Statistics

Dennis Silverman
Mathematical Physics 212C
Department of Physics and Astronomy
U. C. Irvine

Some Examples taken from "Guide to Standard Mathematica Packages", Version 2.2, Wolfram Research

■ Confidence Intervals

■ Confidence Intervals for Data

```
<< Statistics`ConfidenceIntervals`
```

```
data1 = {2.1, 1.2, 0.7, 1.0, 1.1, 3.2, 3.2, 3.3, 2.1, 0.3};
```

```
mean = Mean[data1]
```

```
1.82
```

```
?? VarianceCI
```

```
VarianceCI[list] returns a list {min, max} representing a confidence interval for the
  population variance, using the entries in list as a sample drawn from the population.

Attributes[VarianceCI] = {Protected, ReadProtected}

Options[VarianceCI] = ConfidenceLevel -> 0.95
```

```
VarianceCI[data1]
```

```
{0.593815, 4.1831}
```

```
VarianceCI[data1, ConfidenceLevel -> 0.90]
```

```
{0.667653, 3.39718}
```

■ Confidence Intervals For Distributions

?? NormalCI

```
NormalCI[mean, sd, ConfidenceLevel -> c] returns a list {min, max} representing a confidence
  interval at confidence level c for the population mean, based on the sample mean and
  its standard deviation. This function is used by MeanCI when the population  variance is specified.

Attributes[NormalCI] = {Protected, ReadProtected}

Options[NormalCI] = ConfidenceLevel -> 0.95
```

NormalCI[0, 1]

```
{-1.95996, 1.95996}
```

NormalCI[0, 1, ConfidenceLevel -> 0.90]

```
{-1.64485, 1.64485}
```

?? ChiSquareCI

```
ChiSquareCI[var, dof, ConfidenceLevel -> c] returns a list {min, max} representing a confidence
  interval at confidence level c for the population variance, based on a sample with
  dof degrees of freedom and unbiased variance estimate var.  This function is used by VarianceCI.

Attributes[ChiSquareCI] = {Protected, ReadProtected}

Options[ChiSquareCI] = ConfidenceLevel -> 0.95
```

ChiSquareCI[20, 20, ConfidenceLevel -> 0.67]

```
{15.3701, 28.7551}
```

ChiSquareCI[20, 20, ConfidenceLevel -> 0.90]

```
{12.7346, 36.8636}
```

ChiSquareCI[20, 20]

```
{11.7063, 41.7067}
```

■ Normal Distribution Functions

<< Statistics`NormalDistribution`

? ChiSquareDistribution

```
ChiSquareDistribution[n] represents the chi-square distribution with n degrees of freedom.
```

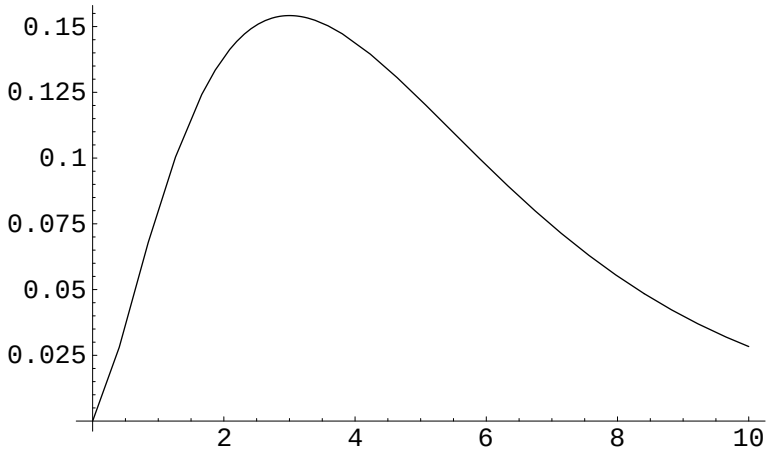
? PDF

```
PDF[distribution, x] gives the probability density function of the specified statistical
  distribution evaluated at x.
```

```
chis = PDF[ChiSquareDistribution[5], x]
```

$$\frac{E^{-x/2}\, x^{3/2}}{3\sqrt{2\,\pi}}$$

```
Plot[chis, {x, 0, 10}]
```



- Graphics -

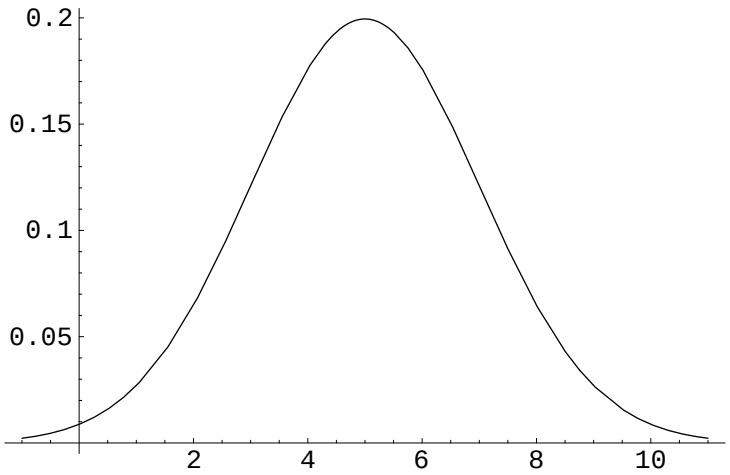
```
?NormalDistribution
```

```
NormalDistribution[mu, sigma] represents the normal (Gaussian) distribution with mean
  mu and standard deviation sigma.
```

```
ndf = PDF[NormalDistribution[5, 2], x]
```

$$\frac{E^{-\frac{1}{8}(-5+x)^2}}{2\sqrt{2\,\pi}}$$

```
Plot[ndf, {x, -1, 11}]
```



- Graphics -

```
Mean[NormalDistribution[5, 2]]
```

```
5
```

```
Variance[NormalDistribution[5, 2]]
```

```
4
```

```
StandardDeviation[NormalDistribution[5, 2]]
```

```
2
```

```
? Quantile
```

```
Quantile[list, q] gives the q-th quantile of the entries in list (the fraction of
  entries in list that are less than this value is q). Quantile[distribution, q] gives
  the q-th quantile of the specified statistical distribution.
```

```
Quantile[NormalDistribution[5, 2], .9]
```

```
7.5631
```

```
Quantile[NormalDistribution[5, 2], .1]
```

```
2.4369
```

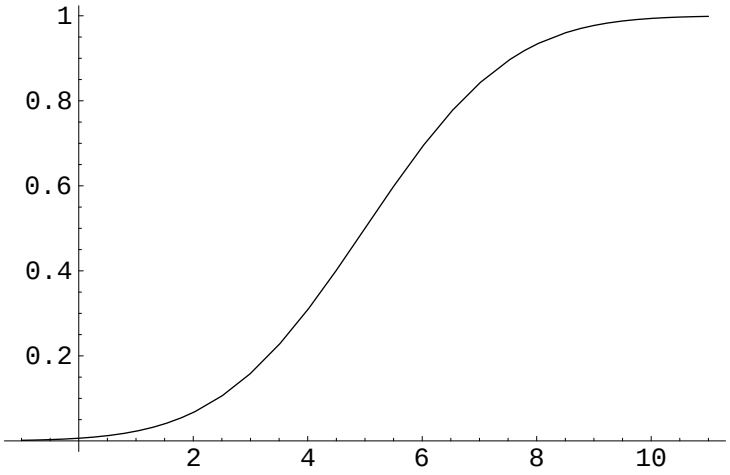
```
? CDF
```

```
CDF[distribution, x] gives the cumulative distribution function of the
  specified statistical distribution evaluated at x.  For continuous distributions,
  this is defined as the integral of the probability density function from the lowest
  value in the domain to x.  For discrete distributions, this is defined as the sum of
  the probability density function from the lowest value in the domain to x.
```

```
ncdf = CDF[NormalDistribution[5, 2], x]
```

$$\frac{1}{2}\left(1+\text{Erf}\left[\frac{-5+x}{2\sqrt{2}}\right]\right)$$

```
Plot[ncdf, {x, -1, 11}]
```




```
- Graphics -
```

```
charndf = CharacteristicFunction[NormalDistribution[5, 2], k]
```

$$E^{5\,I\,k-2\,k^2}$$

■ Discrete Distributions

```
<< Statistics`DiscreteDistributions`
```

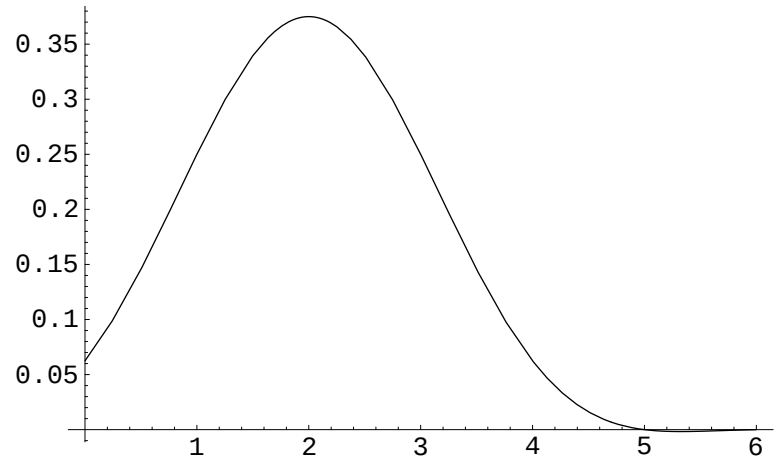
```
? BinomialDistribution
```

```
BinomialDistribution[n, p] represents the binomial distribution for n trials with
  probability p.  A BinomialDistribution[n, p] random variable describes the number of
  successes occuring in n trials, where the probability of success in each trial is p.
```

```
binompdf = PDF[BinomialDistribution[4, .5], m]
```

```
0.0625 Binomial[4, m]
```

```
Plot[binompdf, {m, 0, 6}]
```




```
- Graphics -
```

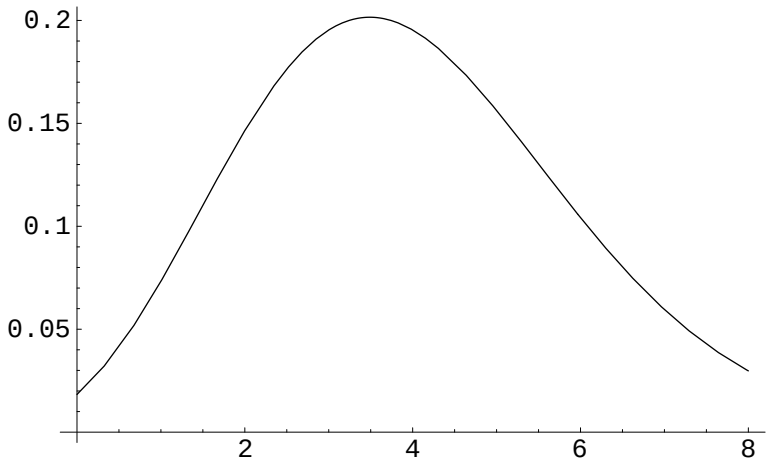
```
? PoissonDistribution
```

```
PoissonDistribution[mu] represents the Poisson distribution with mean mu.
```

```
poissonpdf = PDF[PoissonDistribution[4], m]
```

$$\frac{4^m}{E^4\,m!}$$

```
Plot[poissonpdf, {m, 0, 8}]
```




```
- Graphics -
```

```
N[Exp[-4]]
```

```
0.0183156
```

■ Nonlinear Function Fitting

```
<< Statistics`NonlinearFit`
```

```
? NonlinearFit
```

```
NonlinearFit[data, model, vars, params, (opts)] searches for a least-squares  fit to a list of data
  according to the model containing the variables vars and the parameters params.  Parameters may be
  expressed as a list of symbols  or a list of lists, where each parameter is listed with starting
  value(s) and bounds in one of several different ways: {symbol, start} or {symbol, min, max} or
  {symbol, {start1, start2}} or {symbol, {start1, start2}, min, max}.  The data can have the form
  {{x1, y1, ..., f1}, {x2, y2, ..., f2}, ...},  where the number of coordinates x, y, ... is equal
  to the number of variables in the list vars.  The data can also be of the form {f1,
  f2, ...}, with a single coordinate assumed to take values 1, 2, ....  Method specifies
  the LevenbergMarquardt or FindMinimum search methods.  AccuracyGoal and PrecisionGoal
  specify the number of digits of absolute and relative error allowed in the residual sum of squares.
```

```
data = Table[{n, 3 Cos[2 n]}, {n, 0., 1.5, .5}]
```

```
{{0., 3.}, {0.5, 1.62091}, {1., -1.24844}, {1.5, -2.96998}}
```

```
NonlinearFit[data, b Cos[a x], x, {a, b}]
```

```
2.76709 Cos[2.08175 x]
```

```
Uses Levenberg Marquardt Method going from steepest descents to quadratic minimization.
```