1Escuela Técnica Superior Ingenieros Industriales. Ciudad Real. 2015-04-23

Estadística y probabilidad con Mathematica: Novedades

Guillermo Sánchez (http://diarium.usal.es/guillermo)

Mathematica ha ido incrementando las funciones específicas para cálculos estadísticos y de probabilidad, y ampliando sustancialmente las posibilidades gráficas. En las versiones más recientes se han añadido funciones para cálculos no paramétricos, funciones derivadas, tratamiento de funciones censuradas y truncadas, análisis de clusters, y más posibilidades de ajustes de datos entre otras prestaciones. (quizás incluya más funciones que programas específicos de cálculo estadístico). Al final del capítulo mostramos un ejemplo de cómo debe construirse un paquete orientado al control de calidad.

Esta presentación forma parte del libro: Mathematica más allá de las Mathematicas. 2ª Edición. https://books.google.com/books?id=KjfeBQAAQBAJ.

1.1. Lo más nuevo

Las posibilidades de tratamiento estadístico de datos con *Mathematica* son amplísimas. Una simple enumeración de las mismas excedería el espacio de este capítulo. Si no tiene experiencia previa en cálculos estadísticos y de probabilidad con *Mathematica*, para leer este capítulo quizás le interese empezar en el tutorial tutorial/NumericalOperationsOnDataOverview y desde ahí dirigirse al tema que le interese o puede seleccionar un tema concreto de los que siguen:

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

En este capítulo vamos a referirnos en aquellos aspectos que consideramos más novedosos:

- Funciones de probabilidad (guide/ParametricStatisticalDistributions).- De acuerdo con la información proporcionada en la página del fabricante el número de funciones es sustancialmente superior a las que contienen otros conocidos programas de cálculo estadístico como R, SAS, SPSS, etc.
- Distribuciones derivadas (guide/DerivedDistributions).- Permite construir funciones de probabilidad que son combinación de otras distribuciones.
- Distribuciones no paramétricas (guide/NonparametricStatisticalDistributions).- Permite obtener funciones de distribución no paramétricas a partir de datos empíricos.
- Cálculo automatizado de probabilidades (Probability) y esperanza matemática (Expectation).
- Se amplían las propiedades asociadas a las distribuciones: PDF (distribución de densidad), CDF (función de distribución), SurvivalFunction (función de supervivencia), Moment (momentos), EstimatedDistribution, ...
- Se aumentan las posibilidades de generación de números aleatorios (RandomVariate), de amplio uso en simulación Montecarlo y en numerosos análisis estadísticos, de acuerdo a la función de distribución que se especifique.
- Ajustes de datos para varias variables: regresión no lineal (`NonlinearModelFit`), modelo lineal generalizado (`GeneralizedLinearModelFit`) y modelo de regresión logístico (`ProbitModelFit`).
- Gráficos estadísticos específicos: DiscretePlot3D, Histogram, PairedHistogram, QuantilePlot, ProbabilityPlot, BoxWhiskerChart, DistributionChart, …
- Series temporales: guide/TimeSeriesProcesses.
- Análisis de *clusters* mediante el comando `FindClusters` y otros que se detallan en: guide/DistanceAndSimilarityMeasures.
- Análisis de supervivencia y fiabilidad.
- Procesos estocásticos (guide/RandomProcesses), incluida ecuaciones diferenciales estocásticas.

1.2. Los datos

La primera definición de estadística que encontramos en el diccionario de la Real Academia Española es: “Estudio de los datos cuantitativos de la población, de los recursos naturales e industriales, del tráfico o de cualquier otra manifestación de las sociedades humanas”. Esta definición, sin abarcar todo lo que entendemos por Estadística, incluye una idea importante: sin datos difícilmente hay estadística.

Mathematica nos ofrece la posibilidad de acceder a una amplia base de datos a través de colecciones de datos científicos y técnicos (guide/ScientificAndTechnicalData). Asimismo incluye numerosas herramientas para analizar estos datos u otros.

Veamos un ejemplo con el que simplemente se trata de mostrar algunas de las funciones y herramientas disponibles en *Mathematica*:

- Se pretende analizar cómo se distribuye la población española por municipios según su número de habitantes. No pretende ser un estudio riguroso, simplemente es un esbozo de cómo podríamos afrontar un estudio estadístico.
  Descargamos los datos haciendo uso de `CityData`.

```
poblacionciudades = CityData[#, "Population"] & /@ CityData[{All, "Spain"}];
```

  La sintaxis anterior es equivalente a **Map**[*func[arg, #]&, {ciudad1, ..., ciudadn}*] donde la relación de ciudades se generan con: **CityData**[{All, “Spain”}]

- Podemos comprobar el total de municipios recogidos en **CityData**.

```
Length[poblacionciudades]
```

```
8066
```

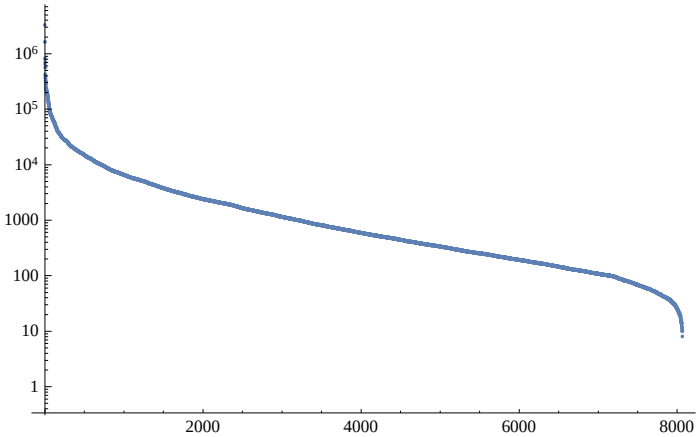
- A continuación hacemos un primer análisis de los datos para lo que suele ser muy útil su representación gráfica. Calculamos los estadísticos más comunes.

```
{Mean[#], StandardDeviation[#], Skewness[#], Kurtosis[#],
    Quantile[#, .6], InterquartileRange[#]} &[poblacionciudades] // N
```

```
{5564.95 people, 47 334.1 people, 47.9311, 3010.32, 980. people, 2168. people}
```

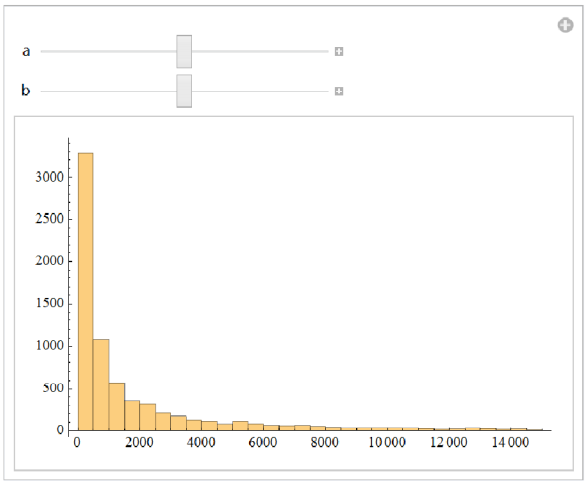
- Observamos que éstos presentan una fuerte asimetría. Podemos verlo mejor si los representamos en escala logarítmica:

```
ListLogPlot[poblacionciudades]
```



- Construimos un histograma. Utilizamos `Manipulate` para ver cómo se comportan los datos en función de los datos que eliminamos por arriba y por abajo.

```
Manipulate[Histogram[Drop[Drop[poblacionciudades, a], -b]],
 {{a, 500}, 0, 1000, 10}, {{b, 500}, 0, 1000, 100}, SaveDefinitions → True]
```



Una primera conclusión es la fuerte asimetría de la distribución de la población española; en particular el elevado número de municipios con una población muy pequeña.

1.2.1. Los datos multivariantes

Muchos datos empíricos vienen representados por más de una variable aleatoria; nos referimos a ellos como datos multivariantes.

En los estudios estadísticos deberían utilizarse datos empíricos reales, pero frecuentemente se recurre a simular los datos para lo que se utiliza generadores de datos aleatorios. En *Mathematica* se utiliza `RandomReal` y `RandomInteger`. Estas funciones emplean algoritmos y por tanto, estrictamente hablando, no proporcionan datos realmente aleatorios; sin embargo, para la mayoría de los casos podemos considerar que realmente lo son.

- Generamos un conjunto de datos que representa una distribución multinormal en la que las medias son iguales (10) y la matriz de covarianzas incluye una correlación entre variables. Si elimina ";" observará que el resultado es una lista de sublistas, que podemos interpretarlo como una matriz de tres columnas:

```
multidata = RandomReal[
   MultinormalDistribution[{10, 10, 10}, {{14, 7, 8}, {7, 6, 8}, {8, 8, 11}}], 100];
```

- Podemos calcular los estadísticos más comunes (media, varianza, etc.) de la forma habitual. En este caso nos encontraremos con tres resultados, uno para cada variable.

```
MedianDeviation[multidata]
```

```
{2.02316, 1.50501, 2.12907}
```

```
Quantile[multidata, .6]
```

```
{11.1392, 11.2133, 11.6519}
```

- Debajo se calcula la covarianza y la correlación para ver si hay relaciones entre las variables. El resultado en ambos casos es una matriz 3×3 de la que se puede concluir que existe correlación entre las variables.

```
Covariance[multidata] // MatrixForm
```

$$\begin{pmatrix} 12.8802 & 6.36327 & 7.24361 \\ 6.36327 & 5.68506 & 7.61568 \\ 7.24361 & 7.61568 & 10.5154 \end{pmatrix}$$

```
Correlation[multidata] // MatrixForm
```

$$\begin{pmatrix} 1. & 0.743619 & 0.622416 \\ 0.743619 & 1. & 0.984982 \\ 0.622416 & 0.984982 & 1. \end{pmatrix}$$

- Puede obtener funciones adicioanales aplicables al tratamiento de datos multivariante utilizando el paquete `MultivariateStatistics`.

```
Needs["MultivariateStatistics`"]
```

- Calculamos la planitud (MultivariateSkewness) y la curtosis (`MultivariateKurtosis`).

```
{MultivariateSkewness[multidata], MultivariateKurtosis[multidata]}
```

```
{0.705951, 15.1111}
```

1.2.2. Gráficos estadísticos

En estadística un aspecto fundamental son la representaciones gráficas. Vamos a mostrar varios tipos de gráficos aplicables a varias variables. Es interesante consultar: guide/StatisticalVisualization.

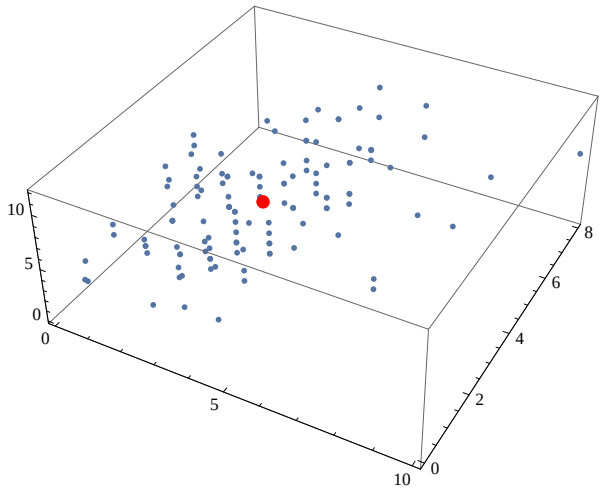
Para construir gráficos estadísticos cargue la paleta : **Palettes ‣ Chart Elements Schemes**.

- Simulamos la generación de un conjunto de tres datos aleatorios procedente de una distribución de Poisson.

```
multidata = RandomVariate[MultivariatePoissonDistribution[1, {2, 3, 4}], 125];
```

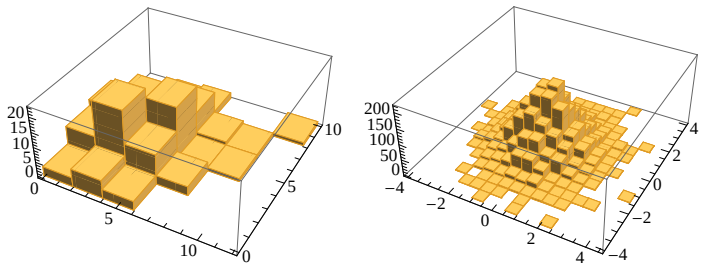
- Representamos cada conjunto de tres datos por un punto. Incluimos un punto en rojo que representa la media de los datos anteriores.

```
Show[{ListPointPlot3D[multidata],
  Graphics3D[{Red, PointSize[Large], Point[Mean[multidata]]}]}]
```



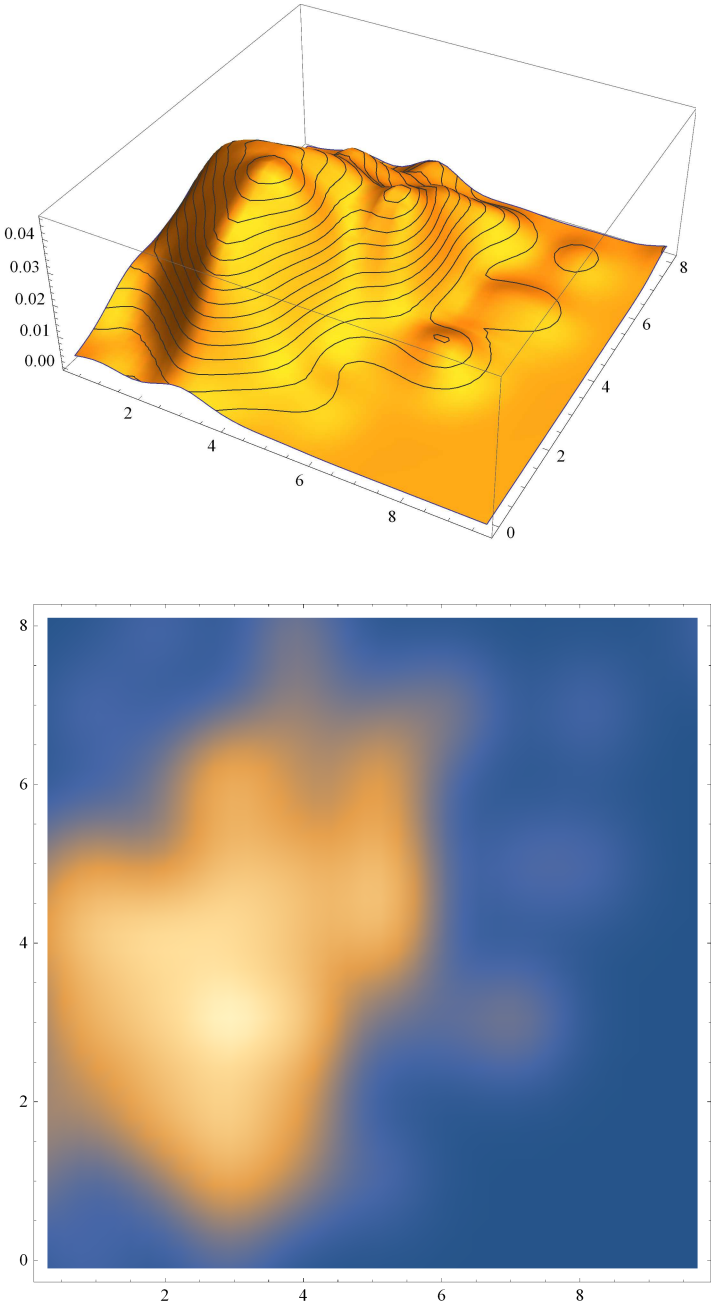
- Aquí mostramos un histograma para las primeras dos dimensiones de los datos que antes hemos generado (**multidata**). (Recuerde que para extraer datos de listas se utiliza: `Part ([[ ...]])`). Lo presentamos conjuntamente con un histograma de 5000 pares de valores aleatorios normalmente distribuidos.

```
GraphicsRow[{Histogram3D[multidata[[All, 1 ;; 2]], 5],
  Histogram3D[RandomReal[NormalDistribution[], {5000, 2}]]}, ImageSize → Medium]
```



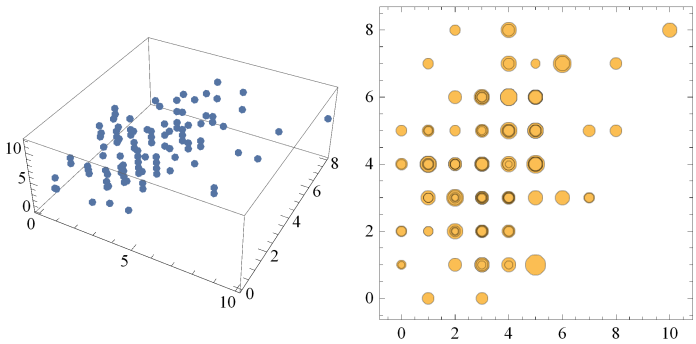
- Otra forma de representar los mismos datos pero suavizando los perfiles es empleando `SmoothHistogram3D` y `SmoothDensityHistogram` (el cambio de color está asociado al valor de la variable).

```
Row[{SmoothHistogram3D[multidata[[All, 1 ;; 2]], ImageSize → Medium],
  SmoothDensityHistogram[multidata[[All, 1 ;; 2]], ImageSize → Medium]}]
```



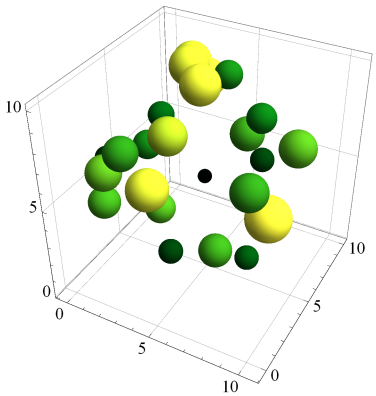
- Aquí un conjunto de datos trivariantes son visualizados como burbujas cuyo tamaño es la coordenada z.

```
GraphicsRow[{ListPointPlot3D[multidata, PlotStyle → PointSize[.025]], BubbleChart[
   multidata, BubbleScale → "Diameter", BubbleSizes → {.025, .075}]}, ImageSize → Medium]
```



- Un conjunto de datos cuadridimensionales pueden ser representados en un gráfico de burbujas (la cuarta dimensión es el color).

```
BubbleChart3D[RandomReal[10, {25, 4}], ColorFunction → "AvocadoColors"]
```



- También podemos crear gráficos personalizados como el ejemplo que sigue en el que el tamaño de las palabras va aumentando según la frecuencia con la que se presentando en un texto.

```
Text[
 Row[With[{data = ReadList[StringToStream["Miré el reloj digital que había en la pared
        de la sala de espera.Marcaba las 16:40 del martes 8 de febrero
        de 2003. El doctor Galán,mi psiquiatra,me había citado a las
        17 h,pero como era habitual en mí,llegué con antelación.Me
        había telefoneado el día anterior pidiéndome que fuese a su
        consulta,aduciendo que circunstancialmente pasaba por Sevilla una
        colega suya.Justo a las 17 horas el Dr.Galán abrió la puerta de su
        despacho y amablemente me invitó a pasar.Nada más entrar observé
        la presencia de una mujer delgada y alta que debía tener una edad
        parecida a la mía.Le presento a la doctora Irina Kuznetsova de
        la Universidad de California,una eminencia en el estudio de la
        consciencia,aunque,como se habrá dado cuenta por el apellido,es de
        origen ruso-dijo el doctor Galán.Ella se dirigió a mí en español,con
        un ligero acento ruso:Encantado de conocerle Sr.Martín o ¿debo
        llamarle doctor Martín? Llámeme por mi nombre:Abel.El Doctor Galán
        nos pidió que continuásemos en ruso,sin preocuparnos porque él no
        pudiese seguirnos;al final podríamos hacerle un resumen"], Word]},
   MapIndexed[Style[#, 2 Count[Take[data, First[#2]], #]] &,
    data]], "
    "]]
```

1.3. Distribuciones de probabilidad

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

1.3.1. Datos y distribuciones de probabilidad

Infinidad de procesos naturales, económicos, medidas experimentales y otros fenómenos pueden representarse por lo que se conoce como distribuciones de probabilidad.

La más conocida y probablemente la que más se presenta en la naturaleza es la distribución Normal o de Gauss (aunque no fue Carl F. Gauss su descubridor) .

- Utilizando el formato lingüístico podemos obtener fácilmente información sobre su definición e historia.

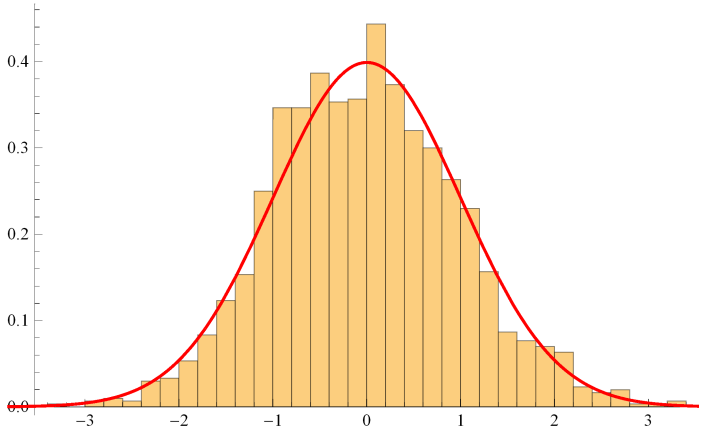
```
WolframAlpha["normal distribution definition",
 {{"DefinitionPod:MathWorldData", 1}, "Content"}]
```

A normal distribution in a variate X with mean μ and variance σ^2 is a statistic distribution with probability density function

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

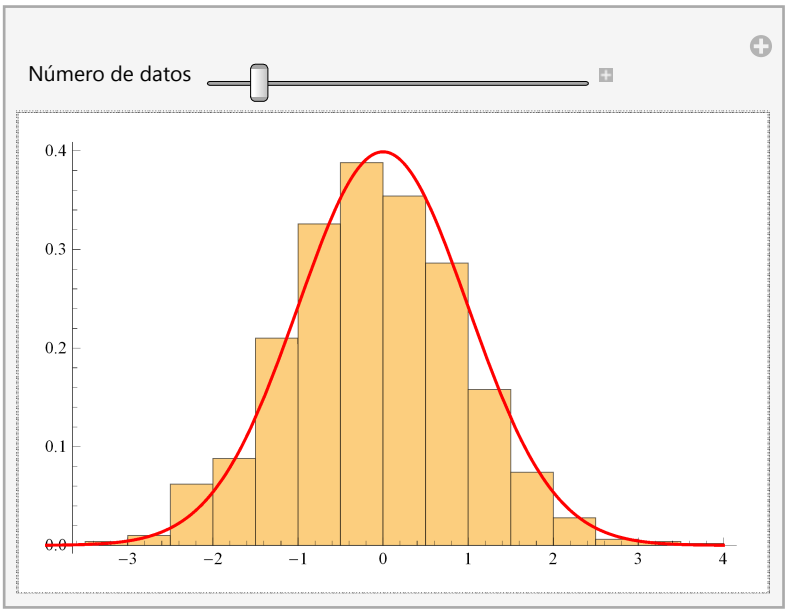
on the domain $x \in (-\infty, \infty)$. While statisticians and mathematicians uniformly use the term "normal distribution" for this distribution, physicists sometimes call it a Gaussian distribution and, because of its curved flaring shape, social scientists refer to it as the "bell curve." Feller uses the symbol $\varphi(x)$ for $P(x)$ in the above equation, but then switches to $n(x)$ in Feller.

- Observe la siguiente sintaxis en la que se comparan datos experimentales (en este ejemplo realmente son datos simulados) con la función de probabilidad normal N(0,1). Aunque aquí se refiere a la distribución normal puede aplicar una sintaxis análoga para otras distribuciones.



Una de las características fundamentales de las distribuciones de probabilidad (en el caso de valores continuos se suele llamar función de densidad) reside en que si determinados datos experimentales siguen una distribución concreta, la diferencia entre la distribución teórica y la experimental será cada vez menor y en un número infinito de datos coincidirán.

- A continuación se compara el histograma de datos experimentales (realmente son datos simulados con la N(0,1), es decir con una función normal de media 0 y desviación estándar 1) con la función teórica de probabilidad. Incremente n y verá la aproximación que se produce de los datos experimentales a la PDF N(0,1).



El número de distribuciones de probabilidad (guide/ParametricStatisticalDistributions) que dispone *Mathematica* es sustancialmente mayor que las disponibles en otros conocidos programas estadísticos, incluidos SPSS, R o SAS. Además *Mathematica* permite operar simbólicamente por lo que podemos construir nuestras propias funciones a partir de las definidas en el programa.

1.3.2. Propiedades

Muchas propiedades de las distribuciones, tales como la función de densidad o de probabilidad (PDF) y la función de distribución (CDF) pueden calcularse fácilmente. Para algunas propiedades de distribuciones multivariantes puede ser necesario cargar el paquete `MultivariateStatistics`.

- En el ejemplo se muestra la función de densidad (o de probabilidad), PDF, y la función de distribución, CDF, de la t de Student para la variable aleatoria X con ν grados de libertad.

```
PDF[StudentTDistribution[ν], x]
```

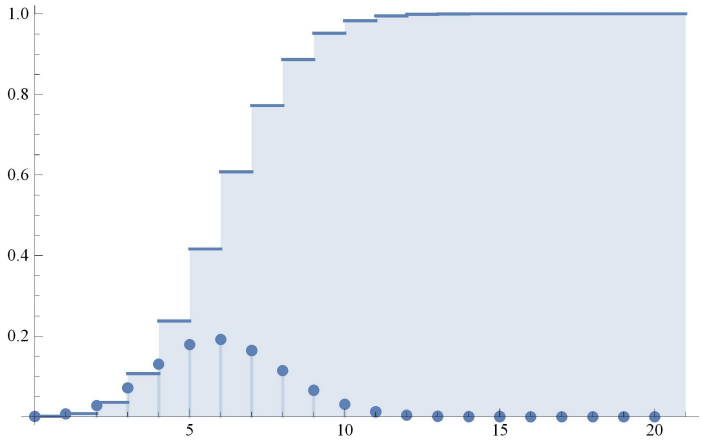
$$\frac{\left(\frac{\nu}{x^2+\nu}\right)^{\frac{1+\nu}{2}}}{\sqrt{\nu}\ \text{Beta}\left[\frac{\nu}{2}, \frac{1}{2}\right]}$$

```
CDF[StudentTDistribution[ν], x]
```

$$\begin{cases} \frac{1}{2}\ \text{BetaRegularized}\left[\frac{\nu}{x^2+\nu}, \frac{\nu}{2}, \frac{1}{2}\right] & x \le 0 \\ \frac{1}{2}\left(1 + \text{BetaRegularized}\left[\frac{x^2}{x^2+\nu}, \frac{1}{2}, \frac{\nu}{2}\right]\right) & \text{True} \end{cases}$$

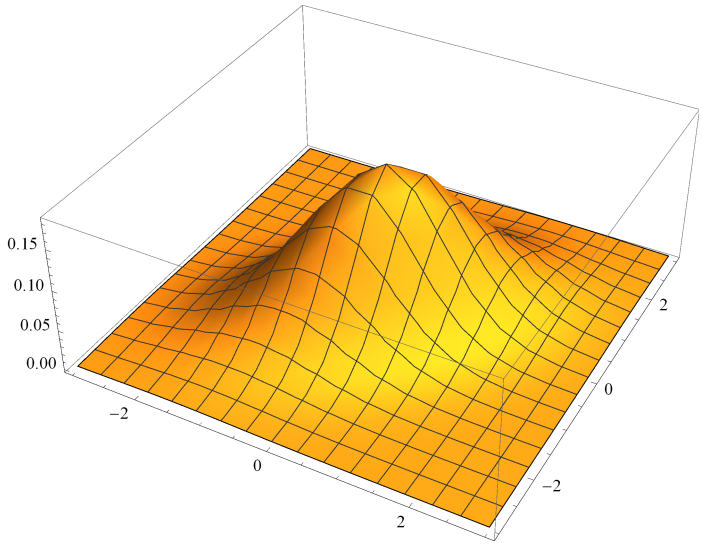
- Debajo representamos la función de probabilidad conjuntamente con la función de distribución para una distribución binomial (`BinomialDistribution`). Observe que se usa `DiscretePlot` que es apropiada cuando la función toma valores discretos, como ocurre con la binomial.

```
Show[DiscretePlot[PDF[BinomialDistribution[20, .3], i], {i, 0, 20}],
 DiscretePlot[CDF[BinomialDistribution[20, .3], i],
  {i, 0, 20}, ExtentSize → Right], PlotRange → All]
```



- Aquí se representa la distribución *t-Student* para dos variables.

```
Plot3D[PDF[MultivariateTDistribution[{{1, 1 / 2}, {1 / 2, 1}}, 20], {x, y}],
 {x, -3, 3}, {y, -3, 3}]
```



- Debajo calculamos algunos parámetros estadísticos (media, varianza, simetría y curtosis) de la distribución χ^2 no central.

```
{Mean[#], Variance[#], Skewness[#], Kurtosis[#]} &[NoncentralChiSquareDistribution[ν, λ]]
```

$$\left\{\lambda+\nu,\ 4\lambda+2\nu,\ \frac{2\sqrt{2}\ (3\lambda+\nu)}{(2\lambda+\nu)^{3/2}},\ 3+\frac{12\ (4\lambda+\nu)}{(2\lambda+\nu)^2}\right\}$$

`Expectation`[*expr*, *x* ≈ *dist*] (para escribir ≈ puede usarse la paleta o el atajo: [Esc] dist [Esc]) calcula la esperanza matemática de *expr* asumiendo que *x* sigue una distribución de probabilidad *dist* que puede aplicarse para distribuciones multivariantes, incluso para el cálculo de esperanza condicionada.

- El momento de orden 2 de una distribución de Poisson.

```
Expectation[x^2, x ≈ PoissonDistribution[μ]]
```

$\mu + \mu^2$

- Si queremos calcular el momento de orden *n* hemos de fijar ciertas restricciones sobre *n* (*n* debe ser un entero mayor que cero).

```
Expectation[x^n, x ≈ PoissonDistribution[μ], Assumptions → n ∈ Integers && n > 0]
```

```
BellB[n, μ]
```

- La función característica de una distribución, como la de Laplace, se calcula como sigue:

```
CharacteristicFunction[LaplaceDistribution[μ, β], t]
```

$$\frac{e^{i t \mu}}{1 + t^2 \beta^2}$$

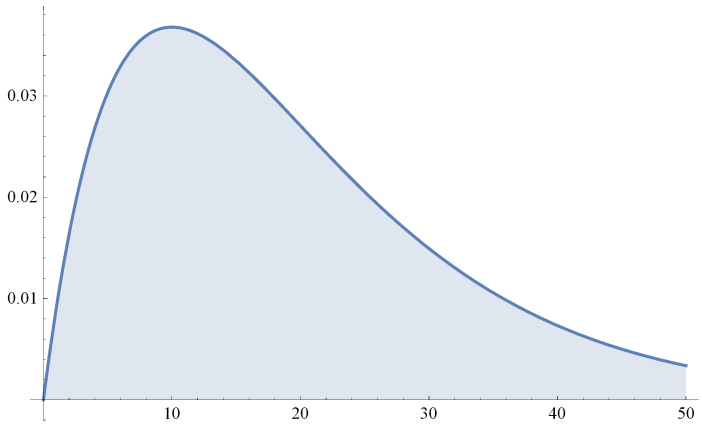
Ejemplo.- Un marcapasos dispone de un sistema redundante formado por unidades independientes. Suponga que la vida media de cada unidad es 10 años y que cuando una unidad falla no se sustituye el marcapasos.

- La distribución típica empleada en el estudio de fallo de equipos y procesos es la distribución gamma (`GammaDistribution`). El caso del ejemplo se puede representar por:

```
gamma = GammaDistribution[2, 10];
```

- La representación de la función de densidad es la siguiente:

```
Plot[PDF[gamma, x], {x, 0, 50}, Filling → Axis]
```



- El tiempo medio de fallo del proceso conjunto es la media de la distribución:

```
Mean[gamma]
```

```
20
```

- La probabilidad de que el marcapasos esté operativo más de 20 años será:

```
Probability[x ≥ 20, x ≈ gamma] // N
```

```
0.406006
```

1.3.3. Distribuciones mezcladas

En este apartado vamos a mostrar con un ejemplo cómo se pueden utilizar las distribuciones mezcladas para resolver casos donde es necesario relacionar varias distribuciones.

Queremos calcular la distribución de la esperanza de vida del conjunto de la población española partiendo de los datos de la esperanza de vida de hombres y mujeres por separado. Se asumirá que la esperanza de vida sigue una distribución normal y se supondrá que las condiciones del presente son aplicables a su proyección futura.

Los datos de la esperanza de vida los vamos a obtener con `CountryData`.

Tenemos dos distribuciones normales: una para hombres y otra para mujeres, y queremos obtener la distribución conjunta. Esto puede hacerse con:

```
MixtureDistribution[{wh, wm}, {N[vh, sh] , N[vm, sm]}]
```

donde

w_h, w_m corresponde a la fracción de hombres y de mujeres.

`N[vh, sh]` y `N[vm, sm]` indica que la esperanza de vida de hombres (h) y mujeres (v) es un distribución normal de media desviación estándar: v_h y s_h para los hombres y v_m, s_m para las mujeres.

- La esperanza de vida de hombres y mujeres en España es la siguiente:

```
{vhombre, vmujer} = QuantityMagnitude[
  Outer[CountryData, {"Spain"}, {"MaleLifeExpectancy", "FemaleLifeExpectancy"}][[1]]]
```

```
{78.866, 85.336}
```

- Con `CountryData` se obtienen las cifras de población de hombres, mujeres y el total.

```
{pophombre, popmujer, poptotal} = QuantityMagnitude[ Outer[CountryData,
    {"Spain"}, {"MalePopulation", "FemalePopulation", "Population"}][[1]]]
```

$\{2.32653 \times 10^7, 2.37773 \times 10^7, 47\,290\,504\}$

- A partir de estos datos obtenemos la fracción de hombres y mujeres en el total de la población.

```
{fh, fm} = {pophombre / poptotal, popmujer / poptotal}
```

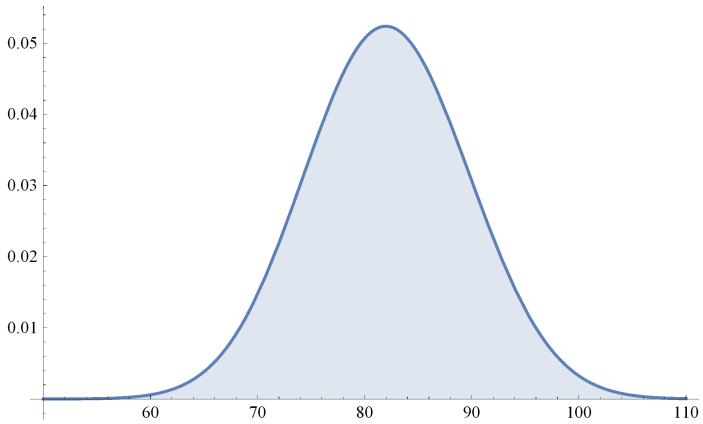
```
{0.491966, 0.502792}
```

Para la desviación típica de la esperanza de vida para edades avanzadas utilizamos (deducido de: http://www.ine.es/): 6.7 años para hombres y 6.9 años para mujeres.

- Con los datos anteriores obtenemos la distribución conjunta para la esperanza de vida de hombres y mujeres como sigue:

```
esperanzavida = MixtureDistribution[{fh, fm},
   {NormalDistribution[vhombre, 6.7], NormalDistribution[vmujer, 6.9]}];
```

```
Plot[PDF[esperanzavida, x], {x, 50, 110}, Filling → Axis]
```



Los resultados podemos aplicarlos a distintos cálculos estadísticos como los que siguen:

- La probabilidad de que una persona, hombre o mujer, muera con 80 años o más es:

```
Probability[x ≥ 80, x ≈ esperanzavida]
```

```
0.608459
```

- Si consideramos exclusivamente los varones, esta probabilidad es considerablemente menor:

```
Probability[x ≥ 80, x ≈ NormalDistribution[vhombre, 6.7]]
```

```
0.432799
```

1.3.4. Distribuciones derivadas

Las distribuciones derivadas se construyen a partir de las distribuciones básicas. Pueden ser combinaciones de varias de ellas o casos especiales de distribuciones básicas como son distribuciones truncadas o censuradas. Al igual que con las distribuciones básicas se pueden calcular distintas propiedades de estas distribuciones derivadas.

Mathematica permite obtener distribuciones trasformadas a partir de la función: `TransformedDistribution`.

- Consideremos una variable aleatoria x, que trasformamos en Exp[x]. Los valores transformados siguen una distribución normal de media μ y desviación estándar σ. Sabemos que una distribución de este tipo es una distribución lognormal y efectivamente ésa es la solución que nos devuelve *Mathematica*:

```
TransformedDistribution[Exp[x], x ≈ NormalDistribution[μ, σ]]
```

```
LogNormalDistribution[μ, σ]
```

La misma instrucción podemos utilizarla para casos más complejos como los que siguen.

Ejemplo: Se trata de calcular la función de distribución de $u + v$ donde u y v son dos variables aleatorias que siguen sendas distribuciones de Poisson de media $\mu 1$ y $\mu 2$,

- La distribución trasformada correspondiente al ejemplo puede calcularse como sigue:

```
TransformedDistribution[u + v, {u ≈ PoissonDistribution[μ1], v ≈ PoissonDistribution[μ2]}]
```

```
PoissonDistribution[μ1 + μ2]
```

Ejemplo.- Un tren AVE recorre una distancia de 340 km a una velocidad media de 245 km/h. Sabemos que la velocidad se distribuye de acuerdo a una distribución normal con desviación estándar 10 km/h. Queremos calcular el tiempo medio que tarda en recorrer dicha distancia.

- Se trata de obtener a partir de la distribución de velocidades una distribución trasformada que represente la función de distribución que sigue el tiempo del recorrido.

```
tiemporecorrido = TransformedDistribution[340 / v, v ≈ NormalDistribution[245, 10]];
```

- Calculamos el tiempo medio del viaje. Como la velocidad está en km/h, multiplicamos el tiempo por 60 para pasarlo a minutos.

```
Chop[60 Mean[tiemporecorrido] // N] "minutos"
```

```
NIntegrate::ncvb :
  NIntegrate failed to converge to prescribed accuracy after 9 recursive bisections in x near {x} = {205.38}. NIntegrate
    obtained 1.3900787018495302` and 0.0025564941721421767` for the integral and error estimates. >>
```

```
83.4047 minutos
```

En medias experimentales es frecuente que ciertos valores no se puedan medir pues caen fuera del rango de detectabilidad de la técnica de medida. Cuando la observación cae por debajo del rango de medida se suele decir que el valor de la medida está por debajo del límite inferior de detección (LID o, en inglés, LLD). También puede ocurrir que esté por encima del rango, como ocurre cuando el equipo se satura. En ese caso decimos que el valor de la medida está por encima del límite superior de detección (LSD).

Cuando se producen estas situaciones, donde no disponemos de los valores que están fuera del intervalo de medida de los equipos, nos encontramos con lo que se conoce como funciones de distribución censuradas.

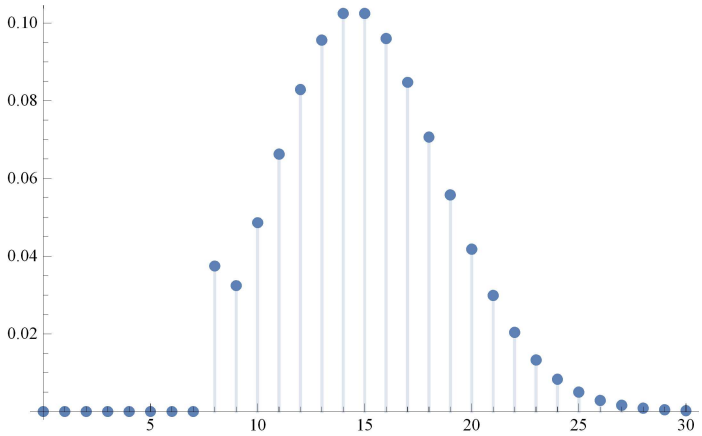
Para cálculos de distribuciones censuradas disponemos de la función
`CensoredDistribution[{{xmin, xmax}, {ymin, ymax}, ...}, dist]`

donde `{{xmin, xmax}, {ymin, ymax}, ...}` representan los límites fuera de los cuales los datos están truncados o censurados.

- Como ejemplo de distribución censurada a la izquierda consideremos el caso de medidas de desintegraciones alfa. Las desintegraciones radiactivas habitualmente siguen una distribución de Poisson y se suelen expresar en bequerelios (Bq) - 1 Bq corresponde a 1 desintegración por segundo.
  Supongamos que la distribución de Poisson tiene de media 15 Bq y que no podemos determinar el valor por debajo de 8 Bq. Este proceso se representa según se muestra a continuación:

```
censuradaIzda = CensoredDistribution[{8, ∞}, PoissonDistribution[15]];
```

```
DiscretePlot[Evaluate[PDF[censuradaIzda, x]], {x, 0, 30}, Filling → Axis]
```



En otras ocasiones ocurre que de un conjunto de medidas eliminamos algunos datos. Por ejemplo: fabricamos un producto y rechazamos los que pesan menos de un valor fijado. Éste es un caso de función de distribución truncada: podemos medir el valor y ello nos sirve para rechazar las unidades que caen por debajo de cierto valor.

Para tratar estas situaciones podemos utilizar la función: `TruncatedDistribution[{{xmin, xmax}, {ymin, ymax}, ...}, dist])`

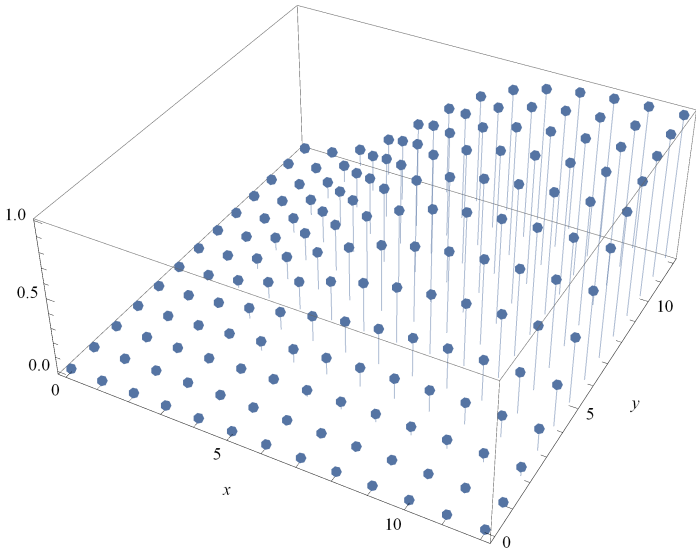
Un caso más complicado es cuando tenemos funciones compuestas de otras distribuciones independientes. En estas situaciones utilizamos `ProductDistribution` que nos da la distribución conjunta. Incluso pueden combinarse de distribuciones censuradas o truncadas.

- Como ejemplo tomemos una distribución compuesta por dos distribuciones de Poisson independientes: una de media 5 que presenta un LSD de 12 y otra de media 6 con un LID de 2.

```
producto = ProductDistribution[PoissonDistribution[5], PoissonDistribution[6]];
censurada = CensoredDistribution[{{-∞, 12}, {2, ∞}}, producto];
```

- Representamos la función de distribución resultante:

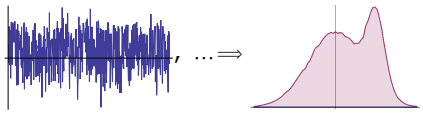
```
DiscretePlot3D[Evaluate[CDF[censurada, {x, y}]],
 {x, 0, 12}, {y, 0, 12}, PlotRange → All, AxesLabel → Automatic]
```



Tal vez le interese explorar estas otras funciones recientemente incoporadas en *Mathematica* : CopulaDistribution, MarginalDistribution y OrderDistribution.

1.3.5. Distribuciones empíricas a partir de los datos

A veces nos encontraremos con un conjunto de datos y queremos agruparlos en una distribución para calcular probabilidades. Esta operación puede ser ejecutada automáticamente con `SmoothKernelDistribution`.



Nos planteamos calcular la probabilidad de que en Buenos Aires las precipitaciones, en cm/mes, exceda 25 cm (o 250 L/m^2) .

- Descargamos las tasas de precipitaciones mensuales de 2000 a 2010 para Buenos Aires.

```
precBA = WeatherData["Buenos Aires",
  "TotalPrecipitation", {{2005, 1, 1}, {2014, 12, 31}, "Month"}]
```

TimeSeries[Time: 01 Jan 2005 to 01 Dec 2014 Data points: 120]

- Nos quedamos sólo con los datos de precipitaciones, sin incluir las fecha. Utilizamos la función `Normal` (nada tiene que ver con la distribución del mismo nombre) para obtener los valores

```
preciBA1 = Transpose[Normal[precBA]][[2]]
```

```
{13.3, 6.43, 8.54, 7.1, 1.4, 4.92, 4.66, 10.01, 3.28, 15.51, 4.91, 1.77, 20.78, 10.33,
 14.01, 5.18, 0.51, 7.76, 4.72, 1.19, 2.51, 16., 4.32, 22.06, 3.25, 8.77, 28.68, 16.4,
 4.19, 4.19, 0.67, 3.62, 8.83, 15.49, 3.2, 2.06, 12.12, 7.09, 16.38, 1.09, 0.53, 5.2,
 3.5, 1.72, 2.36, 28.98, 12.29, 1.85, 1.63, 14.15, 12.94, 11.5, 16.72, 2.03, 10.69,
 1.21, 12.58, 13.2, 18.68, 12.92, 14.88, 28.57, 8.08, 6.51, 12.14, 4.52, 6.64, 1.14,
 11.64, 10.09, 3.2, 7.06, 9.14, 11.13, 3.25, 19.53, 3.99, 12.71, 6.33, 1.78, 1.39,
 4.34, 5.24, 4.03, 16.2, 24.72, 14.25, 21.12, 9.06, 0.72, 2.01, 24.76, 6.61, 26.75,
 11.9, 28.17, 6.55, 9.89, 8.11, 11.23, 10.11, 0.76, 5.97, 1.09, 33.43, 3.1, 13.98,
 0.4, 18.58, 22.23, 21.96, 12.85, 12.78, 10.09, 22.53, 4.82, 20.94, 19.44, 22.79, 9.2}
```

- Trasformamos los datos a una distribución no paramétrica.

```
dist = SmoothKernelDistribution[preciBA1]
```

DataDistribution[Type: SmoothKernel Data points: 120]

- Utilizamos los datos anteriores para calcular la probabilidad de que se superen 25 cm en un mes cualquiera.

```
Probability[x > 25, x ≈ dist]
```

```
0.0613329
```

1.3.6. Cálculo automatizado de probabilidades

En el ejemplo vamos a utilizar las siguientes funciones aplicables al cálculo de probabilidades: `Expectation`, `OrderDistribution` y `Probability`:

3 jugadores (de forma individual e independiente) cuentan el número de coches hasta que ven un coche negro. ¿Cuál es el resultado esperado del ganador si el 10% de los coches son negros?

- Utilizamos la distribución geométrica (`GeometricDistribution[p]`) para modelar la distribución que sigue el número de intentos hasta tener éxito, cuando la probabilidad de éxito es p.

```
Expectation[x, x ≈ OrderDistribution[{GeometricDistribution[0.1], 3}, 3]]
```

```
16.9006
```

- ¿Cuál será la probabilidad de que el ganador cuente menos de 10 coches y al menos 4?

```
Probability[Conditioned[x < 10, x ≥ 4],
 x ≈ OrderDistribution[{GeometricDistribution[0.1], 3}, 3]]
```

```
0.245621
```

1.4 Tratamiento de datos temporales

Otras de las novedades ha sido la incorporación del tratamiento de series temporales. Las novedades son tantas que requeririan un capítulo para tratarlas. Veamos un ejemplo en el que se emplea los datos meteorológicos de temperatura de un lugar concreto para estimar la evolución futura.

- Partimos de las temperaturas medias diaria de Salamanca desde 1973 (la más antigua a la que permite acceder esta función, al menos en el momento de escribir estas líneas) hasta final de 2014 y a continuación se representa gráficamente. Observe la variación estacional.

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

```
tempmediaSalamanca = WeatherData["Salamanca", "MeanTemperature", {{1973}, {2014}, "Day"}]
```

```
TimeSeries[ Time: 01 Jan 1973 to 30 Dec 2014
            Data points: 15 321 ]
```

- La salida lo hace en formato de serie temporal que es muy útil para tratamientos estadísticos posteriores. No obstante pueden mostrarse explicitamente los valores con `Normal`:

```
Normal[tempmediaSalamanca]
```

```
{{2 303 683 200, 0.39 °C}, {2 303 769 600, -2.44 °C}, {2 303 856 000, -2.22 °C},
 {2 303 942 400, -1.44 °C}, {2 304 028 800, -0.39 °C}, {2 304 115 200, -1.72 °C},
 {2 304 201 600, -2 °C}, {2 304 288 000, -2 °C}, ··· 15 306 ···, {3 628 368 000, -0.89 °C},
 {3 628 454 400, 0.56 °C}, {3 628 540 800, 3.61 °C}, {3 628 627 200, 2.33 °C},
 {3 628 713 600, 5.5 °C}, {3 628 800 000, -0.83 °C}, {3 628 886 400, -2.72 °C}}
```

large output | show less | show more | show all | set size limit...

- Observe los primeros números de cada par: Corresponden a los segundos trascurridos desde el 1 de enero de 1900. Es así pues *Mathematica* en muchos casos utiliza lo que denomina `AbsoluteTime`. Por defecto nos da el tiempo absoluto en el momento de ejecutar la función.

```
AbsoluteTime[]
```

$3.6344010033565959 \times 10^9$

- Debajo ajustamos los datos a un modelo típico de serie temporal:

```
tsm = TimeSeriesModelFit[tempmediaSalamanca]
```

TemporalData::rsmplng :

The data is not uniformly spaced and will be automatically resampled to the resolution of the minimum time increment. >>

TimeSeriesModel[Family: ARMA Order: {5, 5}]

- El modelo ajustado lo utilizamos para estimar la temperatura media del mes en el que nos encontramos en el momento de hacer la consulta:

```
today = DateValue[Today, {"Year", "Month"}]
```

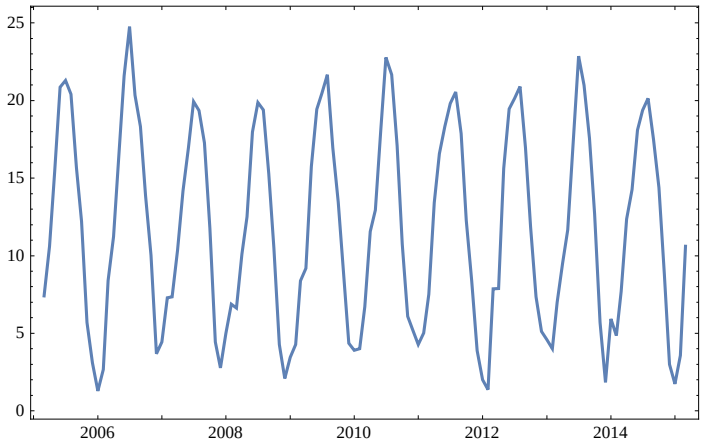
```
{2015, 3}
```

```
tsm[DatePlus[today, {1, "Month"}]]
```

```
7.97269 °C
```

- El ejemplo que sigue estimamos la temperatura medias mensuales para los 12 meses que siguen en el momento de ejecutar las funciones, para ello nos basamos en los datos de los últimos 10 años

```
start = DateValue[DatePlus[Today, {-10, "Year"}], {"Year", "Month"}];
tspec = {start, today, "Month"};
temp = TimeSeries[
   WeatherData["Salamanca", "Temperature", tspec, "Value"], {start, Automatic, "Month"}];
DateListPlot[temp, Joined → True]
```



- Ajustamos los datos a un modelo de serie temporal:

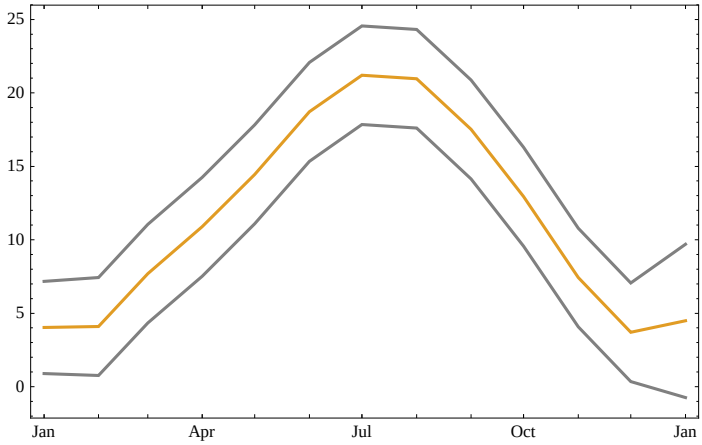
```
tsm = TimeSeriesModelFit[temp]
```

TimeSeriesModel[Family: SARIMA Order: {{1, 0, 0}, {1, 1, 1}$_{12}$}]

- Representamos la estimación de la evolución de la temperatura media mensual para el año que sigue con unas bandas de confianza del 95%:

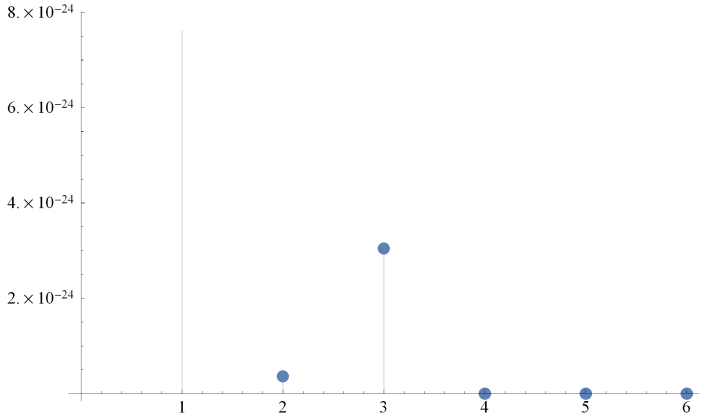
```
fdates = DateRange[today, DatePlus[today, {1, "Year"}], "Month"];
forecast = tsm /@ fdates;
bands = tsm["PredictionLimits"][#] & /@ fdates;
```

```
DateListPlot[{bands[[All, 1]], forecast, bands[[All, 2]]},
 {today, Automatic, "Month"}, Joined → True, PlotStyle → {Gray, Automatic}]
```



- Los datos están altamente autocorrelados,

```
ListPlot[Table[AutocorrelationTest[temp, i], {i, 1, 6}], Filling → Axis]
```



1.5. Modelos de regresión: Ajustes de datos experimentales a funciones

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

En estudios de datos experimentales son frecuentes los casos en los que es conveniente ajustar éstos a una función. Para ello *Mathematica* dispone de numerosas posibilidades. Es recomendable que consulte el tutorial: tutorial/StatisticalModelAnalysis.

En esta sección vamos a mostrar con ejemplos aplicaciones de ajustes en modelos lineales y no lineales. Vamos a utilizar dos ficheros con los datos a ajustar: noisydata.xls y pulsar1257.dat. Estos deben estar en un subdirectorio, llamado Data, del directorio en el que se encuentre el notebook en el que hagamos estos ejercicios.

- Definimos el subdirectorio en el que se encuentran los datos a importar.

```
SetDirectory[FileNameJoin[{NotebookDirectory[], "Data"}]]
```

```
C:\Users\Guillermo\Google Drive\MasAlla10\Data
```

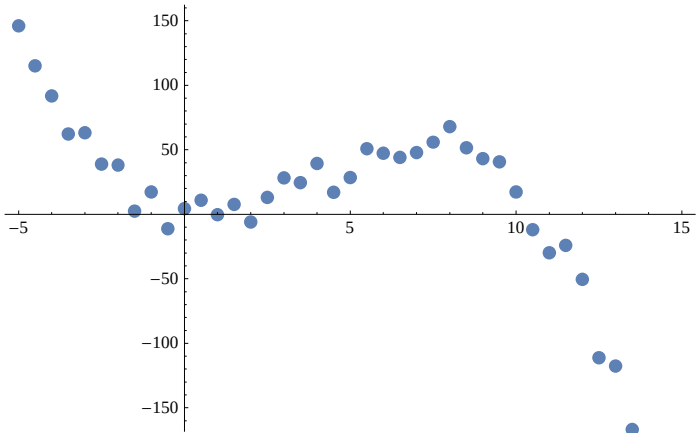
1.5.1. Modelo de regresión lineal

Se van a ajustar los datos importados del fichero noisydata.xls a un polinomio de grado 3 para lo que se utilizará la función `LinearModelFit` aplicable a modelos lineales de regresión lineal y no lineal (por ejemplo: polinomios). Esta función construye un modelo lineal de la forma $\hat{y} = \beta_0 + \beta_1 f_1 + \beta_2 f_2 + \ldots$ (aunque el modelo es lineal las f_i no tienen que ser funciones lineales; puede tratarse, por ejemplo, de polinomios.

Para ser riguroso los datos originales y_i deben ser independientes y estar normalmente distribuidos, con media $\hat{y}_i$ y distribución estándar s común para todos.

- Importamos los datos de un fichero externo en formato *xls*.

```
data = Import["noisydata.xls", {"Data", 1}];
dataplot = ListPlot[data]
```



- Lo ajustamos a un polinomio de tercer grado.

```
fit = LinearModelFit[data, {1, x, x^2, x^3}, x];
```

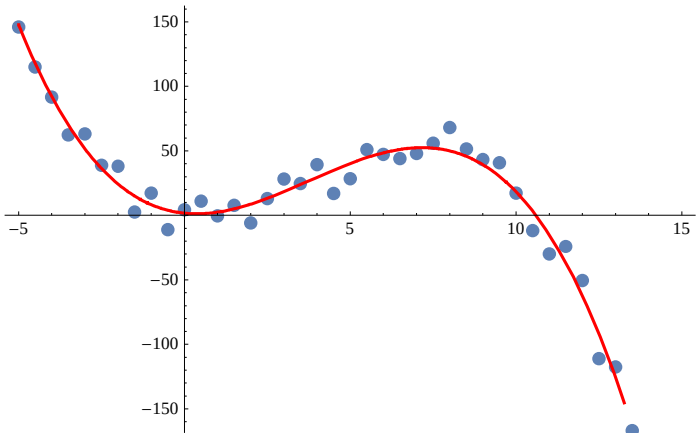
- Mostramos la tabla ANOVA.

```
fit["ANOVATable"]
```

	DF	SS	MS	F- Statistic	P- Value
x	1	165 320.	165 320.	1626.9	3.47061×10^{-32}
x^2	1	58 481.7	58 481.7	575.515	3.78263×10^{-24}
x^3	1	114 956.	114 956.	1131.27	2.41932×10^{-29}
Error	37	3759.8	101.616		
Total	40	342 517.			

- Ahora representamos conjuntamente los datos y la función ajustada.

```
Show[dataplot,
 Plot[fit[x], {x, -5, 15}, PlotStyle → Red]]
```



1.5.2. Modelo de regresión no lineal

Se trata de ajustar los valores experimentales de las medidas de luz realizadas del púlsar PSR1257+12 durante un período de tres años (obtenidos por Alex Wolszczan y disponibles en el fichero pulsar1257.dat) a la función:

$f(t) = \epsilon + \beta \cos(t\,\theta) + \delta \cos(t\,\phi) + \alpha \sin(t\,\theta) + \gamma \sin(t\,\phi)$.

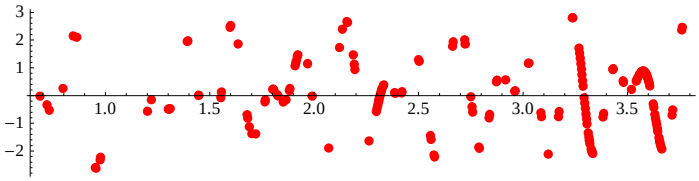
Para ajustes no lineales emplearemos la función `NonlinearModelFit`.

- Importamos el fichero:

```
data = Import["pulsar1257.dat"];
```

- Representamos gráficamente los datos:

```
dataplot = ListPlot[data, AspectRatio → 1/4, PlotStyle → Red]
```



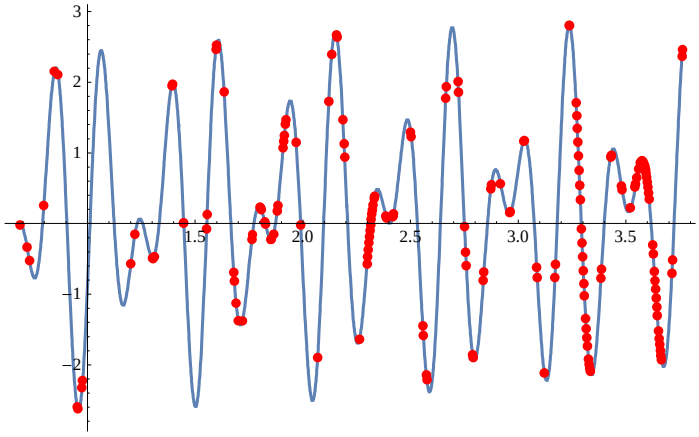
- Los ajustamos a la función $f(t)$ antes indicada:

```
pulsarfit = NonlinearModelFit[data, α Sin[θ t] + β Cos[θ t] + γ Sin[ϕ t] + δ Cos[ϕ t] + ϵ,
  {{α, 1}, {β, 1}, {γ, 1}, {δ, 0}, {θ, 23.31}, {ϕ, 34.64}, {ϵ, 0}}, t]
```

```
FittedModel[ 0.0769581 + 1.33261 ≪1≫ + ≪1≫ ≪1≫ ≪1≫ − 1.29803 Sin[34.5111 t] ]
```

- Lo representamos y vemos el excelente ajuste obtenido:

```
Show[Plot[pulsarfit[t], {t, 0.68, 3.76}],
 dataplot]
```



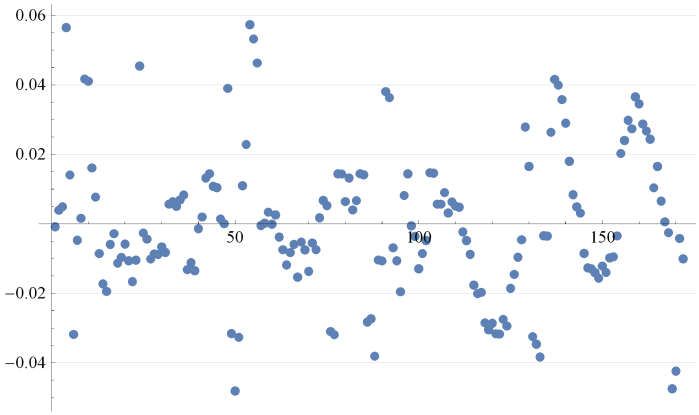
- Podemos realizar distintos tipos de estudios estadísticos. Para ello es conveniente ver las propiedades disponibles para el tipo de ajuste realizado:

```
pulsarfit["Properties"]
```

```
{AdjustedRSquared, AIC, AICc, ANOVATable, ANOVATableDegreesOfFreedom,
 ANOVATableEntries, ANOVATableMeanSquares, ANOVATableSumsOfSquares,
 BestFit, BestFitParameters, BIC, CorrelationMatrix, CovarianceMatrix,
 CurvatureConfidenceRegion, Data, EstimatedVariance, FitCurvatureTable,
 FitCurvatureTableEntries, FitResiduals, Function, HatDiagonal, MaxIntrinsicCurvature,
 MaxParameterEffectsCurvature, MeanPredictionBands, MeanPredictionConfidenceIntervals,
 MeanPredictionConfidenceIntervalTable, MeanPredictionConfidenceIntervalTableEntries,
 MeanPredictionErrors, ParameterBias, ParameterConfidenceIntervals,
 ParameterConfidenceIntervalTable, ParameterConfidenceIntervalTableEntries,
 ParameterConfidenceRegion, ParameterErrors, ParameterPValues, ParameterTable,
 ParameterTableEntries, ParameterTStatistics, PredictedResponse, Properties,
 Response, RSquared, SingleDeletionVariances, SinglePredictionBands,
 SinglePredictionConfidenceIntervals, SinglePredictionConfidenceIntervalTable,
 SinglePredictionConfidenceIntervalTableEntries,
 SinglePredictionErrors, StandardizedResiduals, StudentizedResiduals}
```

- Por ejemplo, aquí se muestran los residuos:

```
ListPlot[pulsarfit["FitResiduals"], GridLines → {None, Automatic}]
```



1.5.3. Ajustes automatizados

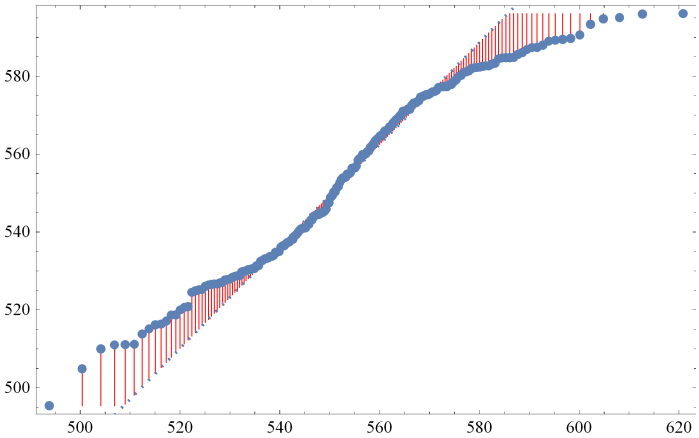
Los datos se ajustan automáticamente a una distribución dada (normalmente es utilizado el método de máxima verosimilitud) empleando la función `EstimatedDistribution`.

- Ajustamos los datos de la cotización de Google a una distribución lognormal. Antes hemos de descargar los datos.

```
googleStock = FinancialData["GOOG", {{2006, 3, 10}, {2015, 1, 1}, "Day"}, "Value"];
dist = EstimatedDistribution[googleStock, LogNormalDistribution[μ, σ]]
LogNormalDistribution[6.31659, 0.0425926]
```

- El ajuste es bueno en su parte central, pero se observa que se aleja en los extremos. De hecho todos los intentos por predecir las cotizaciones en bolsa han sido poco satisfactorios. Pertenecen a lo que Mandelbrot (el padre de la geometria fractal) denomina azar salvaje (la información pasada no nos permite hacer predicciones).

```
QuantilePlot[googleStock, dist, Filling → Automatic, FillingStyle → Red]
```



1.6. Análisis de cluster (grupos)

1.7. Procesos estocásticos

1.8. Análisis de supervivencia y de fiabilidad

1.9. Integración con *RLink*

1.10. Desarrollo de un paquete aplicado a control de calidad

1.11. Recursos adicionales

Para acceder a los tutoriales que se indican a continuación escriba en la ayuda el texto que hay a la derecha de ":".

Guide: Probability and Statistics: guide/ProbabilityAndStatistics

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

Optimization: tutorial/ConstrainedOptimizationOverview

Analisis de supervivencia y fiabilidad:

http://reference.wolfram.com/language/guide/SurvivalAnalysis.html

http://reference.wolfram.com/language/guide/Reliability.html

Series temporales: guide/TimeSeries

Sobre estadística puede encontrar infinidad de demostraciones en:

http://demonstrations.wolfram.com/topic.html?topic=Statistics

http://demonstrations.wolfram.com/topic.html?topic=Clusters.

En el siguiente enlace se muestran ejemplos de cálculos estadísticos realizados por el autor con *Mathematica* y web *Mathematica*. La ayuda incluye una explicación detallada sobre el cálculo de intervalos de tolerancia y otros cálculos estadísticos a los que nos hemos referido en este capítulo.

http://www3.enusa.es/webMathematica/Estadistica/estadistica.htm