Mathematica 8

Roberto Cavaliere

Wolfram Research, Inc.
roberto@wolfram.com

Introduzione

Che *Mathematica* sappia fare i calcoli e offra un linguaggio per implementare modelli matematici di analisi dati è abbastanza noto.

Quello che molti non sanno è che *Mathematica* garantisce un significativo supporto anche in tutti gli altri momenti di un tipico processo e/o contesto di analisi dati in qualsiasi ambito tecnico-scientifico.

In questa presentazione intendo mettere in luce proprio tale aspetto, ossia come *Mathematica* può essere impiegato come strumento di lavoro a partire dalla fase di acquisizione dati fino alla condivisione e pubblicazione dei risultati, attraverso tutti i passi intermedi.

Perchè *Mathematica*? ... principi e vantaggi

Automatismo: decrementa il tempo di prototipazione e sviluppo eliminando passaggi superflui tramite meccanismi di automazione (scelta automatica dell'algoritmo, delle impostazioni di visualizzazione, dei controlli nelle strutture dinamiche (esempio)

Mathematical knowledge: si avvale di una base di conoscenze composta da migliaia di funzioni, algoritmi e metodi (es. NDSolve ha circa 25 differenti algoritmi di risoluzione delle equazioni differenziali) sempre aggiornati agli ultimi risultati della ricerca mondiale (esempio)

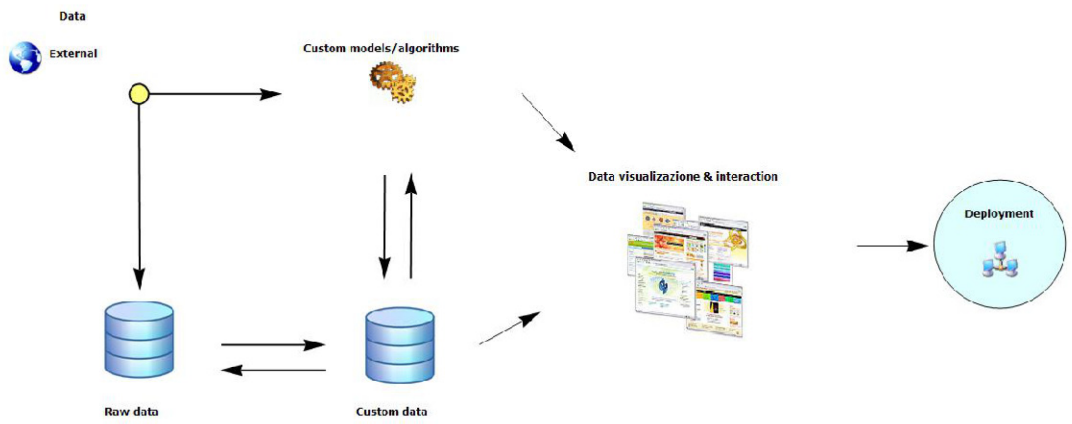
Architettura coerente: incrementa la produttività grazie alla sua capacità di gestire qualsiasi espressione in maniera uniforme e coerente con il principio su cui è stato creato "everything is an expression" (esempio)

Strumento integrato: *Mathematica* include decine di pacchetti specialistici (Database Access Kit, Parallel Computing, Cuda/OpenCL-Link, Control System, Wavelet Analysis, Statistics and Probability, ecc. e dispone di molteplici banche dati estese e curate direttamente accessibili dal codice *Mathematica*

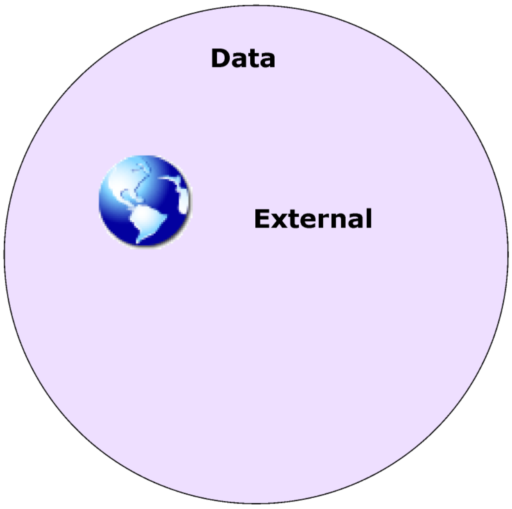
Interfacciabile: *Mathematica* si interfaccia a numerosi altri linguaggi/ambienti: *Mathematica* Link for Excel, *Mathematica* Link for LabView, C, C++, Fortran, Java, .NET, web services, ecc

Linguaggio pluriparadigmatico: *Mathematica* offre set di comandi per diversi paradigmi: procedurale, basato su regole, funzionale

Un tipico processo di flusso e analisi dati



Integrazione con altri linguaggi/ambienti e formati di dati standard



Mathematica consente di importare dati in diversi modi. Il più semplice e comune è tramite il comando Import, che è in grado di riconoscere e trattare oltre cento formati di dati provenienti da altri ambienti e definiti secondo altri standard.

```
In[7]:= SetDirectory[NotebookDirectory[]];
```

```
In[8]:= $ImportFormats
```

```
Out[8]= {3DS, ACO, Affymetrix, AIFF, ApacheLog, ArcGRID, AU, AVI, Base64, BDF, Binary,
  Bit, BMP, Byte, BYU, BZIP2, CDED, Character16, Character8, CIF, Complex128,
  Complex256, Complex64, CSV, CUR, DBF, DICOM, DIF, DIMACS, Directory, DOT,
  DXF, EDF, EPS, ExpressionML, FASTA, FITS, FLAC, GenBank, GeoTIFF, GIF, GPX,
  Graph6, Graphlet, GraphML, GRIB, GTOPO30, GXL, GZIP, HarwellBoeing, HDF,
  HDF5, HTML, ICO, ICS, Integer128, Integer16, Integer24, Integer32, Integer64,
  Integer8, JPEG, JPEG2000, JSON, JVX, KML, LaTeX, LEDA, List, LWO, MAT, MathML,
  MBOX, MDB, MGF, MKV, MMCIF, MOL, MOL2, MPS, MTP, MTX, MX, NASACDF, NB, NDK,
  NetCDF, NEXUS, NOFF, OBJ, ODS, OFF, Package, Pajek, PBM, PCX, PDB, PDF, PGM,
  PLY, PNG, PNM, PPM, PXR, QuickTime, RawBitmap, Real128, Real32, Real64,
  RIB, RSS, RTF, SCT, SDF, SDTS, SDTSDEM, SHP, SMILES, SND, SP3, Sparse6, STL,
  String, SurferGrid, SXC, Table, TAR, TerminatedString, Text, TGA, TGF, TIFF,
  TIGER, TLE, TSV, UnsignedInteger128, UnsignedInteger16, UnsignedInteger24,
  UnsignedInteger32, UnsignedInteger64, UnsignedInteger8, USGSDEM, UUE, VCF, VCS,
  VTK, WAV, Wave64, WDX, XBM, XHTML, XHTMLMathML, XLS, XLSX, XML, XPORT, XYZ, ZIP}
```

Import è in grado di importare sia dal file system su cui è installato *Mathematica* sia da reti intranet/internet, semplicmenete attraverso l'URL.
Questo è un esempio di Import di dati dal sito http://finance.yahoo.com

	France Telecom
Current Values	http://finance.yahoo.com/q?s=FTE
Historical Prices	http://finance.yahoo.com/q/hp?s=FTE+Historical+Prices
Link to time serie	http://ichart.finance.yahoo.com/table.csv?s=FTE&d=8&e=8&f=2010&g=d&a=9&b=20&c=1997&ignore=.csv

In[9]:=
```
fte = Import[
   "http://ichart.finance.yahoo.com/table.csv?s=FTE&d=8&e=8&f=2010&g=d&a=9&b=
    20&c=1997&ignore=.csv"];
```

In[10]:=
```
Export["fte.xls", fte]
```

Out[10]=
```
fte.xls
```

Questo è l'import di dati da un file in formato Comma Separated Value (CSV)

In[11]:=
```
eni = Import["ENI.MI.CSV"];
```

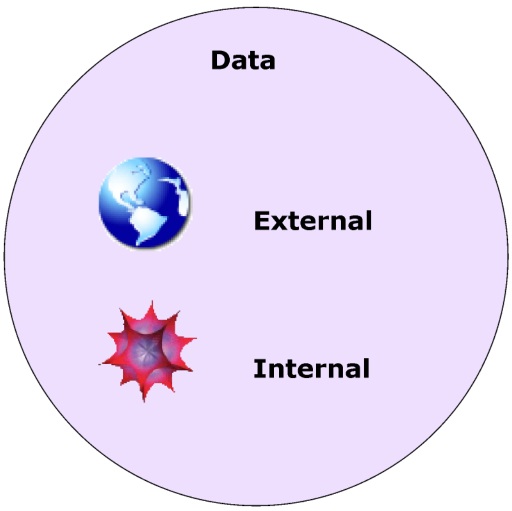
Questo comando visualizza una tabella con le prima 10 righe e prime 6 colonne dell'intera matrice di dati appena importata

In[13]:=
```
Grid[eni[[1 ;; 10, {1, 2, 3, 4, 5, 6}]], Dividers → All]
```

Out[13]=

{2010, 1, 4}	17.94	18.08	17.91	18.05	8.96512×10^6
{2010, 1, 5}	18.12	18.24	18.07	18.23	1.24381×10^7
{2010, 1, 6}	18.14	18.3	18.14	18.25	8.96433×10^6
{2010, 1, 7}	18.23	18.37	18.12	18.36	1.41715×10^7
{2010, 1, 8}	18.37	18.4	18.24	18.37	1.29735×10^7
{2010, 1, 11}	18.51	18.77	18.5	18.56	1.71896×10^7
{2010, 1, 12}	18.5	18.54	18.16	18.19	1.63441×10^7
{2010, 1, 13}	18.1	18.28	18.01	18.23	1.48114×10^7
{2010, 1, 14}	18.37	18.6	18.34	18.5	1.83433×10^7
{2010, 1, 15}	18.55	18.72	18.35	18.35	1.99902×10^7

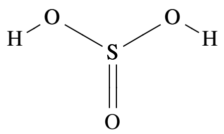
Sorgenti di dati computabili



Oltre alle interfacce con altri ambienti/linguaggi ed al riconoscimento dei principali formati di file, *Mathematica* mette a disposizione una serie di banche dati affidabili, robuste ed aggiornate costantemente. Questo è un elenco completo delle banche dati Computable Data

Vediamo alcuni esempi

```
In[14]:= ChemicalData["Sulfu*"]

Out[14]= {SulfurousAcid, SulfurylFluoride, SulfurTetrafluoride,
  SulfuricAcidSolution, Sulfuretin6Glucoside, Sulfuretin, SulfurylChloride,
  SulfurMonochloride, SulfurTrioxideTrimethylamineComplex, SulfurTrioxide,
  SulfurDioxide, SulfuricAcid, SulfurDichloride, SulfurHexafluoride,
  SulfurTrioxideN,NDimethylformamideComplex, Sulfur32, Sulfur34,
  SulfurylChlorideFluoride, SulfurTrioxidePyridineComplex,
  SulfurChloridePentafluoride, SulfurBromideSSBr2}

In[15]:= ChemicalData["SulfurousAcid"]

Out[15]=
```

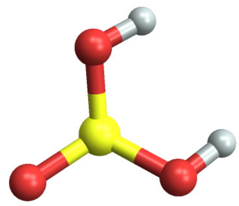


```
In[16]:= ChemicalData["SulfurousAcid", "Properties"]

Out[16]= {AcidityConstant, AcidityConstants, AdjacencyMatrix, AlternateNames,
  AtomPositions, AutoignitionPoint, BeilsteinNumber, BoilingPoint, BondTally,
  CASNumber, CHColorStructureDiagram, CHStructureDiagram, CIDNumber,
  Codons, ColorStructureDiagram, CombustionHeat, CompoundFormulaDisplay,
  CompoundFormulaString, CriticalPressure, CriticalTemperature, Density,
  DensityGramsPerCC, DielectricConstant, DOTHazardClass, DOTNumbers, EdgeRules,
  EdgeTypes, EGECNumber, ElementMassFraction, ElementTally, ElementTypes,
  EUNumber, FlashPoint, FlashPointFahrenheit, FormalCharges, FormattedName,
  GmelinNumber, HBondAcceptorCount, HBondDonorCount, HenryLawConstant,
  HildebrandSolubility, HildebrandSolubilitySI, InChI, IonEquivalents, Ions,
  IonTally, IsoelectricPoint, IsomericSMILES, IUPACName, LogAcidityConstant,
  LowerExplosiveLimit, MDLNumber, MeltingBehavior, MeltingPoint,
  Memberships, MolarVolume, MolecularFormulaDisplay, MolecularFormulaString,
  MolecularWeight, MoleculePlot, Name, NFPAFireRating, NFPAHazards,
  NFPAHealthRating, NFPALabel, NFPAReactivityRating, NonHydrogenCount,
  NonStandardIsotopeCount, NonStandardIsotopeNumbers, NonStandardIsotopeTally,
  NSCNumber, OdorThreshold, OdorType, PartitionCoefficient, pH, Phase,
  RefractiveIndex, Resistivity, RotatableBondCount, RTECSClasses,
  RTECSNumber, SideChainAcidityConstant, SMILES, Solubility, SolubilityType,
  SpaceFillingMoleculePlot, StandardName, StructureDiagram, SurfaceTension,
  TautomerCount, ThermalConductivity, TopologicalPolarSurfaceArea,
  UpperExplosiveLimit, VanDerWaalsConstants, VaporDensity, VaporizationHeat,
  VaporPressure, VaporPressureTorr, VertexCoordinates, VertexTypes, Viscosity}

In[17]:= ChemicalData["SulfurousAcid", "AcidityConstant"]

Out[17]= 0.0169824

In[18]:= ChemicalData["SulfurousAcid", "MolecularFormulaDisplay"]
```

Out[18]= H_2SO_3

In[19]:=
```
ChemicalData["SulfurousAcid", "MoleculePlot"]
```

Out[19]=

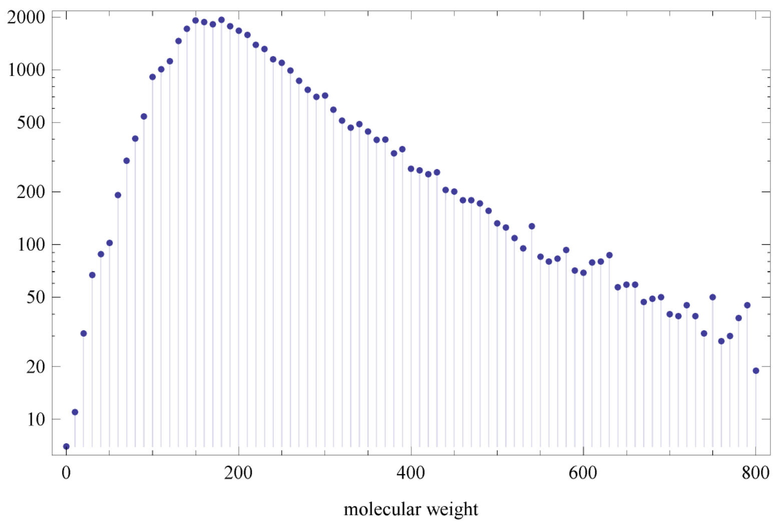


Ovviamente i dati sono disponibili in formato e struttura tali da poter essere immediatamente disponibili in *Mathematica*. Pertanto, si possono programmare anche complesse applicazioni che sfruttano tali dati e creano report, grafici, modelli, ecc. semplicemente richiedendo i dati ai server Wolfram. Ecco un esempio di come una chiamata a ChemicalData si innesta facilmente in una porzione di codice Mathematica

Una distribuzione dei pesi molecolari

In[20]:=
```
ListLogPlot[Transpose[{Table[i, {i, 0, 800, 10}], BinCounts[
     Cases[ChemicalData[#, "MolecularWeight"] & /@ChemicalData[], _Real],
     {-5, 805, 10}]}], Filling → Axis, Frame → True, Axes → None,
  PlotRange → All, FrameLabel → {"molecular weight", None}]
```

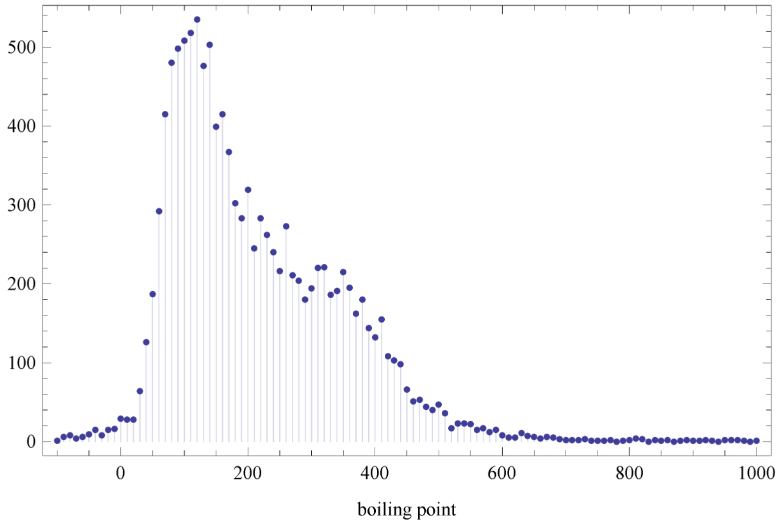
Out[20]=



Una distribuzione dei punti di ebollizione

In[21]:=
```
ListPlot[Transpose[{Table[i, {i, -100, 1000, 10}],
    BinCounts[Cases[ChemicalData[#, "BoilingPoint"] & /@ChemicalData[], _Real],
     {-105, 1005, 10}]}], Filling → Axis, Frame → True,
  Axes → None, PlotRange → All, FrameLabel → {"boiling point", None}]
```

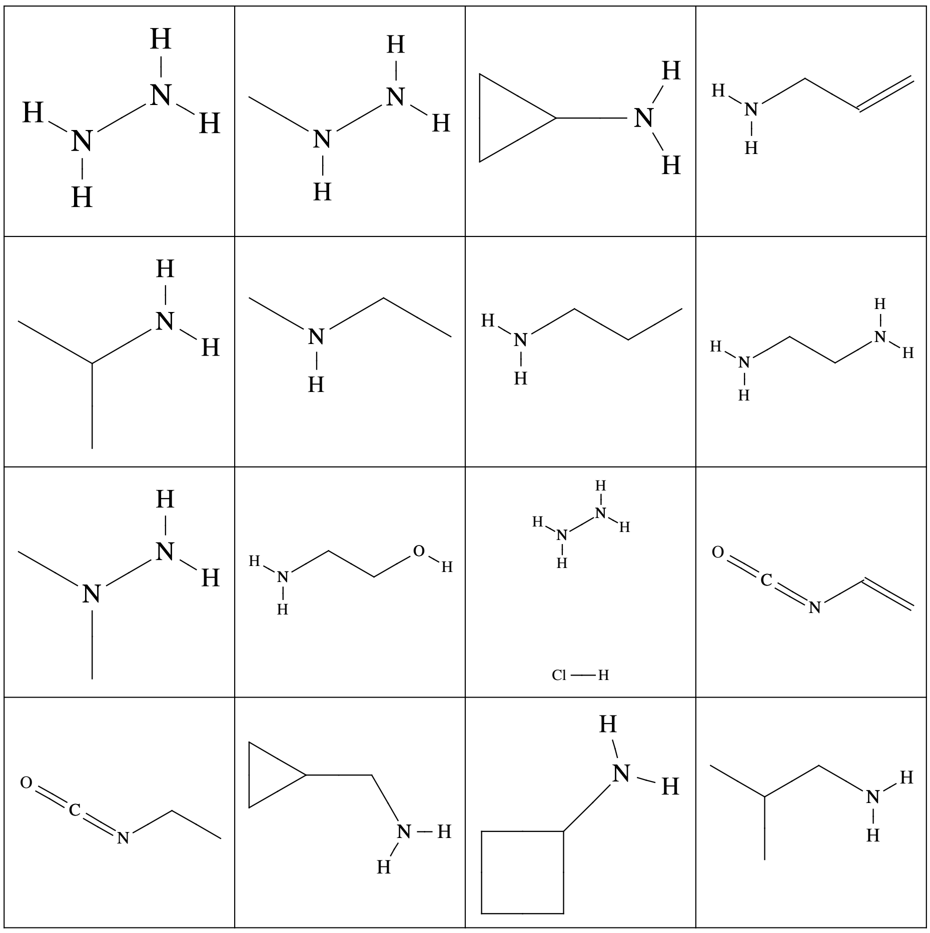
Out[21]=



Una serie di formule

In[22]:=
```
GraphicsGrid[Partition[Show[ChemicalData[#], ImageSize → Tiny] & /@
    Take[ChemicalData["Amines"], 16], 4], Frame → All]
```

Out[22]=

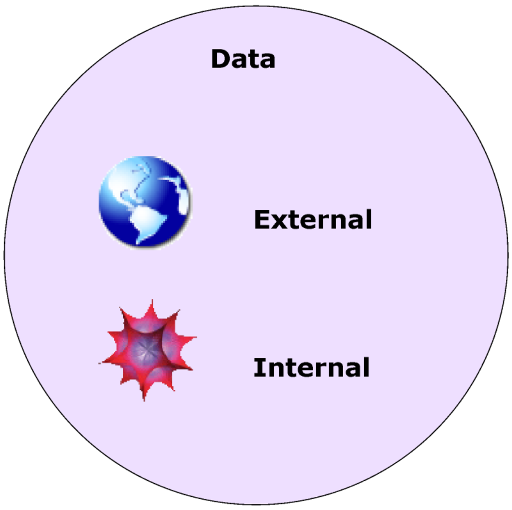


Una tabella formattata

```
vals = Table[ChemicalData[#, prop],
      {prop, {"FormulaDisplay", "MolecularWeight", "AlternateNames"}}] & /@
   ChemicalData[{"Sulfur", "Compound"}];
Text[Grid[Prepend[vals[[1 ;; 10]],
   {"Chemical", "Molecular weight", "Alternate Names"}], Frame -> All,
  Background -> {None, {{{LightBlue, White}}, {1 -> LightYellow}}},
  Alignment -> Left]]
```

Chemical	Molecular weight	Alternate Names
S	32.065	{sulphur}
H_2S	34.081	{sulfane}
32S	34.081	{hydrogen sulfide, sulfane}
^{34}S	35.98375	{sulfane}
D_2S	36.093	{}
BeS	41.077	{beryllium monosulfide, beryllium sulphide}
Li_2S	46.95	{dilithium bisulfide, dilithium sulfanide}
CH_3SNa	48.107	{mercaptomethane, methanethiol, methanethiol sodium salt, methylmercaptan}
CH_3SH	48.107	{mercaptan C1, methyl mercaptan}
NH_4SH	51.111	{ammonium hydrogen sulfide, ammonium hydrosulfide, ammonium sulfide}

Sorgenti di dati computabili



Una particolare sorgente di dati aggiunta in *Mathematica* 8 è quella fornita dal motore di computazione della conoscenza chiamato WolframAlpha. WlA include circa dieci trilioni di data sets sugli argomenti più svariati. Ci sono diversi modi per richiamare WolframApha dall'interno di *Mathematica*. Sia da linea di codice sia programmaticamente. Vediamo alcuni esempi

In[33]:=

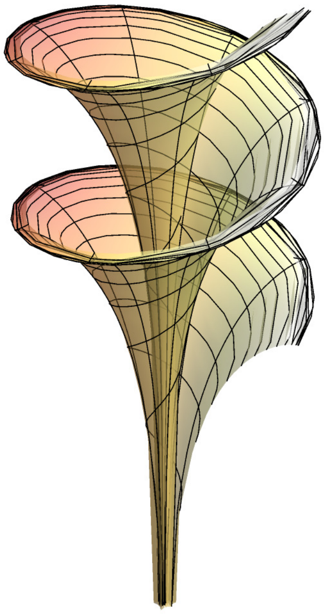
Dini surface

↳ Example plot

```
ParametricPlot3D[{Cos[u] * Sin[v],
  Sin[u] * Sin[v], 0.2 * u + Cos[v] + Log[Tan[0.5 * v]]},
 {u, 0, 4 * Pi}, {v, 0.001, 2},
 {PlotPoints -> 10, MaxRecursion -> Automatic}, PlotLabel ->
  TextCell[Style[Row[{Row[{Style["a", Italic] == 1, ", "}],
      Row[{Style["b", Italic] == 0.2}]}]], GrayLevel[0.5]],
```

$a = 1, b = 0.2$

Out[33]=



In[34]:=

Fermat theorem »

↳ Statement

Out[34]=

No three positive integers a,b,and c can satisfy the equation $a^n+b^n=c^n$ for any integer value of n greater than two.

In[25]:= nutrition facts cheese

Average nutrition facts

Out[25]=

serving size 30 g

total calories 88	fat calories 63
	% daily value*
total fat 6 g	**10%**
saturated fat 4 g	**20%**
trans fat	
cholesterol 21 mg	**7%**
sodium 209 mg	**9%**
total carbohydrates 1 g	**0%**
dietary fiber 12 mg	**0%**
sugar 503 mg	
protein 7 g	**13%**

vitamin A 4%	calcium 17%
iron 1%	vitamin D 2%
thiamin 1%	riboflavin 6%
vitamin B6 1%	vitamin B12 5%
folate 1%	phosphorus 13%
magnesium 2%	zinc 5%

*percent daily values are based on a 2000 calorie diet

(averaged over different types of cheese)

In[26]:=

population history in Italy

```
CountryData["Italy", {"Population", All}]
```

Out[26]=
```
{{{1970, 1, 1, 0, 0, 0}, 5.33593×10^7},
 {{1971, 1, 1, 0, 0, 0}, 5.37396×10^7}, {{1972, 1, 1, 0, 0, 0}, 5.41205×10^7},
 {{1973, 1, 1, 0, 0, 0}, 5.44926×10^7}, {{1974, 1, 1, 0, 0, 0}, 5.48434×10^7},
 {{1975, 1, 1, 0, 0, 0}, 5.51639×10^7}, {{1976, 1, 1, 0, 0, 0}, 5.54505×10^7},
 {{1977, 1, 1, 0, 0, 0}, 5.57048×10^7}, {{1978, 1, 1, 0, 0, 0}, 5.59294×10^7},
 {{1979, 1, 1, 0, 0, 0}, 5.6129×10^7}, {{1980, 1, 1, 0, 0, 0}, 5.63073×10^7},
 {{1981, 1, 1, 0, 0, 0}, 5.64657×10^7}, {{1982, 1, 1, 0, 0, 0}, 5.66037×10^7},
 {{1983, 1, 1, 0, 0, 0}, 5.67201×10^7}, {{1984, 1, 1, 0, 0, 0}, 5.68132×10^7},
 {{1985, 1, 1, 0, 0, 0}, 5.68828×10^7}, {{1986, 1, 1, 0, 0, 0}, 5.69275×10^7},
 {{1987, 1, 1, 0, 0, 0}, 5.69508×10^7}, {{1988, 1, 1, 0, 0, 0}, 5.69624×10^7},
 {{1989, 1, 1, 0, 0, 0}, 5.6975×10^7}, {{1990, 1, 1, 0, 0, 0}, 5.69977×10^7},
 {{1991, 1, 1, 0, 0, 0}, 5.70398×10^7}, {{1992, 1, 1, 0, 0, 0}, 5.70993×10^7},
 {{1993, 1, 1, 0, 0, 0}, 5.71606×10^7}, {{1994, 1, 1, 0, 0, 0}, 5.72009×10^7},
 {{1995, 1, 1, 0, 0, 0}, 5.72068×10^7}, {{1996, 1, 1, 0, 0, 0}, 5.71686×10^7},
 {{1997, 1, 1, 0, 0, 0}, 5.70992×10^7}, {{1998, 1, 1, 0, 0, 0}, 5.70361×10^7},
 {{1999, 1, 1, 0, 0, 0}, 5.70299×10^7}, {{2000, 1, 1, 0, 0, 0}, 5.7116×10^7},
 {{2001, 1, 1, 0, 0, 0}, 5.73063×10^7}, {{2002, 1, 1, 0, 0, 0}, 5.75857×10^7},
 {{2003, 1, 1, 0, 0, 0}, 5.79273×10^7}, {{2004, 1, 1, 0, 0, 0}, 5.82907×10^7},
 {{2005, 1, 1, 0, 0, 0}, 5.86448×10^7}, {{2006, 1, 1, 0, 0, 0}, 5.89819×10^7},
 {{2007, 1, 1, 0, 0, 0}, 5.93047×10^7}, {{2008, 1, 1, 0, 0, 0}, 5.96037×10^7}}
```

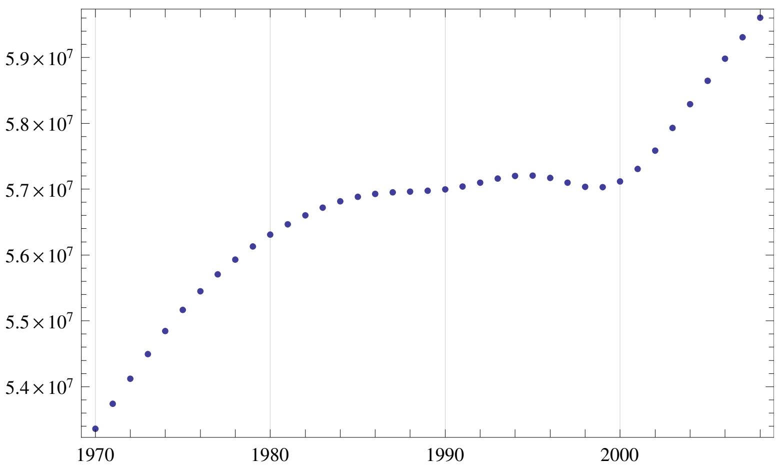
Un esempio di funzioni applicata al risultato di una interrogazione eseguita con WolframAlpha. DateListPlot realizza il grafico dell'andamento della popolazione italiana basandosi sui dati restituiti da WlA

In[36]:=

populatio history in Italy

```
DateListPlot[CountryData["Italy", {"Population", All}]]
```

Out[36]=



In[37]:= **how far is Milan from Rome** »

↳ Distance

Out[37]= 298.1 miles

In[27]:= **GDP history in France**

```
CountryData["France", {"GDP", All}]
```

Out[27]= {{{1970, 1, 1, 0, 0, 0}, 1.46982×10^{11}},
{{1971, 1, 1, 0, 0, 0}, 1.64323×10^{11}}, {{1972, 1, 1, 0, 0, 0}, 2.01841×10^{11}},
{{1973, 1, 1, 0, 0, 0}, 2.62571×10^{11}}, {{1974, 1, 1, 0, 0, 0}, 2.82826×10^{11}},
{{1975, 1, 1, 0, 0, 0}, 3.57042×10^{11}}, {{1976, 1, 1, 0, 0, 0}, 3.68786×10^{11}},
{{1977, 1, 1, 0, 0, 0}, 4.06857×10^{11}}, {{1978, 1, 1, 0, 0, 0}, 5.01787×10^{11}},
{{1979, 1, 1, 0, 0, 0}, 6.06786×10^{11}}, {{1980, 1, 1, 0, 0, 0}, 6.91157×10^{11}},
{{1981, 1, 1, 0, 0, 0}, 6.04412×10^{11}}, {{1982, 1, 1, 0, 0, 0}, 5.7335×10^{11}},
{{1983, 1, 1, 0, 0, 0}, 5.47934×10^{11}}, {{1984, 1, 1, 0, 0, 0}, 5.20232×10^{11}},
{{1985, 1, 1, 0, 0, 0}, 5.43069×10^{11}}, {{1986, 1, 1, 0, 0, 0}, 7.59905×10^{11}},
{{1987, 1, 1, 0, 0, 0}, 9.22339×10^{11}}, {{1988, 1, 1, 0, 0, 0}, 1.00337×10^{12}},
{{1989, 1, 1, 0, 0, 0}, 1.00811×10^{12}}, {{1990, 1, 1, 0, 0, 0}, 1.24442×10^{12}},
{{1991, 1, 1, 0, 0, 0}, 1.24401×10^{12}}, {{1992, 1, 1, 0, 0, 0}, 1.37269×10^{12}},
{{1993, 1, 1, 0, 0, 0}, 1.29113×10^{12}}, {{1994, 1, 1, 0, 0, 0}, 1.36428×10^{12}},
{{1995, 1, 1, 0, 0, 0}, 1.56989×10^{12}}, {{1996, 1, 1, 0, 0, 0}, 1.57369×10^{12}},
{{1997, 1, 1, 0, 0, 0}, 1.4244×10^{12}}, {{1998, 1, 1, 0, 0, 0}, 1.47174×10^{12}},
{{1999, 1, 1, 0, 0, 0}, 1.45742×10^{12}}, {{2000, 1, 1, 0, 0, 0}, 1.32796×10^{12}},
{{2001, 1, 1, 0, 0, 0}, 1.33976×10^{12}}, {{2002, 1, 1, 0, 0, 0}, 1.4574×10^{12}},
{{2003, 1, 1, 0, 0, 0}, 1.79994×10^{12}}, {{2004, 1, 1, 0, 0, 0}, 2.06141×10^{12}},
{{2005, 1, 1, 0, 0, 0}, 2.14653×10^{12}}, {{2006, 1, 1, 0, 0, 0}, 2.26614×10^{12}},
{{2007, 1, 1, 0, 0, 0}, 2.59315×10^{12}}, {{2008, 1, 1, 0, 0, 0}, 2.85653×10^{12}}}

In[28]:= **boiling point of sulphur** »

```
ElementData["Sulfur", "BoilingPoint"]
```

Out[28]= 444.72

In[29]:= **earthquake in Italy 1980**

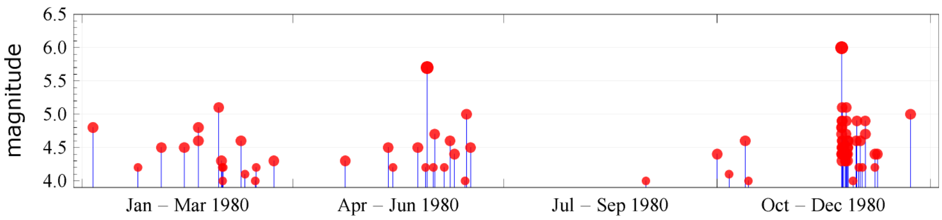
↳ Results (1 of 3)

Input interpretation:

earthquakes	Italy	1980

Results: Show local map | Magnitude > 4 | CST

• Timeline:



• List:

magnitude	time	location
6.0	Sun, Nov 23, 1980 12:34 pm CDT (30 years ago)	6 mi NNW of Pescopagano, Basilicata, Italy
5.7	Wed, May 28, 1980 02:51 pm CDT (30.5 years ago)	32 mi NNW of Santo Stefano Di Camastra, Sicily, Italy
5.1	Tue, Nov 25, 1980 11:06 am CDT (30 years ago)	4 mi S of Muro Lucano, Basilicata, Italy

Out[29]=

Altro esempio di interrogazione da codice

In[30]:=
```
info = WolframAlpha["weather Rome Italy", "PodInformation"];
```

In[31]:=
```
ids = Rest[DeleteDuplicates[#[[1, 1, 1]] & /@ info]];
titles = Map[{{#, 0}, "Title"} &, ids] /. info;
contents = Column[Cases[info, _[{{#, _}, "Content"}, val_] :→ val]] & /@ ids;
MenuView[Thread[titles → contents], ImageSize → Automatic]
```

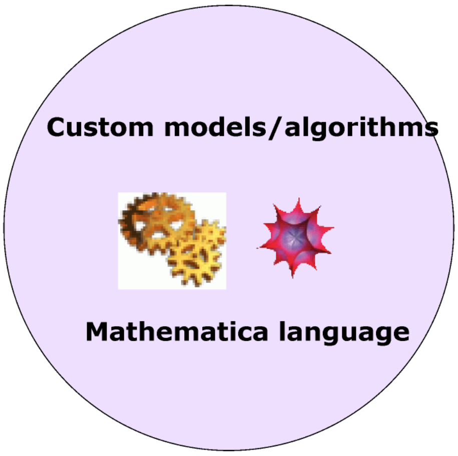
Out[34]=

Latest recorded weather for Rome

temperature	41 °F (wind chill: 39 °F)
conditions	partly cloudy
relative humidity	87% (dew point: 37 °F)
wind speed	3 mph

(13 hours 33 minutes ago)

Il linguaggio *Mathematica*



Mathematica combina il linguaggio simbolico con quello numerico in maniera automatica e spesso trasparente all'utente. Dopo oltre venti anni di sviluppo, *Mathematica* include una notevole conoscenza matematica, fatta di algoritmi, teoremi, regole e metodi per la risoluzione di problemi semplici o complessi.

■ Calcolo numerico

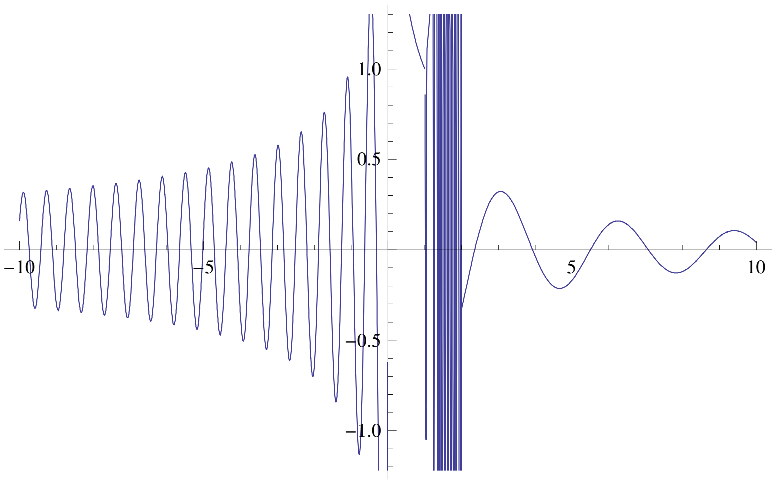
Questo è un esempio di integrazione numerica di una funzione discontinua.

In[35]:=
$$\texttt{fun} = \begin{cases} \frac{\texttt{Sin[10 x]}}{\sqrt{-\texttt{x}}} & -\infty < \texttt{x} < 0 \\ \frac{1}{\sqrt{\texttt{x}}} & 0 < \texttt{x} < 1 \\ \texttt{Sin[2000 x] } \texttt{x}^2 & 1 < \texttt{x} < 2 \\ \frac{\texttt{Cos[2 x]}}{\texttt{x}} & 2 < \texttt{x} < \infty \end{cases};$$

Questo è il grafico della funzione

In[37]:=
```
Plot[fun, {x, -10, 10}]
```

Out[37]=



Ora si effettua il calcolo dell'integrale

In[4]:=
```
NIntegrate[fun, {x, -Infinity, Infinity}]
```

Out[4]= 1.74592

Comparazione di alcuni risultati dell'integrale al variare dell'intervallo di integrazione, tra Mathematica e Matlab

Range	*Mathematica*	Matlab	Symbolic
{0,1}	2.	1.9999	2
{1,2}	0.00127502	0.0015	0.0012750155
{2,3}	0.07292445	0.0729	0.07292445399
{−2,−1}	0.116355743	0.1164	0.11635574348
{−2,3}	1.708268721	1.7006−0.0001i	1.70826880612
{−3,3}	1.685818105	NaN−1.1773i	1.68581826098
{−10,10}	1.817529696	NaN	1.81753064471
{−1,10}	0.186676534	0.1116	0.18667653426

Mathematica permette di controllare la precisione in qualsiasi calcolo

```
In[5]:= NIntegrate[fun, {x, 1, 2}]

Out[5]= 0.00127502

In[6]:= NIntegrate[fun, {x, 1, 2}, WorkingPrecision -> 32]

Out[6]= 0.0012750155303801011420754467680715

In[7]:= RandomReal[100, WorkingPrecision -> 64]

Out[7]= 41.74064907284451200825954770391284837303480091627367277850161570
```

■ Propagazione degli errori e utilizzo precisione arbitraria

```
In[2]:= f[x_] := 11 x - 2
```

Questa funzione ha un punto fisso in x=0.2

```
In[3]:= Reduce[f[x] == x]
```

Out[3]= $x == \frac{1}{5}$

Si osserva come computazioni ripetute applicando la funzione nel suo punto fisso (che scriviamo come 0.2 e non $\frac{1}{5}$) producano risultati inattesi

```
In[4]:= NestList[g, start, 3]

Out[4]= {start, g[start], g[g[start]], g[g[g[start]]]}

In[5]:= Manipulate[NestList[f, 0.2, iterazioni], {iterazioni, 10, 100, 1}]
```

iterazioni

Out[5]=
```
{0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
 0.200005, 0.200051, 0.200557, 0.206132, 0.267457, 0.942028,
 8.36231, 89.9854, 987.84, 10 864.2, 119 505., 1.31455 × 10^6,
 1.446 × 10^7, 1.5906 × 10^8, 1.74966 × 10^9, 1.92463 × 10^10}
```

Il motivo è che mentre 0.2 può essere pensato come un numero decimale esatto (0.20000000000000000000000...) tale numero non può essere rappresentato in maniera esatta con un numero finito di numeri nel sistema binario:

In[6]:=
```
BaseForm[0.2, 2]
```

Out[6]//BaseForm=

$0.0011001100110011001101_2$

Usando la precisione macchina, questo numero viene troncato a 32 bit. Ne segue che, l'ultima cifra troncata è un 1, l'ultima cifra utilizzata viene arrotondata per eccesso.

In[7]:=
```
BaseForm[0.2`20, 2]
```

Out[7]//BaseForm=

$0.00110011001100110011001100110011001100110011001100110011001100110011_2$

In *Mathematica* si può spostare l'errore in avanti e lavorare con la precisione desiderata. L'esempio che segue impiega 30 decimali di precisione

In[8]:=
```
Manipulate[NestList[f, 0.2`50, iterazioni], {iterazioni, 10, 100, 1}]
```

iterazioni

Out[8]=
```
{0.2000000000000000000000000000000000000000000000000,
 0.200000000000000000000000000000000000000000000000,
 0.20000000000000000000000000000000000000000000000,
 0.2000000000000000000000000000000000000000000000,
 0.200000000000000000000000000000000000000000000,
 0.20000000000000000000000000000000000000000000,
 0.2000000000000000000000000000000000000000000,
 0.200000000000000000000000000000000000000000,
 0.20000000000000000000000000000000000000000,
 0.2000000000000000000000000000000000000000,
 0.200000000000000000000000000000000000000,
 0.20000000000000000000000000000000000000,
 0.2000000000000000000000000000000000000,
 0.200000000000000000000000000000000000,
 0.20000000000000000000000000000000000,
 0.2000000000000000000000000000000000,
 0.200000000000000000000000000000000}
```

Si noti che il risultato, ora, è attendibile per un numero più elevato di computazioni. Si noti, inoltre, che *Mathematica* individua il numero di cifre affidabili e mostra solo tali valori.

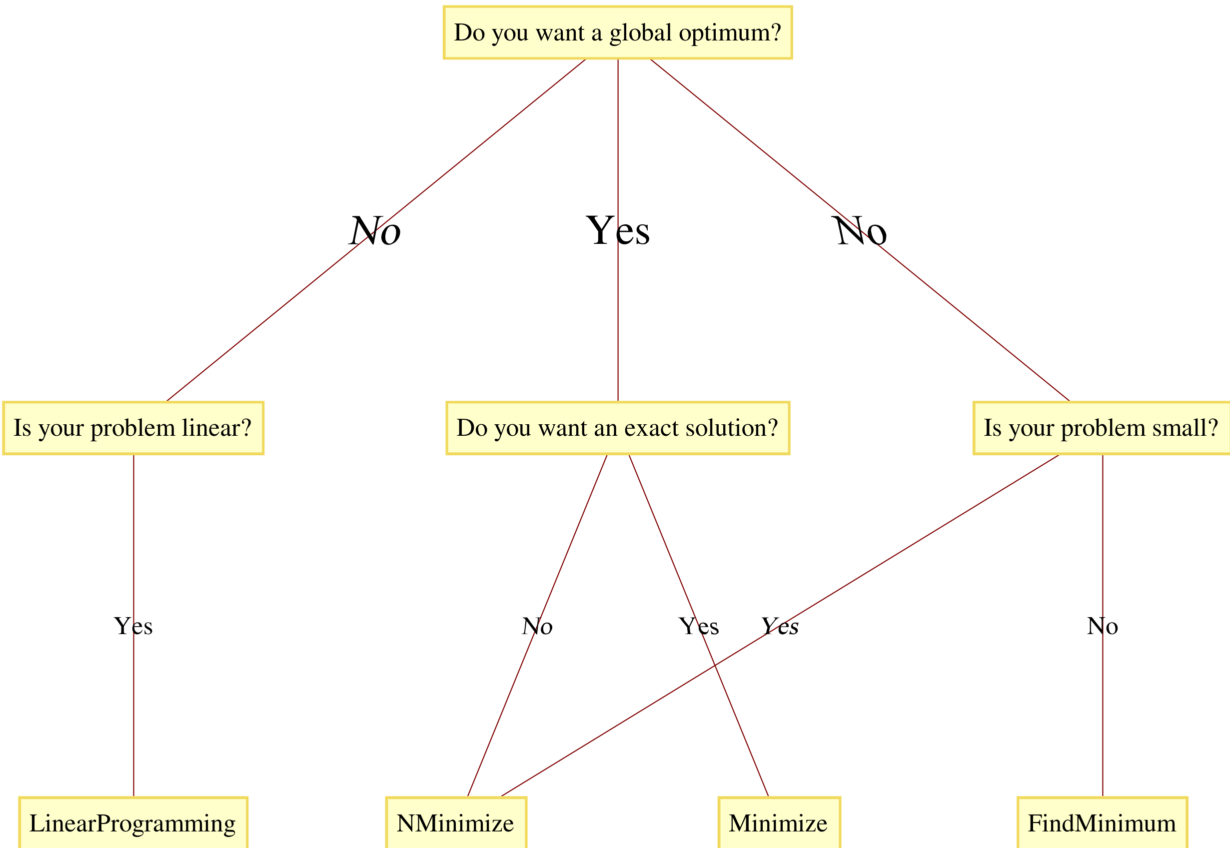
Si osserva così che il primo risultato mostra 50 decimali corretti, mentre gli ultimi due risultati sono completamente inaffidabili.
Ovviamente, con *Mathematica* si può anche usare l'aritmetica esatta.

In[9]:= $\texttt{NestList}\left[\texttt{f}, \frac{1}{5}, 30\right]$

Out[9]= $\left\{\frac{1}{5}, \frac{1}{5}\right\}$

■ Ottimizzazioni e data fitting

Integrate in Mathematica troviamo diverse tecniche di ottimizzazioni locale e globale, sia a livello simbolico che numerico, inclusi l'ottimizzazione con vincoli, il metodo dei punti interni e la programmazone intera.



Mathematica dispone di numerose “macro” già pronte all’uso per l’analisi di dati. Ovviamente, la potenza del suo linguaggio consente anche all’utente di programmare i propri algoritmi laddove si necessita di integrare il sistema con nuovi modelli e/o algoritmi.
Facciamo alcuni esempi. Il fitting dei dati sperimentali: `FindFit`, `LinearModelFit`, `NonlinearModelFit`, `GeneralizedLinearModelFit`

In[17]:=
```
data = {{0, 1}, {1, 0}, {3, 2}, {5, 4}, {6, 4}, {7, 5}};
```

In[18]:=
```
nlm = NonlinearModelFit[data, Log[a + b x^2], {a, b}, x]
```

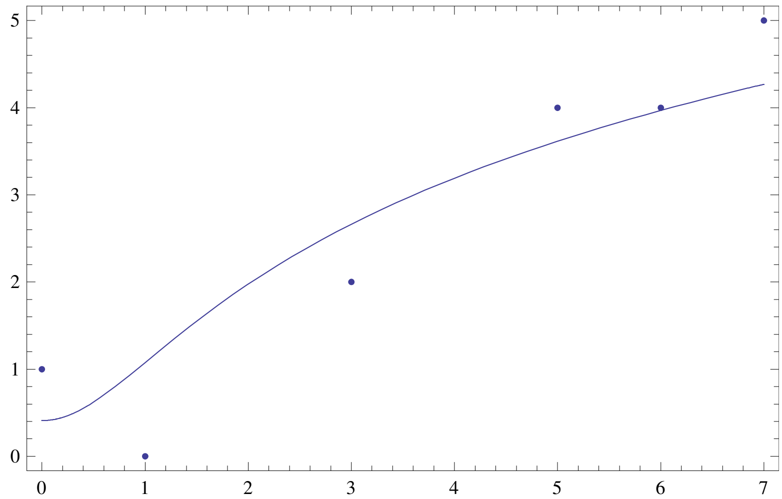
Out[18]= FittedModel[$\boxed{\text{Log}[1.50632 + 1.42633\, x^2]}$]

In[19]:=
```
nlm[2.3]
```

Out[19]= 2.20294

In[20]:=
```
Show[ListPlot[data], Plot[nlm[x], {x, 0, 7}], Frame → True]
```

Out[20]=

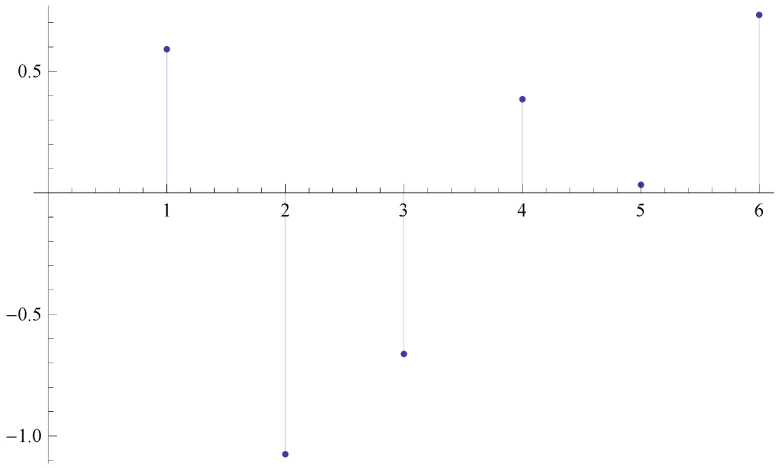


In[21]:=
```
nlm["Properties"]
```

Out[21]=
```
{AdjustedRSquared, AIC, 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}
```

In[22]:=
```
ListPlot[nlm["FitResiduals"], Filling → Axis]
```

Out[22]=



In[23]:=
```
nlm["ANOVATable"]
```

Out[23]=

	DF	SS	MS
Model	2	59.3695	29.6848
Error	4	2.63047	0.657618
Uncorrected Total	6	62.	
Corrected Total	5	19.3333	

Un esempio dinamico

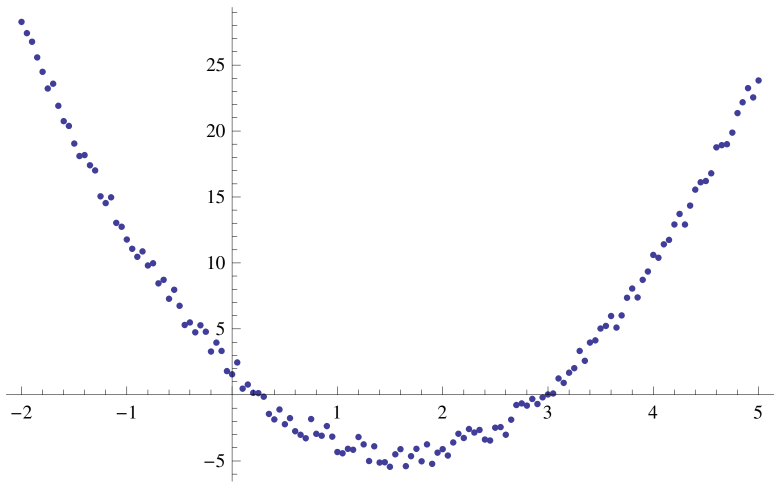
In questo esempio utilizziamo le funzionalità dinamiche per individuare sperimentalmente un fitting opportuno per i dati sperimentali

In[9]:=
```
pData = Table[{x, 2.5 x^2 - 8 x + 1.87 + RandomReal[{-1, 1}]}, {x, -2, 5, 0.05}];
```

Dopo aver generato i dati ne facciamo un primo grafico

In[25]:=
```
ListPlot[pData]
```

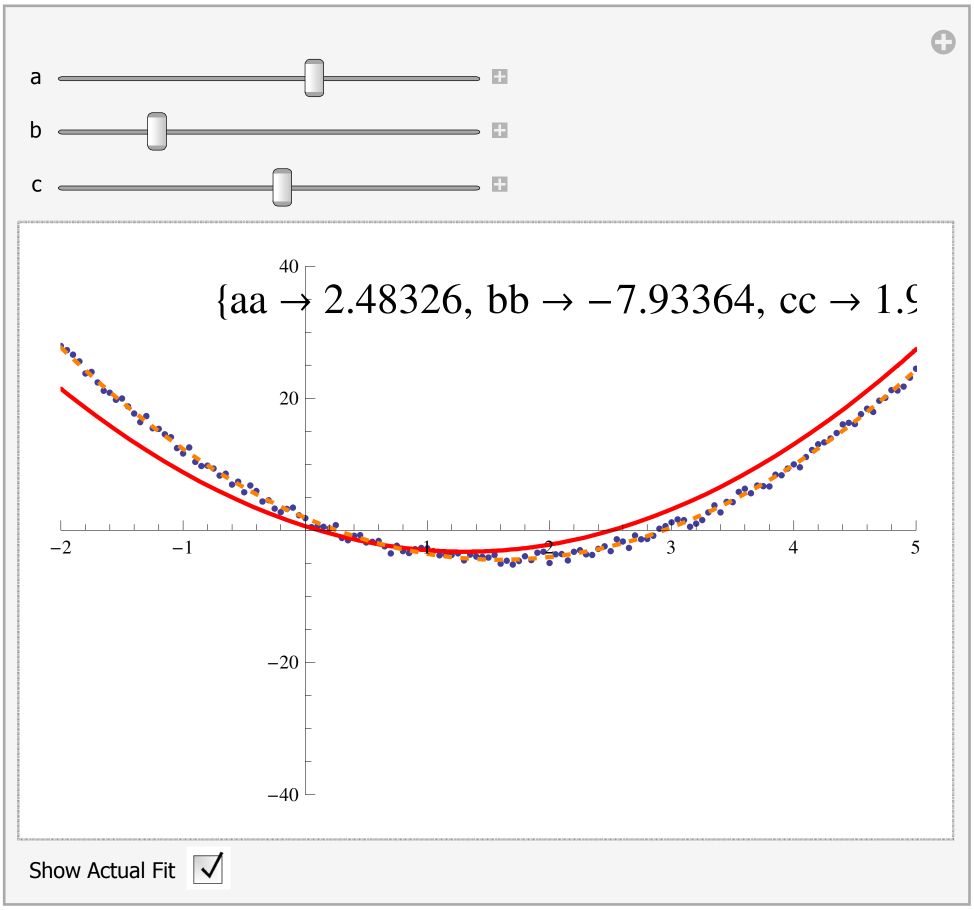
Out[25]=



Ora creiamo l'interfaccia per la manipolazione del fitting

In[13]:=
```
Manipulate[
 Show[ListPlot[pData], Plot[a x^2 + b x + c, {x, -2, 5}, PlotStyle → {Thick, Red}],
  If[fit, Plot[aa x^2 + bb x + cc /. funzione, {x, -2, 5},
    PlotStyle → {Thick, Dashed, Orange}], Graphics[]], ImageSize → Large,
  Epilog → If[fit, Style[Text[funzione, {2.5, 35}], 20], Text[""]],
  PlotRange → {{-2, 5}, {-40, 40}}], {a, -10, 10}, {b, -10, 10}, {c, -10, 10},
 {{fit, False, "Show Actual Fit"}, {True, False}, ControlPlacement → Bottom},
 Initialization :→ (funzione = FindFit[pData, aa x^2 + bb x + cc, {aa, bb, cc}, x])]
```

Out[13]=



▪ Manipolazione di stringhe

In *Mathematica* la programmazione non è intesa solo come la stesura di package articolati e complessi. Spesso ci si trova a programmare e a risolvere problemi semplicemente scrivendo poche linee di codice. Questo è il vantaggio di uno strumento integrato e *general purpose* che mette a disposizione sia i costrutti di base sia una vasta gamma di *macro* che con una sola funzione risolvono task che con altri linguaggi richiederebbero decine o centinaia di righe di codice.

Esempio: manipolazione di stringhe analisi testi e interazione

In[27]:=
```
SetDirectory[NotebookDirectory[]];
```

In[29]:=
```
words = ToLowerCase[StringCases[Import["testo.txt"], WordCharacter ..]];
```

In[30]:=
```
Length[words]
```

Out[30]= 26 733

In[31]:=
```
Length[Union[words]]
```

Out[31]=
```
5638
```

Questo conteggio mostra le dieci parole con maggiore ripetizione nel testo

In[50]:=
```
Take[SortBy[Tally[words], Last], -100]
```

Out[50]=
```
{{dal, 35}, {son, 35}, {subito, 35}, {uno, 35}, {aver, 36}, {conte, 36},
 {nella, 36}, {rodrigo, 37}, {delle, 38}, {fosse, 38}, {ora, 38}, {fatto, 39},
 {ho, 40}, {mai, 40}, {me, 40}, {vi, 40}, {giorno, 41}, {tra, 42},
 {cristoforo, 43}, {lei, 43}, {sempre, 43}, {bene, 45}, {ci, 45}, {far, 46},
 {gran, 46}, {quale, 46}, {tempo, 46}, {chi, 47}, {loro, 47}, {abbondio, 48},
 {anche, 48}, {col, 51}, {quando, 52}, {quella, 52}, {perché, 53},
 {signor, 53}, {tutti, 53}, {qualche, 54}, {nel, 55}, {senza, 55}, {ogni, 56},
 {questa, 56}, {uomo, 56}, {all, 57}, {altro, 57}, {lui, 57}, {ha, 58},
 {lucia, 59}, {tutto, 61}, {de, 62}, {mi, 62}, {ne, 62}, {dell, 63}, {ch, 64},
 {cosa, 65}, {alla, 66}, {poi, 70}, {aveva, 72}, {due, 74}, {padre, 74},
 {così, 76}, {questo, 79}, {renzo, 79}, {sua, 82}, {suo, 87}, {disse, 90},
 {della, 91}, {lo, 91}, {o, 91}, {don, 92}, {io, 98}, {quel, 105}, {s, 106},
 {se, 119}, {al, 126}, {gli, 148}, {i, 148}, {come, 149}, {è, 153}, {era, 164},
 {più, 171}, {del, 173}, {da, 174}, {ma, 181}, {d, 212}, {l, 213}, {una, 223},
 {si, 237}, {le, 238}, {con, 263}, {per, 307}, {la, 393}, {in, 399},
 {non, 414}, {un, 494}, {a, 542}, {il, 552}, {di, 714}, {che, 739}, {e, 1014}}
```

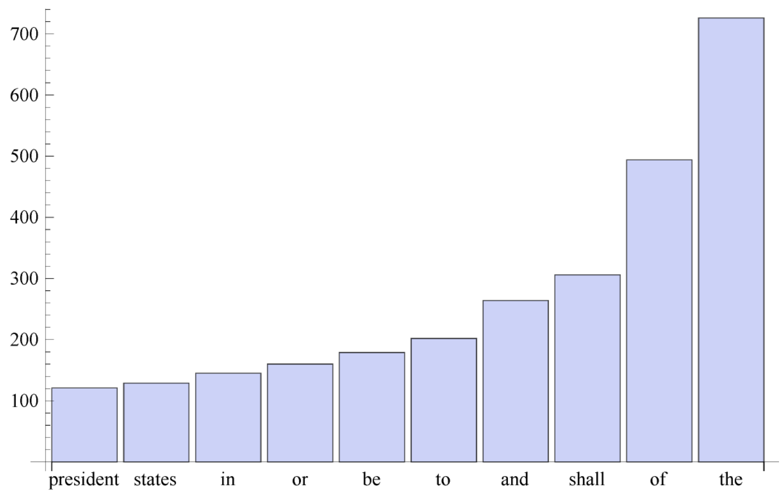
Lo stesso calcolo per il testo della costituzione americana

In[51]:=
```
fr = Take[SortBy[
   Tally[ToLowerCase[StringCases[Import["ExampleData/USConstitution.txt"],
      WordCharacter ..]]], Last], -10]
```

Out[51]=
```
{{president, 121}, {states, 129}, {in, 145}, {or, 160},
 {be, 179}, {to, 202}, {and, 264}, {shall, 306}, {of, 494}, {the, 726}}
```

In[52]:=
```
BarChart[Part[fr, All, 2], ChartLabels → Part[fr, All, 1]]
```

Out[52]=



■ Funzionalità di statistica e probabilità

Mathematica 8 introduce una sezione completamente rivista e potenziata relativa alla nalisi statistica dei dati ed alle probabilità

In[10]:=
```
?? *Distribution
```

▼ System`

ArcSinDistribution	ExpGammaDistribution	LogNormalDistribution	PriceGraphDistribution
BarabasiAlbertGraphDistribution	ExponentialDistribution	LogSeriesDistribution	ProbabilityDistribution
BatesDistribution	ExponentialPowerDistribution	MarginalDistribution	ProductDistribution
BeckmannDistribution	ExtremeValueDistribution	MaxStableDistribution	RayleighDistribution
BenfordDistribution	FisherHypergeometricDistribution	MaxwellDistribution	RiceDistribution
BeniniDistribution	FisherZDistribution	MinStableDistribution	SechDistribution
BenktanderGibratDistribution	FRatioDistribution	MixtureDistribution	SinghMaddalaDistribution
BenktanderWeibullDistribution	FrechetDistribution	MoyalDistribution	SkellamDistribution
BernoulliDistribution	GammaDistribution	MultinomialDistribution	SkewNormalDistribution
BernoulliGraphDistribution	GeometricDistribution	MultinormalDistribution	SmoothKernelDistribution
BetaBinomialDistribution	GompertzMakehamDistribution	MultivariateHypergeometricDistribution	StableDistribution
BetaDistribution	GumbelDistribution	MultivariatePoissonDistribution	StudentTDistribution
BetaNegativeBinomialDistribution	HalfNormalDistribution	MultivariateTDistribution	SurvivalDistribution
BetaPrimeDistribution	HistogramDistribution	NakagamiDistribution	SuzukiDistribution
BinomialDistribution	HotellingTSquareDistribution	NegativeBinomialDistribution	TransformedDistribution
BinormalDistribution	HoytDistribution	NegativeMultinomialDistribution	TriangularDistribution
BirnbaumSaundersDistribution	HyperbolicDistribution	NoncentralBetaDistribution	TruncatedDistribution
BorelTannerDistribution	HypergeometricDistribution	NoncentralChiSquareDistribution	TukeyLambdaDistribution
CauchyDistribution	InverseChiSquareDistribution	NoncentralFRatioDistribution	UniformDistribution
CensoredDistribution	InverseGammaDistribution	NoncentralStudentTDistribution	UniformGraphDistribution
ChiDistribution	InverseGaussianDistribution	NormalDistribution	UniformSumDistribution
ChiSquareDistribution	JohnsonDistribution	OrderDistribution	VonMisesDistribution
CopulaDistribution	KDistribution	ParameterMixtureDistribution	WakebyDistribution
DagumDistribution	KernelMixtureDistribution	ParetoDistribution	WalleniusHypergeometricDistribution
DataDistribution	KumaraswamyDistribution	PascalDistribution	WaringYuleDistribution
DavisDistribution	LandauDistribution	PearsonDistribution	WattsStrogatzGraphDistribution
DegreeGraphDistribution	LaplaceDistribution	PERTDistribution	WeibullDistribution
DirichletDistribution	LevyDistribution	PiecewiseUniformDistribution	WignerSemicircleDistribution
DiscreteUniformDistribution	LindleyDistribution	PoissonConsulDistribution	ZipfDistribution

EmpiricalDistribution	LogGammaDistribution	PoissonDistribution	
ErlangDistribution	LogisticDistribution	PolyaAeppliDistribution	
EstimatedDistribution	LogLogisticDistribution	PowerDistribution	

```
In[3]:= RandomReal[1, {10^6, 5}]; // Timing
Out[3]= {0.171, Null}
```

```
In[4]:= RandomInteger[10, {10^6, 5}]; // Timing
Out[4]= {0.078, Null}
```

Si possono creare numeri casuali secondo qualsiasi distribuzione

```
In[5]:= RandomReal[RayleighDistribution[3], 10^6]; // Timing
Out[5]= {0.047, Null}
```

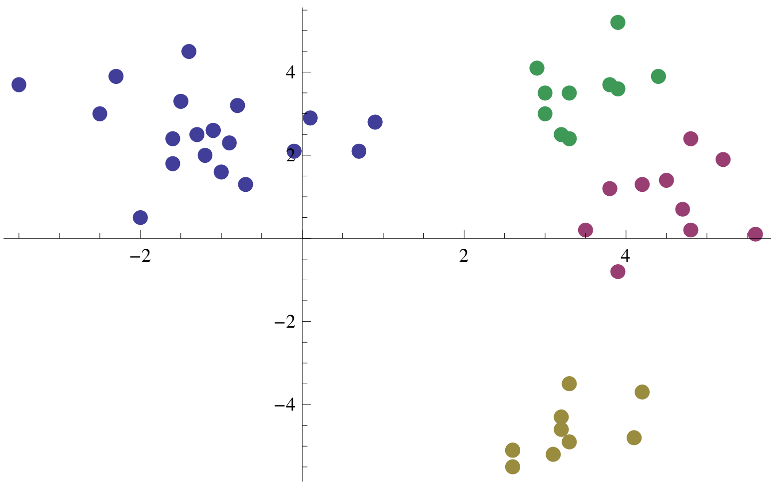
```
In[6]:= RandomInteger[BernoulliDistribution[3 / 10], 10^6]; // Timing
Out[6]= {0.031, Null}
```

Un esempio di clustering

```
In[7]:= data = {{-1.1, 2.6}, {3.9, -0.8}, {4.2, -3.7}, {3.3, 3.5}, {3.9, 5.2},
    {4.1, -4.8}, {3.8, 3.7}, {5.6, 0.1}, {3.1, -5.2}, {-0.9, 2.3}, {2.9, 4.1},
    {-2.3, 3.9}, {-2.5, 3.}, {2.6, -5.5}, {5.2, 1.9}, {-0.7, 1.3},
    {0.9, 2.8}, {-1.5, 3.3}, {3.8, 1.2}, {2.6, -5.1}, {-0.8, 3.2},
    {4.7, 0.7}, {3., 3.}, {3.9, 3.6}, {4.5, 1.4}, {4.2, 1.3}, {-1.1, 2.6},
    {4.8, 2.4}, {3.3, -3.5}, {3.2, -4.6}, {3.3, -4.9}, {3., 3.5}, {0.7, 2.1},
    {3.2, -4.3}, {-2., 0.5}, {-1.2, 2.}, {-1.6, 1.8}, {-3.5, 3.7}, {4.8, 0.2},
    {3.3, 2.4}, {-0.1, 2.1}, {-1.3, 2.5}, {4.4, 3.9}, {3.5, 0.2}, {0.1, 2.9},
    {-1., 1.6}, {-1.4, 4.5}, {3.2, 2.5}, {-1.6, 2.4}, {2.6, -5.1}};
```

```
In[8]:= ListPlot[FindClusters[data], PlotStyle → PointSize[Large]]
```

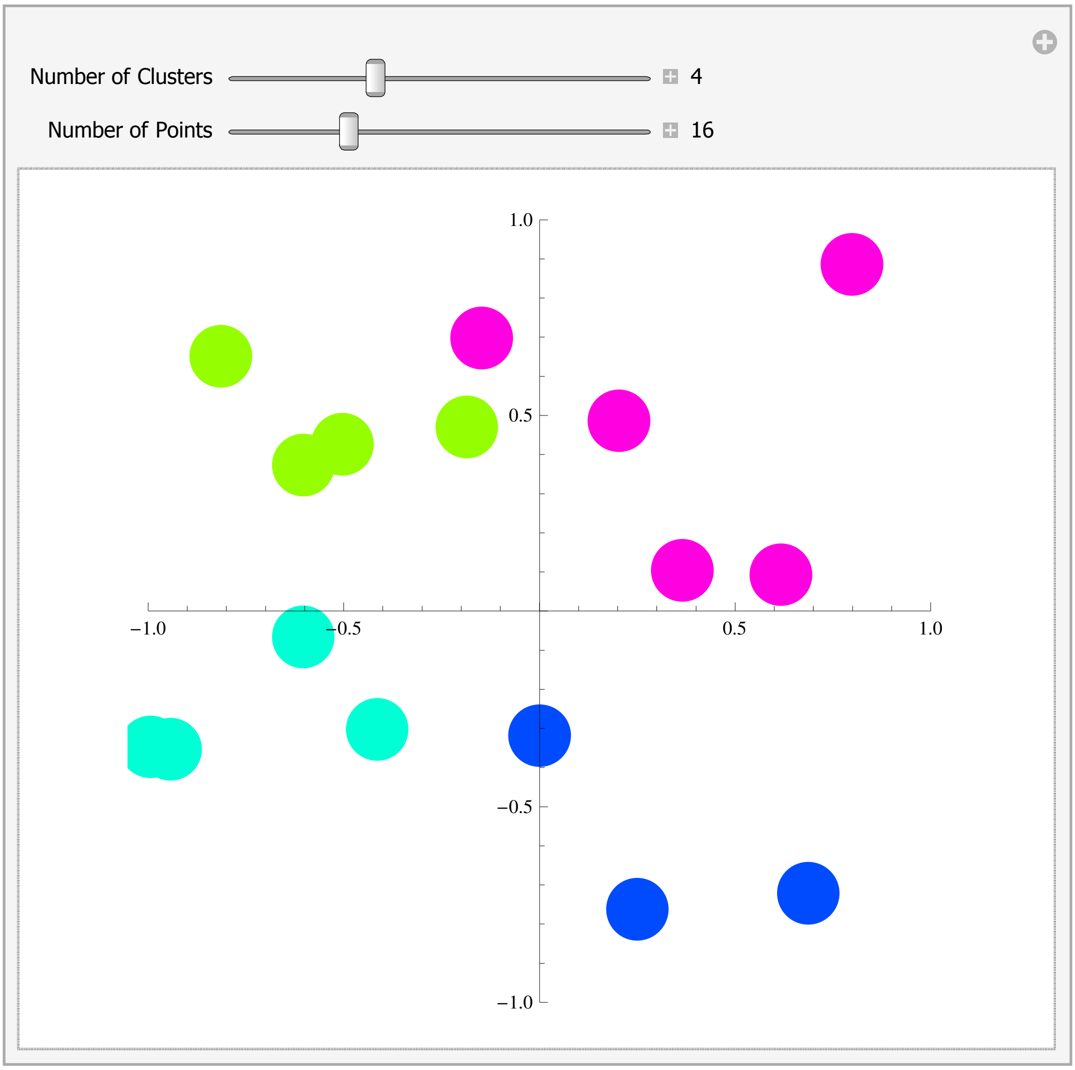
Out[8]=



Il caso dinamico

In[9]:=
```
Manipulate[
r = PadRight[r, If[pt < c, c, pt], RandomReal[{-1, 1}, {If[pt < c, c, pt], 2}]];
 Graphics[{PointSize[.08],
   MapIndexed[{Hue[GoldenRatio First[#2]], Point[#]} &, FindClusters[r, c]]},
  PlotRange → 1, ImageSize → {450, 377}, Axes → True],
 {{r, SeedRandom[64 354];
   RandomReal[{-1, 1}, {20, 2}]}, {-1, -1}, {1, 1}, Locator, Appearance → None},
 {{c, 4, "Number of Clusters"}, 1, 10, 1, Appearance → "Labeled"},
 {{pt, 16, "Number of Points"}, c, 50, 1, Appearance → "Labeled"},
 AutorunSequencing → {1, 2}]
```

Out[9]=



■ Meta-distribuzioni

■ Distribuzioni costruite a partire da altre distribuzioni

Distribution built from a convex combination of component distributions:

Behaves just like a parametric distribution:

In[10]:=
```
𝒟 = MixtureDistribution[{3 / 5, 2 / 5},
  {NormalDistribution[1, 1 / 2], NormalDistribution[2, 1 / 6]}]
```

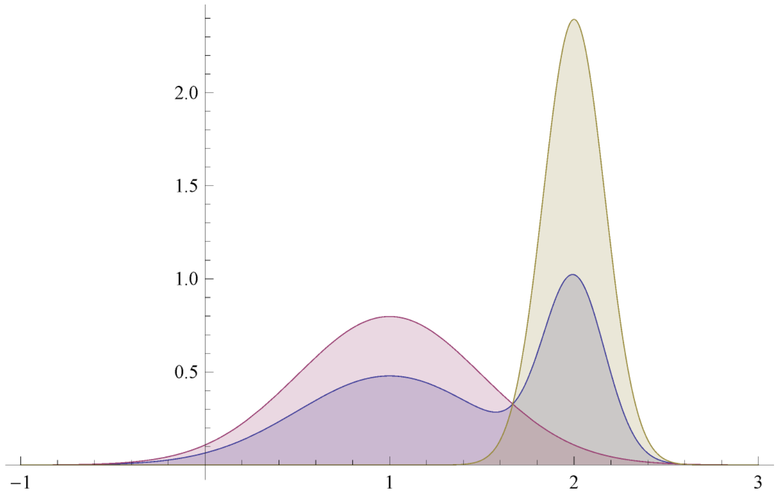
Out[10]= $\text{MixtureDistribution}\left[\left\{\frac{3}{5}, \frac{2}{5}\right\}, \left\{\text{NormalDistribution}\left[1, \frac{1}{2}\right], \text{NormalDistribution}\left[2, \frac{1}{6}\right]\right\}\right]$

In[11]:=
```
PDF[𝒟, x]
```

Out[11]= $\frac{6}{5} e^{-18 (-2+x)^2} \sqrt{\frac{2}{\pi}} + \frac{3}{5} e^{-2 (-1+x)^2} \sqrt{\frac{2}{\pi}}$

In[12]:=
```
Plot[{PDF[𝒟, x], PDF[NormalDistribution[1, 1 / 2], x],
   PDF[NormalDistribution[2, 1 / 6], x]},
  {x, -1, 3}, Filling → Axis, PlotRange → All]
```

Out[12]=



■ **Distribuzioni create dall'utente**

$$\text{ProbabilityDistribution}\left[\begin{cases} e^{x\lambda} p \lambda & x < 0 \\ e^{-x\lambda} (1-p)\lambda & x \geq 0 \end{cases}, \ldots\right] \Longrightarrow$$

Distributions constructed from formulas:

In[13]:=
```
𝒟 = ProbabilityDistribution[
   { e^(x λ) p λ          x < 0
     e^(-x λ) (1 - p) λ   x ≥ 0 , {x, -∞, ∞}, Assumptions → λ > 0 ⋀ 0 < p < 1];
```

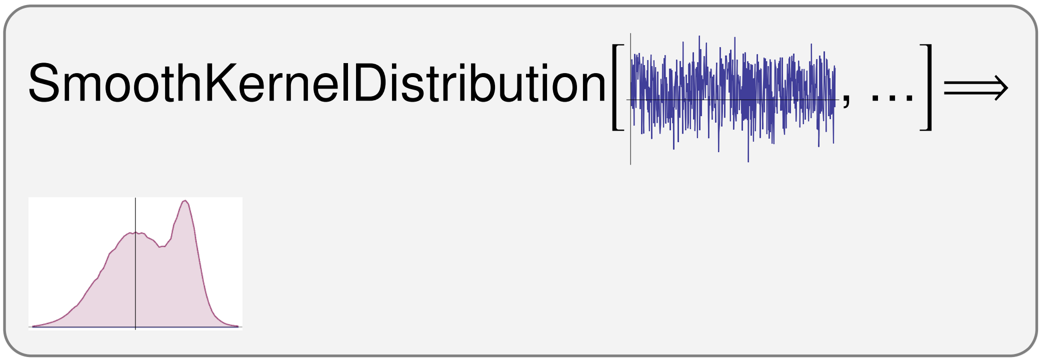
In[14]:=
```
CDF[𝒟, x]
```

Out[14]=
$$\begin{cases} p & x == 0 \\ e^{x\lambda}\, p & x < 0 \\ e^{-x\lambda}\left(-1 + e^{x\lambda} + p\right) & \text{True} \end{cases}$$

In[15]:=
```
Probability[-1 / 2 < X < 1 / 2, X ≈ 𝒟]
```

Out[15]= $1 - e^{-\lambda/2}$

- **Distribuzioni basate sui dati**

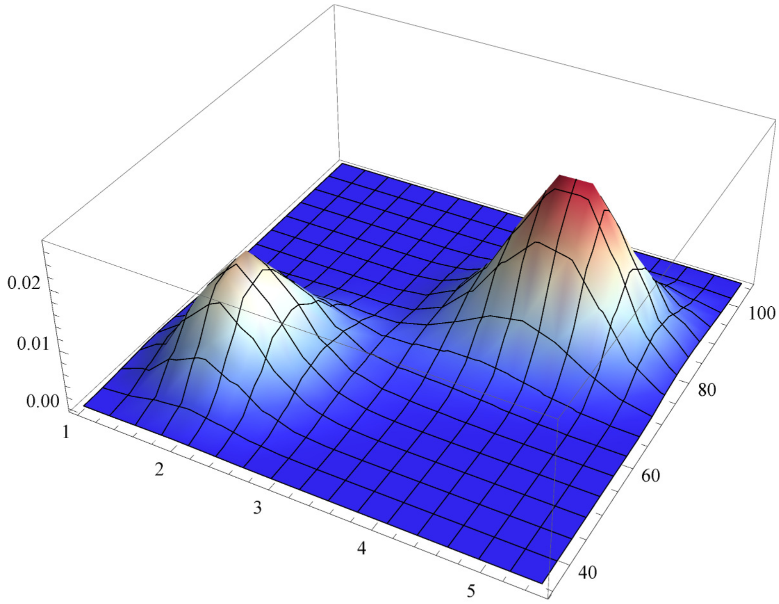


Suppose we have data coming from the following underlying distribution:
Old Faithful geyser data: {duration [minutes], waiting time [minutes]}

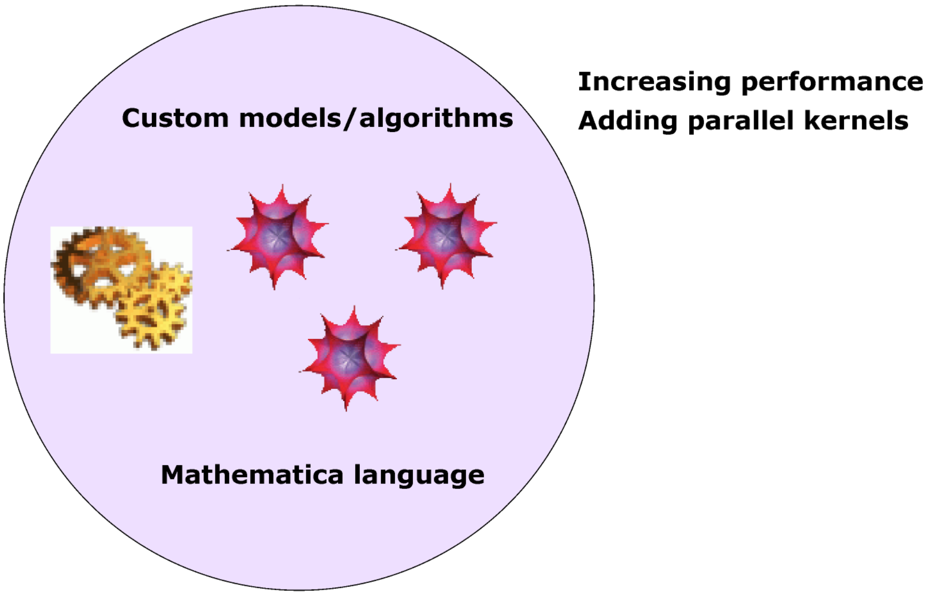
```
In[1]:= OldFaithfulData =
  {{3.6`, 79}, {1.8`, 54}, {3.333`, 74}, {2.283`, 62}, {4.533`, 85}, {2.883`, 55},
   {4.7`, 88}, {3.6`, 85}, {1.95`, 51}, {4.35`, 85}, {1.833`, 54}, {3.917`, 84},
   {4.2`, 78}, {1.75`, 47}, {4.7`, 83}, {2.167`, 52}, {1.75`, 62}, {4.8`, 84},
   {1.6`, 52}, {4.25`, 79}, {1.8`, 51}, {1.75`, 47}, {3.45`, 78}, {3.067`, 69},
   {4.533`, 74}, {3.6`, 83}, {1.967`, 55}, {4.083`, 76}, {3.85`, 78}, {4.433`, 79},
   {4.3`, 73}, {4.467`, 77}, {3.367`, 66}, {4.033`, 80}, {3.833`, 74},
   {2.017`, 52}, {1.867`, 48}, {4.833`, 80}, {1.833`, 59}, {4.783`, 90},
   {4.35`, 80}, {1.883`, 58}, {4.567`, 84}, {1.75`, 58}, {4.533`, 73},
   {3.317`, 83}, {3.833`, 64}, {2.1`, 53}, {4.633`, 82}, {2, 59}, {4.8`, 75},
   {4.716`, 90}, {1.833`, 54}, {4.833`, 80}, {1.733`, 54}, {4.883`, 83},
   {3.717`, 71}, {1.667`, 64}, {4.567`, 77}, {4.317`, 81}, {2.233`, 59},
   {4.5`, 84}, {1.75`, 48}, {4.8`, 82}, {1.817`, 60}, {4.4`, 92}, {4.167`, 78},
   {4.7`, 78}, {2.067`, 65}, {4.7`, 73}, {4.033`, 82}, {1.967`, 56}, {4.5`, 79},
   {4, 71}, {1.983`, 62}, {5.067`, 76}, {2.017`, 60}, {4.567`, 78}, {3.883`, 76},
   {3.6`, 83}, {4.133`, 75}, {4.333`, 82}, {4.1`, 70}, {2.633`, 65},
   {4.067`, 73}, {4.933`, 88}, {3.95`, 76}, {4.517`, 80}, {2.167`, 48},
   {4, 86}, {2.2`, 60}, {4.333`, 90}, {1.867`, 50}, {4.817`, 78}, {1.833`, 63},
   {4.3`, 72}, {4.667`, 84}, {3.75`, 75}, {1.867`, 51}, {4.9`, 82}, {2.483`, 62},
   {4.367`, 88}, {2.1`, 49}, {4.5`, 83}, {4.05`, 81}, {1.867`, 47}, {4.7`, 84},
   {1.783`, 52}, {4.85`, 86}, {3.683`, 81}, {4.733`, 75}, {2.3`, 59},
   {4.9`, 89}, {4.417`, 79}, {1.7`, 59}, {4.633`, 81}, {2.317`, 50}, {4.6`, 85},
   {1.817`, 59}, {4.417`, 87}, {2.617`, 53}, {4.067`, 69}, {4.25`, 77},
   {1.967`, 56}, {4.6`, 88}, {3.767`, 81}, {1.917`, 45}, {4.5`, 82}, {2.267`, 55},
   {4.65`, 90}, {1.867`, 45}, {4.167`, 83}, {2.8`, 56}, {4.333`, 89},
   {1.833`, 46}, {4.383`, 82}, {1.883`, 51}, {4.933`, 86}, {2.033`, 53},
   {3.733`, 79}, {4.233`, 81}, {2.233`, 60}, {4.533`, 82}, {4.817`, 77},
   {4.333`, 76}, {1.983`, 59}, {4.633`, 80}, {2.017`, 49}, {5.1`, 96},
   {1.8`, 53}, {5.033`, 77}, {4, 77}, {2.4`, 65}, {4.6`, 81}, {3.567`, 71},
   {4, 70}, {4.5`, 81}, {4.083`, 93}, {1.8`, 53}, {3.967`, 89}, {2.2`, 45},
   {4.15`, 86}, {2, 58}, {3.833`, 78}, {3.5`, 66}, {4.583`, 76}, {2.367`, 63},
   {5, 88}, {1.933`, 52}, {4.617`, 93}, {1.917`, 49}, {2.083`, 57}, {4.583`, 77},
   {3.333`, 68}, {4.167`, 81}, {4.333`, 81}, {4.5`, 73}, {2.417`, 50}, {4, 85},
   {4.167`, 74}, {1.883`, 55}, {4.583`, 77}, {4.25`, 83}, {3.767`, 83},
   {2.033`, 51}, {4.433`, 78}, {4.083`, 84}, {1.833`, 46}, {4.417`, 83},
   {2.183`, 55}, {4.8`, 81}, {1.833`, 57}, {4.8`, 76}, {4.1`, 84}, {3.966`, 77},
   {4.233`, 81}, {3.5`, 87}, {4.366`, 77}, {2.25`, 51}, {4.667`, 78}, {2.1`, 60},
   {4.35`, 82}, {4.133`, 91}, {1.867`, 53}, {4.6`, 78}, {1.783`, 46},
   {4.367`, 77}, {3.85`, 84}, {1.933`, 49}, {4.5`, 83}, {2.383`, 71},
   {4.7`, 80}, {1.867`, 49}, {3.833`, 75}, {3.417`, 64}, {4.233`, 76},
   {2.4`, 53}, {4.8`, 94}, {2, 55}, {4.15`, 76}, {1.867`, 50}, {4.267`, 82},
   {1.75`, 54}, {4.483`, 75}, {4, 78}, {4.117`, 79}, {4.083`, 78}, {4.267`, 78},
   {3.917`, 70}, {4.55`, 79}, {4.083`, 70}, {2.417`, 54}, {4.183`, 86},
   {2.217`, 50}, {4.45`, 90}, {1.883`, 54}, {1.85`, 54}, {4.283`, 77},
   {3.95`, 79}, {2.333`, 64}, {4.15`, 75}, {2.35`, 47}, {4.933`, 86}, {2.9`, 63},
   {4.583`, 85}, {3.833`, 82}, {2.083`, 57}, {4.367`, 82}, {2.133`, 67},
   {4.35`, 74}, {2.2`, 54}, {4.45`, 83}, {3.567`, 73}, {4.5`, 73}, {4.15`, 88},
   {3.817`, 80}, {3.917`, 71}, {4.45`, 83}, {2, 56}, {4.283`, 79}, {4.767`, 78},
   {4.533`, 84}, {1.85`, 58}, {4.25`, 83}, {1.983`, 43}, {2.25`, 60}, {4.75`, 75},
   {4.117`, 81}, {2.15`, 46}, {4.417`, 90}, {1.817`, 46}, {4.467`, 74}};
```

```
In[17]:= 𝒟data = SmoothKernelDistribution[OldFaithfulData];

In[18]:= Plot3D[PDF[𝒟data, {x, y}], {x, 1, 5.5}, {y, 35, 105},
          PlotRange → All, ColorFunction → "ThermometerColors"]
```

Out[18]=

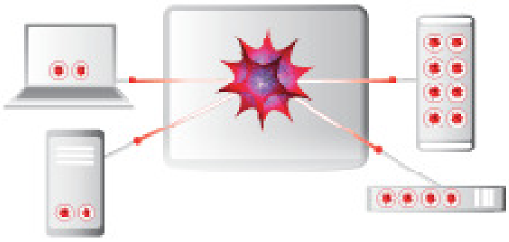


Controllo delle performance



■ Il calcolo parallelo

Mathematica contiene un ambiente di calcolo parallelo (del tipo master-slave) completamente integrato ed automatizzato.



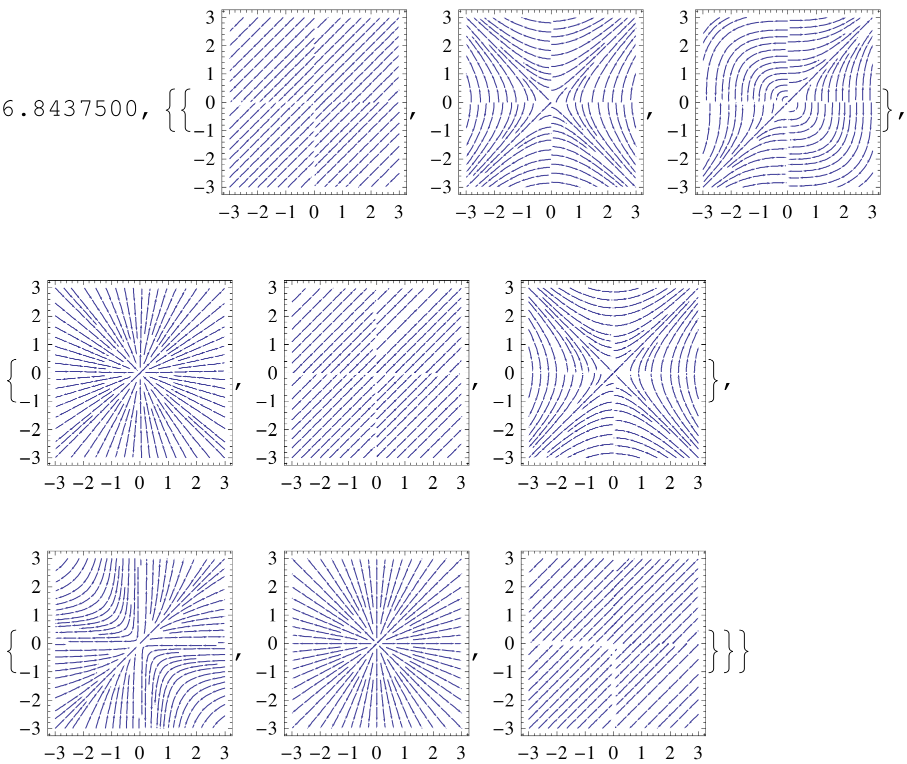
Le richieste eseguite alle varie banche dati sono tipicamente sequenziali. Dunque, quando si fa un uso intensivo delle sorgenti integrate ritorna utile sfruttare i comandi paralleli

```
In[16]:= Map[FinancialData["MI:BUL", #] &,
           {"Company", "Open", "Close", "Volume", "High", "Low"}] // AbsoluteTiming

Out[16]= {4.4218750, {Bulgari, 7.45, 7.45, 753 910, 7.51, 7.34}}
```

```
In[17]:= ParallelMap[FinancialData["MI:BUL", #] &,
           {"Company", "Open", "Close", "Volume", "High", "Low"}] // AbsoluteTiming

Out[17]= {0.9531250, {Bulgari, 7.45, 7.45, 753 910, 7.51, 7.34}}
```

Esempio di calcolo

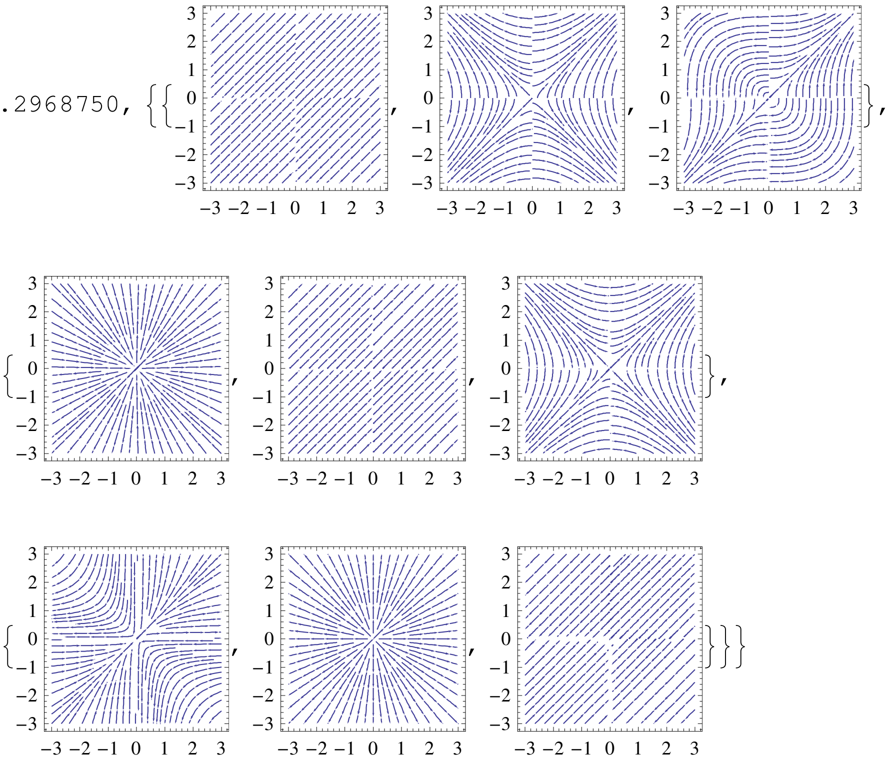
```
In[12]:= Table[StreamPlot[{x^i y^j, x^j y^i}, {x, -3, 3}, {y, -3, 3}, ImageSize → 100],
           {i, 3}, {j, 3}] // AbsoluteTiming

Out[12]= {16.8437500, {{
```



Se vogliamo parallelizzare questa computazione, possiamo aggiungere il comand Parallelize all precedente istruzione

In[13]:=
```
Parallelize[Table[StreamPlot[{x^i y^j, x^j y^i}, {x, -3, 3},
    {y, -3, 3}, ImageSize → 100], {i, 3}, {j, 3}]] // AbsoluteTiming
```

Out[13]= {3.2968750, {{



}}}

■ Altre novità della versione 8 in merito all'incremento delle performance

■ Compile

Ora è possibile compilare in maniera ancora più efficiente una parte significativa dei nostri algoritmi

■ CUDALink

Questa tabella ci mostra i fattori di guadagno relativi ad un calcolo eseguito con *Mathematica* e poi con *Mathematica* usando codice CUDALink

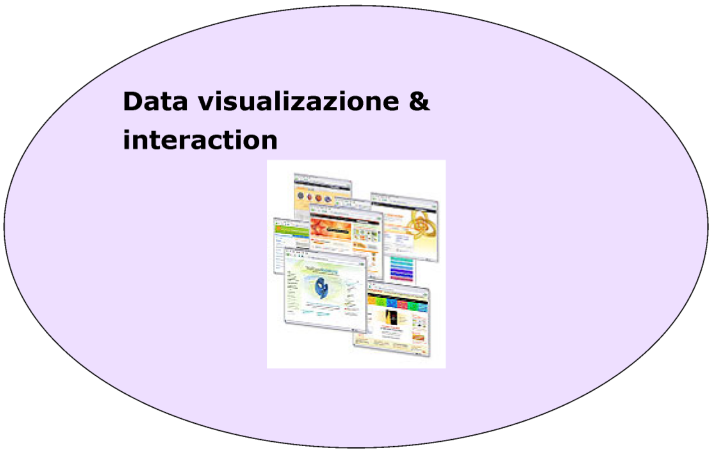
Caratteristiche della macchina

CPU: Core i7 950, quad core 3.06GHz
Memory: 12GB DDR3
GPU: NVIDIA Tesla C2050 (Fermi)
OS: Windows 7

Valori

Option	Method	Enhancement
Vanilla American Options	Binomial	62
American Quanto Fixed Exchange Options	Binomial	68
Asian Arithmetic Options	Monte Carlo	130

Dynamic, Manipulate e altre funzionalità dedicate alle GUI interattive



L'interfaccia di *Mathematica* è basata su documenti (Notebook). *Mathematica* consente di computare, sviluppare ed utilizzare le applicazioni tramite un potente documento chiamato "Notebook" (.nb)

Principali Vantaggi

I Notebooks sono platform independent (Windows/Macintosh/Linux/Unix)
I Notebooks sono completamente personalizzabili e programmabili per adeguarsi a qualsiasi esigenza di workflow (es. le palette sono notebook)

■ Interattività e dinamicità

Mathematica ha rivoluzionato il concetto di computazione interattiva e dinamica, introducendo funzioni dinamiche che istantaneamente creano interfacce intuitive e interattive. Le computazioni sottostanti vengono eseguite in run-time

In[22]:=
```
Integrate[1 / (x^3 + 1), x]
```

Out[22]=
$$\frac{\text{ArcTan}\left[\frac{-1+2x}{\sqrt{3}}\right]}{\sqrt{3}} + \frac{1}{3}\,\text{Log}[1+x] - \frac{1}{6}\,\text{Log}\left[1-x+x^2\right]$$

In[23]:=
```
Manipulate[Integrate[1 / (x^a + 1), x], {a, 1, 10, 1}]
```

a

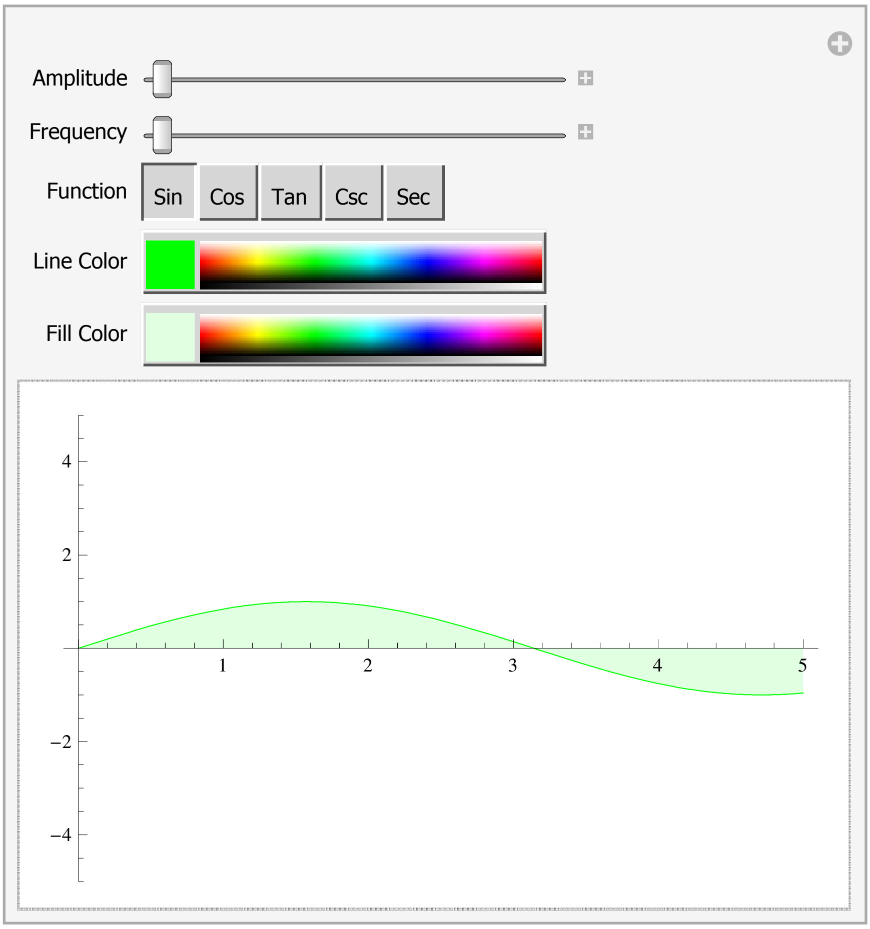
Out[23]=

$$\frac{1}{4\sqrt{2}}\left(-2\,\mathrm{ArcTan}\left[1-\sqrt{2}\,x\right]+2\,\mathrm{ArcTan}\left[1+\sqrt{2}\,x\right]-\mathrm{Log}\left[1-\sqrt{2}\,x+x^2\right]+\mathrm{Log}\left[1+\sqrt{2}\,x+x^2\right]\right)$$

Manipulate e gli altri comandi dinamici possono trattare qualsiasi espressione di *Mathematica*

In[24]:=
```
Manipulate[Plot[amp function[freq x], {x, 0, 5}, PlotRange → {-5, 5},
  Filling → Axis, PlotStyle → pcol, FillingStyle → fcol],
 {{amp, 1, "Amplitude"}, 1, 5},
 {{freq, 1, "Frequency"}, 1, 5},
 {{function, Sin, "Function"}, {Sin, Cos, Tan, Csc, Sec}},
 {{pcol, Green, "Line Color"}, Red},
 {{fcol, LightGreen, "Fill Color"}, LightRed}]
```

Out[24]=



Si possono sviluppare interfacce anche molto complesse - Esempio 1 - Esempio 2

■ Programmabilità dei documenti

Si possono anche creare report da programma.

In[25]:=
```
Button["Generate Report",
 Quiet[CreateDocument[CellGroup[{TextCell[CountryData[#, "Name"], "Section"],
       ExpressionCell[GraphicsRow[{CountryData[#, "Flag"], Quiet[
           DateListPlot[CountryData[#, {{"Population"}, {1980, 2008}}]]]}],
         Frame → All, ImageSize → All], "Output"]}, Closed] & /@
    CountryData["Africa"], WindowTitle → "Africa", StyleDefinitions →
    "Creative/NaturalColor.nb"]], Method -> "Queued"]
```

Out[25]= Generate Report

Una tabella

In[26]:=
```
Manipulate[
 Grid[Prepend[Table[{i, m, i^m}, {i, 1, n}],
   {Style["n", Bold, Red], Style["m", Bold, Red], Style["n^m", Bold, Red]}],
  Alignment → {{Left, Left, Right}, Automatic}, Frame → All,
  ItemStyle → {Blue, FontFamily → "Helvetica"}, Background → LightGray],
 {n, 1, 20, 1}, {m, 1, 100, 1}]
```

Out[26]=

n

m

n	m	n^m
1	43	1
2	43	8 796 093 022 208
3	43	328 256 967 394 537 077 627

Le proprietà di questo documento

In[27]:=
```
Manipulate[(SetOptions[ButtonNotebook[], Background → RGBColor[r, g, b]];
  "Modifica il colore dello sfondo"), {{r, 0.9, "Red"}, 0, 1},
 {{g, 0.6, "Green"}, 0, 1}, {{b, 0.3, "Blue"}, 0, 1}]
```

Out[27]=

Red

Green

Blue

Modifica il colore dello sfondo

Infine, bisogna sottolineare ancora una volta il carattere di strumento altamente integrato: tutte le funzionalità viste sin ora si integrano in maniera intuitiva ed immediata, senza dover scrivere pagine a pagine di codice.
Esempio di tabella realizzata direttamente caricando dati da una sorgente interna a *Mathematica*: tabella dei 20 paesi del mondo più grandi in base all'area

```
In[28]:= biggestA =
           Last /@ Take[Reverse[Sort[{CountryData[#, "Area"], #} & /@ CountryData[]]], 20];
         biggestP = Last /@ Take[Reverse[
              Sort[{CountryData[#, "Population"], #} & /@ CountryData[]]], 20];
         Row[{Text[Grid[Prepend[{CountryData[#, "Name"], CountryData[#, "Area"],
                 CountryData[#, "Population"]} & /@ biggestA, {"", "area", "population"}],
              Frame → All, Background → {None, {Gray, {Yellow, White}}}]], Spacer[5],
           Text[Grid[Prepend[{CountryData[#, "Name"], CountryData[#, "Population"],
                 CountryData[#, "Area"]} & /@ biggestP, {"", "population", "area"}],
              Frame → All, Background → {None, {Gray, {White, Green}}}]]}]
```

Out[30]=

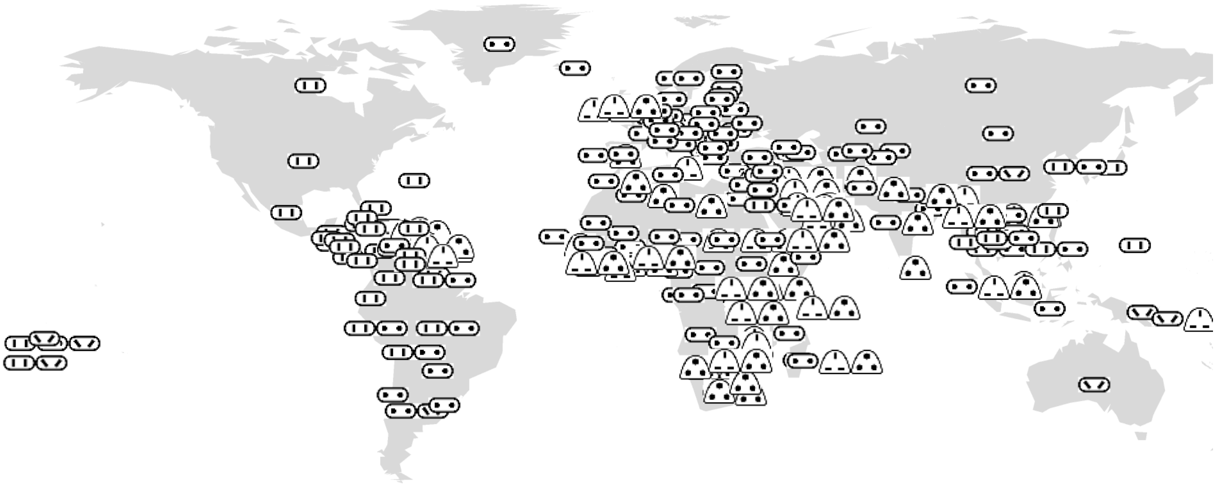
	area	population
Russia	1.70752×10^7	1.41394×10^8
Canada	9.98467×10^6	3.32593×10^7
United States	9.63142×10^6	3.11666×10^8
China	9.59696×10^6	1.31436×10^9
Brazil	8.51488×10^6	1.91972×10^8
Australia	7.68685×10^6	2.10744×10^7
India	3.28726×10^6	1.18141×10^9
Argentina	2.76689×10^6	3.9883×10^7
Kazakhstan	2.7249×10^6	1.55215×10^7
Sudan	2.50581×10^6	4.13477×10^7
Algeria	2.38174×10^6	3.43734×10^7
Democratic Republic of the Congo	2.34486×10^6	6.42566×10^7
Greenland	2.16609×10^6	57 307.
Mexico	1.96438×10^6	1.08555×10^8
Saudi Arabia	1.96058×10^6	2.52005×10^7
Indonesia	1.90457×10^6	2.27345×10^8
Libya	1.75954×10^6	6.29418×10^6
Iran	1.648×10^6	7.33118×10^7
Mongolia	1.56412×10^6	2.64122×10^6
Peru	1.28522×10^6	2.88367×10^7

	population	area
China	1.31436×10^9	9.59696×10^6
India	1.18141×10^9	3.28726×10^6
United States	3.11666×10^8	9.63142×10^6
Indonesia	2.27345×10^8	1.90457×10^6
Brazil	1.91972×10^8	8.51488×10^6
Pakistan	1.76952×10^8	796 095.
Bangladesh	1.6×10^8	143 998.
Nigeria	1.51212×10^8	923 768.
Russia	1.41394×10^8	1.70752×10^7
Japan	1.27293×10^8	377 835.
Mexico	1.08555×10^8	1.96438×10^6
Philippines	9.03484×10^7	300 000.
Vietnam	8.70959×10^7	329 560.
Germany	8.22643×10^7	357 022.
Egypt	8.15272×10^7	1.00145×10^6
Ethiopia	8.07134×10^7	1.12713×10^6
Turkey	7.39143×10^7	780 580.
Iran	7.33118×10^7	1.648×10^6
Thailand	6.73864×10^7	513 120.
Democratic Republic of the Congo	6.42566×10^7	2.34486×10^6

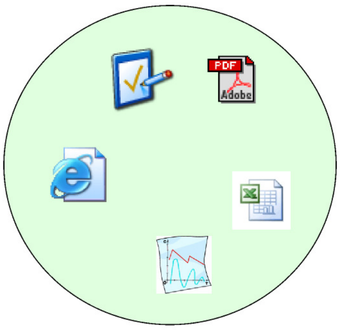
Prima di mettersi in viaggio ...

In[31]:=
```
Graphics[{LightGray, CountryData[#, "SchematicPolygon"] & /@ CountryData[],
  Inset[Row[Show[#, ImageSize → 15] & /@
       (CountryData[#, "ElectricalGridPlugImages"] /. _Missing → {})],
     Reverse[CountryData[#, "CenterCoordinates"]]] & /@
   CountryData[]}, ImageSize → 600]
```

Out[31]=



Export verso altri ambienti/applicazioni



■ Formati di dati gestiti in esportazione

In[32]:= `$ExportFormats`

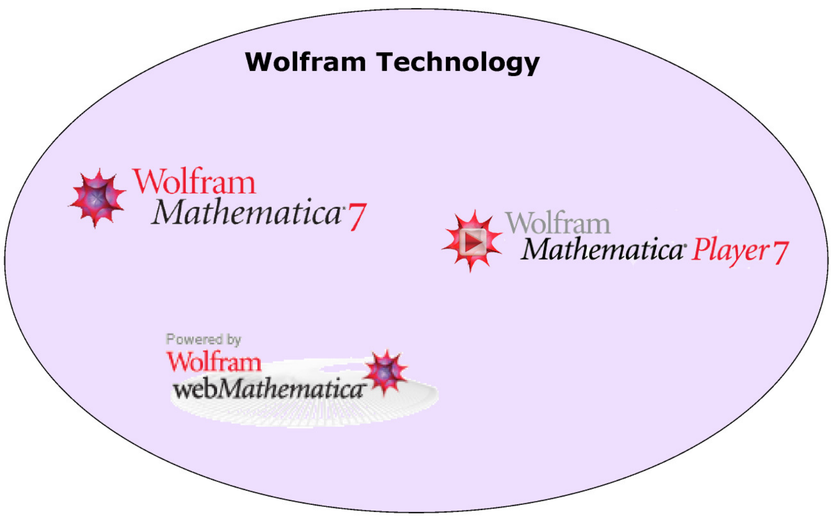
```
Out[32]= {3DS, ACO, AIFF, AU, AVI, Base64, Binary, Bit, BMP, Byte, BYU, BZIP2, C,
  Character16, Character8, Complex128, Complex256, Complex64, CSV, DICOM, DIF,
  DIMACS, DOT, DXF, EMF, EPS, ExpressionML, FASTA, FITS, FLAC, FLV, GIF, Graph6,
  Graphlet, GraphML, GXL, GZIP, HarwellBoeing, HDF, HDF5, HTML, Integer128,
  Integer16, Integer24, Integer32, Integer64, Integer8, JPEG, JPEG2000, JSON,
  JVX, KML, LEDA, List, LWO, MAT, MathML, Maya, MGF, MIDI, MOL, MOL2, MTX, MX,
  NASACDF, NB, NetCDF, NEXUS, NOFF, OBJ, OFF, Package, Pajek, PBM, PCX, PDB,
  PDF, PGM, PICT, PLY, PNG, PNM, POV, PPM, PXR, QuickTime, RawBitmap, Real128,
  Real32, Real64, RIB, RTF, SCT, SDF, SND, Sparse6, STL, String, SurferGrid,
  SVG, SWF, Table, TAR, TerminatedString, TeX, Text, TGA, TGF, TIFF, TSV,
  UnsignedInteger128, UnsignedInteger16, UnsignedInteger24, UnsignedInteger32,
  UnsignedInteger64, UnsignedInteger8, UUE, VideoFrames, VRML, VTK, WAV,
  Wave64, WDX, WMF, X3D, XBM, XHTML, XHTMLMathML, XLS, XLSX, XML, XYZ, ZIP, ZPR}
```

Esempio di report HTML generato da una applicazione *Mathematica* (genera report da TradingStrategy)

■ Interazione con altri strumenti mediante link dedicati

- ***Mathematica* Link for Excel**

L'impiego di tecnologie Wolfram per il deployment delle soluzioni



■ *Mathematica* come ambiente di lavoro per gli end-user

In questa soluzione l'utente finale deve acquistare *Mathematica,* e questo gli consente di eseguire non solo tutte le funzionalità dell'applicativo che gli è stato sviluppato dagli ingegneri, ma anche di creare ex-novo qualsiasi altro strumento di calcolo di cui necessita.

- ***Mathematica* Player**

L'utente final non deve acquistare *Mathematica* perchè l'applicativo è stato realizzato per essere seguito trami il software *Mathematica* Player, nella sua versione Free o Professional. Ci sono alcune differenze tecniche tra le due versioni `link`

- **web*Mathematica* per rendere più facile lo sfruttamente dei risultati ottenuti con *Mathematica***

L'utente finale ha bisogno solo di un browser web

Conclusioni

- ***Mathematica* fornisce un ambiente integrato per lo sviluppo e la fruizione di soluzioni anche complesse sia dal punto di vista del calcolo che delle interfacce utente**
- ***Mathematica,* grazie anche alla sua natura di linguaggio simbolico, consente di esprimere i problemi nel linguaggio tradizionale della letteratura scientifica**
- **Offre un linguaggio di programmazione di alto livello**
- **Include banche dati aggiornate, affidabili e complete**
- **Si integra e/o interagisce facilmente con altri ambienti e sistemi informativi aziendali**
- **Infine ... uno sguardo ad alcuni prezzi del segmento Educational**