

Análisis Exploratorio de Datos en R

Felipe José Bravo Márquez

12 de diciembre de 2019

Análisis Exploratorio de Datos

- El análisis exploratorio de datos o (EDA) engloba un conjunto de técnicas para poder comprender de manera rápida la naturaleza de una colección de datos o **dataset**.
- Fue creado 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.

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:
 - 1 El tipo de flor como variable categórica.
 - 2 El largo y el ancho del pétalo en cm como variables numéricas.
 - 3 El largo y el ancho del sépalo en cm como variables numéricas.



El dataset Iris



Figura: Virginica - Setosa - Versicolor

- El dataset se encuentra disponible en R:

```
> data(iris) # cara cargar el dataset al workspace  
> names(iris)  
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length"  
     "Petal.Width"  "Species"
```

- Para poder acceder a las variables directamente usamos el comando `attach(iris)`.

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.

Frecuencia y Moda

- 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
> vec<-c(1,1,1,0,0,3,3,3,3,2)
> table(vec)
vec
0 1 2 3
2 3 1 4
```

- Ejercicio: Calcular las frecuencias porcentuales del vector anterior.

```
> table(vec)/length(vec) # Frecuencia porcentual
vec
0    1    2    3
0.2 0.3 0.1 0.4
```

Frecuencia y Moda (2)

- 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))  # Elementos con el valor  
  names(valor)  
}  
> my_mode(vec)  
[1] "3"  
> my_mode(iris$Sepal.Length)  
[1] "5"
```

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

Medidas de Tendencia Central

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

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

- El mayor problema de la media es que es una medida muy sensible a **outliers** o valores atípicos.
- Ejemplo: Tomamos un vector aleatorio de media 20 y luego le agregamos un elemento aleatorio que proviene de una distribución de media mucho mayor. Vemos que la media es fuertemente afectada por el ruido:

```
> 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 (2)

- 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 (3)

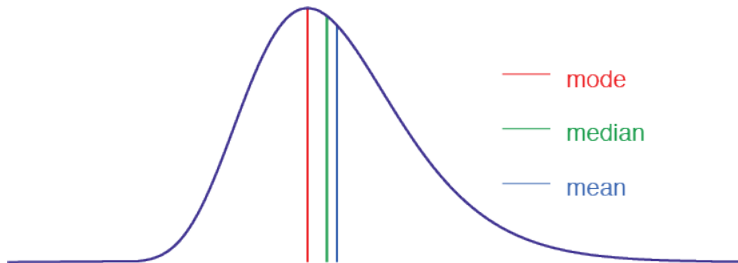
- La mediana representa de posición central de la variable que separa la mitad inferior y la mitad superior de las observaciones.
- Intuitivamente, consiste el valor donde para una mitad de las observaciones todos los valores son mayores que ésta, y para la otra mitad todos son menores.

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

- 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
```

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))
```

- Además es muy común hablar de los **cuartiles** que son tres percentiles específicos:
 - El primer cuartil Q_1 (lower quartile) es el percentil con $k = 25$.
 - El segundo cuartil Q_2 es con $k = 50$ que equivale a la mediana.
 - El tercer cuartil Q_3 (upper quartile) es con $k = 75$.

```
# 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 resumir varias estadísticas de resumen de una variable o de un data.frame 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)
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?

```
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 que tan distintas o similares tienden a ser las observaciones respecto a un valor particular. Generalmente este valor 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
```

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

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

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

```
> var(Sepal.Length)
[1] 0.6856935
> sd(Sepal.Length)
[1] 0.8280661
```

Medidas de Dispersión (2)

- 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 media absoluta (AAD) como:

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

- Programe la función `add` en R, como una función que recibe un vector `x` y una función de media central `fun`. El valor absoluto se calcula con el comando `abs`:

```
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 (3)

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

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

- 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] 0.7
```

- Finalmente, se define el rango inter-cuartil (IQR) como la diferencia entre el tercer y el primer cuartil ($Q_3 - Q_1$).

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

Estadísticas de Resumen Multivariadas

- Para comparar como 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 :

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

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

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

- Si le damos una matriz o un data.frame de variables numéricas, calcula una matriz de covarianzas:

```
> 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 (2)

- 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:

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

- 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). Si la correlación es cercana a cero tenemos independencia lineal. Ojo que eso no implica que no pueda haber una relación no-lineal entre las variables.
- En R se calcula con el comando `cor`.

```
> 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

Tablas de Contingencia

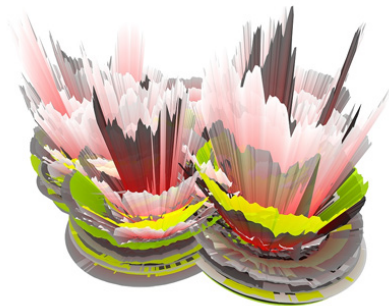
- Para analizar la relación entre variables de naturaleza categórica usamos **tablas de contingencia**.
- La tabla se llena con las frecuencias marginales de todos los pares de valores entre dos variables categóricas.
- En R se crean usando el comando `table` que usábamos para frecuencias, pero entregándole 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 los elementos del dataset.
- La visualización permite que las personas reconozcan patrones o tendencias en base a su criterio o expertiz en el dominio particular.



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.

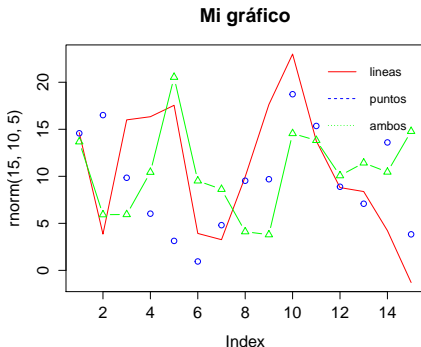
Graficando en R

- 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 todos los gráficos les podemos agregar parámetros adicionales como: `main` para el título, `xlab` e `ylab` para el nombre del eje x y del el eje y.
- Otras propiedades son `col` para definir el color, `type` para definir el tipo de gráfico: (p) para puntos o (l) para líneas.
- Además podemos agregarle 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()
```

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)
```

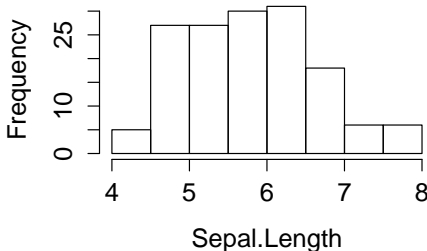


Histogramas

- Muestran la distribución de los valores de una variable.
- Los valores de los elementos se dividen en contenedores (bins) y se crean gráficos de barra por cada contenedor.
- La altura de cada barra indica el número de elementos o frecuencia del contenedor.
- En R se crean con el comando `hist`.

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

Histogram of Sepal.Length

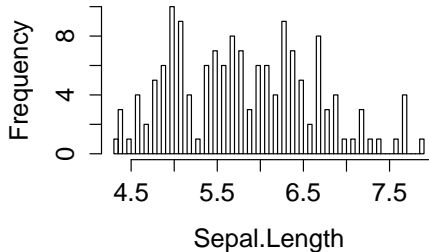


Histogramas (2)

- La forma del histograma depende de el número de contenedores.
- En R se puede definir esa cantidad con el parámetro `nclass`.

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

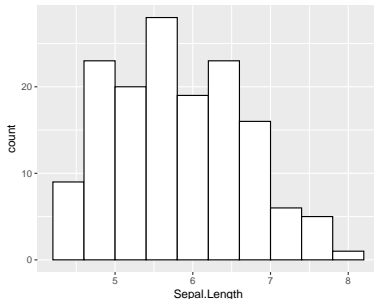
Histogram of Sepal.Length



Histogramas (3)

- 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)  
>ggplot(iris, aes(x=Sepal.Length))  
+ geom_histogram(bins = 10, color="black", fill="white")
```

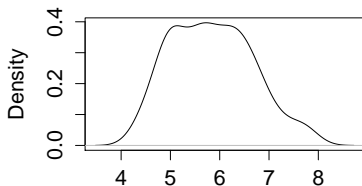


Densidad

- Otra forma de visualizar como se distribuyen los datos es estimando una densidad.
- Se calculan usando técnicas estadísticas no paramétricas llamadas estimación de densidad de **kernel**.
- La 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.
- En R se crean con el comando `density`, para luego visualizarlas con el comando `plot`.

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

Densidad de Sepal.Length

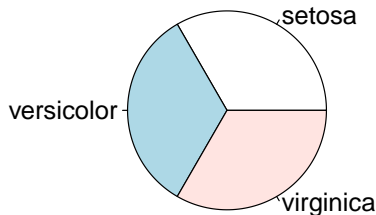


N = 150 Bandwidth = 0.2736

Gráficos de Torta o Pie Charts

- 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:

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



Boxplots

- Los Boxplots o diagramas de caja se construyen a partir de los percentiles.
- Se construye un rectángulo usando entre el primer y el tercer cuartil (Q_1 y Q_3).
- La altura del rectángulo es el rango intercuartil RIC ($Q_3 - Q_1$).
- La mediana es una línea que divide el rectángulo.
- Cada extremo del rectángulo se extiende con una recta o brazos de largo $Q_1 - 1,5 \cdot \text{RIC}$ para la recta inferior y $Q_3 + 1,5 \cdot \text{RIC}$ 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.

Boxplots (2)

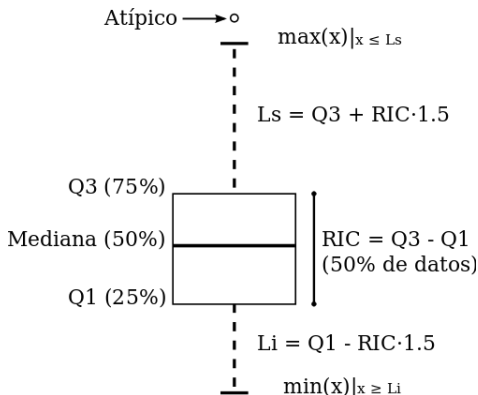
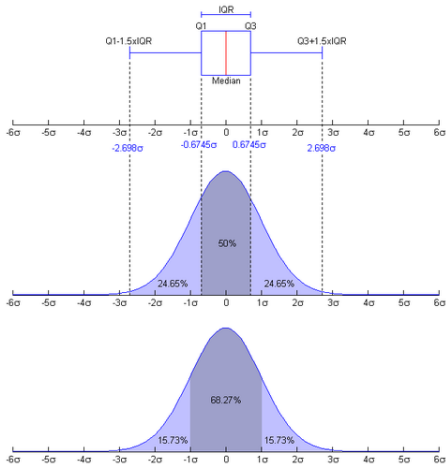


Figura: Fuente:

<http://commons.wikimedia.org/wiki/File:Boxplot.svg>

Boxplots (3)

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

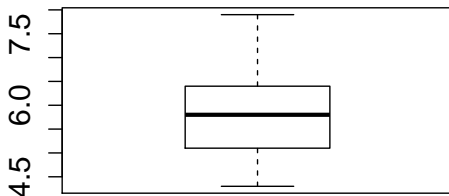


Boxplots (4)

- En R los boxplots se grafican con el comando `boxplot`:

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

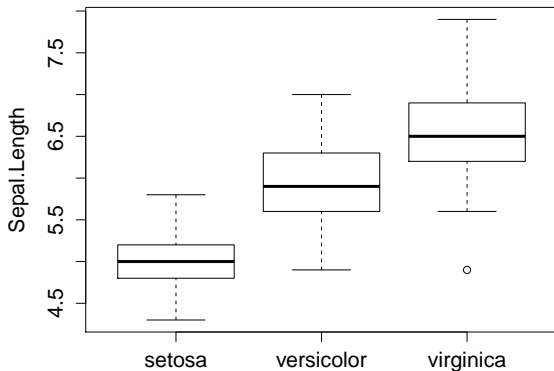
Boxplot Sepal.Length



Boxplots (4)

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

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

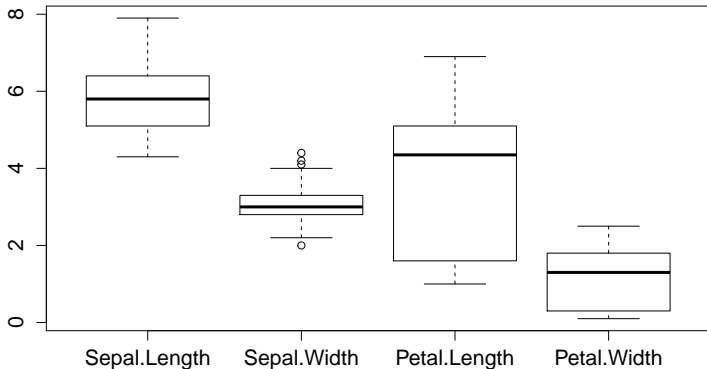


Boxplots (5)

- También podemos comparar varios boxplots en un mismo gráfico:

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

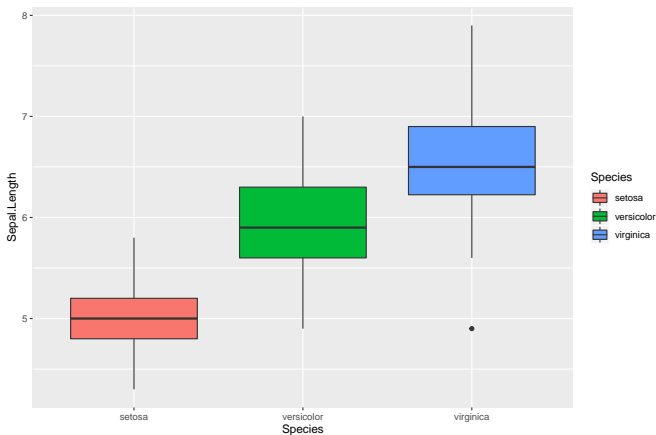
Boxplots Iris



Boxplots (6)

- Ahora usando *ggplot2*:

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



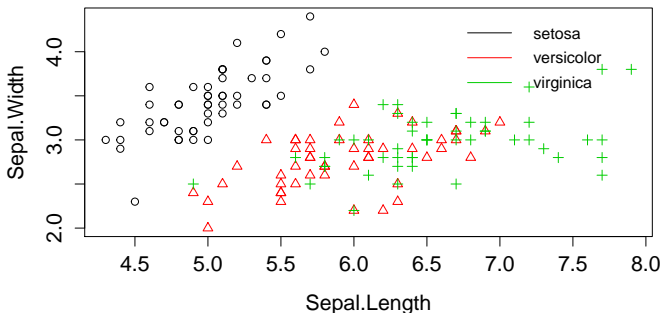
Diagramas de Dispersión

- Los diagramas de dispersión o scatter plots 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 usarse para definir 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 $y \sim x$.
- De esta manera el comando `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)`.

Diagramas de Dispersión (2)

● Ejemplos:

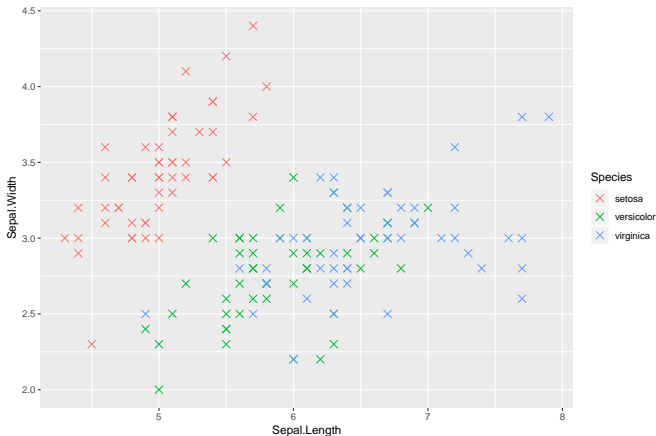
```
# El ancho del sépalos vs el largo del sépalos  
plot(Sepal.Width~Sepal.Length, col=Species)  
# Equivalente  
plot(Sepal.Length, Sepal.Width,col=Species,  
      pch=as.numeric(Species))  
# Le agregamos una leyenda  
legend('topright', levels(Species) ,  
       lty=1, col=1:3, bty='n', cex=.75)
```



Diagramas de Dispersión (3)

- Lo mismo usando *ggplot2*:

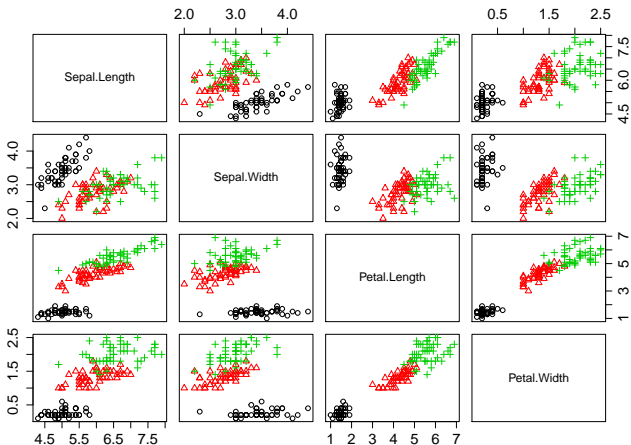
```
ggplot(iris, aes(x=Sepal.Length,  
y=Sepal.Width, color=Species)) +  
geom_point(size=3, shape=4)
```



Diagramas de Dispersión (4)

- 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)
```



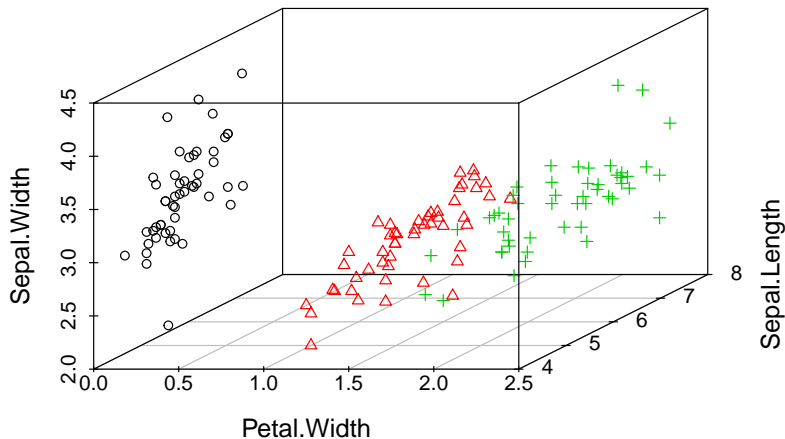
Diagramas de Dispersión (5)

- También se pueden crear scatterplots en tres dimensiones.
- Se debe instalar la librería `scatterplot3d` usando el siguiente comando:

```
install.packages("scatterplot3d", dependencies=T)
```
- Luego cargan la librería escribiendo `library(scatterplot3d)`.
- Un scatterplot 3d para el ancho del pétalo, el largo del sépalo y el ancho del sépalo:

```
scatterplot3d(iris$Petal.Width, iris$Sepal.Length,  
              iris$Sepal.Width, color=as.numeric(iris$Species),  
              pch=as.numeric(iris$Species))
```

Diagramas de Dispersión (6)



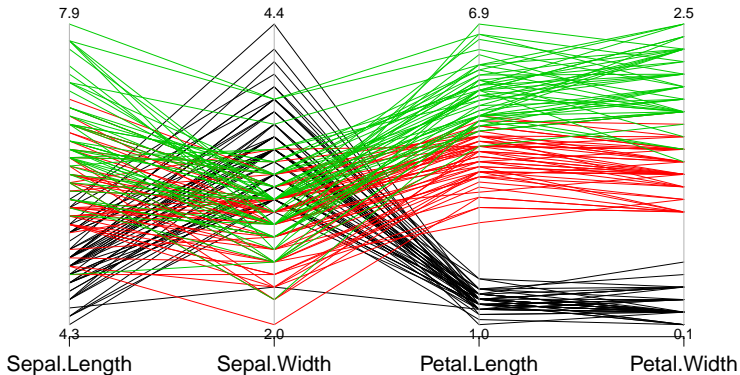
Gráficos de Coordenadas Paralelas

- Los gráficos de coordenadas paralelas son otra forma de visualizar datos multi-dimensionales.
- En vez de usar ejes perpendiculares (x-y-z) usamos varios ejes paralelos entre sí.
- Cada atributo es representado por uno de los ejes paralelo con sus respectivos valores.
- Los valores de los distintos atributos son escalados para que cada eje tenga la misma altura.
- Cada observación representa una línea que une los distintos ejes de acuerdo a sus valores.
- De esta manera, objetos similares entre sí tienden a agruparse en líneas con trayectoria similar.
- En muchas ocasiones es necesario realizar un re-ordenamiento de los ejes para poder visualizar un patrón.

Gráficos de Coordenadas Paralelas (2)

- En R podemos crear gráficos de coordenadas paralelas con el comando `parcoord` de la librería `MASS`.
- Ejemplo:

```
library(MASS)  
parcoord(iris[1:4], col=iris$Species, var.label=T)
```



Otras técnicas de Visualización

Gráficos de Estrellas

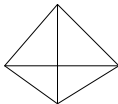
- También conocidos como **gráficos radiales**.
- Cada estrella representa una observación, formando un polígono a partir de cada variables con dirección a las agujas del reloj hasta formar un polígono.
- El tamaño de cada línea respecto al centro de la estrella corresponde al valor re-escalado de la variable.
- Sirve para comparar objetos o detectar valores atípicos.

Caras de Chernoff

- Enfoque creado por Herman Chernoff basado en la capacidad humana para distinguir rostros.
- Cada atributo corresponde a alguna característica de la cara (boca, ojos, nariz, etc..)
- El valor de los atributos determina la apariencia de la característica facial.
- Cada observación es una cara.

Ejemplo Star Plot

```
iris_sample1<-iris[sample(1:dim(iris)[1],size=6,replace=F),]  
rownames(iris_sample1)<-paste(as.character(iris_sample1$Species),1:6)  
stars(iris_sample1[1:4])
```



versicolor 1



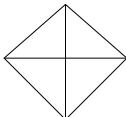
setosa 2



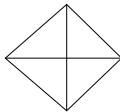
versicolor 3



versicolor 4



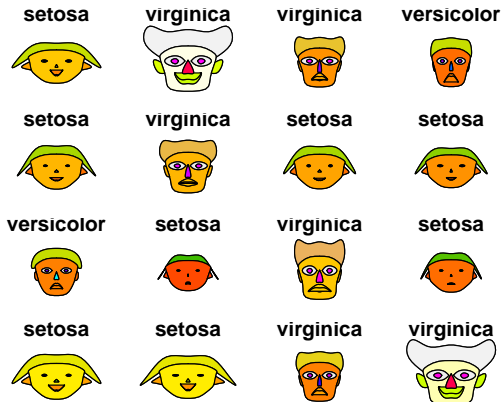
virginica 5



virginica 6

Ejemplo Chernoff Face

```
library("aplpack")  
iris_sample<-iris[sample(1:dim(iris)[1],size=16,replace=F),]  
faces(iris_sample[1:4],face.type=1,labels=iris_sample$Species)
```



Bibliografía I



Venables, William N., David M. Smith, and R Development Core Team. *An introduction to R.*, 2002.



Tan, P. N., Steinbach, M., & Kumar, V. *Introduction to Data Mining*, 2005.



<http://cran.r-project.org/doc/contrib/grafi3.pdf>