Modelos probabilísticos y análisis estadístco / Métodos Estadísticos para Data Analytics

Introducción al uso de R / Análisis descriptivos / Gráficas

Rodrigo Gil Castañeda

rodrigo.gil@utadeo.edu.co

Área de Ciencias Básicas y Modelado
Facultad de Ciencias Naturales e Ingeniería
Universidad de Bogotá Jorge Tadeo Lozano
Módulo 6, oficina 403

Tabla de contenidos

Introducción - ¿Qué y por qué?

R es un lenguaje de programación, basado en S, otro lenguaje de programación estadístico desarrollado por los laboratorios Bell (Bell Labs) desde 1976. S fue desarrollado para apoyar proyectos de investigación y analizar datos de alta complejidad.

Actualemte, S a evolucionado a S-Plus, que requiere la compra de la licencia.

R, es una plataforma libre similar a S desarrollada por **R**obert Gentleman y **R**oss Ihaka (U. de Auckland, NZ) durante la década de los 90s. Desde 1997 se creó un equipo internacional de desarrolladores del núcleo de R.

Introducción- ¿Qué y por qué?

Una de las principales ventajas que tiene R es que los usuarios pueden, de manera fácil, escribir sus propios códigos y personalizar las funciones.

La sintaxis usada por R es considerada por programadores de otros lenguajes como extremadamente fácil de aprender, incluso para usuarios que no tienen experiencia programando.

Una vez que se logra entender y manipular las estructuras básicas de R, este lenguaje se convierte en una herramienta poderosa para manipular y analizar casi cualquier tipo de datos.

Interfaz gráfica de usuario : Rstudio

- Se puede trabajar con varios ficheros de comandos de R (libretos-scripts) simultáneamente, agrupados por pestañas (en la ventana superior izquierda).
- El editor de comandos esta pensado para programar en R.
- Autocompletado de comandos. Si no recordamos la sintaxis exacta de un comando de R, basta con escribir las primeras letras, pulsar la tecla Tab, y RStudio despliega información sobre ese comando. Funciona igual para las variables, ficheros y demás objetos que se hayan creado.
- Gestión de los gráficos de R mucho más eficaz.
- Mayor facilidad para instalar paquetes, consultar ayudas, etcétera.
- Cuando se tienen varias versiones de R instaladas, es fácil seleccionar la que utilizará en cada sesión.

Páginas web recomendadas

- Sitio oficial de R
- Sitio oficial de RStudio
- Introducción a R (español)
- Stack Overflow
- Blog de R
- Datasets for Data Science Projects

Lenguaje de programación orientada a objetos

Objeto: Es toda variable, conjunto de datos, funciones*, resultados y otros, que se guardan en la memoria activa del computador y que tienen asignado un nombre. Los usuarios manipulan los objetos con operadores (aritméticos, lógicos y comparativos) o a través de funciones*.

Objeto:

Nombre <- datos de la variable o información almacenada

*Las funciones son un tipo especial de objetos diseñados para llevar a cabo operaciones. Las funciones emplean argumentos, con base en los cuales se genera un resultado después de ejecutar una o un conjunto de operaciones.

Funciones en R

En R, las funciones realizan un procedimiento en específico con base en unos argumentos, los caules están, generalmente, definidos por defecto pero pueden ser modificados a través de las opciones disponibles.

En R las funciones son creadas por la función: `function()`, y son alamcenadas como objetos en R. En R las funciones tienen la siguiente estructura general:

```
 F <- function(argumento 1, argumento 2, ....)
{comandos}
```

donde `F` es la designación que se le da a la función; `function()` es la instrucción que le indica a R que se esta creando una nueva función; `argumento 1 , argumento 2` son las entradas que se emplearán para realizar los procedimientos, y `comandos` son los procedimientos específicos que desarrolla R con base en los argumentos definidos.

Ejemplo de una función

Programar en R una función para normalizar un conjunto de datos numéricos

$$Normalizado = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

```
normalizar <- function(x){
numerador <- x-min(x)
denominador <- max(x)-min(x)
normalizada <- numerador/denominador
return(normalizada)
}
datos<-c(23,29,30.5,12, 0.5, 67)
normalizar(datos)
```

Características de las funciones en R

En R las funciones son **un objeto de primera clase**, lo cual significa que pueden ser usandas sin restricciones y como cualquier otro objeto. En la práctica esto trae como consecuencia que:

- Una función pueda ser pasada como argumento a otra función.
- Las funciones tienen argumentos con nombres específicos pueden tener valores por defecto.
- Para usar algunas funciones no es necesario espeficicar valores para todos los argumentos.

Características de las funciones en R

La función `formals()` devuelve una lista de todos los argumentos formales de la función. Los argumentos formales son los argumentos incluidos en la definición de la función. Los argumentos de una función en R pueden ser organizados por posición o por nombre. A manera de ejemplo se muestran diferentes maneras de aplicar la función `sd()` que son equivalentes:

Función `sd` - Desviación estándar

- `mydata <- rnorm(100)`
- `sd(mydata)`
- `sd(x=mydata)`
- `sd(x=mydata, na.rm=FALSE)`
- `sd(na.rm=FALSE, x=mydata)`

Características de las funciones en R

A pesar de que cualquiera de las maneras presentadas es correcta, no es recomendable alterar demasiado el orden de los argumentos debido a que se puede generar confusión.

El uso de los nombres de los argumento es útil cuando se emplean funciones que tienen un gran número de argumentos, pero sólo se desea modificar unos pocos y los demás se quieren dejar con los valores por defecto. Además, los argumentos con nombres facilitan el trabajo en R debido a que es más fácil recordar el nombre del argumento que la posición en la cual se encuentra.

Asignar nombre y elementos a un nuevo objeto en R

Asignación de nombres

- Debe iniciar con un letra mayúscula o minúscula de A a Z.
- Puede contener dígitos y para fijar periodos, se suele usar el punto (.). Ejemplo: `Enero.2004`, `Monitoreo.1`, etc.
- R distingue minúsculas de mayúsculas en nombres de objetos (incluidas las funciones) y nombres de argumentos, por ejemplo: `mydata` y `MyData` no son objetos iguales.

Asignación de elementos

- Asignar elementos a un objeto: Posterior al nombre del objeto se usan los símbolos <- (< menor que y - giuon, unidos) para indicar qué elementos se asignarán a ese objeto, así: `x<-1:4`
- La asignación de elementos a un objeto también se puede realizar con la función `assign ()`. así:
  `assign("nombre del objeto", elementos)`

Aritmética vectorial

Sobre los vectores se pueden ejecutar operaciones aritméticas, las cuales se realizarán elemento a elemento. Si dos vectores involucrados en una misma operación no tienen la misma longitud (número de elementos) el resultado se obtendrá reciclando los valores del más corto tantas veces como sea necesario hasta que coincida con el más largo, aunque en este caso R advertirá mediante un mensaje que la longitud de los vectores no es la misma.

- `x<-c(1,3,5,6,8,10,12)`
- `y<-c(5,6,7)`
- `x+y`

Aritmética vectorial

Cuadro: Operadores aritméticos en R

Operador	Descripción
+	Adición
-	Sustracción
*	Multiplicación
/	División
** ó ^	Exponencial
%/%	Resultado entero en una división

Aritmética vectorial

Cuadro: Operadores lógicos en R

Operador	Descripción
$<$	Menor a
$<=$	Menor o igual a
$>$	Mayor a
$>=$	Mayor o igual a
$==$	Exactamente igual a
!	Negación
x | y	x o y, unión (“disyunción”)
x & y	x y y, intersección (“conjunción”)
isTRUE(x)	Probar si x es verdadero

Aritmética vectorial

Cuadro: Algunas funciones comúnmente empleadas en R

Operador	Descripción
`log()`	Logaritmo natural
`exp()`	Valor exponencial, e^x
`sin()`	Seno
`cos()`	Coseno
`tan()`	Tangente
`sqrt()`	Raíz cuadrada
`max()`	Valor máximo
`min()`	Valor mínimo
`range()`	Rango
`length()`	Número de elementos
`sum()`	Sumatoria de los elementos
`prod()`	Productoria de los elementos

Generación de sucesiones

En R existen varias funciones para generar sucesiones numéricas. El operador más básico es : (dos puntos), el cual genera una sucesión de números desde el primer número hasta el valor del segundo número, haciendo saltos de una (1) unidad. Si el primer número es mayor que el segundo, la sucesión generada tendrá orden descendente.

```
x<-1:5
y<-100:90
```

Para crear sucesiones más complejas se emplea la función seq(), en la cual los dos primeros argumentos corresponden al comienzo y el final de la sucesión, si la función se ejecuta con solo estos dos argumentos el resultado será idéntico al que se obtendría usando el operador :

```
x<-seq(1,5)
y<-seq(100,90)
```

Generación de sucesiones

En la función `seq()` los argumentos son: `from=` que corresponde al valor inicial de la sucesión, `to=` que corresponde al valor final, `by=` que especifica el paso o salto de la sucesión, `length.out=` que determina la longitud que tendrá la sucesión; la última es `along.with=` que corresponde a un vector, se emplea como único argumento y crea una sucesión `1, 2, ..., length(vector)`.

```
 x<-seq(0,1.5,0.2)
 r<-c(8,9,3,10,12,0,37,48,50)
 y<-seq(min(r),max(r),(max(r)-min(r))/100)
z<-seq(along.with=r)
```

Una función relacionada con el uso de `seq()` es `rep()`, la cual sirve para duplicar objetos o elementos.

Valores faltantes

Algunas veces en un vector aparecen elementos con valores que no son conocidos, en estos casos se les denomina "valor faltante" y se les asigna un valor especial, `NA` (Not Available). Cuando se opera un vector de contiene elementos con valores faltantes, en general, el resultado será `NA`. Esto se debe a que al no poder especificar de manera completa la operación el resultado no puede ser conocido. Para verificar si algún elemento del vector contiene valores faltantes se emplea la función `is.na()`, la cual evalúa si los elementos que componen el vector corresponden a datos faltantes o no.

```
x<-c(1:3,NA)
ind<-is.na(x)
```

Tenga en cuenta que la expresión lógica `x == NA` es distinta de la función `is.na(x)`, debido a que `NA` no es un valor, sino un indicador de que un elemento no esta disponible.

Hojas de datos

Hojas de datos (data frames) son estructuras similares a una matrix pero cada columna puede ser de un tipo distinto. Las hojas de datos (como las de Excel) son apropiadas para describir "matrices de datos" donde cada fila representa a un individuo y cada columna una variable, que a su vez puede ser numérica o categórica. Las hojas de datos pertenecen a la clase `data.frame`, y pueden entenderse como matrices en las cuales las columnas pueden tener diferentes modos y atributos. La selección sobre hojas de datos, filas y columnas, se puede hacer con la misma estructura de indexación de matrices. Aunque las hojas de datos se pueden crean en R usando la función `data.frame()` lo más común es leer este tipo de objetos desde un archivo usando funciones como `read.table()` o `read.csv()` .

```
 NombreObjeto<-read.table("ruta.archivo.txt", header=TRUE,
sep="", dec=",")
 NombreObjeto<-read.csv("ruta.archivo.csv", header=TRUE,
sep=",", dec=",")
```

¿Y si quiero leer una tabla de wikipedia?

```
library(rvest)
DatainURL <- "http: ..."
temp <- DatainURL%>%
html%>%
html_nodes("table")
Test<-as.data.frame(html_table(temp[3], fill = TRUE))
```

Estadística descriptiva - ¿Qué y por qué?

La construcción de modelos con base en datos experimentales requiere un conocimiento básico de su estructura, de manera que se pongan en evidencia características sobresalientes o inesperadas.

Además permite resumir los datos y tener una primera impresión acerca de su comportamiento general.

El análisis exploratorio, mediante técnicas descriptivas, debe ser la primera etapa en la construcción de un modelo basado en datos experimentales.

Medidas de tendencia central

Media aritmética o promedio aritmético

La media aritmética de un conjunto de observaciones es igual a la suma de los valores de las observaciones dividido en el número de observaciones:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{(Media aritmética de una muestra)}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \text{(Media aritmética de una población)}$$

La media de una población es una cantidad fija, mientras que la media de una muestra no lo es, es decir, las medias obtenidas a partir de diferentes muestras tomadas de una población , generalmente, son diferentes.

Características de la média aritmética

Características

- La media aritmética se expresa en las mismas unidades de medida de los datos originales
- La sumatoria de las diferencias entre la media aritmética y cada uno de los datos es cero.
- La precisión de la media dependerá de la representatividad de la muestra
- La media de una muestra tiende, en general, a tener valores diferentes cuando se calcula con diferentes muestras de un mismo tamaño y de la mismas población.
- La medía aritmética se ve afectada por valores extremos dentro del conjunto de datos

Características de la média aritmética

Ejemplo para verificar las características

Se creará un población ficticia de 1000000 de individuos, hombres varones que habitan un determinado departamento del país. La variable que se midió a la población fue la estatura en centímetros y se determinó que la media poblacional tiene un valor de 180, y que los valores se distribuye de forma acampanada (distribución normal).

La primera caractaréstica que se verificará será: **La precisión de la media dependerá de la representatividad de la muestra**. Tomaremos tres muestras de los siguientes tamaños: $n_1 = 10000$, $n_2 = 100$ y $n_3 = 10$, calcularemos la media para cada una de esas muestras y la compararemos con respecto a la media de la población que es conocidas.

Características de la média aritmética

Ejemplo para verificar las características de la media en R

```
N<-1000000; population <- rnorm(N, 180)
plot(density(population, na.rm = T))
abline(v = mean(population), lwd=1,col="black",lty=2)
n1<-10000; n2<-100; n3<-10
X1 <- sample(population, n1)
X2 <- sample(population, n2)
X3 <- sample(population, n3)
abline(v = mean(X1), lty = 1, col="green")
abline(v = mean(X2), lty = 1, col="blue")
abline(v = mean(X3), lty = 1, col=red")
legend("topright", c("Media Poblacional","Media
n=10000","Media n=100","Media n=10"), cex=0.6, lty =
c(2,rep(1,3)),col=c("black","green","blue","red"))
```

Características de la média aritmética

Ejemplo para verificar las características

Ahora, se verificará la característica con relación a: **La media de una muestra tiende, en general, a tener valores diferentes cuando se calcula con diferentes muestras de un mismo tamaño y de la mismas población**. Tomaremos los mismos tamaños de muestras ($n_1 = 10000$, $n_2 = 100$ y $n_3 = 10$), pero esta vez se seleccionarán 500 muestras para cada tamaño, se calculará la media para cada muestra y se comparará con respecto a la media de la población.

Características de la média aritmética

Ejemplo para verificar las características de la media en R

```
plot(density(population, na.rm = T))
abline(v = mean(population), lwd=1,col='black',lty=2)
n1<-10000; n2<-100; n3<-10; sim<-500
for (j in 1:sim){
    X1 <- sample(population, n1)
    X2 <- sample(population, n2)
    X3 <- sample(population, n3)
    abline(v = mean(X1), lty = 1, col="grey40")
    abline(v = mean(X2), lty = 1, col="grey60")
    abline(v = mean(X3), lty = 1, col="grey80") }
legend("topright", c("Media Poblacional","Medias
n=10000","Medias n=100","Medias n=10"), cex=0.6, lty =
c(2,rep(1,3)),col=c("black","grey40","grey60","grey80"))
```

Medidas de tendencia central

Función `apply`

Devuelve un vector, matriz o lista de valores obtenidos al aplicar una función a los márgenes de una matriz o una hojas de datos.La función `apply` cuenta con tres argumentos:

- `X`: Matrix o hojas de datos
- `MARGIN`: Un vector, si es 1 la función se aplica sobre las filas, si es 2 se aplica sobre las columnas y si es `c(1, 2)` sobre ambas.
- `FUN`: Las función que se desea aplicar

Ejemplo para verificar las características

Finalmente, se verificará la característica con relación a: **La medía aritmética se ve afectada por valores extremos dentro del conjunto de datos**. Se insertarán a la población 100000 datos de estaturas con valores que corresponderán a una secuencia desde 180 hasta 220 cm, se verificará el desplazamiento de la media .

Caracteristicas de la média aritmética

Ejemplo para verificar las características de la media en R

- `N<-1000000; inusuales<-100000`
- `population <- c(rnorm(N, 180), seq(180,220,length.out= inusuales))`
- `plot(density(population, na.rm = T))`
- `abline(v = mean(population), lwd=1,col="black",lty=2)`
- `n1<-10000; n2<-100; n3<-10`
- `X1 <- sample(population, n1)`
- `X2 <- sample(population, n2)`
- `X3 <- sample(population, n3)`
- `abline(v = mean(X1), lty = 1, col="green")`
- `abline(v = mean(X2), lty = 1, col="blue")`
- `abline(v = mean(X3), lty = 1, col="red")`

Medidas de tendencia central

Mediana

La mediana es el valor de la variable que se encuentra en la posición central en un conjunto de datos ordenados. Por esta razón el 50 % de los datos tendrá un valor menor que la mediana y el restante 50 % un valor mayor. Cuando se dispone de un conjunto de datos de una muestra ordenada en orden creciente $X = \{x_{(1)}, x_{(2)}, x_{(3)}, \ldots, x_{(n)}\}$, la mediana M_e se cálcula de dos maneras dependiendo de si el número de observaciones es par o impar.

$$M_e = x_{(\frac{n+1}{2})} \qquad \text{(Mediana cuando n es impar)}$$

$$M_e = \frac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2} \qquad \text{(Mediana cuando n es par)}$$

La propiedad más importante de la mediana es que no se ve afectada por los valores extremos como si sucede con el promedio.

Ejemplo de la mediana en R

Ejemplo para verificar la característica

Usando la misma población ficticia que incluyó los datos inusualmente altos se calculará la media y la mediana de la población anteriormente definida, verifique que la mediana se ve menos influenciada por datos inusuales.

Ejemplo para verificar las características de la mediana en R

- `N<-1000000; inusuales<-100000`
- `population <- c(rnorm(N, 180), seq(180,220,length.out= inusuales))`
- `plot(density(population, na.rm = T))`
- `abline(v = median(X1), lty = 1, col="blue")`
- `abline(v = mean(X1), lty = 1, col="green")`

Ejemplo de la mediana en R

Ejemplo del cálculo de la mediana

Usando los datos de clase anterior (DatosEjercicios) cálcule la mediana de las variables cuantitativas.

Cálculo de la mediana en R

- `cuantitativas<-datos[,c(2,3,5,7)]`
- `medianas<-apply(cuantitativas,2,median)`

Ejemplo de la mediana en R

Ejemplo del cálculo de la mediana

Ahora usando los mismos datos (DatosEjercicios) cálcule la mediana para la variable cuantitativa *WormDesnity* pero separando por la variable cualitativa *Damp*.

En R

- `medianas.2<-tapply(datos[,7],datos[,6],median)`

Función `tapply`

Aplica una función a subconjutos de diferentes longitudes; los subconjuntos son determinados por los niveles de ciertos factores (típicamente variables categóricas).La función `tapply` cuenta con tres argumentos:

- `X`: Matrix o hojas de datos
- `INDEX`: Vector, de la misma longitud que X, de uno o más factores.
- `FUN`: Las función que se desea aplicar

Medidas de tendencia central

Moda

La moda es el valor que más se repite dentro de un conjunto de datos, puede no existir cuando todos los valores son diferentes o tienen la misma frecuencia.

Ejercicios en R

Calcule la moda para la variable *WormDesnity*

- `moda.WormDesnity<- table(datos[,7])`
- `moda.WormDesnity[moda.WormDesnity == max(moda.WormDesnity)]`
- `names(moda.WormDesnity)[moda.WormDesnity == max(moda.WormDesnity)]`

Medidas de dispersión

Las medidas de dispersión cuantifican la separación, la dispersión, la variabilidad de los valores de una muestra; lo más común es que lo realicen con respecto a la media u otra medida de tendecia central.

Rango

El rango R es la diferencia entre el máximo $x_{(n)}$ y el mínimo valor $x_{(1)}$de un conjunto de datos ordenados.

Ejercicio en R

Calcule los rangos de las variables cuantitativas de hoja de datos *DatosEjercicios*.

- `rangos<-apply(cuantitativas,2,range)`

El rango se interpreta de manera simple, así: la diferencia entre el valor mínimo y el máximo es de ...

Medidas de dispersión

Varianza

La varianza es quizá la medida de variabilidad más importante en el análisis estadístico. Entre más grande es la variabilidad de los datos, mas grande será la incertidumbre de los valores de los parámetros estimados a partir de ellos, y menor será la capacidad de distinguir diferencias entre conjuntos de datos contrastantes.

La varianza de una muestra es determinada como una función de: *la suma de los cuadrados de las diferencias entre los datos y la media arítmetica*. Esta operación se conoce como suma de cuadrados.

$$SS = \sum_{i=1}^{n} (x_i - \overline{x})^2 \qquad \text{(Suma de cuadrados)}$$

Naturalmente, esta cantidad se incrementará cada vez que se introduzcan nuevos datos. Una manera obvia de resolver este problema es dividir entre el número de datos, como en el promedio, pero ....

Medidas de dispersión

. . . para poder calcular la suma de cuadrados se necesita conocer la media arítmetica. Acá se introduce el concepto de grados de libertad, el esta definido con la siguiente expresión :

$$d.f. = n - k \qquad \text{(Grados de libertad)}$$

donde, n es el tamaño de la muestra y k es el número de parámetros estimados a partir de los datos. Para la varianza se usa un parámetro estimado a partir de los datos, la media $\overline{x}$; así que los grados de libertad para el cálculo de la varianza serán $n - 1$. La varianza se calculará así:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1} \qquad \text{(Varianza)}$$

Medidas de dispersión

Cálculo de la varianza

Cálculo de la varianza para las variables cuantitativas del archivo *DatosEjercicios*.

En R

- `varianzas<-apply(cuantitativas,2,var)`

Características de la varianza

- La varianza toma valores positivos o cero. ¿En qué casos será cero?
- La varianza, es sensible a la presencia de datos con los valores extremos.
- Si no es posible determinar la media aritmética tampoco será posible hallar la varianza
- La varianza no viene expresada en las mismas unidades que los datos

Medidas de dispersión

Desviación estándar

La desviación estándar s es una medida de dispersión cálculada a partir de la varianza que se caracteriza por tener las mismas unidades de la variable original.

$$s = \sqrt[2]{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \qquad \text{(Desviación estándar)}$$

Características de la desviación estándar

- La desviación estándar, al igual que la media y la varianza, es sensible a la presencia de datos con valores inusuales.
- Cuanto más pequeña sea la desviación estándar mayor será concentración de datos alrededor de la media.

Medidas de dispersión

Cálculo y visualización de la desviación estándar en R

- `desv.est<-apply(cuantitativas,2,sd)`
- `medias<-apply(cuantitativas,2,mean)`
- `x<-seq(along.with=medias)`
- `plot(x,medias, xaxt="n", ylab="Valores", xlab="",ylim=c(-2,8), las=2, pch=15)`
- `axis(1,at=x, labels=names(medias))`
- `arrows(x, medias-desv.est,x, medias+desv.est, code=3, angle=90,length=0.2)`

Medidas de dispersión

Coeficiente de variación

El coeficiente de variación se emplea para comparar la variabilidad relativa entre grupos que tienen distintas (o las mismas) unidades, datos que tienen medias diferentes o que pertenecen a categorias diferentes.

$$c.v. = \frac{s}{\overline{x}} * 100 \qquad \text{(Coeficiente de variación)}$$

Coeficiente de variación en R

- La desviación estándar, al igual que la media y la varianza, es sensible a la presencia de datos con valores inusuales.
- Cuanto más pequeña sea la desviación estándar mayor será concentración de datos alrededor de la media.

Presentación de información en tablas

Cuadro: Resumen de algunos estadísticos descriptivos (*DE* desviación estándar, *CV* Coeficiente de variación) de las variables: área (Ha), Pendiente (%), pH del Suelo (adimensional), Densidad de lombrices ($individuos/m^2$)

Variable	Media	Mediana	*DE*	*CV*
Area	2.99	3.00	1.07	35.86
Pendiente	3.50	2.00	3.65	104.26
pH del Suelo	4.56	4.60	0.58	12.76
Densidad de lombrices	4.35	4.00	2.62	60.26

Medidas de posición relativa

Medidas de posición relativa

Las medidas de posición relativa tienen como propósito describir el comportamiento de una variable cuantitativa dividiendo la serie de valores en un número determinado de partes que sean porcentualmente iguales, los más comunes son: los cuartiles (cuatro partes), lo deciles (diez partes) y los centiles o percentiles (cien partes).

Cuartiles

Dividen al conjunto de datos (ordenado) en cuatro partes porcetualmente iguales (25 %). Hay tres cuartiles: Q_1, Q_2 y Q_3.

Deciles

Dividen al conjunto de datos (ordenado) en diez partes porcetualmente iguales (10 %). Los deciles se denotan: D_1, D_2, ..., D_9. El decil 5 corresponde al cuartil 2 que es igual a la mediana.

Medidas de posición relativa

Percentiles o centines

Dividen al conjunto de datos (ordenado) en 100 partes porcetualmente iguales (1 %). Los deciles se denotan: P_1, P_2, ..., P_{99}. El percentil 50 coincide con el decil 5 que a su vez corresponde al cuartil 2 que es igual a la mediana de los datos.

Cálculo de cuantiles, deciles y centiles en R

- `cuartiles<-quantile(cuantitativas[,1])`
- `deciles<-quantile(cuantitativas[,1], probs=seq(0,1,by=0.1))`
- `centiles<-quantile(cuantitativas[,1], probs=seq(0,1,by=0.01))`

Medidas de posición relativa

Prueba númerica para los cuantiles

Escriba y ejecute las siguientes líneas de código para verificar que se cumple el enunciado: *"describir el comportamiento de una variable cuantitativa dividiendo la serie de valores en un número determinado de partes que sean porcentualmente iguales"*

- `DT<-rnorm(1000)`
- `Qs<-quantile(DT)`
- `length(which(DT>=Qs[1] & DT<Qs[2]))`
- `length(which(DT>=Qs[2] & DT<Qs[3]))`
- `length(which(DT>=Qs[3] & DT<Qs[4]))`
- `length(which(DT>=Qs[4] & DT<=Qs[5]))`
- `diff(cuartiles)`

Medidas de posición relativa

Representación gráfica de los cuartiles

Los cuartiles, junto con el máximo y el mínimo, se representan mediante un gráfico llamado: diagrama de caja y bigotes (box-plot). Este diagrama está compuesto por un rectángulo (caja) y dos brazos (bigotes), pero en el siguiente código adiciona el promedio arítmetico como una cruz azul y los datos originales como puntos rojos.

- `library(reshape)`
- `reshape.cuanti<-melt(cuantitativas)`
- `reshape.cuanti$id<-rep(seq(1,4),each=20)`
- `boxplot(cuantitativas, las=1)`
- `points(reshape.cuanti[,3],reshape.cuanti[,2],`
  `cex=0.7, pch=16, col=red")`
- `points(c(1:4),apply(cuantitativas,2, mean), pch=3,`
  `col="blue", cex=1.5)`

Medidas de posición relativa

Datos atípicos o inusuales

Un valor atípico o inusual (outlier, en inglés) corresponde a una observación que presenta un valor distante con respecto al resto de los datos. Las estadísticas calculadas con conjuntos de datos que incluyen valores inusuales pueden resultar poco veraces.

Para determinar qué datos son atípicos dentro de un conjunto de observaciones se suele tomar como referencia la diferencia entre el tercer y primer cuartil ($Q_3 - Q_1$), el cual se conoce como el rango intercuartílico (RIQ). Un valor es considerado como atípico cuando:

$$< Q_1 - 1.5 \cdot \mathrm{RIQ} \qquad \text{(Límite inferior, Li)}$$

ó

$$> Q_3 + 1.5 \cdot \mathrm{RIQ} \qquad \text{(Límite superior, Ls)}$$

Medidas de posición relativa

Diagrama de caja y bigotes con datos atípicos

Cuando se representa un conjunto de datos con valores atípicos mediante un diagrama de cajas y bigotes, estos aparecerán como puntos por encima o por debajo del límite del bigote. En estos casos el valor máximo y mínimo se re-definirán como:

$$\text{Maximo} = \text{Max}(X) \leq \text{Ls} \qquad \text{(Máximo en presencia de atípicos)}$$

$$\text{Minimo} = \text{Min}(X) \geq \text{Li} \qquad \text{(Mínimo en presencia de atípicos)}$$

¿A qué se deben los datos atípicos?

- Errores de procedimiento.
- Acontecimientos extraordinarios.

Diagrama de caja y bigotes con datos atípicos

- `Dat.Inu<-c(13, 16.3, 20.5, 18.7, 18, 18, 18.8, 22.3, 19.7, 18.1, 20, 24)`
- `boxplot(Dat.Inu, pch=16, cex=0.7, ylim=c(12,25))`
- `cuartiles<-quantile(Dat.Inu)`
- `RIQ<-cuartiles[4] - cuartiles[2]`
- `Li<-cuartiles[2]-(1.5*RIQ)`
- `Ls<-cuartiles[4]+(1.5*RIQ)`
- `abline(h=Ls, lwd=2,lty=3, col=red")`
- `abline(h=Li, lwd=2,lty=3, col=red")`

¿Qué hacer con los datos atípicos?

Los datos atípicos distorsionan los resultados de los análisis, por esta razón se deben identificar y tratar de manera adecuada. Generalmente se excluyen del análisis.

Correlación

Covarianza

La covarianza es una medida del grado de variación conjunta de dos variables aleatorias. El estimador de la covarianza $COV_{(X,Y)}$ de dos variables aleatorias x y y es:

$$COV_{(X,Y)} = \frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y}) \qquad \text{(Covarianza)}$$

Características de la covarianza

- El signo de la covarianza representa la tendencia general de la relación lineal entre las variables.
- Cuando la covarianza entre dos variables (X y Y) es cero, es porque son independientes
- La covarianza es simetría, es decir: $COV_{(X,Y)} = COV_{(Y,X)}$

Correlación

Cálculo de la Covarianza en R

En R la covarianza se puede calcular usando la función `var()` o `cov()` si se emplean como argumentos el vector con los datos de la variable X y el vector con la variable Y. Si se dispone de una matriz numérica, ésta se puede ingresar como único argumento a cualquiera de las dos funciones y el resultado será una matriz cuadrada que tendrá en la diagonal los valores de la varianza y en los triángulos las covarianzas.

Cálculo de la covarianza en R

- `var(cuantitativas[,3],cuantitativas[,4])`
- `cov(cuantitativas[,3],cuantitativas[,4])`
- `Varianzas<-diag(cov(cuantitativas))`
- `Covarianzas<-cov(cuantitativas)`

Correlación

Correlación

Coeficiente de correlación de Pearson es una medida de la relación lineal entre dos variables aleatorias **cuantitativas**. Se define como la covarianza de dos variables, dividido en el producto de sus desviaciones estándar.

$$\rho_{(X,Y)} = \frac{COV_{(X,Y)}}{S_X * S_Y} \qquad \text{(Coeficiente de correlación de Pearson)}$$

Correlación y estadística de prueba en R

- `cor(cuantitativas[,3],cuantitativas[,4])`
- `cor.test(cuantitativas[,3],cuantitativas[,4])`
- `plot(cuantitativas[,3],cuantitativas[,4], pch=16)`
- `text(3.8, 8, paste("r",round(cor(cuantitativas[,3], cuantitativas[,4]),3),sep="="))`

Correlación

Diagrama de dispersión (scatter-plot ó XY-plot)

En la diapositiva anterior se empleó la función `plot()` para generar un **diagrama de dispersión**. El diagrama de dispersión utiliza coordenadas cartesianas para mostrar los valores de dos variables de un conjunto de datos.

El diagrama de dispersión se emplea en dos circunstancias, cuando una variable está bajo el control del experimentador y cuando no. En el contexto de la correlación ninguna de las variables está bajo control; por lo tanto, cualquiera se puede representar en cada eje y el diagrama de dispersión mostrará el grado de correlación (no causalidad) entre las dos variables.

El diagrama de dispersión pone en evidencia algunas características que pueden tener los datos, tales como: relaciones no lineales entre las variables, existencia de agrupaciones definidas por las variables entre otras.

Correlación

Interpretación de la correlación

En la práctica el coeficiente de correlación de Pearson (r) es un índice que mide el grado de relación (asociación) de dos variables. La correlación toma valores entre -1 y 1; un valor de 0 indica la ausencia de relación.

La fuerza de la correlación no depende del signo. Por lo tanto, $r = 0.9$ y $r = -0.9$ son iguales en cuanto al grado de asociación de las variables. Un r positivo indica que un aumento en la variable X corresponde con un aumento en la variable Y; existe una relación directa entre ellas. Una correlación negativa indica una relación inversa, mientras una variable aumenta la otra disminuye.

Cuando la r tiene valores de 1 ó -1, se le llama correlación lineal perfecta. Sin embargo, en la vida real, siempre hay variaciones aleatorias en las observaciones; por lo tanto, una relación lineal perfecta es extremadamente rara.

Correlación

Evaluación de la asociación entre dos muestras pareadas

Se requiere aplicar un prueba para evaluar si el valor de la correlación es significativamente diferente de cero. Ya que un valor de cero, o que no es significativamente diferente de cero, indica que no hay asociación entre las variables.

Cuando se emplea el coeficiente de correlación de Pearson, la estadística de prueba se basa en el supuesto de que la muestra ha sido extraída de manera aleatoria, que las dos variables se distribuyen de manera normal; si es así, el estadístico T se distribuye de acuerdo a un modelo de probabilidad $t - students$ con $n - 2$ grados de libertad.

Si el el valor de p ($p - value$) es menor que 0.05 se acepta que el valor de la correlación es significativamente diferente de cero.

Correlación

Errores en la interpretación de la correlación

La correlación tiene limitaciones y puede en algunos casos malinterpretarse; como por ejemplo cuando se presentan asociaciones accidentales, lo que ha llevado a afirmaciones como: **es un error creer en una hipótesis de investigación solo porque el valor de p indica la existencia de significancia estadística**. Existen circunstancias como la contaminación de los datos, errores en los instrumentos de medición, sesgos en la elección de los individuos o elementos de la muestra o un diseño experimental pobre que pueden afectar la confiabilidad de estadísticos como la correlación.

Uno de los usos erróneos más frecuentes y serios con respecto al análisis de la correlación es interpretar una alta correlación entre variables como una relación de causa y efecto. El análisis de correlación mide una relación o asociación, no determina la explicación o los fundamentos de esa asociación.

Métodos no-paramétricos para el estudio de la correlación

El coeficiente de correlación de Pearson aplica cuando las variables son cuantitativas y siguen una distribución gaussiana; si esto no se cumple existen otros métodos no-paramétricospara determinar la correlación: correlación de ρ **(rho) de Spearman** y la correlación τ **(tau) de Kendall**.

Correlación **rho de Spearman**

Es un coeficiente que permite medir la correlación (asociación) de dos variables cuando las mediciones se realizan en una escala ordinal. La correlación de Spearman también se usa cuando aun siendo variables cuantitativas continuas, no tienen una distribución semejante a la curva normal, o cuando ambas variables son discretas. En estos casos los datos deben ser ordenados y reemplazados por su respectivo orden. Se calcula así:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}. \qquad \text{(Coeficiente de correlación de Spearman)}$$

donde $d_i = x_i - y_i$, es la diferencia entre rangos y n es el número de parejas

Ejemplo para la correlación **rho de Spearman**

Los machos del ave fragata magnífica (Fregata magnificens) tienen una gran bolsa roja en la garganta. Ellos exhiben esta bolsa y la usan para hacer un sonido similar al producido por un tambor cuando están en búsqueda de pareja. Madsen et al. (2004) se plantearon como objetivo determinar si las hembras, que presuntamente eligen a sus compañeros en función del tamaño de la bolsa, usan el tono del sonido como un indicador del tamaño de la bolsa. Los autores estimaron el volumen de la bolsa y la frecuencia del sonido de tambores en 18 machos (Fragata.csv).

- `Fragata<-read.csv(/Users/.../Fragata.csv");`
- `par(mfrow=c(1, 2))`
- `hist(Fragata[,1], breaks=6 ,main="Volumen")`
- `hist(Fragata[,2], breaks=6 ,main="Frecuencia")`
- `cor.test(Fragata[,1],Fragata[,2], method="spearman")`

Ejemplo para la correlación **rho de Spearman**

Un investigador está interesado en saber si el desarrollo mental de un niño se asocia a la educación formal del padre. De esta manera, obtiene la calificación de desarrollo mental en la escala de Gesell de ocho niños elegidos aleatoriamente y se consulta acerca del grado de escolaridad del padre.

Escolaridad	Rango	Desarrollo del niño
Primero de bachillerato	5	90
Primero de primaria	2	87
Técnico	8	89
Sexto de primaria	4	80
Tercero de Bachillerato	6	85
Tercero de primaria	3	84
Analfabeta	1	75
Bachiller	7	91

Métodos no-paramétricos para el estudio de la correlación

Ejemplo para la correlación **rho de Spearman**

- `x <- c(5,2,8,4,6,3,1,7)`
- `y <- c(90,87, 89, 60, 85, 84, 75, 91)`
- `cor.test(x, y, method = "kendall")`

Prueba τ de Kendall

Es usada para medir la asociación entre dos cantidades medidas en condiciones no-paramétricas. La τ de Kendall, que se basa en contar el número de pares concordantes y discordantes. Sea $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ un conjunto de observaciones de las variables aleatorias X y Y, respectivamente; de tal manera que todos los valores de (x_i) y (y_i) son únicos. Cualquier par de observaciones (x_i, y_i) y (x_j, y_j) se consideran concordantes o discordantes de acuerdo a:

$$\text{Concordantes} = \begin{cases} x_i > x_j & \text{y } y_i > y_j, \\ x_i < x_j & \text{y } y_i < y_j, \end{cases} \tag{2}$$

$$\text{Discordantes} = \begin{cases} x_i > x_j & \text{y } y_i < y_j, \\ x_i < x_j & \text{y } y_i > y_j, \end{cases} \tag{3}$$

Si $x_i = x_j$ o $y_i = y_j$, la pareja no es ni concordante ni discordante.

Prueba τ de Kendall

El valor del coeficiente de correlación τ de Kendall se determina así:

$$\tau = \frac{N_C - N_D}{\frac{1}{2}n(n-1)}. \qquad \text{(Coeficiente de correlación tau de Kendall)}$$

donde N_C, es el número de parejas concordantes y N_D es el número de parejas discordantes. En una relación monótona perfecta, todas las parejas son concordantes o todas son discordantes. Aunque esta es una situación extraña cuando se trabajan con variables que son aleatorias. Debido a que hay que determinar la concordancia o discordancia de las todas las parejas, éste es un método intensivo desde el punto de vista computacional.

Por qué usar ggplot2?

Las gráficas normales pueden ser

- Feas y requerir demasiado trabajo
- En R hay mejores maneras de construir visualizaciones estadísticas.

Qué ventajas me ofrece ggplot2?

- Sigue una gramática, como cualquier lenguaje de programación.
- Define los componentes básicos que componen una sentencia. En este caso, la gramática define los componentes de una gráfica.
- Gramática de los gráficos originalmente propuesta por Lee Wilkinson
- Requiere cierto grado de esfuerzo para comienzar pero con la práctica se puede construir sin mayor esfuerzo gráficas complejas para publicaciones de alto nivel.

Alguna terminología

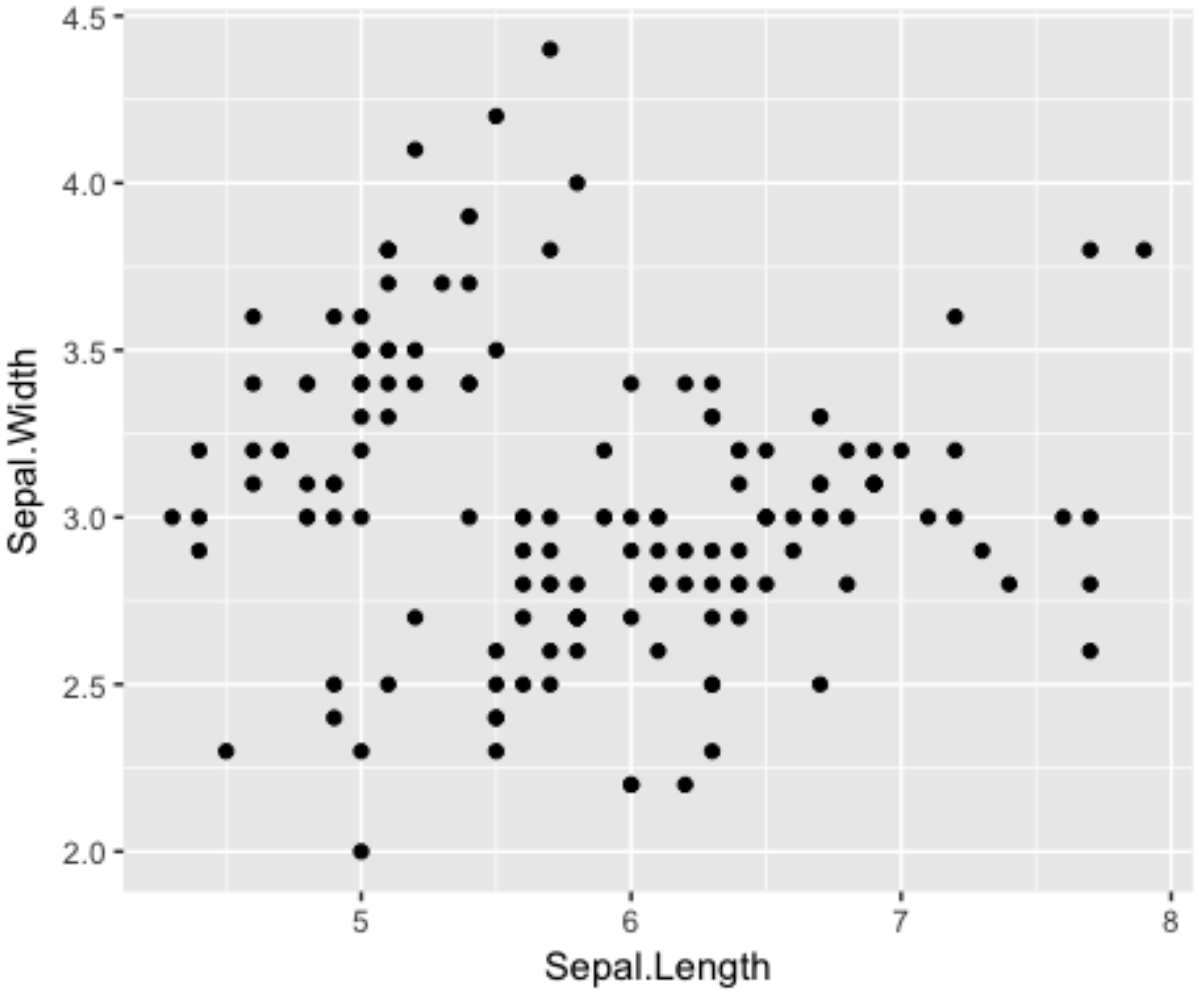
- **ggplot** - La función principal donde se especifica el conjunto de datos y las variables a graficar
- **geoms** - Objetos geométricos
  - geom_point(), geom_bar(), geom_density(), geom_line(), geom_area()
- **aes** - componente estético
  - shape, transparency (alpha), color, fill, linetype.
- **scales** Definir cómo se trazarán los datos
  - *continuous*, *discrete*, *log*

El conjunto de datos **iris**

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Hagamos un primer ejemplo

```
ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width)) +
geom_point()
```



Hagamos un primer ejemplo

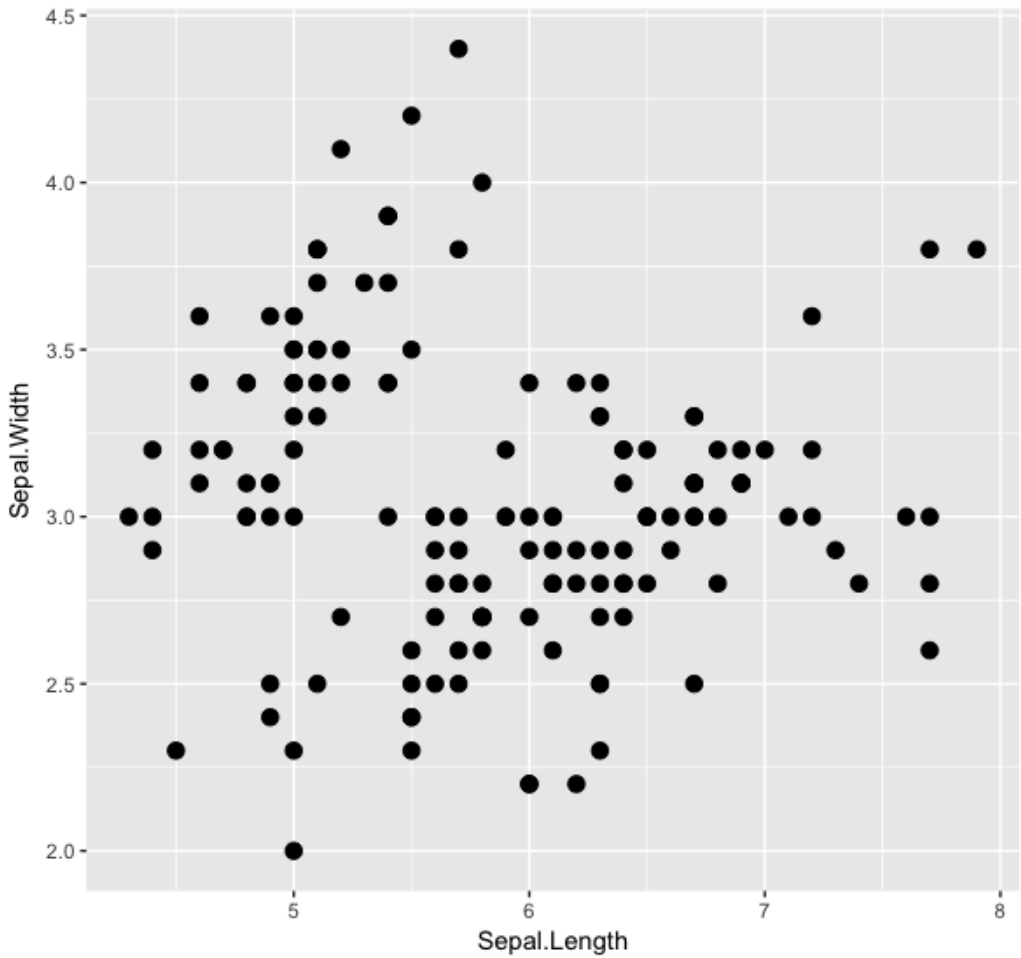
```
ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width)) +
geom_point()
```

```
myplot<-ggplot(data=iris, aes(x=Sepal.Length,
y=Sepal.Width)) +
myplot + geom_point()
```

- Especifique los datos y las variables en la función `ggplot`.
- A continuación, agregue capas de objetos geométricos, modelos estadísticos y paneles.

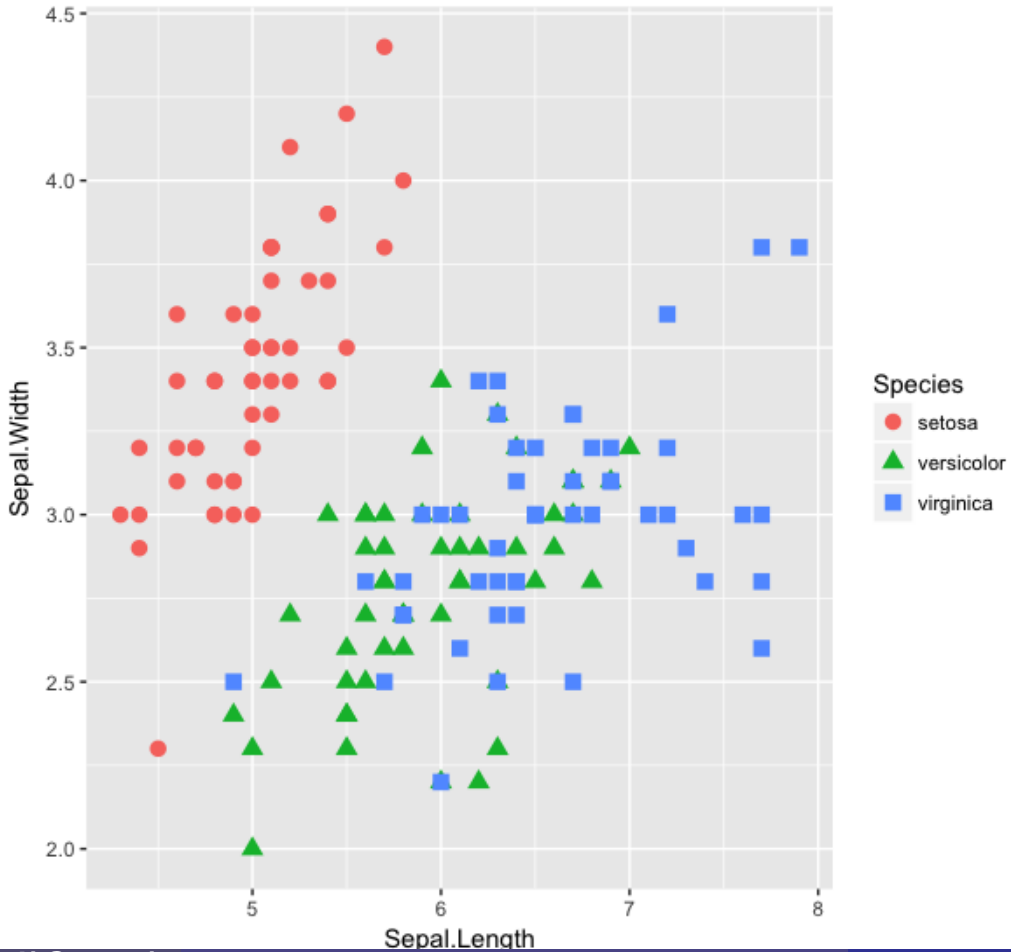
Personalizar el gráfico

```
ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width)) +
geom_point(size=3)
```



Personalizar el gráfico

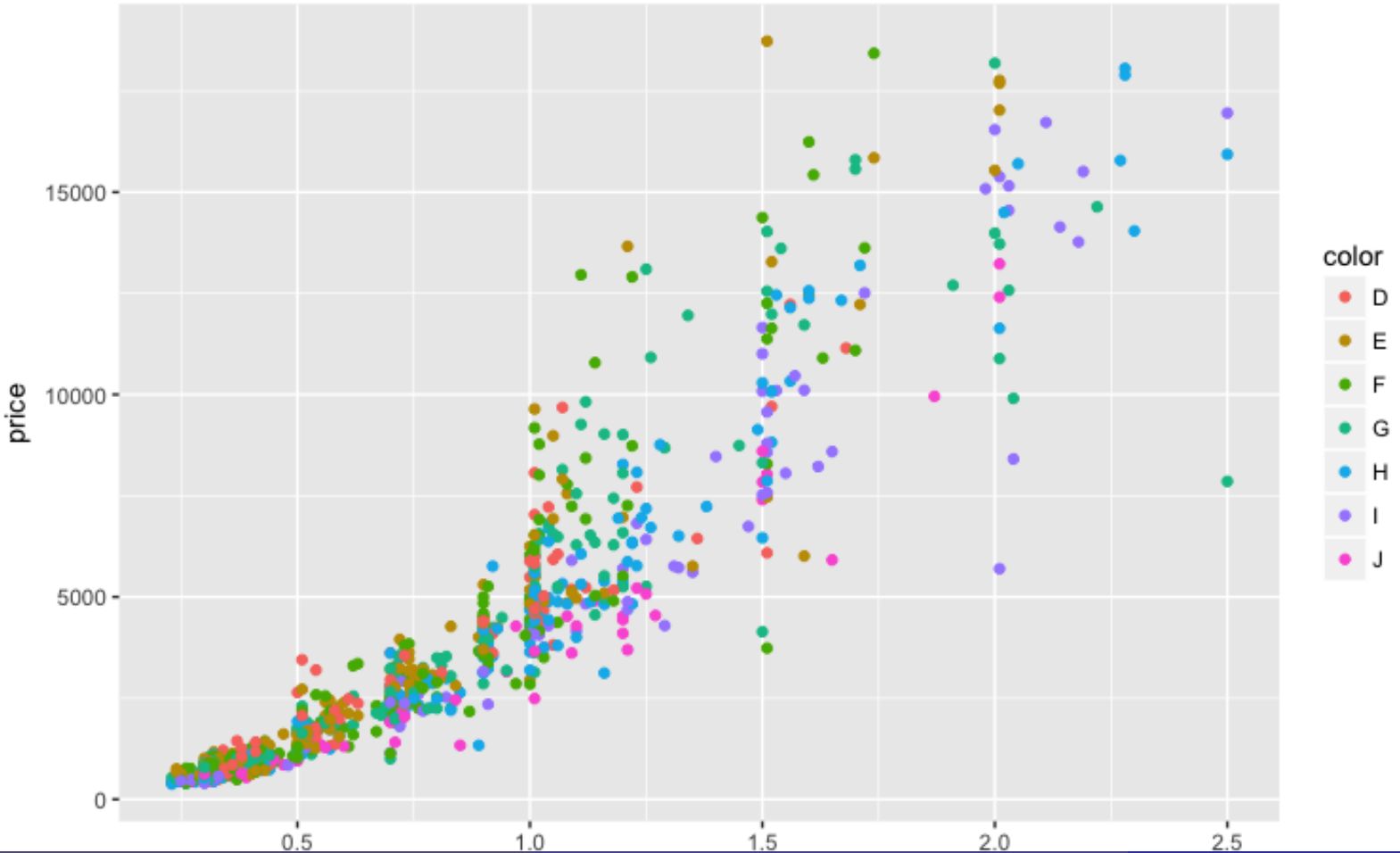
```
 ggplot(data=iris, aes(x=Sepal.Length, y=Sepal.Width,
color=Species)) +
 geom_point(aes(shape=Species), size=3)
```



Ejercicio 1

```
data(diamonds)
d2 <- diamonds[sample(1:dim(diamonds)[1], 1000), ]
```

Generen esta gráfica a partir de los datos llamados `diamonds`



Solución 1

- `ggplot(data=d2) + geom_point(aes(x=carat, y=price))`
- `ggplot(data=d2) + geom_point(aes(x=carat, y=price, col=color))`
- `ggplot() + geom_point(aes(x=carat, y=price, col=color))`

Paquete dplyr

- dplyr es un paquete de R con funciones para transformar y resumir datos tabulares: con filas y columnas.
- Incluye un conjunto de funciones para filtrar filas, seleccionar columnas, reordenar filas, agregar columnas nuevas y resumir datos (estadísticas).
- Además, dplyr contiene una función útil para realizar una tarea común, que es el concepto "dividir-aplicar-combinar".
- Comparadas con las funciones base de R, las funciones en dplyr tienen una ventaja al ser más fáciles de usar, más consistentes en la sintaxis y tienen como objetivo analizar conjuntos de datos en lugar de vectores.

Instalación de dplyr y principales funciones

```
install.packages("dplyr")
library(dplyr)
head(iris)
```

- **select()** - Seleccionar columnas
- **filter()** - Filtrar filas
- **arrange()** - Reordenar o organizar filas
- **mutate()** - Crear nuevas columnas
- **summarize()** - Resumir valores
- **group_by()** - Permite operar bajo el concepto "dividir-aplicar-combinar"

Seleccionar columnas con **select()**

Para seleccionar un conjunto de columnas, podemos usar:

```
head(select(iris, Sepal.Length))
```

Para seleccionar todas las columnas excepto una columna específica, use el operador - (sustracción):

```
head(select(iris, -Sepal.Length))
```

Para seleccionar un rango de columnas por nombre, use el operador : (dos puntos)

```
head(select(iris, Sepal.Length:Petal.Length))
```

Para seleccionar todas las columnas que comienzan con la cadena de caracteres S, use la función `starts_with ()`

```
head(select(iris, starts_with("S")))
```

Seleccionar columnas con **Filter()**

Para filtrar las filas que cumplan con la siguiente condición: `Sepal.Length> = 4.6`, puede usar:

```
filter(iris, Sepal.Length >= 4.6)
```

Para filtrar las filas que cumplan simultáneamente con las siguientes condiciones: `Sepal.Length> = 4.6` y `Petal.Width >= 0.5`, puede usar:

```
filter(iris, Sepal.Length >= 4.6, Petal.Width>=0.5)
```

Operador pipe: %>%

Este operador permite canalizar la salida de una función a la entrada de otra función, conectarlas.

```
iris %>% select(Sepal.Length, Sepal.Width) %>% head
```

Organizar filas **arrange()**, crear nuevas columnas **mutate()** y resumir datos **summarize()**

arrange()

Para organizar las filas de una columna en particular.

```
iris %>% arrange(Sepal.Width) %>% head
```

mutate()

Esta función agregará nuevas columnas al conjunto de datos

```
iris %>% mutate(proportion = Sepal.Length/Sepal.Width)
```

summarize()

La función **summarize ()** calcula estadísticas para resumir la información en una columna determinada

```
iris %>% summarize(avg_slength = mean(Sepal.Length))
```

Operaciones grupales usando **group_by ()**

group_by ()

La función **group_by ()** es de las más importantes y útiles en **dplyr**. Queremos dividir el conjunto de datos por alguna variable (por ejemplo, Sepal.Length), aplicar una función a los datos agrupados y luego combinar el resultado.

```
iris %>% group_by(Sepal.Length) %>%
summarise(avg_slength = mean(Sepal.Length),
min_slength = min(Sepal.Length),
max_slength = max(Sepal.Length), total = n())
```