

---

# Introducción a la estadística con *Mathematica*.

por *Guillermo Sánchez*

<http://web.usal.es/guillermo>

Universidad de Salamanca

<http://www.usal.es>

Actualizado : 2012 - 05-11

Este tutorial procede del libro (en prensa) *Mathematica más allá de las matemáticas* (editado por <http://www.ad-dlink.es>)

---

## Estadística descriptiva

```
Clear["Global`*"]
```

Cuando se trabaja con gran número de datos uno de sus destinos más frecuentes es su tratamiento estadístico, del que forma parte fundamental la representación gráfica de los datos. En esta sección vamos a utilizar alguna de las funciones más comunes para estudios estadístico básicos: `tutorial/BasicStatistics`.

La instrucción `RandomReal[]` permite generar números aleatorios reales que siguen una distribución uniforme (aquella que toma un determinado valor constante  $min < x < max$ , y es 0 para  $x < min$  o  $x > max$ ). En el ejemplo generamos dos listas con 20 números comprendidos entre 0 y 10.

```
{data1, data2} = {RandomReal[10, 20], RandomReal[10, 20]}  
  
{7.33108, 1.08321, 3.46697, 3.79476, 3.03221, 8.47467,  
 0.157354, 6.24064, 1.03391, 6.47265, 0.203767, 0.426067, 2.19163,  
 0.786015, 3.9021, 3.08666, 5.52376, 9.6152, 8.56477, 3.53023},  
{8.18757, 1.9374, 6.65654, 2.1352, 9.72071, 3.66957, 2.21482,  
 6.46996, 1.28604, 6.86611, 0.022393, 9.04415, 2.98033,  
 0.679684, 6.3707, 3.04171, 2.09199, 6.25824, 9.81779, 0.731796}
```

Para ser rigurosos deberíamos referirnos a números pseudoaleatorios pues son algoritmos quienes lo generan. Sin embargo, para la gran mayoría de aplicaciones prácticas pueden considerarse a todos los efectos números aleatorios.

Calculamos algunos de los estadísticos más comunes de la primera lista (`data1`). Observe que empleamos la sintaxis `{f1[#], ..., fn[#]}&[data1]` que es equivalente a `{f1[data1], ..., fn[data1]}`, simplificando la escritura. A esta forma de definir las funciones, que se conoce como funciones puras, nos referiremos en el próximo capítulo.

```
{Mean[#], Variance[#], Skewness[#], Kurtosis[#],  
  StandardDeviation[#], MeanDeviation[#], MedianDeviation[#],  
  Quantile[#, .6], InterquartileRange[#]} &[data1]  
  
{3.94588, 9.05616, 0.41116, 1.97251, 3.00934, 2.46016, 2.58864, 3.79476, 5.29808}
```

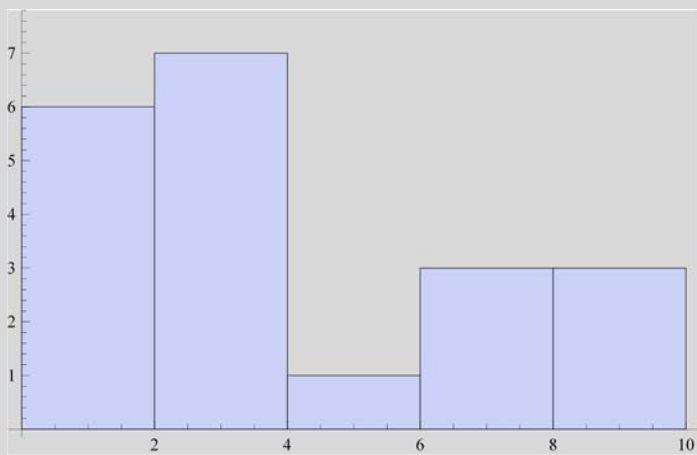
Calculamos la covarianza y la correlación entre las dos listas antes creadas.

```
{Covariance[data1, data2], Correlation[data1, data2]}
```

```
{4.63659, 0.477302}
```

Representamos los datos utilizando Histogram. Si posiciona el cursor dentro de cada barra se le mostrará el valor que toma ésta.

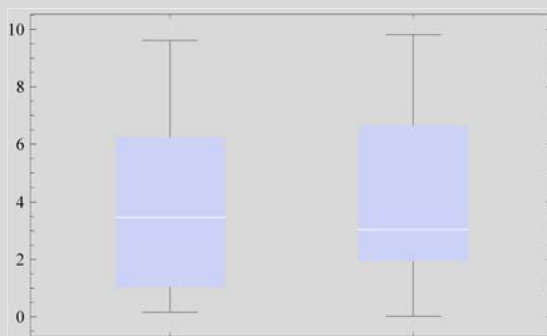
```
Histogram[data1, 5]
```



Podemos también representar los datos en un gráfico BoxWhiskerPlot .

Pruebe a poner el cursor dentro del gráfico y verá toda la información estadística asociada a cada caja.

```
BoxWhiskerChart[{data1, data2}]
```



Para los gráficos de tallo y hojas (StemLeafPlot) necesitamos cargar el paquete StatisticalPlots` .

```
Needs["StatisticalPlots`"]
```

Creamos un gráfico "stem-and-leaf" (Tallo y hojas) para comparar las dos listas antes creadas.

```
StemLeafPlot[data1, data2]
```

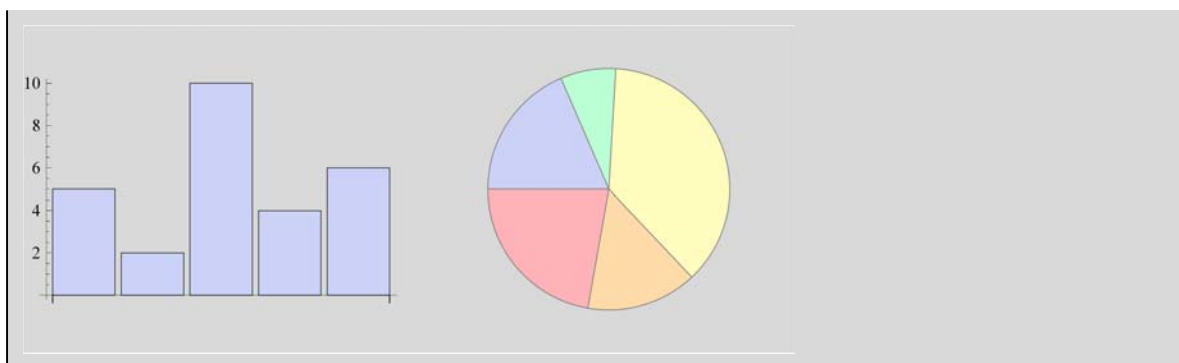
Leaves	Stem	Leaves
8422	0	077
10	1	39
2	2	112
985510	3	007
5	5	
52	6	34579
3	7	
65	8	2
6	9	078

```
Stem units: 1
```

Debajo mostramos un ejemplo en el que representamos los datos en un diagrama de barras y en un diagrama de tarta. Los mostramos uno junto a otro, en horizontal, para lo que utilizamos la función `GraphicsRow`.

```
data3 = {5, 2, 10, 4, 6};
```

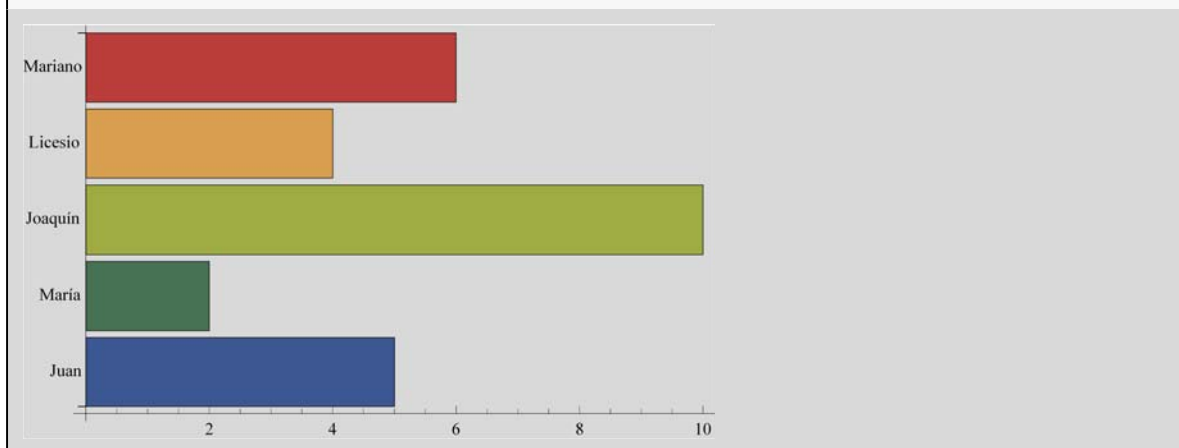
```
GraphicsRow[{BarChart[data3], PieChart[data3]}, ImageSize -> Large]
```



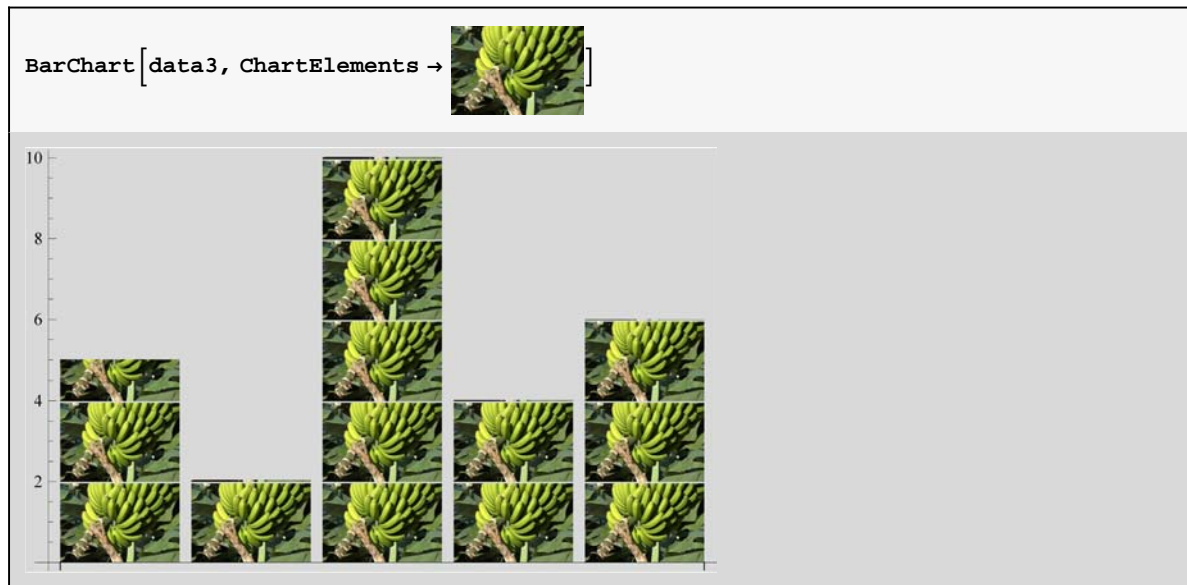
Podemos contruir gráficos personalizados como los que se muestran a continuación.

Añadimos una etiqueta a cada barra y elegimos una paleta de colores (en este caso "DarkRainbow"). Además giramos el gráfico para que las frecuencias que por defecto se representan en el eje OY se nos muestren en el OX.

```
BarChart[data3, BarOrigin -> Left, ChartStyle -> "DarkRainbow",  
ChartLabels -> {"Juan", "María", "Joaquín", "Licesio", "Mariano"}]
```



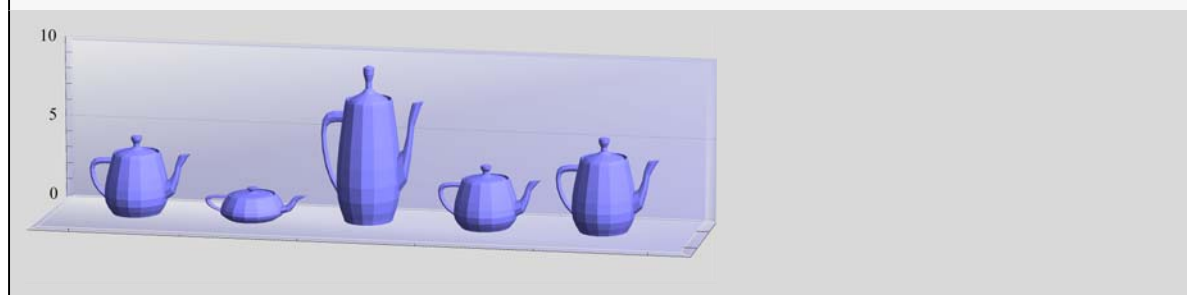
Podemos utilizar cualquier imagen para representar un diagrama de barras.



Aquí utilizamos una figura, conocida como “tetera de Utah”, que se incluye en **ExampleData**, para construir un diagrama de barras 3D.

```
g = ExampleData[{"Geometry3D", "UtahTeapot"}];
```

```
BarChart3D[data3, ChartElements -> g, BoxRatios -> {4, 1, 1}]
```



Los siguientes tutoriales probablemente sean de su interés:

Basic Statistics: [tutorial/BasicStatistics](#)

Descriptive Statistics: [tutorial/DescriptiveStatistics](#)

Continuous Distributions: [tutorial/ContinuousDistributions](#)

Discrete Distributions: [tutorial/DiscreteDistributions](#)

Descriptive Statistics: [tutorial/DescriptiveStatistics](#)

Convolutions and Correlations: [tutorial/ConvolutionsAndCorrelations](#)