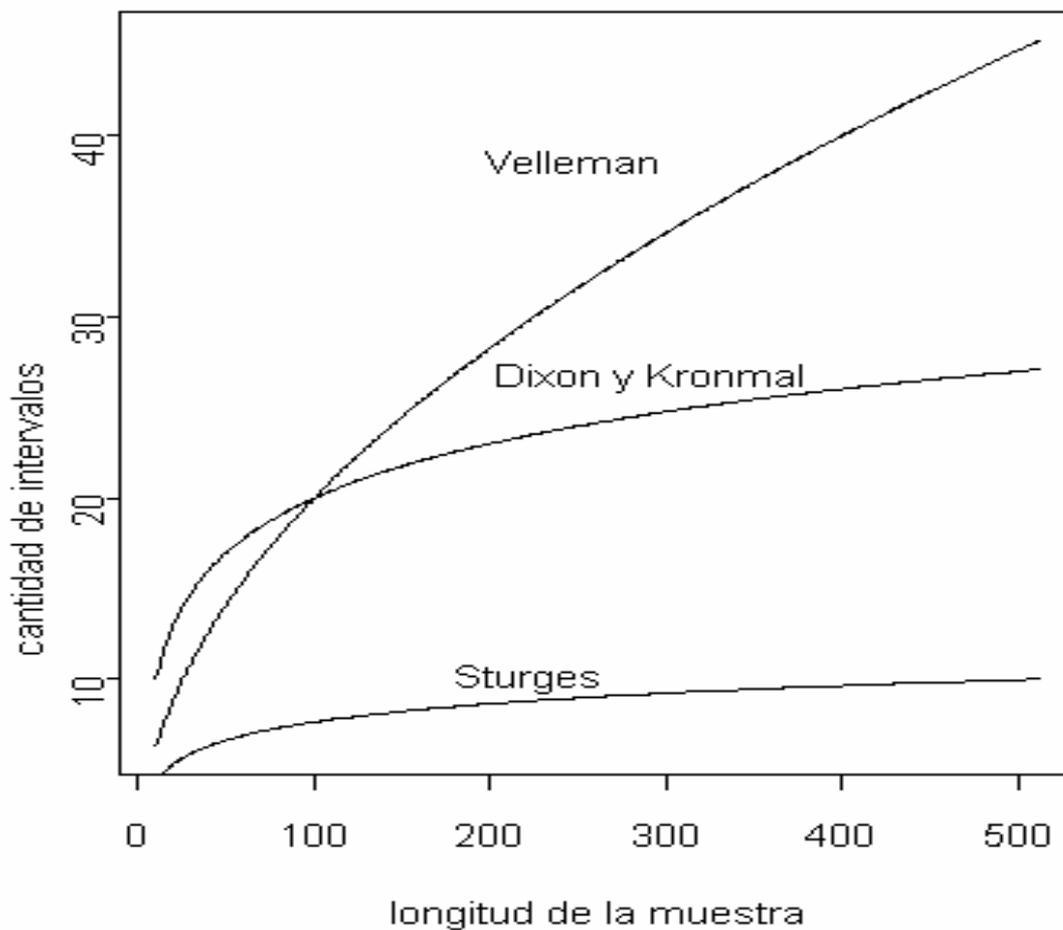


## Hemos visto

Tres reglas para elegir la cantidad de intervalos de un histograma basadas en la cantidad de datos  $n$ .

- i)  $10 \log_{10} n$ , Dixon y Kronmal(1965)
- ii)  $2 n^{1/2}$ , Velleman (1976)
- iii)  $1 + \log_2 n$ , Sturges (1926).



Para obtener el gráfico anterior en S-PLUS utilizamos los siguientes comandos

```

> sturges<-0
> for(i in c(10:512)) (sturges[i]<-
                        (1+log(i,2)))

> velle<-0
> for(i in c(10:512)) (velle[i]<- 2*i^0.5)

> dixkron<-0
> for(i in c(10:512)) (dixkron[i]<-
                        (10*log(i,10)))

```

Definimos un vector, para cada una de las reglas (**sturges**, **velle**, **dixkron**) que dará las componentes y de los pares (x,y) a graficar y el vector

**size** que tiene las coordenadas x para las tres reglas

```

> size<-c(10:512)

```

Graficamos la primera curva en base a la función **plot**

```

> plot(size,velle[10:512],lty=1,type="l",
+ xlab="longitud de la muestra",
+ ylab="cantidad de intervalos")

```

**size**: vector que contiene las coordenadas x

**velle[10:512]**: vector que contiene las coordenadas y

**lty=1**: el help de **par (Graphical Parameters)** dice

lty=x line type, device dependent. Normally type 1 is solid, 2 and up are dotted or dashed. A few devices have only one line type.

**type="l"**: el help dice type="c" type of plot desired.

Values are "p" for points, "l" for lines, "b" for both points and lines (lines miss the points), "o" for overlaid points and lines, "s" or "S" for stairsteps, "n" for no plotting, and "h" for high-density vertical line plot.

Superponemos dos curvas al gráfico anterior.

```
> lines(size,sturges[10:512],lty=1,type="l")
> lines(size,dixkron[10:512],lty=1,type="l")
```

Indicamos a qué regla corresponde a cada curva

```
> cartel<-c("Velleman", "Dixon y Kronmal",
"Sturges")
> text(locator(1),cartel[1])
> text(locator(1),cartel[2])
> text(locator(1),cartel[3])
```

La función, `locator(1)`, da las coordenadas de 1 punto donde se clickea el mouse. Estas coordenadas son leídas por la función `text` que escribe el contenido de `cartel[i]` en el punto del gráfico especificado por dichas coordenadas.

### Hemos visto también:

Dos reglas para elegir el ancho del intervalo de un histograma como

i)  $h_n = 3.49 s n^{-1/3}$  Scott (1979),

ii)  $h_n = 2 R n^{-1/3}$  Freedman & Diaconis (1981),

$R$  es el rango intercuartil.

Para comparar estas reglas con las tres dadas anteriormente debemos transformarlas a cantidad de intervalos.

Una manera de realizar esta conversión es seguir suponiendo datos gaussianos y dividir el rango medio de muestras gaussianas ( $EW_n$ ) de tamaño  $n$  por  $h_n$ .

Una tabla dada en Harter (1970) permite calcular  $EW_n$  fácilmente y la tabla 2 muestra los resultados.

**Tabla 2**

$n$	$EW_n$	$EW_n / h_n$	
		Scott	Friedman-Diaconis
10	3.078	1.90	2.46
20	3.735	2.91	3.76
30	4.086	3.64	4.71
40	4.322	4.23	5.48
50	4.498	4.95	6.14
75	4.806	5.81	7.51
100	5.015	6.67	8.63
150	5.298	8.07	10.43
200	5.492	9.20	11.90
300	5.756	11.04	14.28

El rango ( $W_n$ ) de una muestra aleatoria es una variable aleatoria.

Se puede estimar fácilmente  $EW_n$  utilizando la función

```
rnrangle(n,size)
```

que genera al azar  $n$  valores del rango de  $n$  muestras, de tamaño **size** de la distribución Normal estándar.

Por ejemplo mediante

```
> rnrangle(n=2, size=200)  
[1] 5.905620 5.607135
```

obtenemos los rangos de dos muestras aleatorias de tamaño 200.

Generamos los rangos de 200 muestras de tamaño 10 y calculamos su media.

```
> rango <-rnrangle(n=200,size=10)  
> mean(rango)  
[1] 2.963612
```

Generamos los rangos de 200 muestras de tamaño 100, calculamos su media y su mediana.

```
> rango<-rnrangle(n=200,size=100)  
> mean(rango)  
[1] 5.044275  
> median(rango)  
[1] 4.989066
```

Estos valores son coherentes con los de la tabla 2.

En R no disponemos de la función `rnrange` pero podemos calcular el rango de un vector mediante las funciones `range` y `diff`

`range`: calcula el máximo y el mínimo de un vector  
`diff`: calcula la diferencia de dos valores

```
> 2.5:10
[1] 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
> range(2.5:10)
[1] 2.5 9.5
> diff(range(2.5:10))
[1] 7
```

Generamos los rangos de 200 muestras de tamaño 10 y calculamos su media.

```
> a <- matrix( rnorm(2000),byrow=T, ncol=10)
> mean(apply(apply(a,1,range),2,diff)) #primero
aplicamos la función rango por filas
[1] 3.125263
```

Generamos los rangos de 200 muestras de tamaño 100, calculamos su media y su mediana.

```
> a <- matrix( rnorm(20000),byrow=T, ncol=100)
> mean(apply(apply(a,1,range),2,diff))
[1] 4.993768
> median(apply(apply(a,1,range),2,diff))
[1] 4.963381
```

Nuevamente obtenemos valores coherentes con los de la tabla 2.