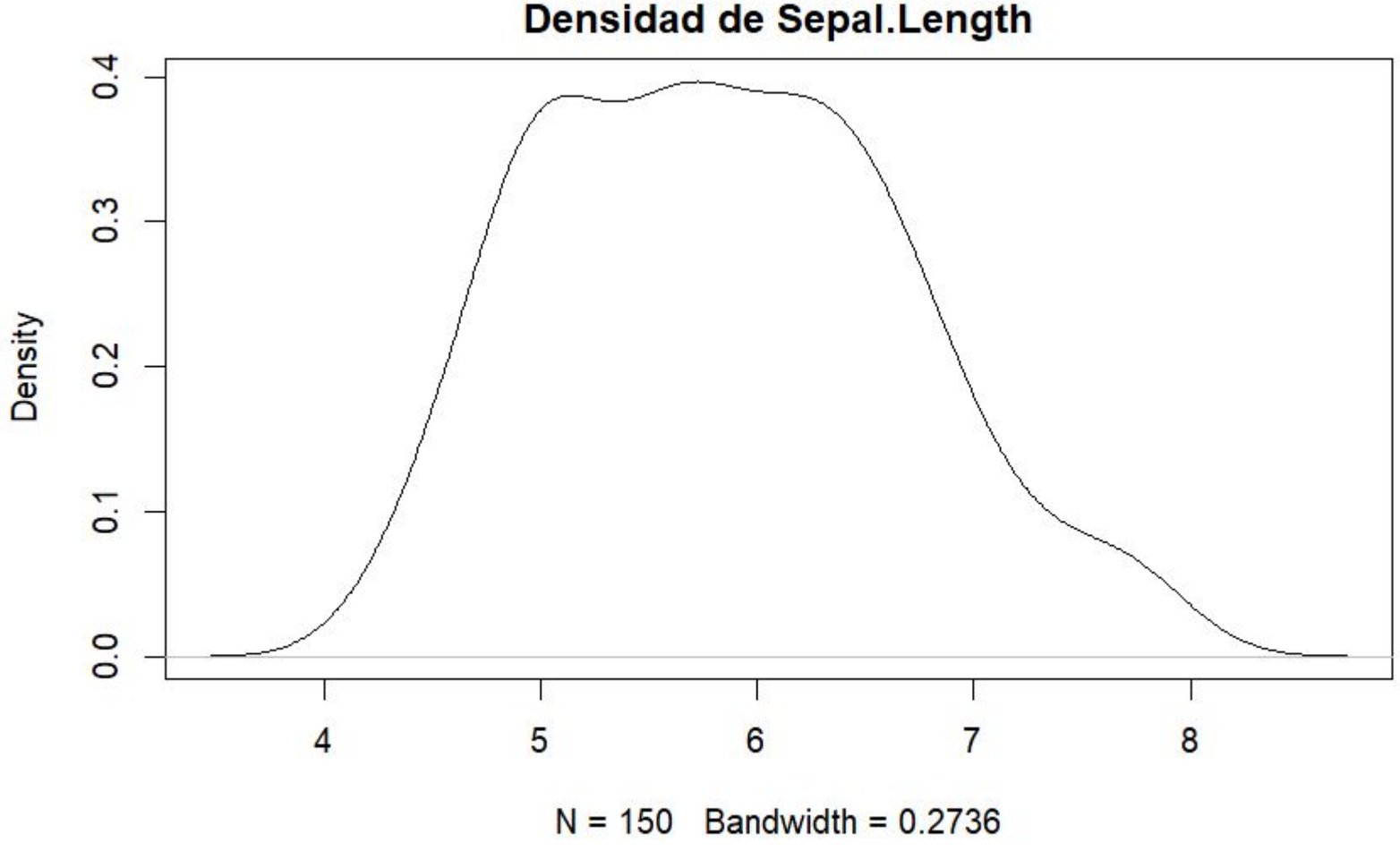
- Se calculan usando técnicas estadísticas no paramétricas llamadas estimación de densidad de **kernel** (ej: gaussiano).
- Las curvas de densidad se suelen escalar de forma que el área bajo la curva sea igual a uno.

En R se crean con el comando `density`, para luego visualizarlas con el comando `plot`.

Visualizar distribuciones

Densidad

```
> plot(density(iris$Sepal.Length), main="Densidad
de Sepal.Length")
```



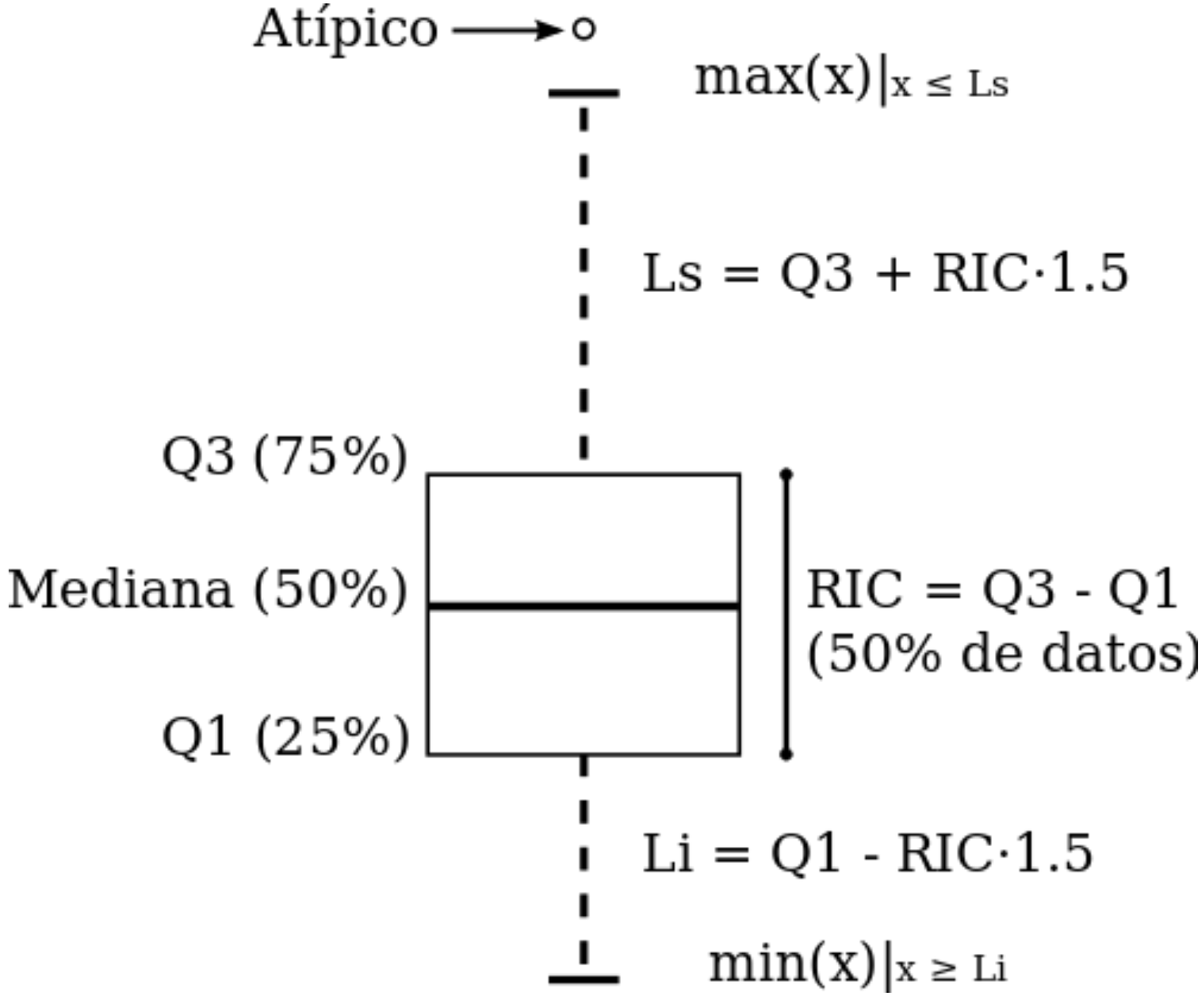
Visualizar distribuciones

Gráficos de cajas (boxplots)

Muestra la distribución de los datos cuantitativos de forma que facilita las comparaciones entre variables o entre niveles de una variable categórica.

- La altura del rectángulo es el rango intercuartil RIC (Q3 – Q1).
- La mediana es una línea que divide el rectángulo.
- Cada extremo del rectángulo se extiende con una recta o brazos de largo Q1 – 1,5*RIC para la recta inferior y Q3 + 1,5*RIC para la recta superior.
- Los valores más extremos que el largo de los brazos son considerados atípicos.
- El boxplot nos entrega información sobre la simetría de la distribución de los datos.
- Si la mediana no está en el centro del rectángulo, la distribución no es simétrica.
- Son útiles para ver la presencia de valores atípicos u outliers.

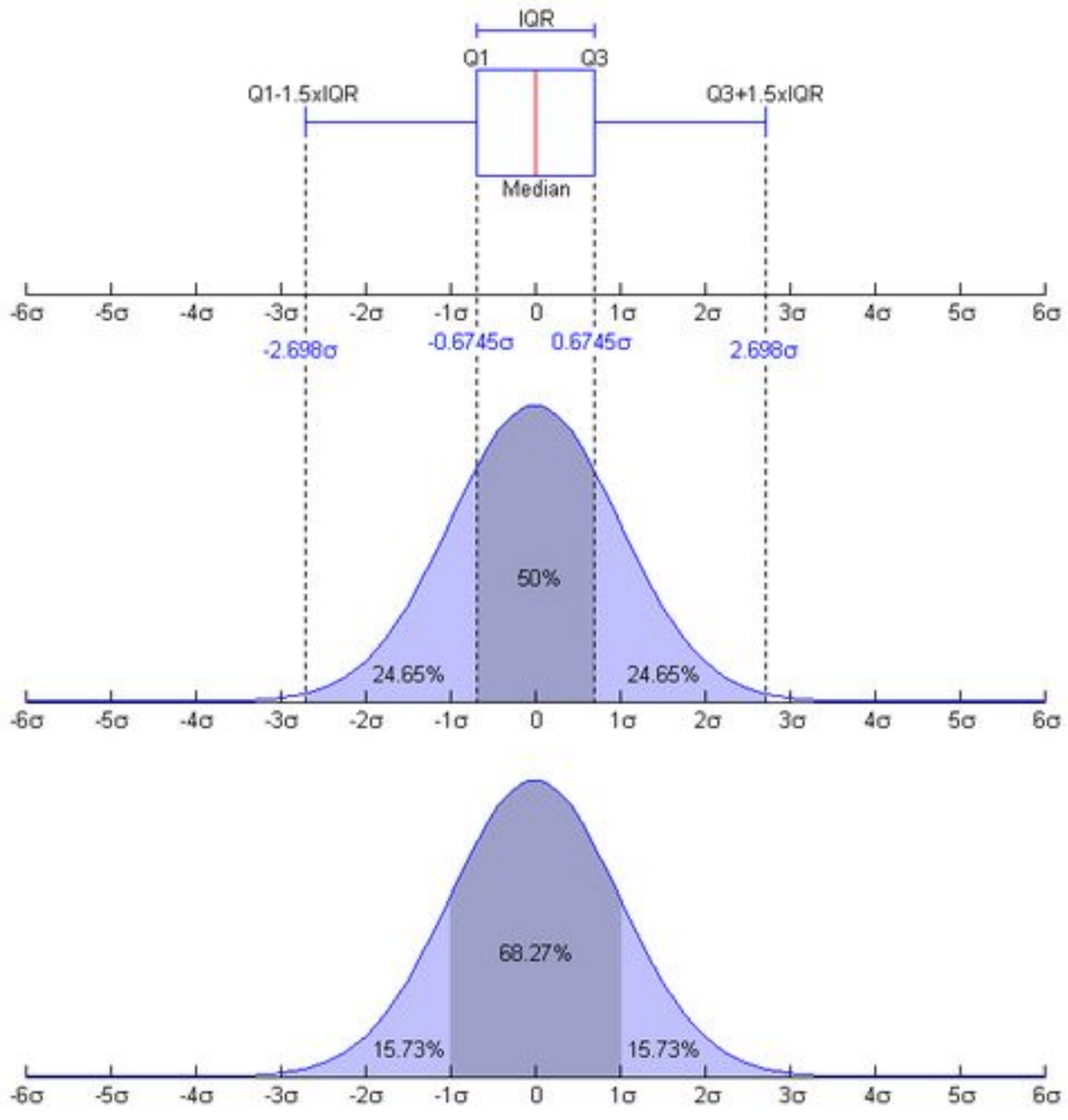
Gráficos de cajas (boxplots)



Fuente:
http://commons.wikimedia.org/wiki/File:Boxplot.svg

Gráficos de cajas (boxplots)

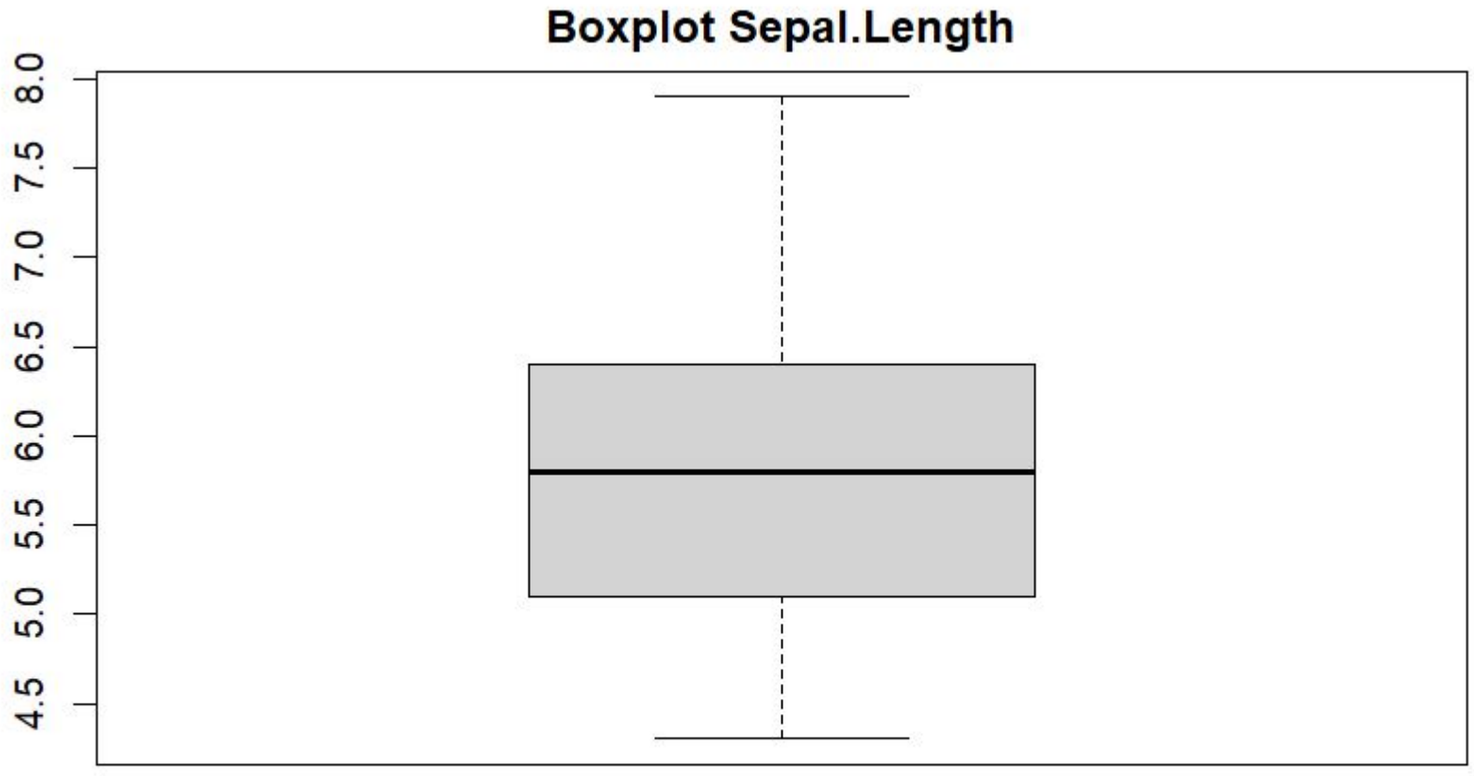
El largo de los brazos así como el criterio para identificar valores atípicos se basa en el comportamiento de una normal.



Visualizar distribuciones

Gráficos de cajas (boxplots)

```
> boxplot(Sepal.Length, main="Boxplot Sepal.Length")
```

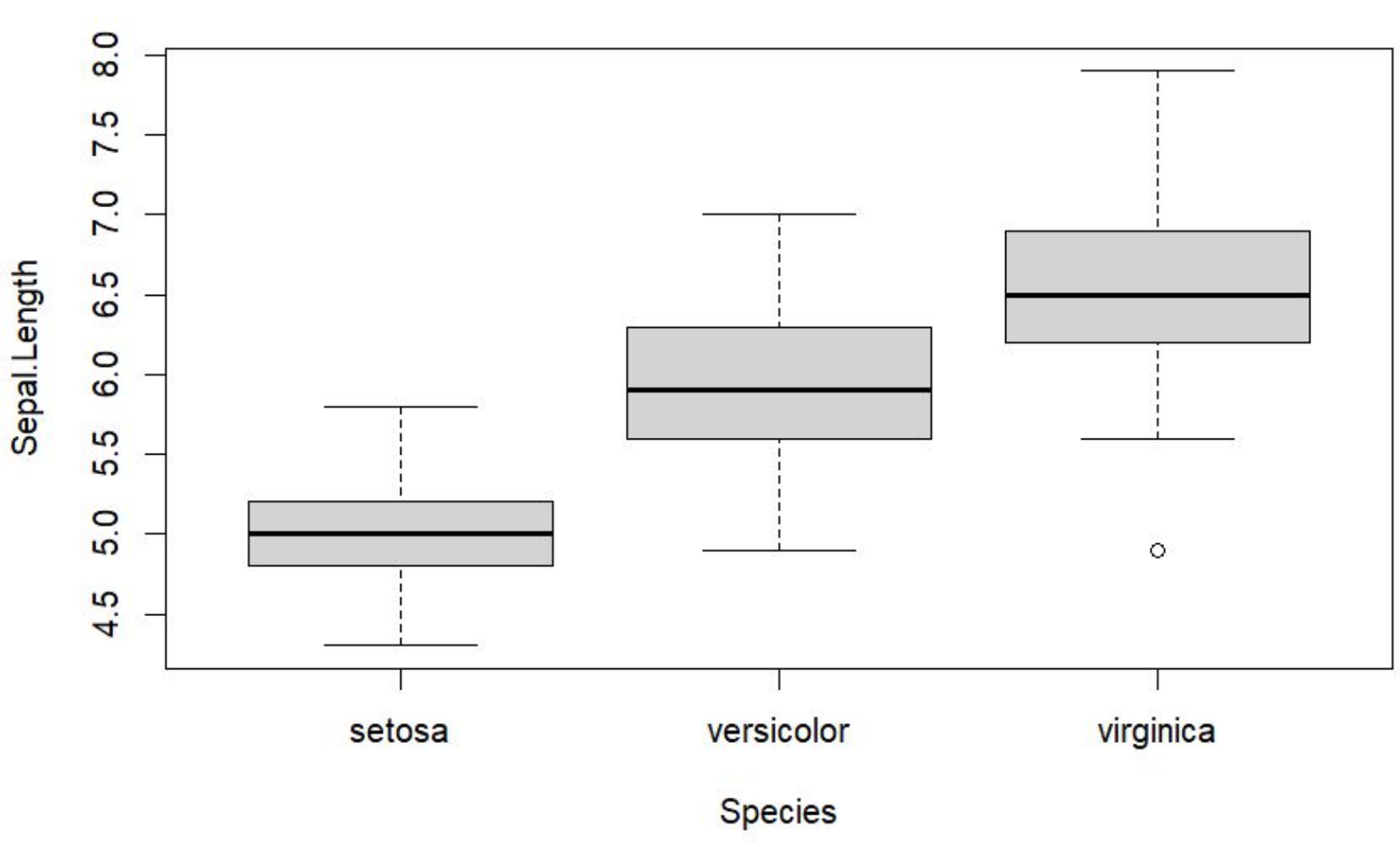


Visualizar distribuciones

Gráficos de cajas (boxplots)

Si tenemos una variable factor podemos crear un boxplot para cada categoría de la siguiente manera:

```
> boxplot(Sepal.Length~Species,ylab="Sepal.Length")
```

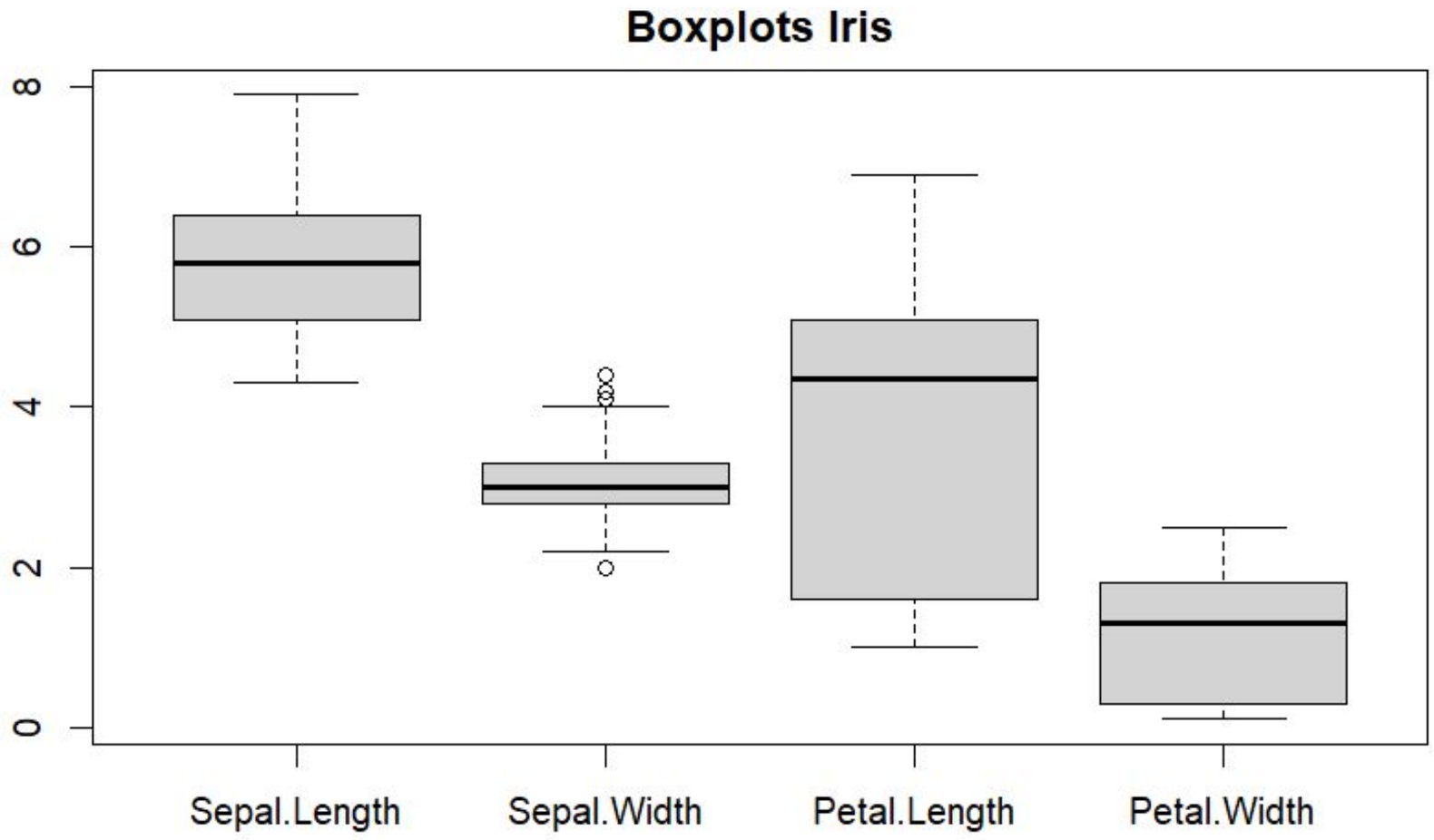


Visualizar distribuciones

Gráficos de cajas (boxplots)

También podemos podemos comparar varios boxplots en un mismo gráfico:

```
> boxplot(x=iris[,1:4], main="Boxplots Iris")
```

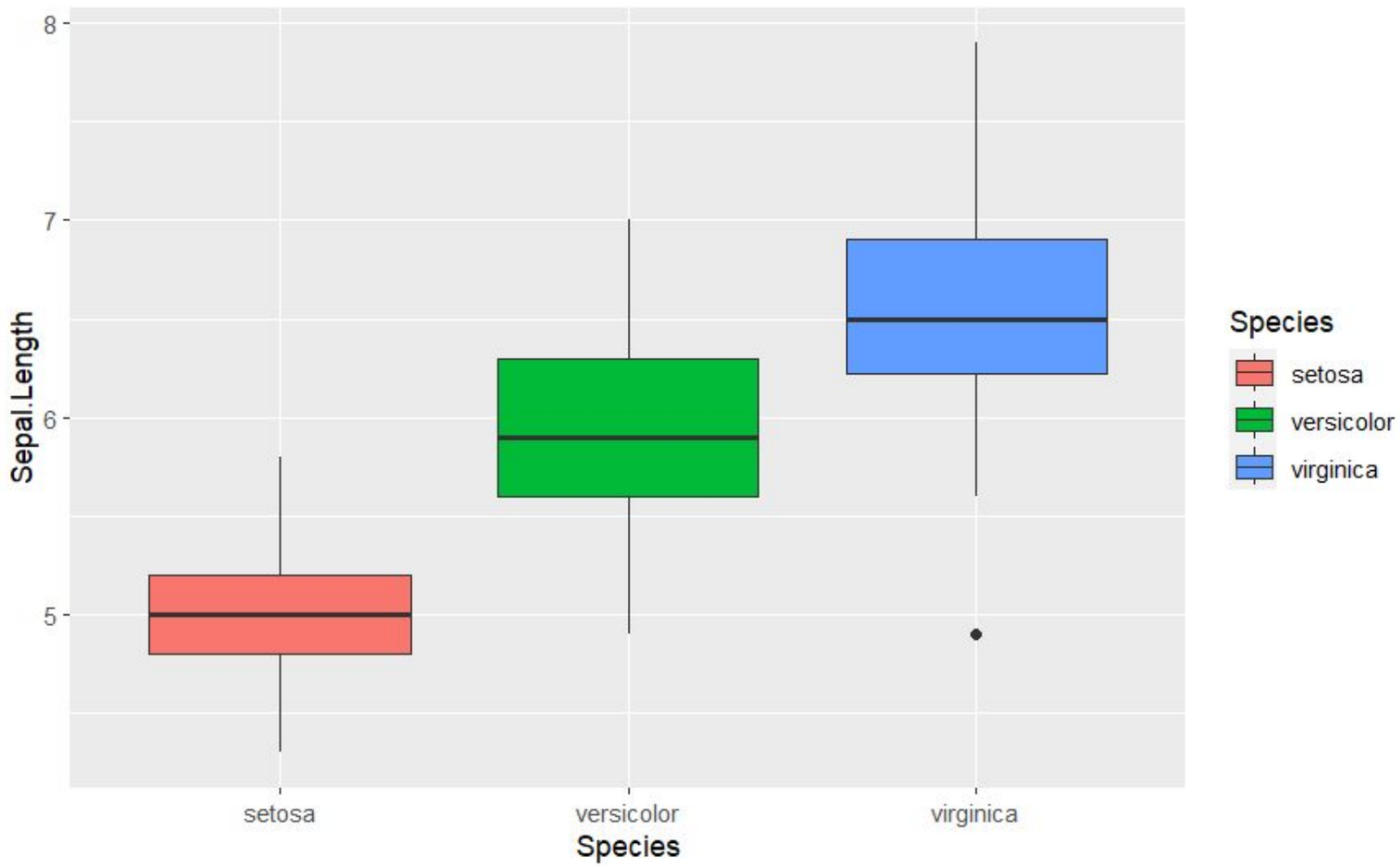


Visualizar distribuciones

Gráficos de cajas (boxplots)

Ahora usando ggplot2:

```
> ggplot(iris, aes(x = Species, y = Sepal.Length,
   fill = Species)) +  geom_boxplot()
```



Visualizar relaciones

Gráficos de dispersión (o scatter plots)

Permiten visualizar las asociaciones entre dos o más variables cuantitativas.

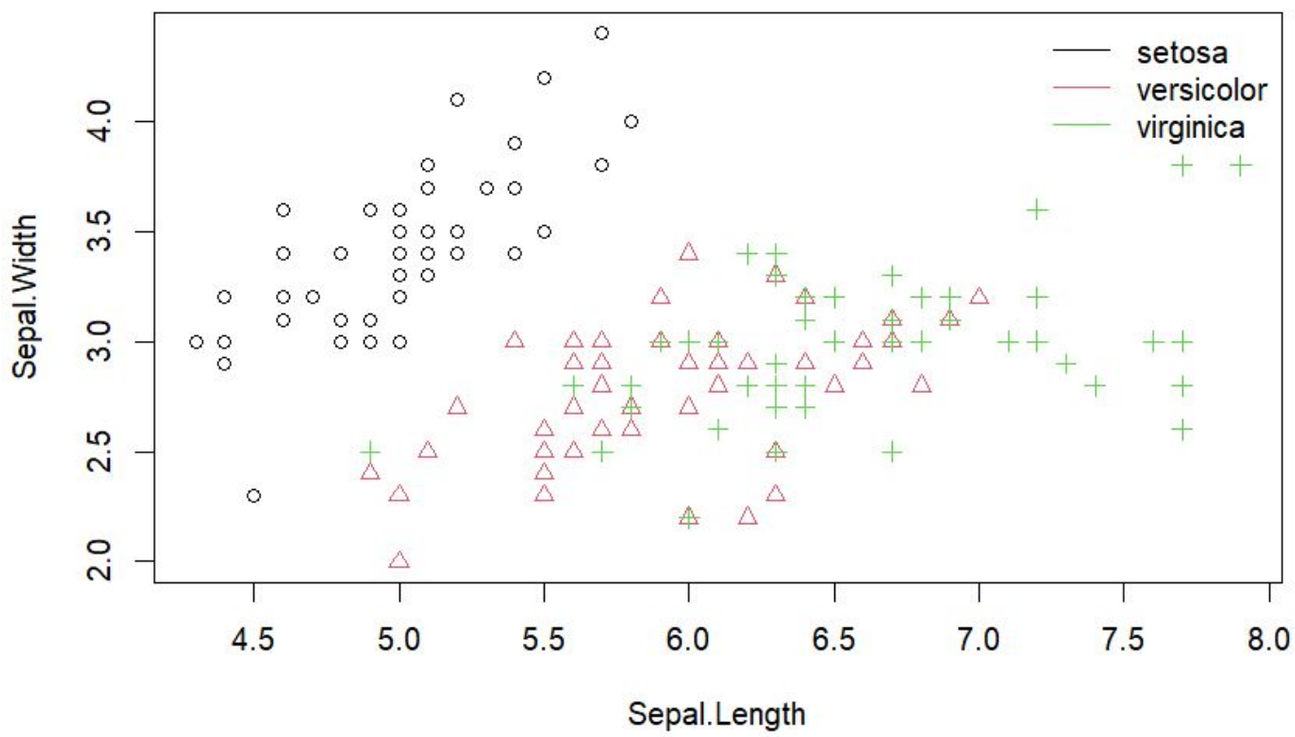
- Usan coordenadas cartesianas para mostrar los valores de dos variables numéricas del mismo largo.
- Los valores de los atributos determinan la posición de los elementos.
- Otros atributos pueden codificarse mediante el tamaño, la forma o el color de los objetos.
- En R podemos graficar un scatterplot de dos variables numéricas usando el comando `plot(x,y)`, que sería `y` vs `x`.
- También se pueden definir fórmulas `f(x)=y` usando la notación `y~x`.
- De esta manera, `plot(y~x)` es equivalente a `plot(x,y)`.
- Si tenemos un data.frame o matriz numérica podemos ver los scatterplots de todos los pares usando el comando `pairs(x)`.

Visualizar relaciones

Gráficos de dispersión (o scatter plots)

```
# El ancho del sépalo vs el largo del sépalo
plot(Sepal.Width~Sepal.Length, col=Species)

# Equivalente
plot(Sepal.Length, Sepal.Width, col=Species,
   pch=as.numeric(Species))
# Le agregamos una leyenda
legend('topright', levels(Species), lty=1, col=1:3, bty='n')
```

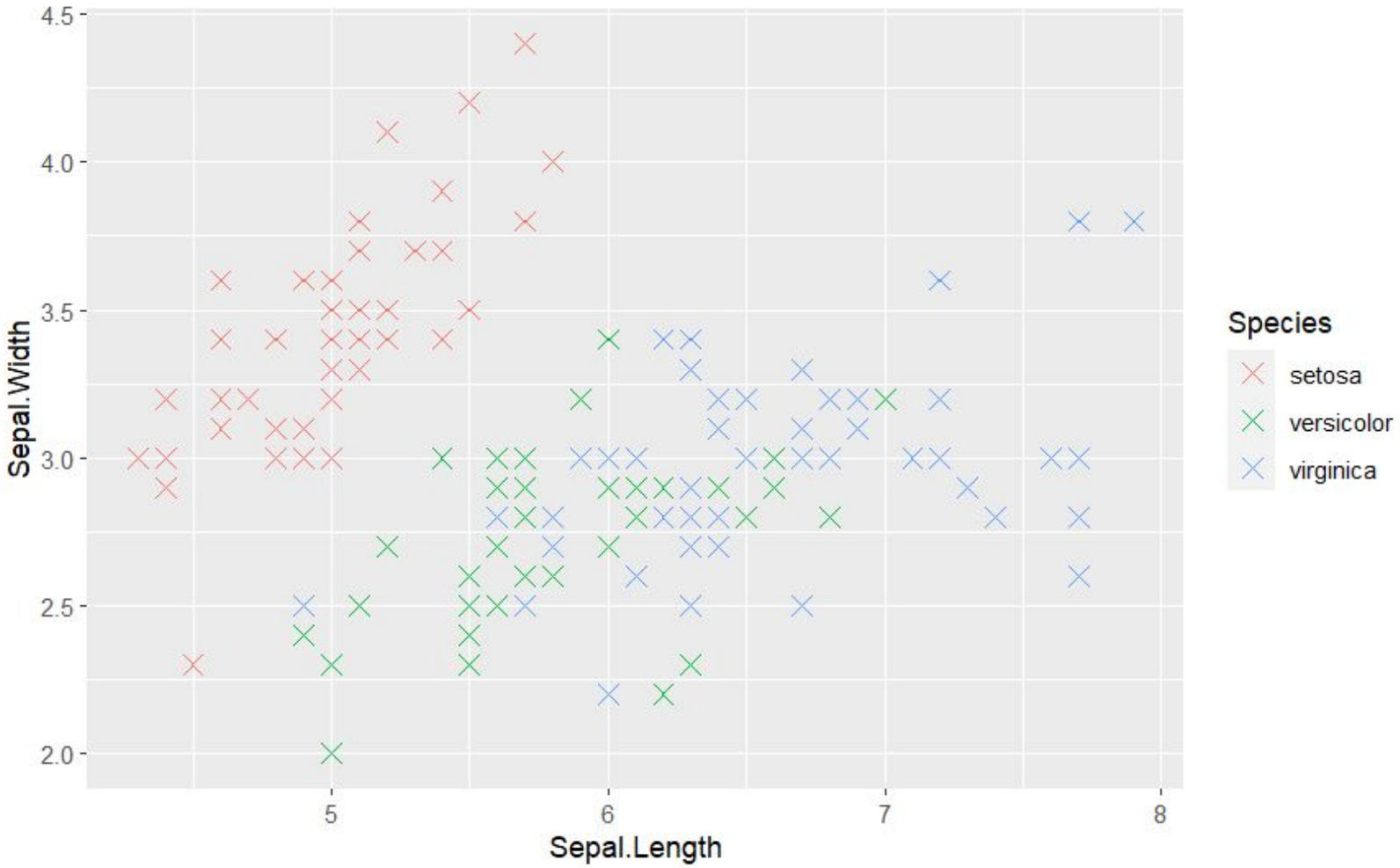


Visualizar relaciones

Gráficos de dispersión (o scatter plots)

Lo mismo usando ggplot2:

```
ggplot(iris, aes(x=Sepal.Length,
y=Sepal.Width, color=Species)) +
geom_point(size=3,shape=4)
```

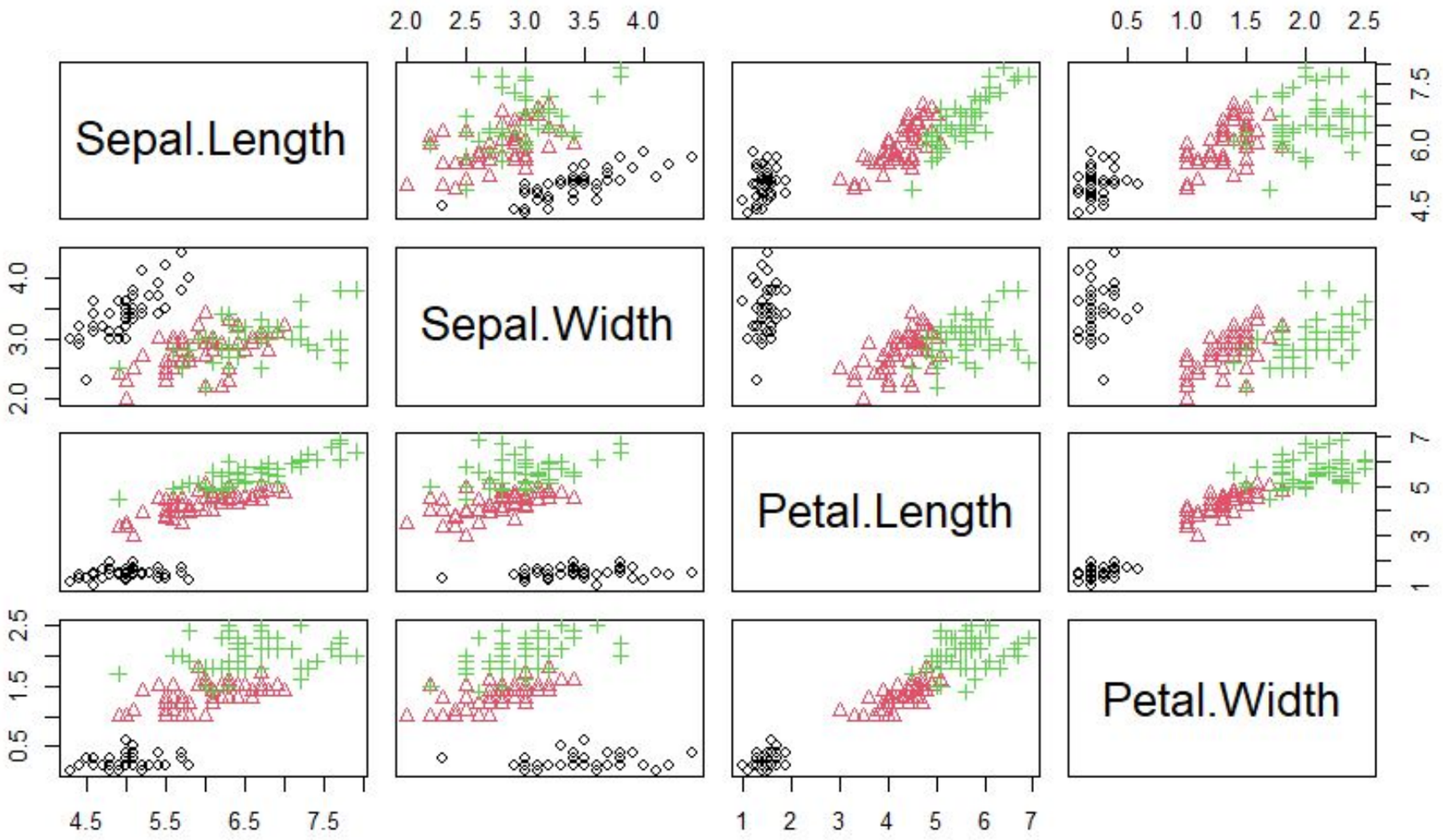


Visualizar relaciones

Gráficos de dispersión (o scatter plots)

Ahora grafiquemos todos los pares de las 4 variables del dataset iris usando un color y un carácter distinto para cada especie:

```
pairs(iris[,1:4],pch=as.numeric(iris$Species),col=iris$Species)
```



Galería de visualizaciones (con código)

- https://r-graph-gallery.com/index.html
- https://plotly.com/r/
- https://www.data-to-viz.com/
- https://datavizcatalogue.com/
- https://datavizproject.com/

dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl

/ DCCUCHILE