Wolfram Mathematica
Il software di riferimento per la Didattica, la Ricerca e lo Sviluppo

ADALTA
Distributore ufficiale per l'Italia di Wolfram Research
www.adalta.it/wolfram

Lezione 5

Statistica e analisi dei dati

Crescenzio Gallo – Università di Foggia
crescenzio.gallo@unifg.it

Note:

- Il materiale visualizzato durante questo seminario è disponibile per il download all'indirizzo http://www.crescenziogallo.it/unifg/seminario-mathematica-2014/
- Il materiale utilizzato è tratto dai webinar pubblicati da Adalta e prodotti dal dott. Roberto Cavaliere (*Mathematica* Technical Sales Manager, *r.cavaliere@adalta.it*)

Agenda

Introduzione

- I dati

Funzionalità di statistica e analisi dati

- Panoramica
- Esempi di statistica descrittiva
- Esempi di distribuzioni statistiche
- Model fitting e analisi

Conclusioni

I dati

L'introduzione dei dati nel sistema di calcolo *Mathematica* può avvenire in diversi modi e impiegando varie tecnologie.

- Integrazione del kernel con altri ambienti (es. C, C++, Fortran, Java, .NET ed altri)
- Import dei file tramite formati standard
- Accesso ai database (MySQL, Oracle, ...)
- Dati curati (FinancialData, WeatherData, CountryData, ...)

Integrazione con altri ambienti

Si veda Systems Interfaces & Deployment per una lista completa delle modalità di integrazione tra *Mathematica* ed altri linguagg

Formati disponibili

$ImportFormats

{3DS, ACO, Affymetrix, AgilentMicroarray, AIFF, ApacheLog, ArcGRID, AU, AVI, Base64, BDF, Binary, Bit, BMP, Byte, BYU, BZIP2, CDED, CDF, Character16, Character8, CIF, Complex128, Complex256, Complex64, CSV, CUR, DBF, DICOM, DIF, DIMACS, Directory, DOT, DXF, EDF, EPS, ExpressionML, FASTA, FASTQ, FCS, FITS, FLAC, GenBank, GeoTIFF, GIF, GPX, Graph6, Graphlet, GraphML, GRIB, GTOPO30, GXL, GZIP, HarwellBoeing, HDF, HDF5, HIN, HTML, ICC, ICNS, ICO, ICS, Integer128, Integer16, Integer24, Integer32, Integer64, Integer8, JCAMP−DX, JPEG, JPEG2000, JSON, JVX, KML, LaTeX, LEDA, List, LWO, MAT, MathML, MBOX, MDB, MGF, MIDI, MMCIF, MOL, MOL2, MPS, MTP, MTX, MX, NASACDF, NB, NDK, NetCDF, NEXUS, NOFF, OBJ, ODS, OFF, OpenEXR, Package, Pajek, PBM, PCX, PDB, PDF, PGM, PLY, PNG, PNM, PPM, PXR, QuickTime, RawBitmap, Real128, Real32, Real64, RIB, RSS, RTF, SCT, SDF, SDTS, SDTSDEM, SFF, 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}

$ExportFormats

{3DS, ACO, AIFF, AU, AVI, Base64, Binary, Bit, BMP, Byte, BYU, BZIP2, C, CDF, Character16, Character8, Complex128, Complex256 Complex64, CSV, CUR, DICOM, DIF, DIMACS, DOT, DXF, EMF, EPS, ExpressionML, FASTA, FASTQ, FCS, FITS, FLAC, FLV, GIF, Graph6, Graphlet, GraphML, GXL, GZIP, HarwellBoeing, HDF, HDF5, HTML, ICNS, ICO, 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, X3D, XBM, XHTML, XHTMLMathML, XLS, XLSX, XML, XYZ, ZIP, ZPR}

Import

Sintassi di base della Import

- `Import["`*file*`",` *format*`]`: importa dati dal file specificato restituendono la versione Mathematica
- `Import["`*file*`", {`*format*`,` *elements*`}]`: importa dal file solo gli elementi specificati.
- `Import["http://`*url*`",…]`: importa da un qualsiasi URL accessibile (anche se sono richieste credenziali di accesso)

La maggior parte dei casi si può omettere il formato del file e la Import sarà comunque in grado di identificare il formato automat camente dal file stesso

```
Import["C:\\temp\\Fiat.xls"]
```

oppure direttamente dalla rete

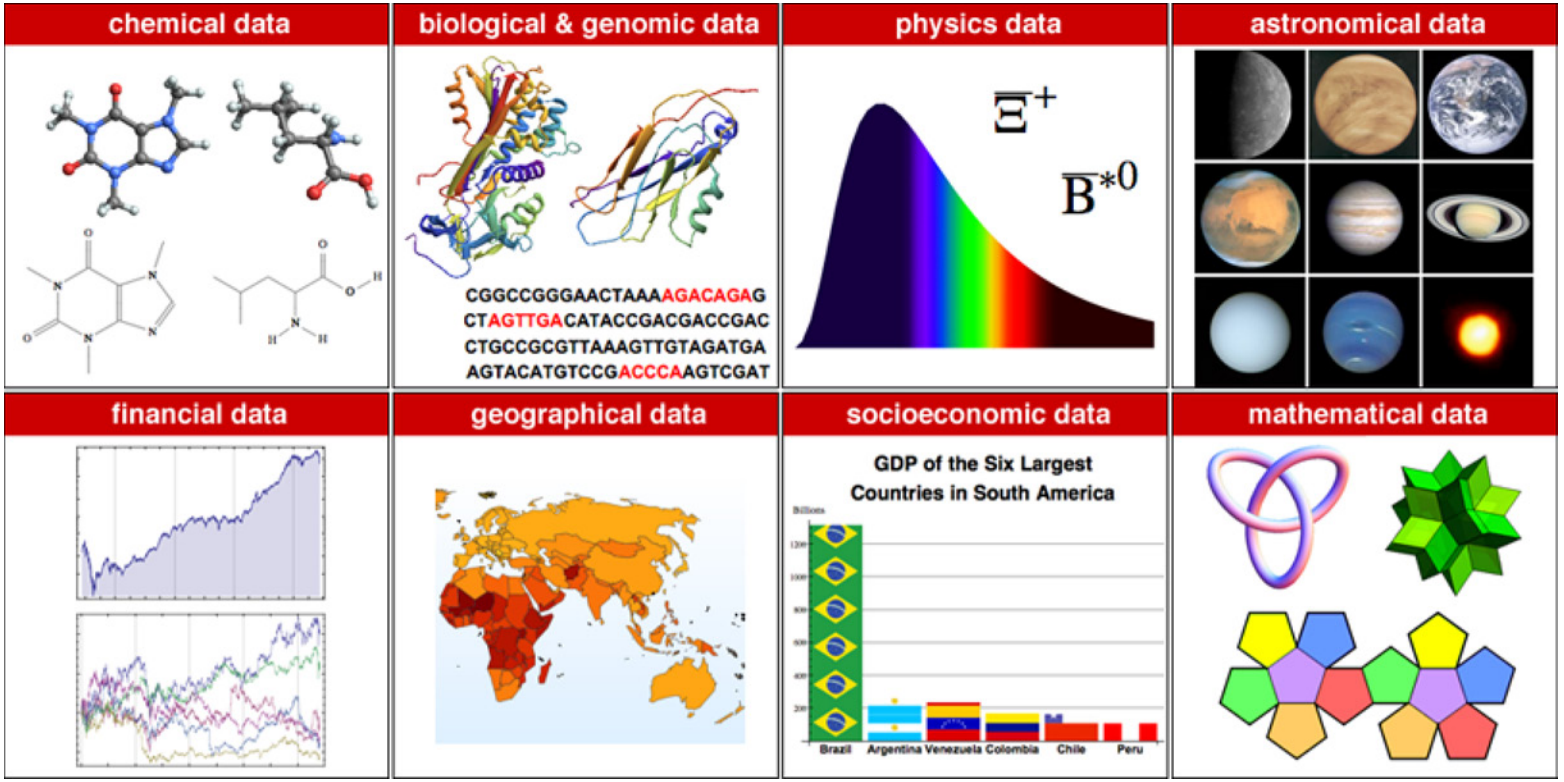
```
Import[
 "http://ichart.finance.yahoo.com/table.csv?s=AAPL&d=10&e=7&f=2011&g=d&a=8&b=7&c=1984&ignore=.
   csv"]
```

Collegamento con i database

Un altro importantissimo aspetto legato alla gestione dei dati è quello dei database. Anche in questo caso *Mathematica* com prende la tecnologia necessaria per l'integrazione immediata ed altamente flessibile: Database Connectivity

Banche dati interne a *Mathematica*

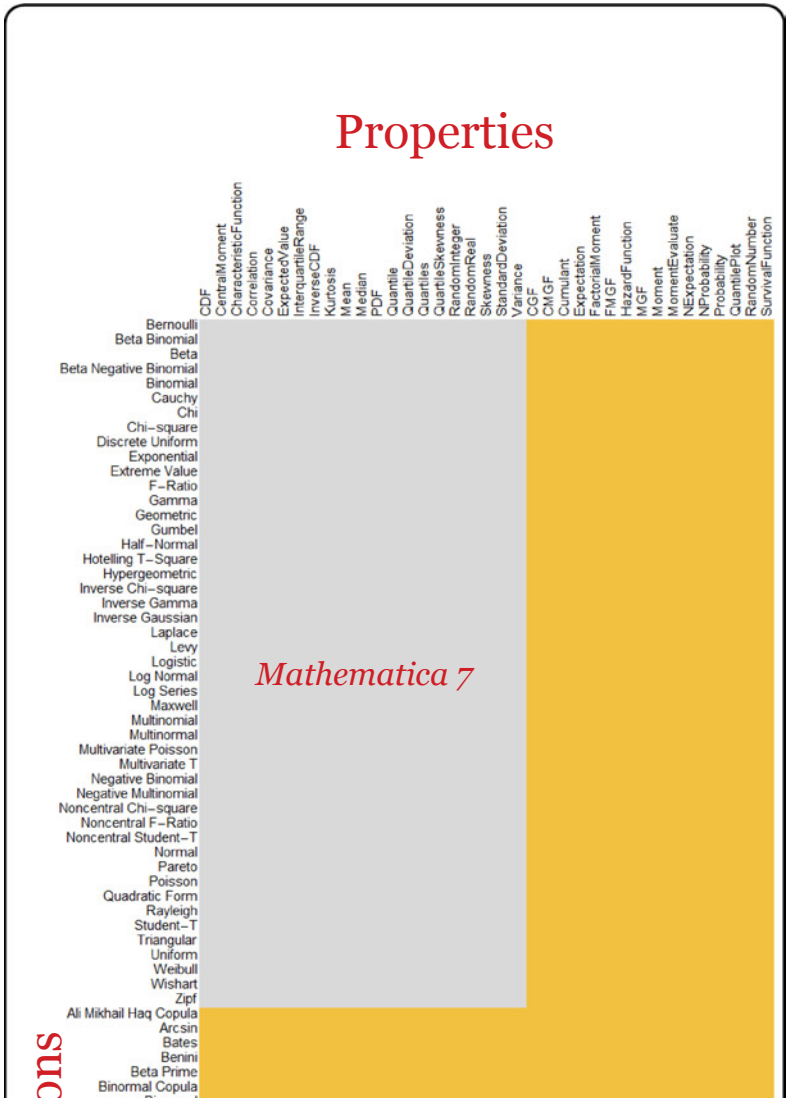
Oltre alle interfacce con altri ambienti/linguaggi ed al riconoscimento dei principali formati di file, *Mathematica* mette a disposizione una serie di banche dati affidabili, robuste ed aggiornate costantemente. Questo è un elenco completo delle banche dati:
Computable Data

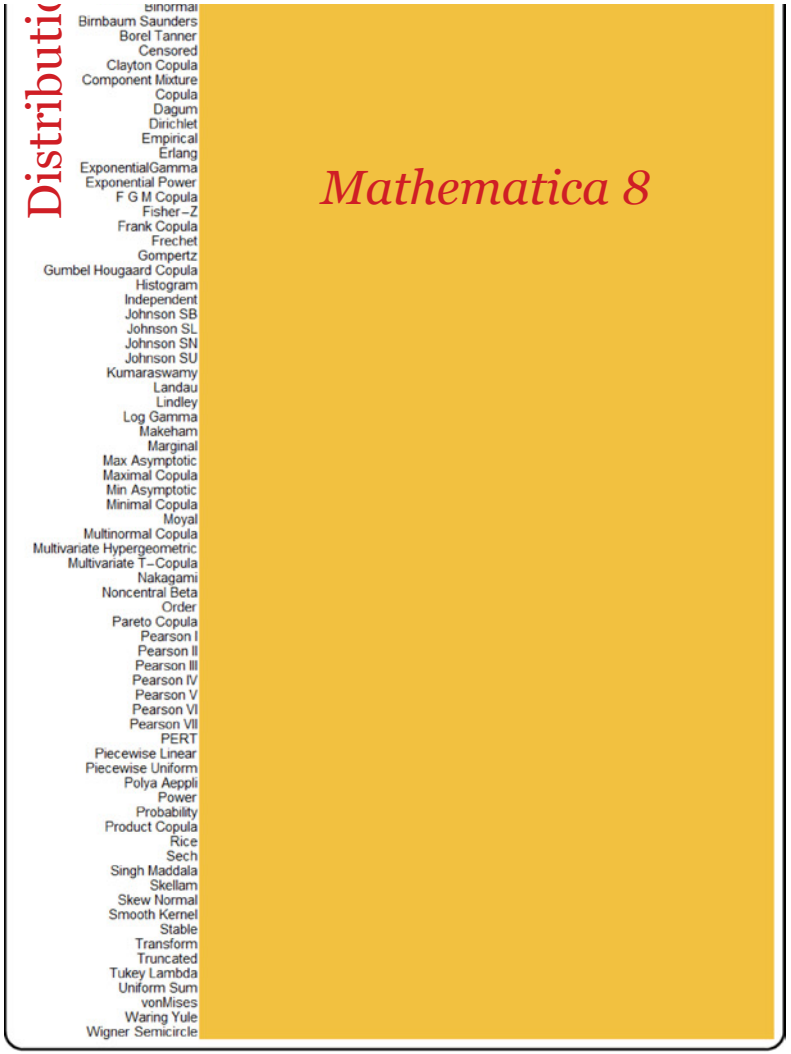


Funzionalità di statistica e analisi dati: panoramica

La versione 8 di *Mathematica* ha completamente rinnovato l'area delle funzioni dedicate alla Statistica. Le due immagini cł seguono indicano in sintesi i livelli di funzionalità ad oggi disponibili.

Mathematica 7 vs. *Mathematica* 8





Sintesi delle distribuzioni

```
?*Distribution*
```

▼ System`

ArcSinDistribution	EstimatedDistribution	LogLogisticDistribution	ReliabilityDistribution
BarabasiAlbertGraphDistr-ibution	ExpGammaDistribution	LogMultinormalDistributio-n	RiceDistribution
BatesDistribution	ExponentialDistribution	LogNormalDistribution	SechDistribution

BeckmannDistribution	ExponentialPowerDistribution	LogSeriesDistribution	SinghMaddalaDistribution
BenfordDistribution	ExtremeValueDistribution	MarginalDistribution	SkellamDistribution
BeniniDistribution	FailureDistribution	MaxStableDistribution	SkewNormalDistribution
BenktanderGibratDistribution	FindDistributionParameters	MaxwellDistribution	SliceDistribution
BenktanderWeibullDistribution	FirstPassageTimeDistribution	MeixnerDistribution	SmoothKernelDistribution
BernoulliDistribution	FisherHypergeometricDistribution	MinStableDistribution	SpatialGraphDistribution
BernoulliGraphDistribution	FisherZDistribution	MixtureDistribution	SplicedDistribution
BetaBinomialDistribution	FRatioDistribution	MoyalDistribution	StableDistribution
BetaDistribution	FrechetDistribution	MultinomialDistribution	StandbyDistribution
BetaNegativeBinomialDistribution	GammaDistribution	MultinormalDistribution	StationaryDistribution
BetaPrimeDistribution	GeometricDistribution	MultivariateHypergeometricDistribution	StudentTDistribution
BinomialDistribution	GompertzMakehamDistribution	MultivariatePoissonDistribution	SurvivalDistribution
BinormalDistribution	GraphPropertyDistribution	MultivariateTDistribution	SuzukiDistribution
BirnbaumSaundersDistribution	GumbelDistribution	NakagamiDistribution	TransformedDistribution
BorelTannerDistribution	HalfNormalDistribution	NegativeBinomialDistribution	TriangularDistribution
CauchyDistribution	HistogramDistribution	NegativeMultinomialDistribution	TruncatedDistribution

CensoredDistribution	HotellingTSquareDistribution	NoncentralBetaDistribution	TsallisQExponentialDistribution
ChiDistribution	HoytDistribution	NoncentralChiSquareDistribution	TsallisQGaussianDistribution
ChiSquareDistribution	HyperbolicDistribution	NoncentralFRatioDistribution	TukeyLambdaDistribution
CompoundPoissonDistribution	HyperexponentialDistribution	NoncentralStudentTDistribution	UniformDistribution
CopulaDistribution	HypergeometricDistribution	NormalDistribution	UniformGraphDistribution
CoxianDistribution	HypoexponentialDistribution	OrderDistribution	UniformSumDistribution
DagumDistribution	InverseChiSquareDistribution	ParameterMixtureDistribution	VarianceGammaDistribution
DataDistribution	InverseGammaDistribution	ParetoDistribution	VoigtDistribution
DavisDistribution	InverseGaussianDistribution	PascalDistribution	VonMisesDistribution
DegreeGraphDistribution	JohnsonDistribution	PearsonDistribution	WakebyDistribution
DirichletDistribution	KDistribution	PERTDistribution	WalleniusHypergeometricDistribution
DiscreteUniformDistribution	KernelMixtureDistribution	PoissonConsulDistribution	WaringYuleDistribution
DistributionChart	KumaraswamyDistribution	PoissonDistribution	WattsStrogatzGraphDistribution
DistributionDomain	LandauDistribution	PolyaAeppliDistribution	WeibullDistribution
DistributionFitTest	LaplaceDistribution	PowerDistribution	WignerSemicircleDistribution

DistributionParameterAssumptions	LevyDistribution	PriceGraphDistribution	ZipfDistribution
DistributionParameterQ	LindleyDistribution	ProbabilityDistribution	
EmpiricalDistribution	LogGammaDistribution	ProductDistribution	
ErlangDistribution	LogisticDistribution	RayleighDistribution	

Apri la guida delle distribuzioni statistiche.

Le funzioni di visualizzazione.

Apri la guida Statistical Visualization

Esempi di statistica descrittiva

Le funzioni di statistica descrittiva di *Mathematica* si applicano a dati espliciti, elementi simbolici, dataset ibridi simbolic numerici, rappresentazioni simboliche di distribuzioni statistiche.
Le funzioni di statistica descrittiva sono elencate nella guida **Descriptive Statistics** (guida) ed un tutorial si trova **Descriptive Statistics** (tutorial).
Vediamo alcuni esempi utilizzando dei dati forniti dalla funzione **ExampleData**.

I dati della sorgente geyser chiamata “OldFaithful”

```
intestazioni = ExampleData[{"Statistics", "OldFaithful"}, "ColumnHeadings"]
```

{Duration, WaitingTime}

Misure descrittive dei primi quattro momenti, della mediana e del decimo e novantesimo percentile relativamente alla durata di un getto:

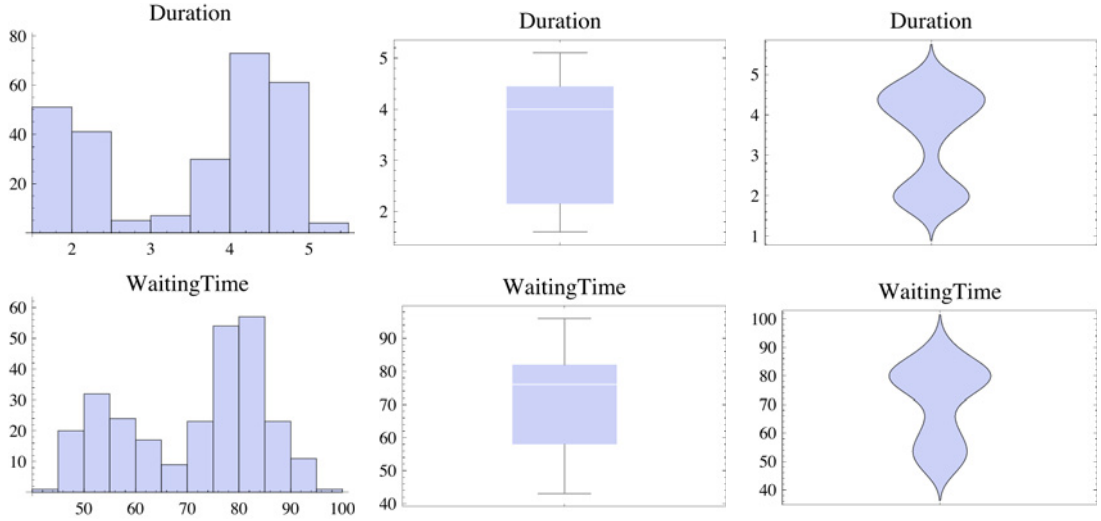
```
serie = ExampleData[{"Statistics", "OldFaithful"}];
Table[f[serie[[All, 1]]],
 {f, {Mean, Variance, Skewness, Kurtosis, Median, Quantile[#, .1] &, Quantile[#, .9] &}}]
```

{3.48778, 1.30273, −0.415841, 1.4994, 4., 1.85, 4.7}

Esempi di visualizzazione.

Grafici di tipo istogramma, box-whisker e a violino:

```
Table[f[serie[[All, i]], PlotLabel → intestazioni[[i]]],
  {i, 2}, {f, {Histogram, BoxWhiskerChart, DistributionChart}}] // Grid
```



Alcuni indici descrittivi di forma, posizione e dispersione

Calcoliamo alcune misure descrittive relative alle coppie di valori precedenti (durata, attesa).

```
Panel[Grid[Table[{f[[1]], f[[2]][N[serie]]},
    {f, {{"Media", Mean}, {"Varianza", Variance}, {"Deviazione standard", StandardDeviation},
      {"Skewness", Skewness}, {"Curtosi", Kurtosis}, {"Mediana", Median},
      {"10° percentile", Quantile[#, .1] &}, {"90° percentile", Quantile[#, .9] &}}}],
  Alignment → Left, BaseStyle → {"Times", 16, Blue}]]
```

Media	{3.48778, 70.8971}
Varianza	{1.30273, 184.823}
Deviazione standard	{1.14137, 13.595}
Skewness	{−0.415841, −0.416319}
Curtosi	{1.4994, 1.85737}
Mediana	{4., 76.}
10° percentile	{1.85, 51.}
90° percentile	{4.7, 86.}

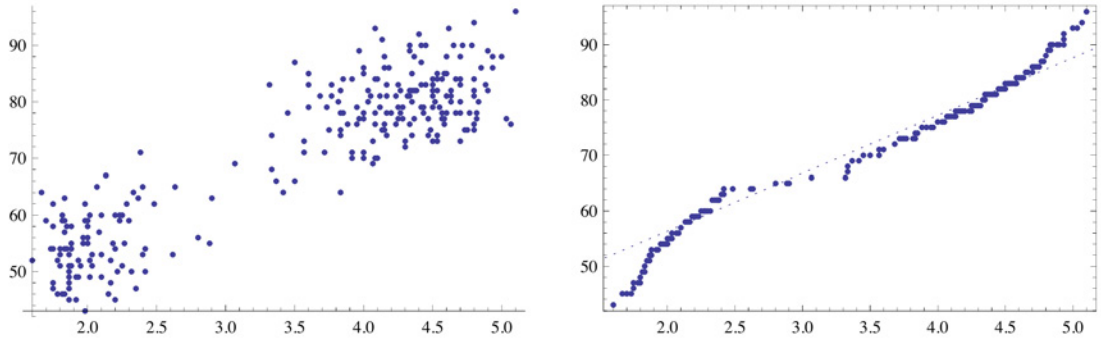
Correlazione tra la durata e l'attesa:

```
Correlation[serie] // MatrixForm
```

$$\begin{pmatrix} 1. & 0.900811 \\ 0.900811 & 1. \end{pmatrix}$$

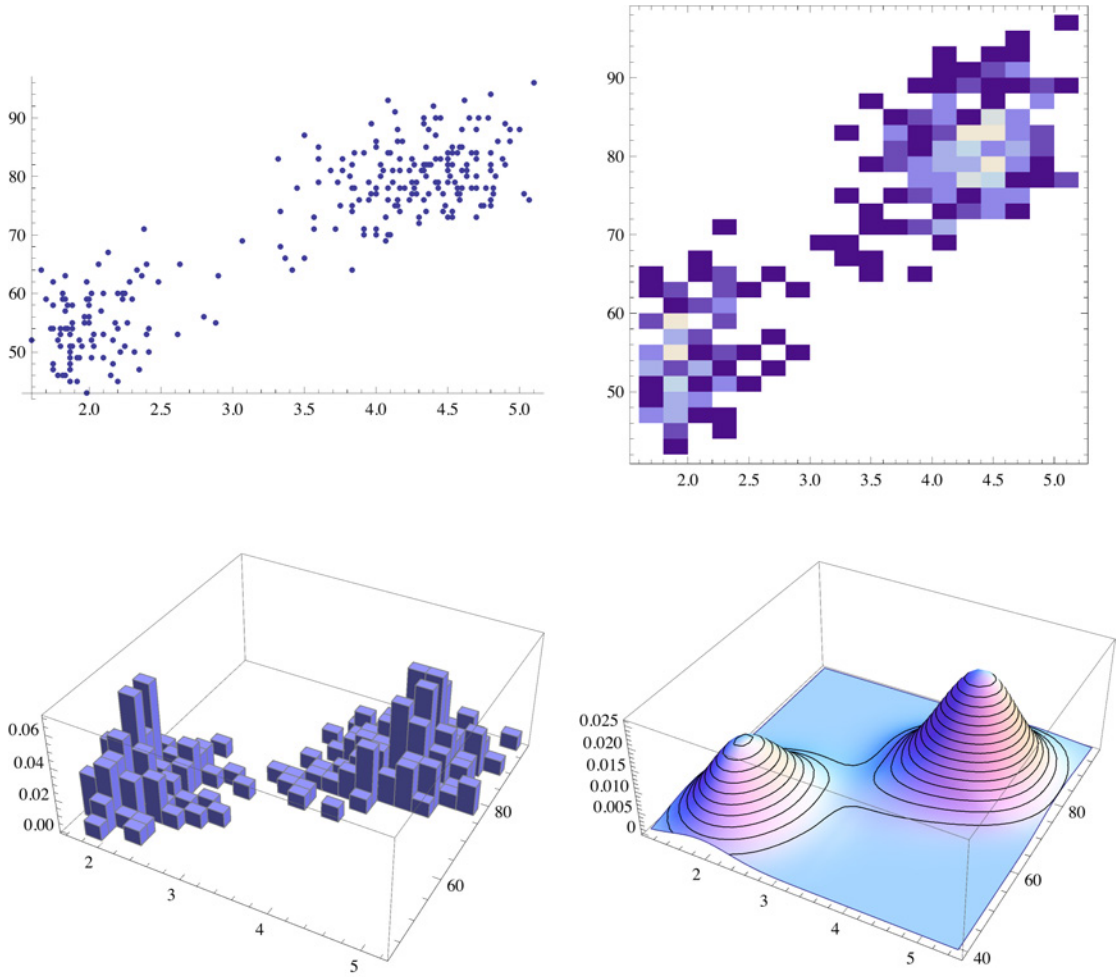
Come confermato anche dal grafico, le due varaibili sono fortemente correlate:

```
GraphicsRow[{ListPlot[serie], QuantilePlot[serie[[All, 2]], serie[[All, 1]]]}, ImageSize → Large]
```



Altre possibili visualizzazioni. Le funzioni `DensityHistogram` e `SmoothHistogram3D`, sono state aggiunte in *Mathematica* 8.

```
GraphicsGrid[{
   {ListPlot[serie], DensityHistogram[serie, 20]},
   {Histogram3D[serie, 20, "PDF"], SmoothHistogram3D[serie]}
  }, ImageSize → Large]
```



Le stesse funzioni operano anche su dati simbolici o ibridi:

```
Correlation[{{a, b}, {c, d}}] // TraditionalForm
```

$$\begin{pmatrix} 1 & \frac{(a-c)\left(b^*-d^*\right)}{\sqrt{(a-c)\left(a^*-c^*\right)}\ \sqrt{(b-d)\left(b^*-d^*\right)}} \\ \frac{(b-d)\left(a^*-c^*\right)}{\sqrt{(a-c)\left(a^*-c^*\right)}\ \sqrt{(b-d)\left(b^*-d^*\right)}} & 1 \end{pmatrix}$$

Categorie e clustering

» BinCounts

Possiamo scomporre i dati in categorie, per esempio in base all'intervallo di durata del getto.

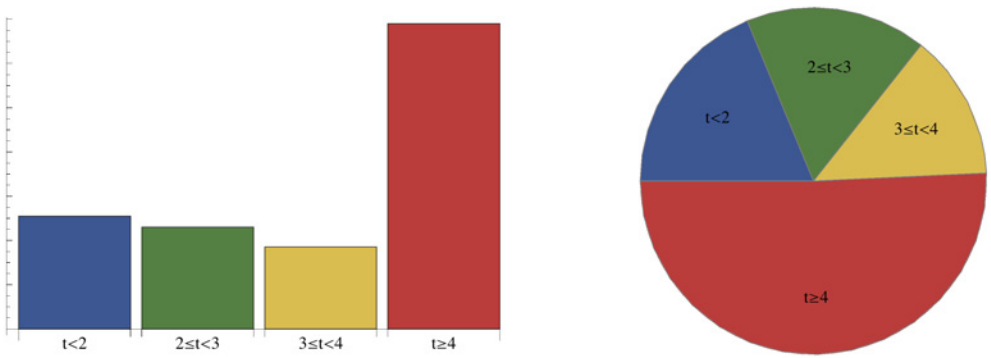
`BinCounts` viene usata per calcolare il tempo t tale che $t < 2$, $2 \le t < 3$, $3 \le t < 4$, e $t \ge 4$, rispettivamente.

```
bc = BinCounts[serie[[All, 1]], {{0, 2, 3, 4, 10}}]
```

{51, 46, 37, 138}

Visualizzazione in grafico a barre o a torta:

```
GraphicsRow[
 Table[f[bc, ChartStyle → "DarkRainbow", ChartLabels → {"t<2", "2≤t<3", "3≤t<4", "t≥4"}],
  {f, {BarChart, PieChart}}], ImageSize → Large]
```



» FindClusters

Il clustering è una tecnica spesso utilizzata per separare i dati in gruppi o identificare sottogruppi all'interno di un data set. I principale funzione di *Mathematica* per il clustering è `FindClusters`, mentre nella guida Distance and Similarity Measures

trovano tutte le possibili distanze disponibili e che si possono utilizzare per raggruppare i dati in FindClusters.

Generiamo 10 cluster campione da una distribuzione normale bivariata:

```
origclusters =
  BlockRandom[
   SeedRandom[123 456 789];
   muvals = RandomReal[{-10, 10}, {10, 2}];
   sigmavals = Table[Block[{s1 = RandomReal[], s2 = RandomReal[], s12},
      s12 = RandomReal[] * Sqrt[s1 * s2]; {{s1, s12}, {s12, s2}}], {10}];
   Join[MapThread[RandomReal[MultinormalDistribution[#1, #2], RandomInteger[{20, 75}]] &,
     {muvals, sigmavals}]]];
```

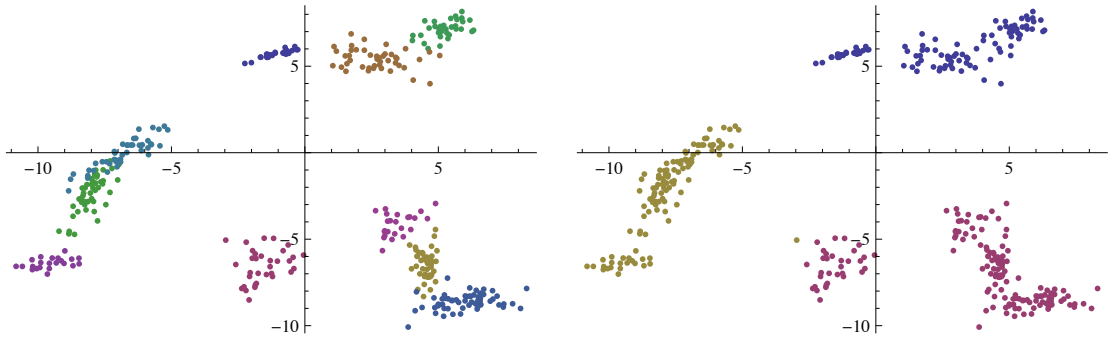
Mischiamo gli elementi così da non averli già in gruppi, e poi usiamo **FindClusters** per identificare i cluster nei dati complessivi

```
newclusters = BlockRandom[SeedRandom[1]; RandomSample[Flatten[origclusters, 1]]];
```

```
clusters = FindClusters[newclusters];
```

`FindClusters` identifica bene 3 distinti cluster, il che sembra ragionevole dato che i valori di partenza sono molto sparpagliati

```
GraphicsGrid[{{ListPlot[origclusters], ListPlot[clusters]}}, ImageSize → Large]
```



◂ | ▸

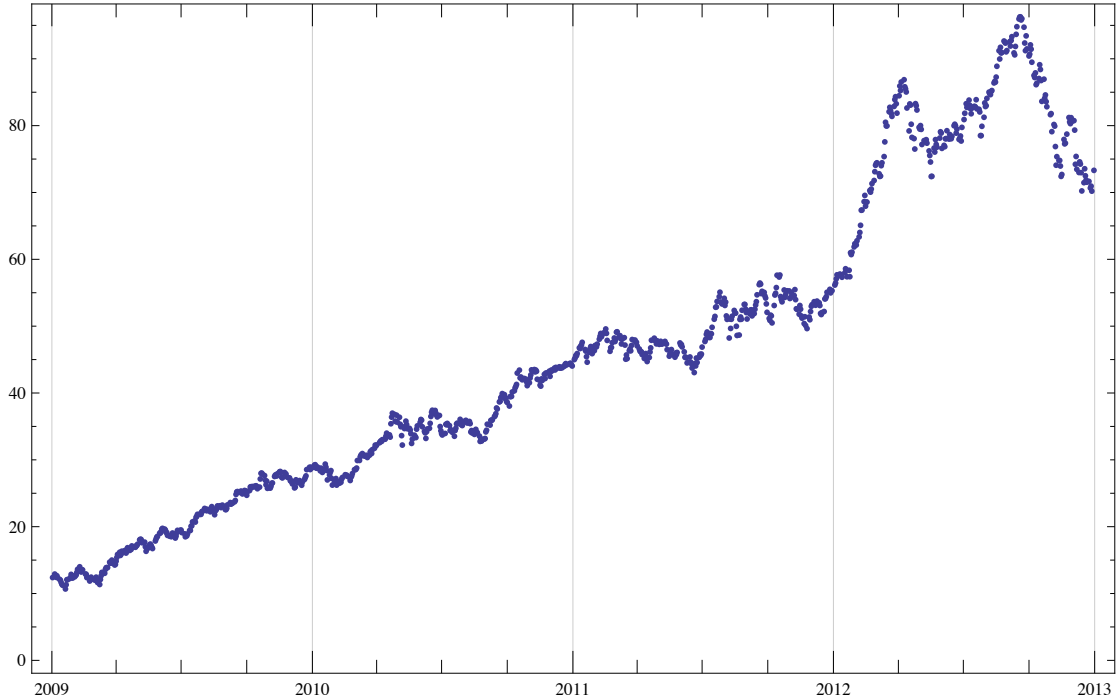
◂ | ▸

Analisi di serie storiche

Vediamo un esempio di funzioni quali `MovingAverage`, `MovingMedian`, ed `ExponentialMovingAverage`

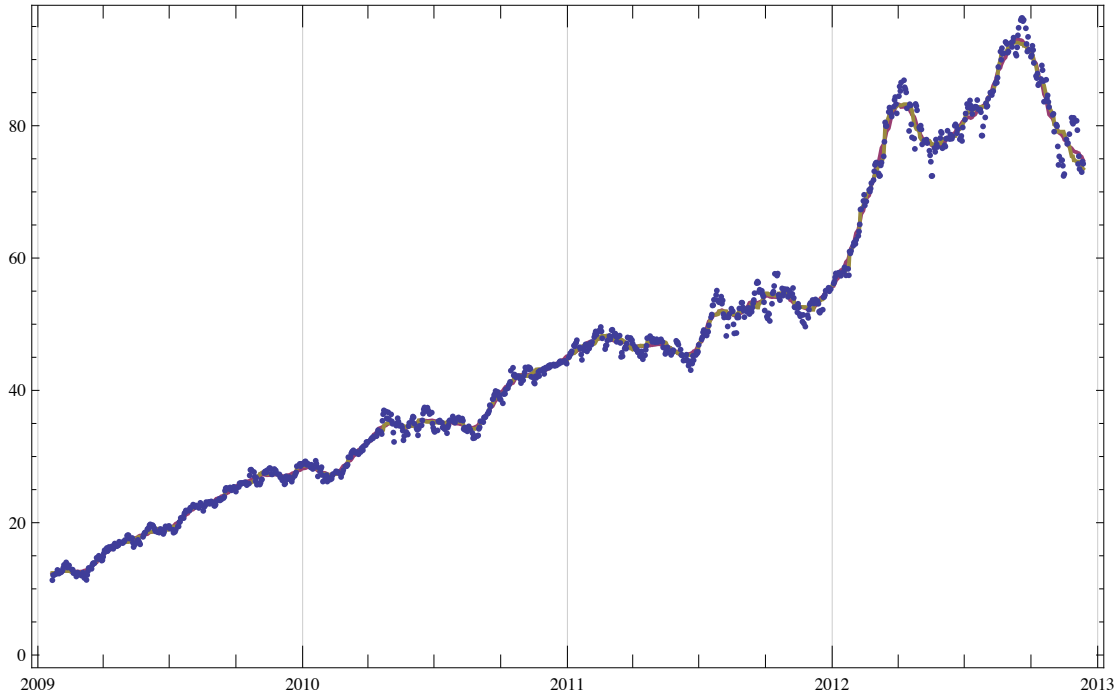
Serie storica della quotazione Apple:

```
apple = FinancialData["AAPL", {{2009}, {2013}}];
DateListPlot[apple, ImageSize → Large]
```



Questo grafico mostra una media mobile a 25 termini ed una mediana mobile

```
movavgdata = Transpose[{apple[[13 ;; -13, 1]], MovingAverage[apple[[All, 2]], 25]}];
movmeddata = Transpose[{apple[[13 ;; -13, 1]], MovingMedian[apple[[All, 2]], 25]}];
DateListPlot[{apple[[13 ;; -13]], movavgdata, movmeddata},
 Joined → {False, True, True}, PlotStyle → {Automatic, Thick, Thick}, ImageSize → Large]
```



Possiamo usare elementi dinamici per osservare le variazioni della media mobile al variare del numero di elementi presi:

```
Clear[a, b, c, d, e, f]
```

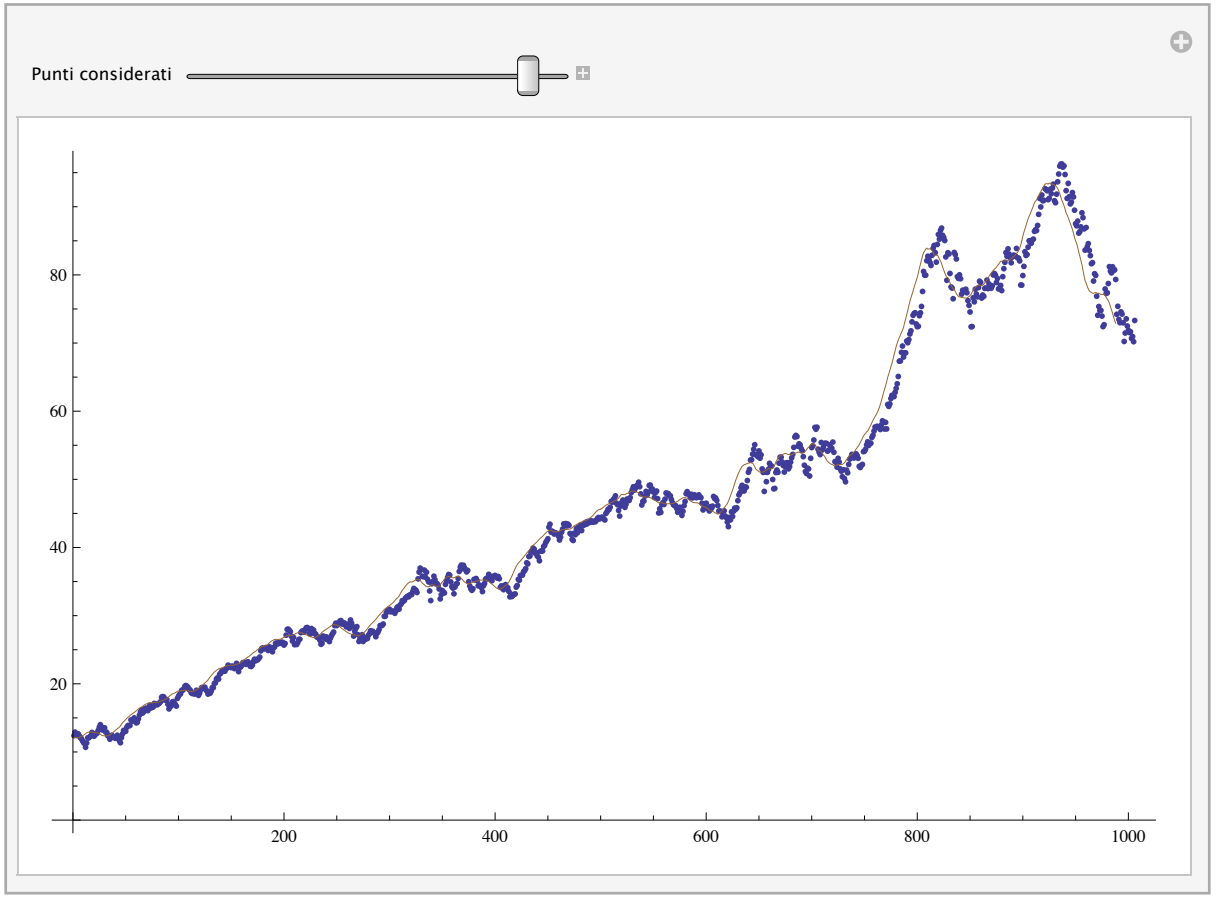
```
MovingAverage[{a, b, c, d, e, f}, 2]
```

$$\left\{\frac{a+b}{2}, \frac{b+c}{2}, \frac{c+d}{2}, \frac{d+e}{2}, \frac{e+f}{2}\right\}$$

```
MovingAverage[{a, b, c, d, e, f}, 3]
```

$$\left\{\frac{1}{3}(a+b+c), \frac{1}{3}(b+c+d), \frac{1}{3}(c+d+e), \frac{1}{3}(d+e+f)\right\}$$

```
Manipulate[Show[
  {ListPlot[apple[[All, 2]]], ListLinePlot[MovingAverage[apple[[All, 2]], n], PlotStyle → {Brown}]},
  ImageSize → Large], {{n, 1, "Punti considerati"}, 1, 20, 1}]
```



Esempio di media mobile esponenziale usata per regolarizzare dati sperimentali:

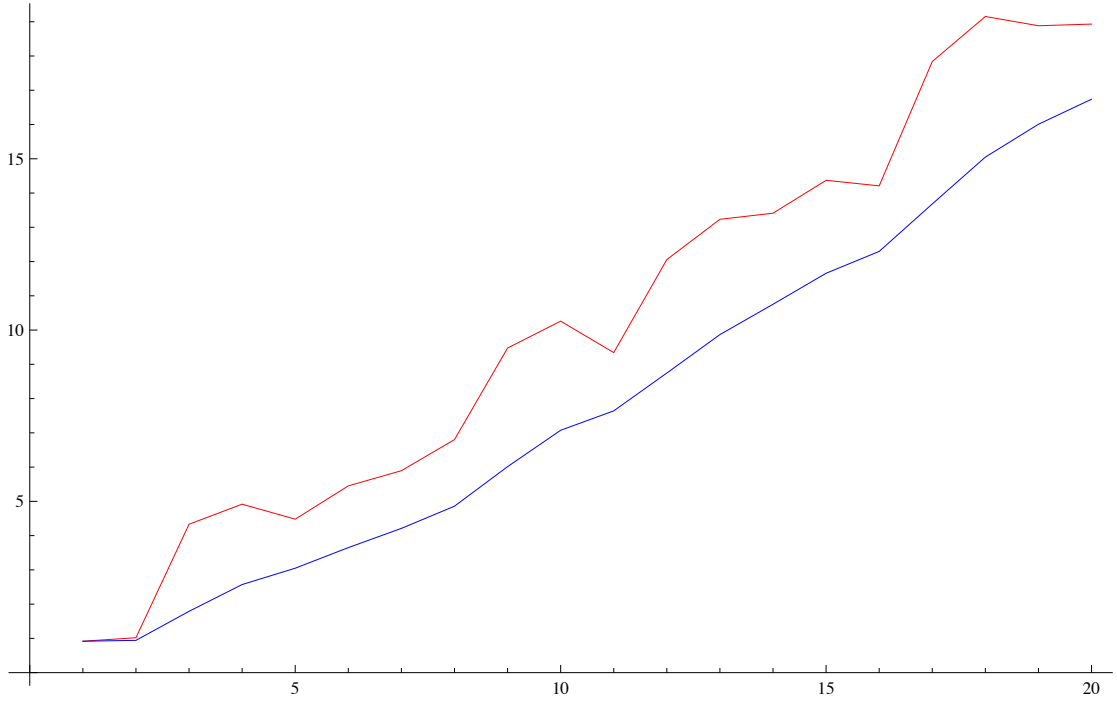
```
data = Range[20] + RandomReal[{-2, 2}, 20]
```

{0.917289, 1.01917, 4.33262, 4.91728, 4.47982, 5.4515, 5.89517, 6.80248, 9.47525,
10.2606, 9.34279, 12.0575, 13.2326, 13.4109, 14.3715, 14.2096, 17.8379, 19.1531, 18.8832, 18.9316}

```
res = ExponentialMovingAverage[data, 1 / 4]
```

{0.917289, 0.942758, 1.79022, 2.57199, 3.04895, 3.64958, 4.21098, 4.85886, 6.01295,
7.07487, 7.64185, 8.74576, 9.86746, 10.7533, 11.6579, 12.2958, 13.6813, 15.0493, 16.0078, 16.7387}

```
ListPlot[{data, res}, PlotStyle → {Red, Blue}, Joined → True, ImageSize → Large]
```



Funzionalità di statistica e analisi dati: esempi di distribuzioni statistiche

» Distribuzioni parametriche

- Discrete univariate e multivariate
- Continue univariate e multivariate

» Distribuzioni non parametriche

- EmpiricalDistribution
- HistogramDistribution
- SmoothKernelDistribution
- SurvivalDistribution

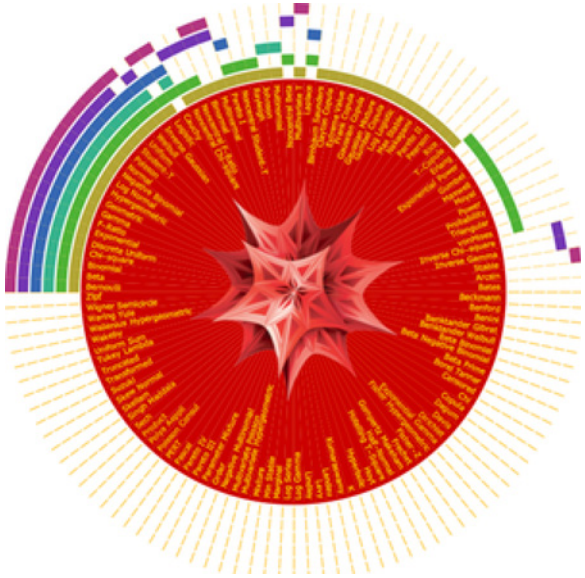
» Distribuzioni basate su formule

- ProbabilityDistribution (da PDF, CDF, SF)

» Distribuzioni derivate

- TransformedDistribution
- OrderDistribution
- MixtureDistribution
- ParameterMixtureDistribution
- CensoredDistribution
- TruncatedDistribution
- MarginalDistribution
- ProductDistribution
- CopulaDistribution

Confronto con altri software



distribution coverage in Mathematica

Funzionalità di statistica e analisi dati: esempi di distribuzioni statistiche

Per tutte le distribuzioni si possono utilizzare

» Funzioni di alto livello

- Probability, NProbability
- Expectation, NExpectation
- RandomVariate
- EstimatedDistribution, DistributionFitTest
- MomentConvert, MomentEvaluate

» Proprietà delle distribuzioni

- PDF, CDF, SurvivalFunction (SF), HazardFunction (HF), Quantile (QF), InverseCDF, InverseSurvivalFunction
- Moment / MGF, CentralMoment / CMGF, FactorialMoment / FMGF, Cumulant / CGF, CF
- Mean, Variance, StandardDeviation, Skewness, Kurtosis, Covariance, Correlation, ...
- Median, Quartiles, QuartileDeviation, QuartileSkewness, ...
- Likelihood, LogLikelihood, ...

Esempio di calcolo delle proprietà di varie distribuzioni

Calcolo di proprietà di una distribuzione `NakagamiDistribution` con parametri simbolici

```
dist = NakagamiDistribution[μ, ω];
Panel[
 Grid[{Map[Style[#, Bold, 16] &, {"Media", "Funzione caratteristica",
     "Funzione generatrice dei momenti", "Probabilità", "Valore atteso"}],
   Map[TraditionalForm, {Mean[dist], CharacteristicFunction[dist, t], MomentGeneratingFunction[
     dist, t], Probability[Log[x] > 1 / 2, x ≈ dist], Expectation[Sqrt[x], x ≈ dist]}]},
  Alignment → Center, Dividers → Directive[Orange], ItemSize → 25]]
```

Media	Funzione caratteristica	Funzione generatrice dei momenti	Probabilità	Valore
$\frac{\sqrt{\omega}\ (\mu)_{\frac{1}{2}}}{\sqrt{\mu}}$	${}_1F_1\left(\mu;\frac{1}{2};-\frac{t^2\omega}{4\mu}\right)+\frac{i\,t\sqrt{\frac{\omega}{\mu}}\ \Gamma\left(\mu+\frac{1}{2}\right){}_1F_1\left(\mu+\frac{1}{2};\frac{3}{2};-\frac{t^2\omega}{4\mu}\right)}{\Gamma(\mu)}$	${}_1F_1\left(\mu;\frac{1}{2};\frac{t^2\omega}{4\mu}\right)+\frac{t\sqrt{\frac{\omega}{\mu}}\ \Gamma\left(\mu+\frac{1}{2}\right){}_1F_1\left(\mu+\frac{1}{2};\frac{3}{2};\frac{t^2\omega}{4\mu}\right)}{\Gamma(\mu)}$	$\frac{\Gamma\left(\mu,\frac{\epsilon\mu}{\omega}\right)}{\Gamma(\mu)}$	$\frac{\sqrt[4]{\frac{\omega}{\mu}}\ \Gamma$

◄ | ►

Esempio di calcolo di probabilità e valore atteso

`Probability` e `Expectation` sono funzioni generali che operano con algoritmi simbolici. Esistono comunque le corrispondenti funzioni numeriche che secondo la convenzione di *Mathematica* assumono lo stesso nome solo che sono precedute dalla N: NProbability e NExpectation. Chiaramente i calcoli numerici possono risultare più veloci mentre quelli simbolici sono certamente più esatti.

Considerando la distribuzione usata prima, assegnamo due valori ai parametri μ e ω e calcoliamo una probabilità e poi un valore atteso:

```
With[{dist = NakagamiDistribution[1, 5]},
 {NProbability[Log[x] > 1 / 2, x ≈ dist], NExpectation[Sqrt[x], x ≈ dist]}]
```

{0.580621, 1.35539}

Verifichiamo il risultato simbolico:

```
With[{dist = NakagamiDistribution[1, 5]},
 {Probability[Log[x] > 1 / 2, x ≈ dist], Expectation[Sqrt[x], x ≈ dist]}]
```

$$\left\{e^{-e/5}, \frac{4\,\Gamma\left(\frac{9}{4}\right)}{5^{3/4}}\right\}$$

```
N[%]
```

{0.580621, 1.35539}

Ovviamente si può sfruttare la precisione arbitraria di *Mathematica* per calcolare una probabilità con una precisione maggiore di quella macchina.
Calcoliamo la probabilità che lanciando 200 volte una moneta esca un numero di volte il simbolo testa che varia tra 60 e 80:

```
Probability[60 ≤ x ≤ 80, x ≈ BinomialDistribution[200, 1 / 2]]
```

$$\frac{285\,490\,103\,818\,418\,400\,682\,713\,759\,027\,385\,648\,382\,592\,499\,707\,372\,504\,741}{100\,433\,627\,766\,186\,892\,221\,372\,630\,771\,322\,662\,657\,637\,687\,111\,424\,552\,206\,336}$$

Ora calcoliamo il corrispondente valore numerico ad una precisione arbitraria

```
N[%, 50]
```

0.00284257484438428988288218216563172723901785379274 61

Altri semplici esempi di probabilità

Prabablità univariata discreta:

```
Probability[x ≥ 1, x ≈ BinomialDistribution[n, p]]
```

$$1 - (1 - p)^n$$

Condizioni non lineari:

```
Probability[(x - 3)^2 ≥ x, x ≈ ZipfDistribution[ρ]]
```

$$\frac{2^{-2\rho-2}\,15^{-\rho-1}\left(-2^{2\rho+2}\,3^{\rho+1} - 2^{2\rho+2}\,5^{\rho+1} - 15^{\rho+1} - 30^{\rho+1} + 2^{2\rho+2}\,15^{\rho+1}\,\zeta(\rho+1)\right)}{\zeta(\rho+1)}$$

Combinazioni logiche:

```
Probability[k^2 > 10 || k < 2, k ≈ PoissonDistribution[λ]]
```

$$\frac{1}{6}\,e^{-\lambda}\left(-\lambda^3 - 3\,\lambda^2 + 6\,e^{\lambda}\right)$$

Probabilità multivariata:

```
Probability[X ≤ 1 / 2 && Y ≤ 1, {X, Y} ≈ DirichletDistribution[{1, 2, 3}]]
```

$$\frac{31}{32}$$

Probabilità condizionata:

```
Probability[x^2 > 12 ⧐ x > 2, x ≈ PoissonDistribution[λ]]
```

$$\frac{\lambda^3 + 3\,\lambda^2 + 6\,\lambda - 6\,e^{\lambda} + 6}{3\,\lambda^2 + 6\,\lambda - 6\,e^{\lambda} + 6}$$

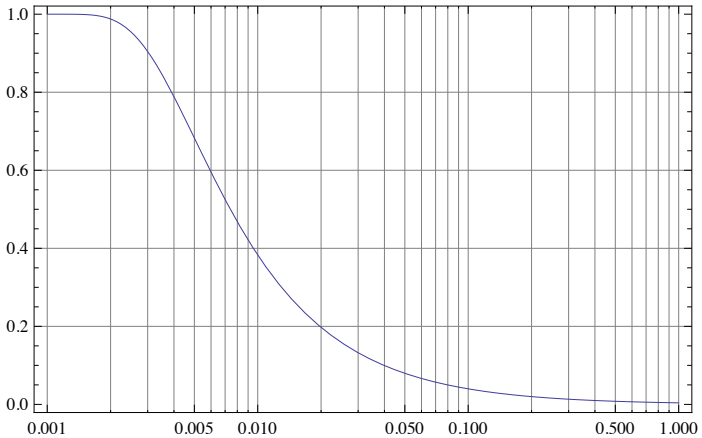
» Applicazioni immediate

Una fabbrica produce chiodi la cui lunghezza è un valore con distribuzione normale e media 0.5 pollici. Se il 50% della produzio ha una lunghezza compresa tra 0.495 e 0.505, trovare la deviazione standard:

```
prob = Probability[0.495 < x < 0.505, x ≈ NormalDistribution[0.5, σ]]
```

$$\frac{1}{2}\left(\text{erfc}\left(-\frac{0.00353553}{\sigma}\right)-\text{erfc}\left(\frac{0.00353553}{\sigma}\right)\right)$$

```
LogLinearPlot[prob, {σ, 10^-3, 1}, GridLines → Automatic, Frame → True]
```




```
NSolve[prob == 0.5 && 10^-3 < σ < 10^-1, σ]
```

Solve::ratnz : Solve was unable to solve the system with inexact coefficients.

The answer was obtained by solving a corresponding exact system and numericizing the result. ≫

$\{\{\sigma \to 0.00741301\}\}$

Creare data set sperimentali

Molte volte risulta fondamentale anche la generazione di numeri pseudo-causali secondo una certa distribuzione. In *Mathematic* 8 è stata creata una nuova funzione appositamente per la generazione di campioni casuali, chiamata `RandomVariate`

Simuliamo il lancio di un dado:

```
RandomVariate[DiscreteUniformDistribution[{1, 6}], 1]
```

{4}

Di due dadi:

```
RandomVariate[DiscreteUniformDistribution[{1, 6}], 2]
```

{2, 3}

Generiamo un campione di 100 lanci di due dadi:

```
RandomVariate[DiscreteUniformDistribution[{1, 6}], {100, 2}]
```

$$\begin{pmatrix} 3 & 4 \\ 5 & 6 \\ 2 & 2 \\ 1 & 6 \\ 5 & 6 \\ 1 & 5 \\ 5 & 6 \\ 4 & 5 \\ 2 & 4 \\ 5 & 6 \\ 3 & 6 \\ 5 & 1 \\ 6 & 3 \\ 1 & 3 \\ 1 & 4 \\ 6 & 3 \end{pmatrix}$$

4	2
3	5
4	3
5	6
3	5
4	2
6	6
1	1
2	4
5	2
2	5
5	1
2	2
1	5
3	1
4	1
5	2
3	1
3	4
5	2
1	6
3	6
2	1
3	4
2	6
1	4
4	2
2	5
5	4
6	4
3	4

1	2
5	3
4	4
2	3
3	2
5	1
1	6
4	5
1	3
3	1
6	1
4	6
5	6
5	2
3	6
6	5
5	4
3	3
3	6
2	5
3	2
3	4
6	6
3	3
6	1
4	1
6	3
6	1
1	5
6	5
3	4
4	4

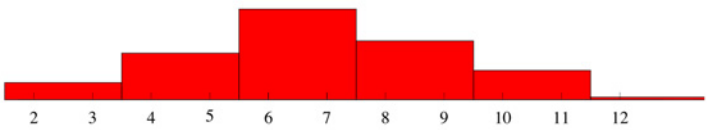
$$\begin{pmatrix} 4 & 4 \\ 3 & 5 \\ 3 & 3 \\ 1 & 3 \\ 6 & 2 \\ 4 & 5 \\ 4 & 2 \\ 2 & 6 \\ 6 & 4 \\ 4 & 6 \\ 3 & 5 \\ 2 & 6 \\ 5 & 1 \\ 4 & 6 \\ 4 & 5 \\ 2 & 4 \\ 5 & 6 \\ 6 & 3 \\ 3 & 6 \\ 2 & 5 \\ 3 & 5 \\ 5 & 1 \end{pmatrix}$$

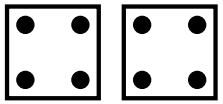
Ancora una volta è interessante riportare l'attenzione su altre funzionalità integrate in *Mathematica*, come ad esempio quelle dedicate alla creazione di semplici interfacce:

new set of trials

total trials 1000

number of trials 100





100 trials out of 1000

Modelli statistici

Mathematica dispone anche di funzioni ad hoc per l'analisi statistica tramite modelli lineari e non lineari (si veda il tutorial Statis cal Model Analysis).

`LinearModelFit`	modello lineare
`GeneralizedLinearModelFit`	modello lineare generalizzato
`LogitModelFit`	modello regressione logistica
`ProbitModelFit`	modello regressione probit
`NonlinearModelFit`	modello non lineare dei minimi quadrati

La caratteristica importante delle funzioni in questa categoria è che generano un oggetto di tipo `FittedModel` ossia uno model completo con tutti i parametri diagnostici utili alla piena comprensione del fitting eseguito. Facciamo un esempio.

Importiamo dei dati da un file di test:

```
SetDirectory[NotebookDirectory[]];
```

```
lineardata = {{8.71, 6.92, 18.89}, {6.05, 5.97, 15.08},
   {6.24, 0.99, 5.92}, {8.25, 3.37, 11.39}, {6.58, 8.22, 20.77}, {4.14, 9., 21.09},
   {4.35, 9.94, 24.32}, {8.99, 4.47, 13.79}, {2.82, 3.91, 10.68}, {5.14, 0.4, 3.82}};
```

Facciamone una regressione lineare usando `LinearModelFit`

```
lm = LinearModelFit[lineardata, {x, y}, {x, y}]
```

FittedModel[0.340391 x + 2.08429 y + 1.40308]

```
Normal[lm]
```

$0.340391\, x + 2.08429\, y + 1.40308$

`LinearModelFit` produce un oggetto di tipo `FittedModel` che include tutta l'analisi eseguita sui dati e che possiamo ottenere semplicemente esprimendo le richieste:

```
lm["Properties"]
```

{AdjustedRSquared, AIC, AICc, ANOVATable, ANOVATableDegreesOfFreedom, ANOVATableEntries, ANOVATableFStatistics, ANOVATableMeanSquares, ANOVATablePValues, ANOVATableSumsOfSquares, BasisFunctions, BetaDifferences, BestFit, BestFitParameters, BIC, CatcherMatrix, CoefficientOfVariation, CookDistances, CorrelationMatrix, CovarianceMatrix, CovarianceRatios, Data, DesignMatrix, DurbinWatsonD, EigenstructureTable, EigenstructureTableEigenvalues, EigenstructureTableEntries, EigenstructureTableIndexes, EigenstructureTablePartitions, EstimatedVariance, FitDifferences, FitResiduals, Function, FVarianceRatios, HatDiagonal, MeanPredictionBands, MeanPredictionConfidenceIntervals, MeanPredictionConfidenceIntervalTable, MeanPredictionConfidenceIntervalTableEntries, MeanPredictionErrors, ParameterConfidenceIntervals, ParameterConfidenceIntervalTable, ParameterConfidenceIntervalTableEntries, ParameterConfidenceRegion ParameterErrors, ParameterPValues, ParameterTable, ParameterTableEntries, ParameterTStatistics, PartialSumOfSquares, PredictedResponse, Properties, Response, RSquared, SequentialSumOfSquares, SingleDeletionVariances, SinglePredictionBands, SinglePredictionConfidenceIntervals, SinglePredictionConfidenceIntervalTable, SinglePredictionConfidenceIntervalTableEntries, SinglePredictionErrors, StandardizedResiduals, StudentizedResiduals, VarianceInflationFactors}

Il valore del modello nel punto $\{x, y\} = \{1, 3\}$.

```
lm[1, 3]
```

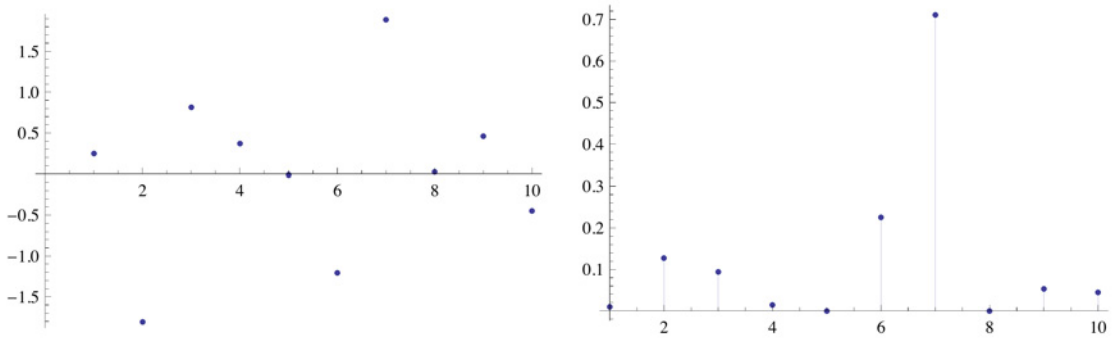
7.99634

I residui standardizzati e la distanza di Cook:

```
{stresids, cookd} = lm[{"StandardizedResiduals", "CookDistances"}]
```

$$\begin{pmatrix} 0.248592 & -1.80966 & 0.815043 & 0.368262 & -0.0132824 & -1.20686 & 1.88487 & 0.0250828 & 0.459402 & -0.446663 \\ 0.00970292 & 0.127308 & 0.0936223 & 0.0143911 & 0.0000147254 & 0.225197 & 0.71086 & 0.0000955904 & 0.052746 & 0.044873 \end{pmatrix}$$

```
GraphicsRow[
 {ListPlot[stresids], ListPlot[cookd, Filling → Bottom, PlotRange → All]}, ImageSize → Large]
```



Un esempio reale di calcolo statistico

La potenza del vento

▲ Image courtesy of Wikipedia: User Fanny Schertzer

Questo esempio è tratto da un articolo a cura di Jon McLoone, della Wolfram Research

La Statistica dell'Energia Eolica

Come può contribuire *Mathematica* nel campo dell'energia sostenibile? Vedremo che con una combinazione di dati built-in funzionalità statistiche *Mathematica* può dare un buon supporto all'analisi energetica.

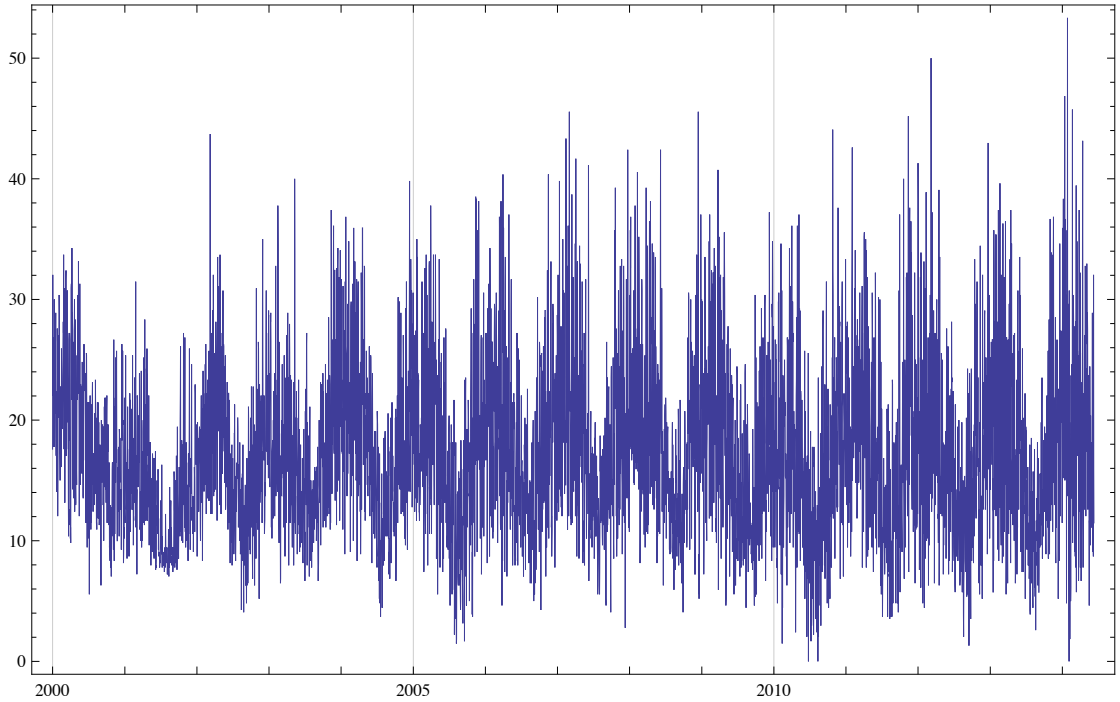
In questo esempio faremo uso di informazioni tratte da WeatherData e studieremo la possibilità di utilizzare il vento per genera elettricità. [Verranno anche utilizzati dati provenienti da www.vestas.com/en/wind-power-plants/procurement/turbin overview/v90-1.8/2.0-mw.aspx#/vestas-univers]

In primo luogo otteniamo la velocità media del vento giornaliera per la stazione meteo più vicina a Bloomington IL, a partire dal 2000.

```
wsData = WeatherData[{"Bloomington", "Illinois"}, "MeanWindSpeed", {{2000, 1, 1}, Date[], "Day"}];
```

Possiamo dare un'occhiata alla forma dei dati e visualizzare immediatamente i dati della serie temporale:

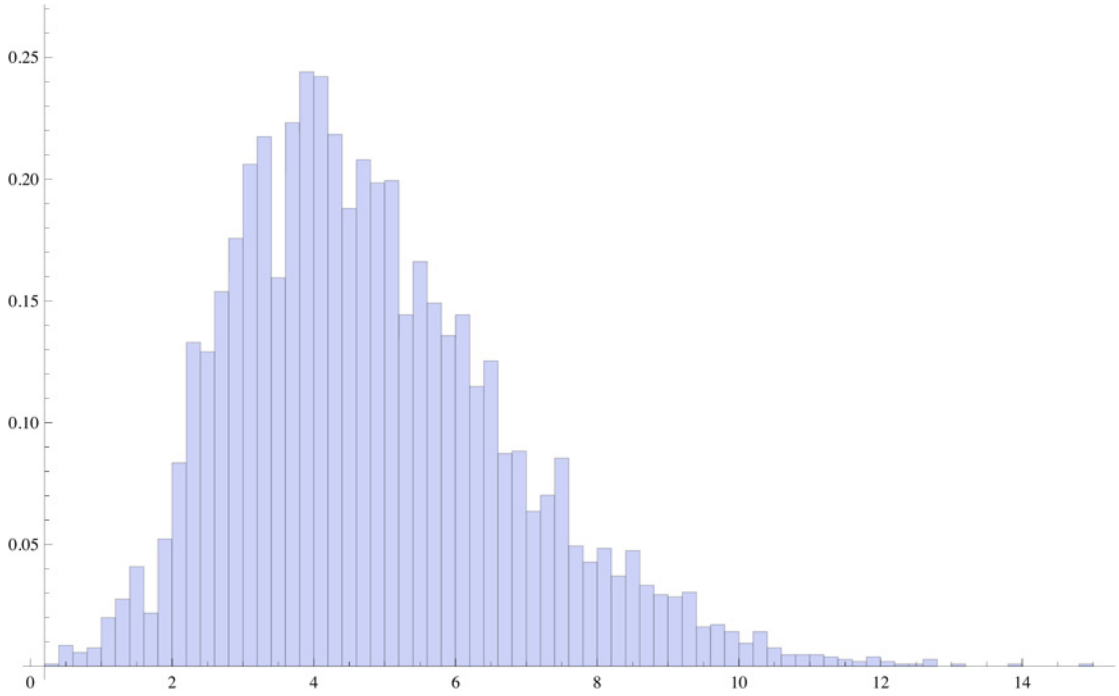
```
DateListPlot[wsData, Joined → True, ImageSize → Large]
```



Le voci sono coppie ordinate consistenti di data e velocità media del vento per la giornata. Avremo bisogno di estrarre le velocità dai dati.

Prendiamo la seconda parte di tutte le coppie e convertiamo da km/h a m/s; quindi visualizziamo l'istogramma.

```
wndData = Select[1000. wsData[[All, 2]] / 60.^2, # > 0 &];
histplot = Histogram[wndData, 45, "PDF", ImageSize → Large]
```

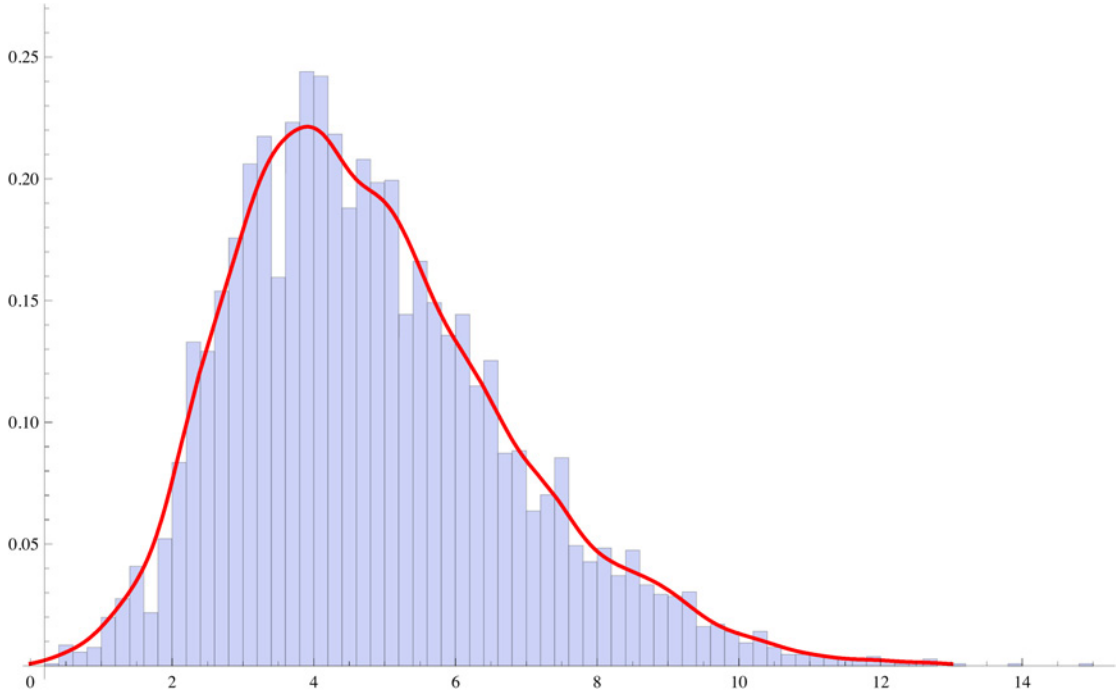


La Statistica dell'Energia Eolica

Poiché la velocità del vento può essere assunta variare continuamente, possiamo rappresentare la distribuzione per le nostre osse vazioni utilizzando la funzione SmoothKernelDistribution.

Visualizziamo la distribuzione con l'istogramma originale.

```
dist = SmoothKernelDistribution[wndData];
Show[{histplot, estplot = Plot[PDF[dist, x], {x, 0, 13}, PlotStyle → {Thick, Red}]}]
```



Possiamo chiedere informazioni su tale distribuzione. Per esempio, qual è la probabilità che la velocità del vento sia maggiore di 3,5 m/s?

```
Probability[speed > 3.5, speed ≈ dist]
```

0.725259

La Statistica dell'Energia Eolica

Comunque, stiamo assumendo che i dati seguano una ExtremeValueDistribution, e li utilizzeremo per stimare i parametri del distribuzione.

```
params = FindDistributionParameters[wndData, ExtremeValueDistribution[α, β]]
```

$\{\alpha \to 3.90168, \beta \to 1.63055\}$

```
distEV = ExtremeValueDistribution[α, β] /. params
```

ExtremeValueDistribution[3.90168, 1.63055]

Ed ora verifichiamo la nostra ipotesi.

```
tstData = DistributionFitTest[wndData, distEV, "HypothesisTestData"];
```

```
{tstData["AutomaticTest"], tstData["TestConclusion"]}
```

{CramerVonMises, The null hypothesis that the data is distributed according to the ExtremeValueDistribution[3.90168, 1.63055] is not rejected at the 5 percent level based on the Cramér-von Mises test.}

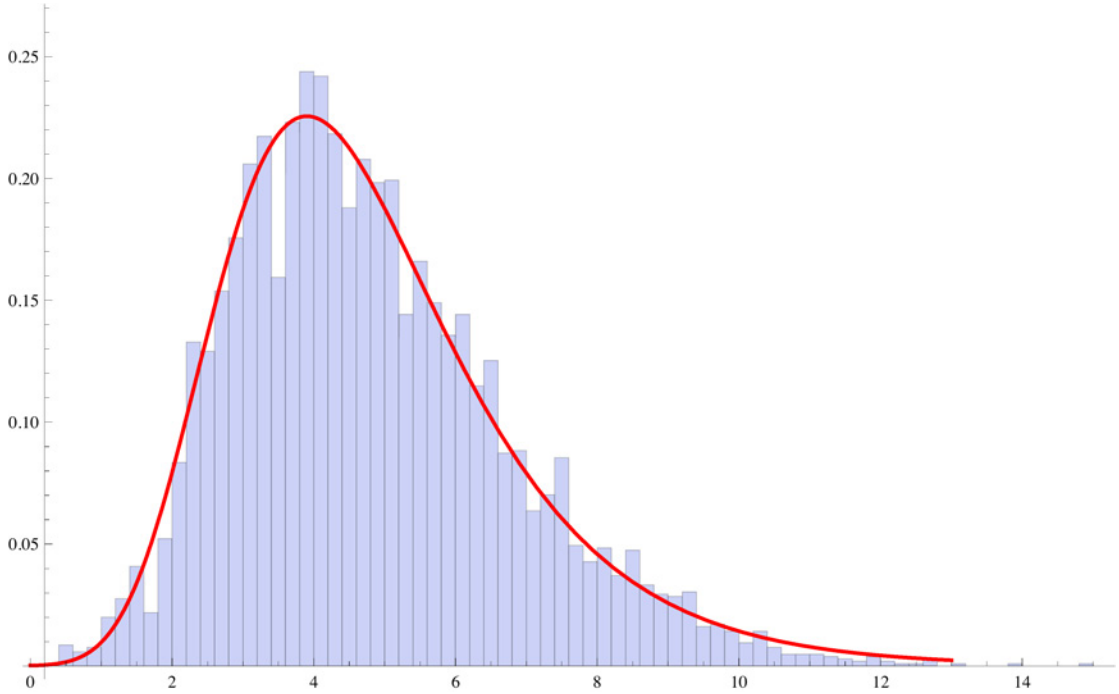
Ecco la statistica del test ed il p-Value.

```
tstData["TestDataTable"]
```

	Statistic	P-Value
Cramér-von Mises	0.223041	0.226948

Possiamo ora visualizzare la nostra stima della distribuzione.

```
Show[{histplot, estplot = Plot[PDF[distEV, x], {x, 0, 13}, PlotStyle → {Thick, Red}]},
 ImageSize → Large]
```



Il generatore che stiamo usando per la nostra analisi richiede un vento di 3,5 m/s per generare elettricità. Possiamo usare la fu
zione Probability per stimare per quale percentuale di tempo il generatore contribuirà alla rete.

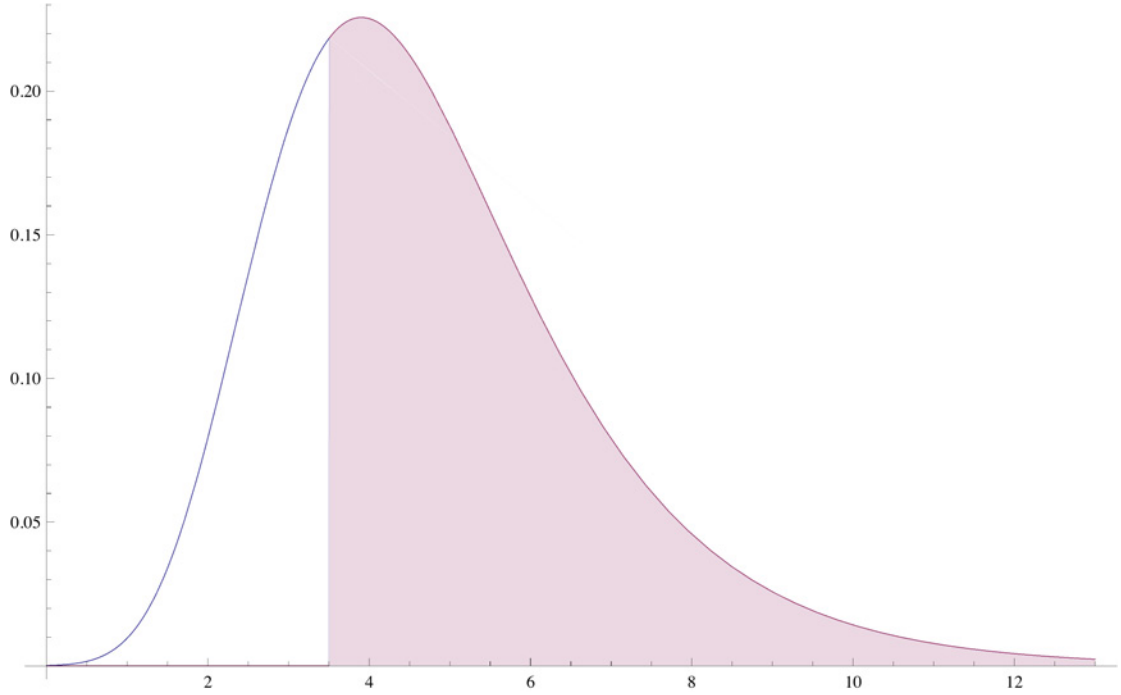
Sulla base della distribuzione dei dati, è possibile calcolare la probabilità che la velocità del vento superi la soglia di 3,5 m/s:

```
Probability[x > 3.5, x ≈ distEV]
```

0.721779

Visualizziamo il risultato.

```
Plot[Evaluate[PDF[distEV, x] {1, Boole[x > 3.5]}],
  {x, 0, 13}, Filling → {{2 → Axis}}, ImageSize → Large]
```



La Statistica dell'Energia Eolica

Dalla scheda tecnica del produttore per il generatore V90-1.8/2.0 MW possiamo fare una tabella che indichi i livelli di prestazior attesi per la posizione individuata.

```
Grid[{
   {Style["Wind Speed m/s", FontWeight → "Bold"],
    Style["Probability", FontWeight → "Bold"], Style["Estimated Power MW", FontWeight → "Bold"]},
   {"< 3.5", Probability[x < 3.5, x ≈ distEV], 0.0},
   {"Between 3.5 and 7", Probability[3.5 < x < 7, x ≈ distEV], "0 to .8"},
   {"Between 7 and 13", Probability[7 < x < 13, x ≈ distEV], ".8 to 1.65"},
   {"> 13", Probability[13 < x, x ≈ distEV], "1.65"}
 }, BaseStyle → {"Menu", 19}, Alignment → {Left},
 Background → LightYellow, Frame → {False, All}, Spacings → {2, 0.62}]
```

Wind Speed m/s	Probability	Estimated Power MW
< 3.5	0.278221	0.
Between 3.5 and 7	0.58288	0 to .8
Between 7 and 13	0.135133	.8 to 1.65
> 13	0.00376579	1.65

Siamo in grado di calcolare la potenza che questo generatore accumulerà nel periodo di tempo dal 2000 fino ad oggi:

```
power = Total[{
    Probability[3.5 < x < 7, x ≈ distEV] Mean[{0, 0.8}],
    Probability[7 < x < 13, x ≈ distEV] Mean[{1.65, 0.8}],
    Probability[13 < x, x ≈ distEV] 1.65}]
```

0.404903

hours in a year

Result

8760 hours

cost of electricity in IL

Result

10.73¢/kW·h (US cents per kilowatt hour) (01/03/2014)

Se avessimo acquistato energia elettrica per l'ammontare calcolato, avremmo speso il seguente importo in $ per un anno (ques solo per un generatore eolico):

```
valuePerYear = "$" power * 8760 * 10.73 / 100 * 1000
```

380 588. $

Conclusioni

Anche per la statistica e l'analisi dei dati *Mathematica* si conferma un sistema altamente competitivo anche rispetto a quei sof
ware dedicati che offrono esclusivamente funzioni in tale aree. Questo è ancora un altro esempio di come *Mathematica* stia dive
tando sempre più uno strumento integrato e completo.

In relazione all'analisi dei dati ed alla statistica con *Mathematica* esistono moltissime fonti informative tra cui:

Tutorial

- Basic Statistics
- Continuous Distributions
- Discrete Distributions
- Descriptive Statistics
- Convolutions and Correlations
- Curve Fitting
- Importing and Exporting Data
- Pseudorandom Numbers
- Random Number Generation
- Numerical Operations on Data

Seminari e Corsi Wolfram Education Group

- S18: Import and Export Formats in *Mathematica*
- S24: Working with Imported Data in *Mathematica*
- S41: Statistical Visualization with *Mathematica*
- M215: Applied Statistical Analysis with *Mathematica*
- M221: Introduction to Programming in *Mathematica*

Sul Web

- Examples on the Web: Statistical Visualization
- Wolfram Demonstrations Project: demonstrations.wolfram.com
- Wolfram Library Archive: library.wolfram.com
- Training from Wolfram Education Group: www.wolfram.com/weg