

Introducción y Lectura de Datos en R:

1) En forma directa:

```
> edad = c(19, 34, 26, 18, 28, 21, 23, 22, 27, 31, 25, 34, 19, 27, 36, 23, 25, 23, 21, 27, 26, 21, 28, 23)
```

2) Como una secuencia:

```
> ss = 1:25
```

3) Lectura de archivos de datos texto, los cuales se crea con un editor de textos (como el block de notas) e forma de filas y columnas separados por blancos, opcionalmente el nombre de las columnas en la primera fila:

```
> setwd("c:/datos")
```

```
> datos=read.table("hemoglobina.dat", header = TRUE)
```

4) Introducción de datos por teclado mediante scan():

```
> y = scan()
```

Introduzca los datos separados por espacios o la tecla enter, para finalizar presione otra vez enter.

5) Carga de datos de otros paquetes (por defecto del paquete "datasets"), data() proporciona una lista de los datos del paquete y data(nombredata, package="nombre paquete")

```
> data(iris)
```

Para cargar los datos de tres especies de la planta iris del paquete "datasets".

6) Introducción de datos mediante el editor de data.frame:

```
> xnew <- edit(data.frame())
```

Se mostrará el editor del data.frame (una matriz de columnas o variables y filas), en el mismo puede colocar los nombres de variables. Para salir cierre la ventana. Para posterior modificaciones utilice:

```
> xnew = edit(xnew)
```

Lista de los datos creados con los ejemplos anteriores:

```
> ls()
```

```
[1] "cars" "datos" "edad" "iris" "peso" "ss" "x" "xnew" "y"
```

Estadística Descriptiva con R:

Vamos a calcular algunos estadísticos a los datos almacenados en edad:

- **Tamaño o número de los datos**

```
> length(edad)
[1] 24
```

- **media y mediana de los datos**

```
> mean(edad)
[1] 25.29167
> median(edad)
[1] 25
```

- **Percentil k de los datos**

```
> quantile(edad, .10)
10%
19.6
> quantile(edad, .75)
75%
27.25
> quantile(edad, c(0.1, 0.25, 0.5, 0.75, 0.8, 0.9, 0.95))
10% 25% 50% 75% 80% 90% 95%
19.60 21.75 25.00 27.25 28.00 33.10 34.00
```

- **Rango, varianza, desviación estándar y rango inter-cuartílico de los datos**

```
> range(edad)
[1] 18 36
> var(edad)
[1] 23.60688
> sd(edad)
[1] 4.858692
> IQR(edad)
[1] 5.5
```

- Produce un resumen de 5 estadísticos: mínimo, cuartil 1, media, cuartil 3 y máximo de los datos

```
> fivenum(edad)
[1] 18.0 21.5 25.0 27.5 36.0
```

- Produce un resumen de 6 estadísticos: mínimo, cuartil 1, mediana, media, cuartil 3 y máximo de los datos

```
> summary(edad)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
18.00   21.75   25.00   25.29   27.25   36.00
```

Gráficos con R:

- Diagrama de tallo y hojas:

```
> stem(edad)
```

- Gráfico de caja y bigotes

```
> boxplot(edad)
```

- Histograma:

```
> hist(edad)
```

Opciones para los gráficos:

xlab=cadena, ylab=cadena, define etiquetas para los ejes

main = cadena, define título del gráfico, aparece en la parte superior con tamaño de letra grande.

sub = cadena, define sub-título del gráfico, aparece debajo del eje X con tamaño de letra pequeña.

col = cadena, color del gráfico (barras)

```
> hist(edad,xlab="Años", main="Distr. por Edad", ,sub="2015",
col="red")
```

- Gráfico de Barras:

```
> barplot(table(datos$Esc), main="Distr. Por Escuela",
ylab="frecuencias", col="blue")
```

- Gráfico de Torta o sectores:

```
> pie(table(datos$Esc), main="Marcas preferidas")
```

Table(), produce una tabla de frecuencia de la variable.

Análisis de Regresión con R

Para empezar vamos a leer el conjunto de datos “datos_pers.txt” en un data.frame que contiene tres variables: edad, peso y estatura.

```
> setwd("c:/datos")
```

```
> datos=read.table("datos_pers.txt", header = TRUE)
```

Para acceder estas variables se hace referencia a datos\$edad, datos\$peso y datos\$estatura. Para no tener que usar estos nombres largos, vamos a utilizar el comando:

```
> attach(datos)
```

Lo primero en un análisis de regresión es realizar un gráfico de dispersión con la función plot()

```
> plot(Peso, Estatura)
```

Los parámetros de la recta de regresión (b0 y b1) se calculan manualmente como sigue:

```
> b1=cov(Peso, Estatura)/var(Peso)
```

```
> b1
```

```
[1] 0.004123804
```

```
> b0=mean(Estatura) - b1 * mean(Peso)
```

```
> b0
```

```
[1] 1.421818
```

Así que la recta de regresión sería $Estatura = 1.421818 + 0.004123804 * Peso$

Para ver esta recta incluida en el gráfico de dispersión utilizamos la función abline()

```
> abline(a = b0, b = b1)
```

La función lm()

Una de las tareas más importantes de la Estadística es la construcción de modelos para describir la relación entre variables. Y los modelos lineales los más importantes.

En R se usa la función `lm()` para construir estos modelos.

```
> lmXY = lm(y ~ x)
```

Ejemplo:

```
> reg = lm(Estatura ~ Peso)
```

```
> reg
```

Call:

```
lm(formula = Estatura ~ Peso)
```

Coefficients:

(Intercept)	Peso
1.421818	0.004124

El modelo se almacena en la variable `reg` (este nombre puede ser cualquier otro). Uno puede pensar que no hemos ganado mucho, pero no es así. La variable `reg` que hemos creado es un objeto que tiene muchas más propiedades. Vamos:

```
> summary(reg)
```

Call:

```
lm(formula = Estatura ~ Peso)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.108112	-0.016798	-0.005379	0.015985	0.100135

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
--	----------	------------	---------	----------

```
(Intercept) 1.421818 0.089615 15.866 2.42e-10 ***
Peso        0.004124 0.001405 2.935 0.0109 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05363 on 14 degrees of freedom
Multiple R-squared: 0.3809, Adjusted R-squared: 0.3367
F-statistic: 8.614 on 1 and 14 DF, p-value: 0.01086
```

La función `summary()` provee todas las propiedades que se almacenan en la variable `reg`. Además podemos acceder a una de ellas usando `reg$`, veamos como:

```
> lmXY$coefficients
(Intercept)      Peso
1.421817597    0.004123804
```

Muestra los coeficientes `b0` (Intercepto) y `b1` (pendiente) del modelo. Si necesitamos almacenar estos valores:

Para guardar el intercepto u ordenada en el origen

```
> b0 = lmXY$coefficients[1]
> b0
(Intercept)
1.421818
```

Para guardar el intercepto u ordenada en el origen

```
> b1 = lmXY$coefficients[2]
> b1
Peso
0.004123804
```

Resumen de comandos en R:

Comando	Descripción
<code>getwd()</code>	Devuelve el directorio de trabajo
<code>setwd()</code>	Fija el directorio de trabajo
<code>sink("archivo")</code>	la salida de los comandos posteriores se almacene en "archivo"
<code>sink()</code>	Devuelve la salida a la consola
<code>ls()</code>	nos proporciona un listado de los objetos que hay actualmente en el espacio de trabajo (área de memoria donde se guardan los objetos creados en la sesión de R)
<code>rm(objectos)</code>	Eliminar uno o varios objetos del espacio de trabajo
<code>data()</code>	Ver listado de los datos que vienen con la instalación de R
<code>data(nombre)</code>	Cargar el conjunto de datos "nombre"
<code>read.table()</code>	Leer archivos de texto externos en forma de un <code>data.frame</code>
<code>scan()</code>	Leer datos desde el teclado
<code>edit(data.frame)</code>	Crear un <code>data.frame</code> con el editor de R
<code>edit(nombre)</code>	Editar "nombre" con el editor de R
<code>length(datos)</code>	Calcular el tamaño de un dato (número de elementos)
<code>mean(datos)</code>	Media o promedio de los datos
<code>median(datos)</code>	Mediana de los datos
<code>quantile(datos,)</code>	
<code>range(datos)</code>	Rango de los datos
<code>var(datos)</code>	Varianza de los datos
<code>sd(datos)</code>	Desviación estándar de los datos
<code>cov(x,y)</code>	Covarianza entre 2 variables.
<code>cor(x,y)</code>	Correlación entre 2 variables.
<code>fivenum (datos)</code>	Calcula los siguientes 5 estadísticos: mínimo, cuartil 1, mediana, cuartil 3 y máximo
<code>summary(datos)</code>	Además de los 5 anteriores también la media
<code>boxplot(datos,opciones)</code>	Gráfico de caja y bigotes
<code>stem(datos,opciones)</code>	Gráfico de tallo y hojas
<code>hist(datos,opciones)</code>	Histograma
<code>barplot(datos,opciones)</code>	Gráfico de barras
<code>pie(datos,opciones)</code>	Gráfico de torta o sectores
<code>lm(y ~ x)</code>	Modelo lineal
<code>plot(x,y)</code>	Gráfico de dispersión entre las variables x – y
<code>abline(a= , b=)</code>	Dibuja una recta con pendiente b e intercepto a