Statistics

Physics 212 C
Dennis Silverman

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

■ Confidence Intervals

■ Confidence Intervals for Data

```
In[1]:= << Statistics`ConfidenceIntervals`

In[3]:= data1 = {2.1, 1.2, 0.7, 1.0, 1.1, 3.2, 3.2, 3.3, 2.1, 0.3};

In[4]:= mean = Mean[data1]

Out[4]= 1.82

In[7]:= ?? 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

In[8]:= VarianceCI[data1]

Out[8]= {0.593815, 4.1831}

In[9]:= VarianceCI[data1, ConfidenceLevel -> 0.90]

Out[9]= {0.667653, 3.39718}
```

■ Confidence Intervals For Distributions

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

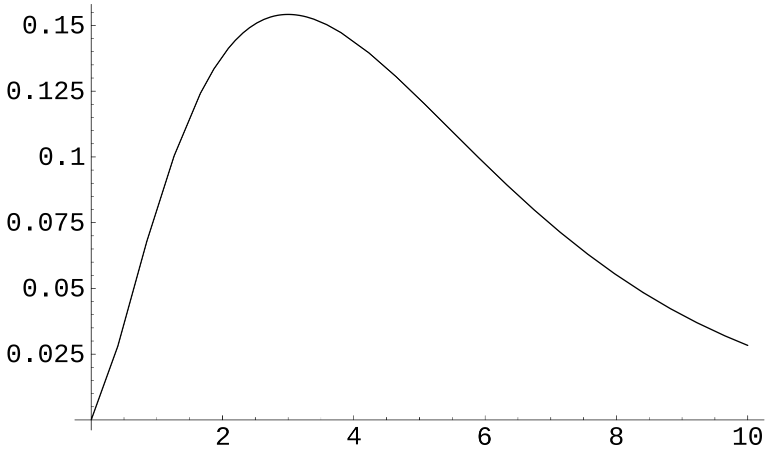
```
In[11]:= NormalCI[0, 1]

Out[11]= {-1.95996, 1.95996}

In[12]:= NormalCI[0, 1, ConfidenceLevel -> 0.90]

Out[12]= {-1.64485, 1.64485}

In[13]:= ?? 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

In[14]:= ChiSquareCI[20, 20, ConfidenceLevel -> 0.67]

Out[14]= {15.3701, 28.7551}

In[15]:= ChiSquareCI[20, 20, ConfidenceLevel -> 0.90]

Out[15]= {12.7346, 36.8636}

In[16]:= ChiSquareCI[20, 20]

Out[16]= {11.7063, 41.7067}
```

■ Normal Distribution Functions

```
In[18]:= << Statistics`NormalDistribution`

In[19]:= ? ChiSquareDistribution

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

In[20]:= ? PDF

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

In[21]:= chis = PDF[ChiSquareDistribution[5], x]
```

Out[21]= $\frac{E^{-x/2}\, x^{3/2}}{3\sqrt{2\pi}}$

```
In[22]:= Plot[chis, {x, 0, 10}]
```



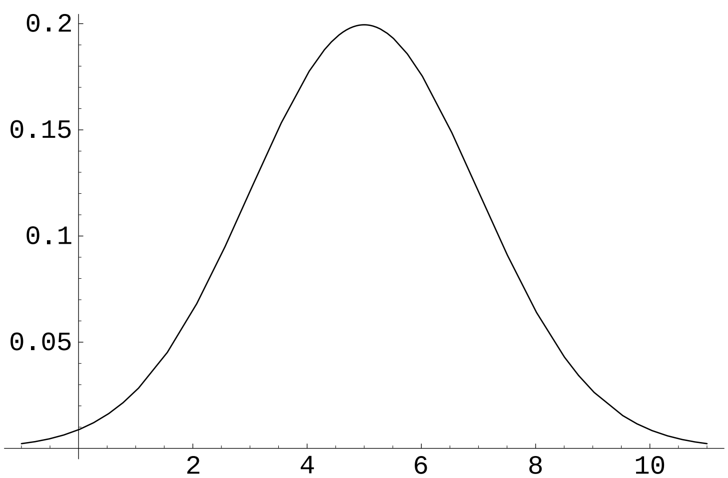

```
Out[22]= - Graphics -

In[23]:= ?NormalDistribution
```

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

```
In[25]:= ndf = PDF[NormalDistribution[5, 2], x]
```

Out[25]= $\frac{E^{-\frac{1}{8}(-5+x)^2}}{2\sqrt{2\pi}}$

```
In[26]:= Plot[ndf, {x, -1, 11}]
```

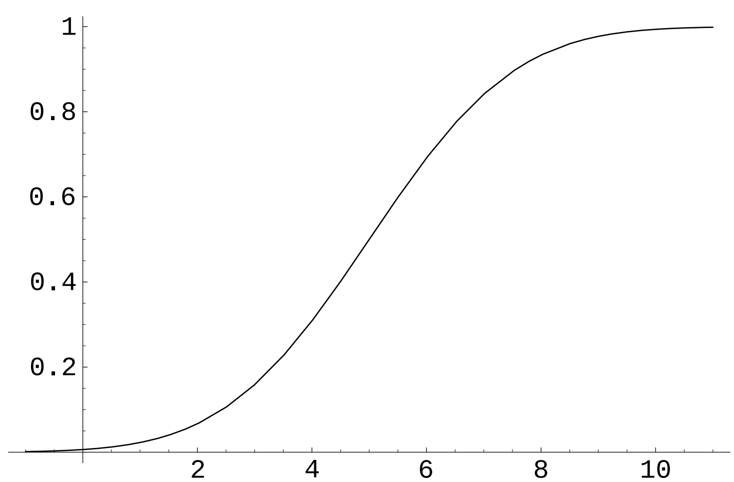



```
Out[26]= - Graphics -

In[27]:= Mean[NormalDistribution[5, 2]]

Out[27]= 5
```

```
In[36]:= Variance[NormalDistribution[5, 2]]

Out[36]= 4

In[28]:= StandardDeviation[NormalDistribution[5, 2]]

Out[28]= 2

In[29]:= ? 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.

In[30]:= Quantile[NormalDistribution[5, 2], .9]

Out[30]= 7.5631

In[31]:= Quantile[NormalDistribution[5, 2], .1]

Out[31]= 2.4369

In[32]:= ? 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.

In[33]:= ncdf = CDF[NormalDistribution[5, 2], x]
```

Out[33]= $\frac{1}{2}\left(1 + \mathrm{Erf}\left[\frac{-5 + x}{2\sqrt{2}}\right]\right)$

```
In[35]:= Plot[ncdf, {x, -1, 11}]
```




```
Out[35]= - Graphics -
```

In[38]:= **charndf = CharacteristicFunction[NormalDistribution[5, 2], k]**

Out[38]= $E^{5\,I\,k-2\,k^2}$

■ Discrete Distributions

In[39]:= **<< Statistics`DiscreteDistributions`**

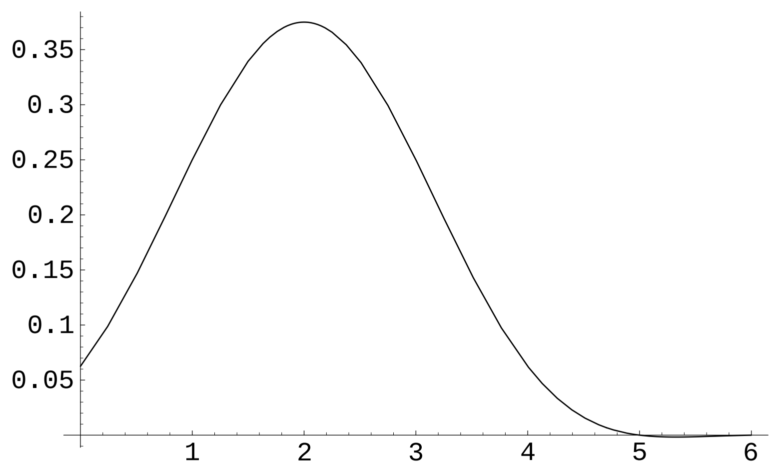
In[40]:= **?BinomialDistribution**

```
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.
```

In[46]:= **binompdf = PDF[BinomialDistribution[4, .5], m]**

Out[46]= 0.0625 Binomial[4, m]

In[48]:= **Plot[binompdf, {m, 0, 6}]**



Out[48]= - Graphics -

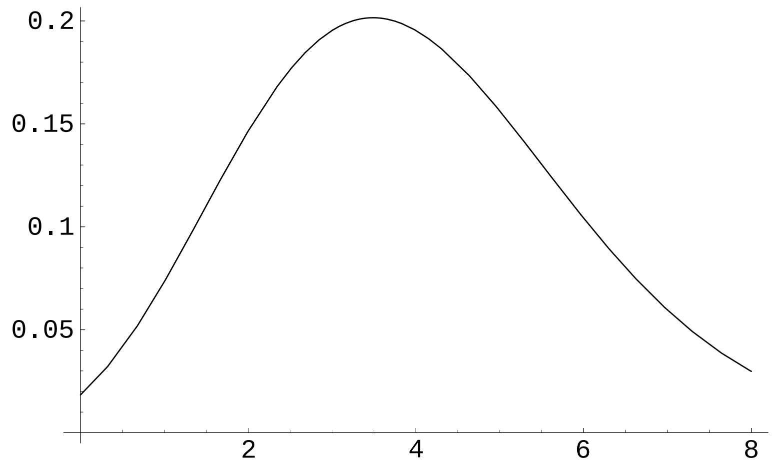
In[49]:= **?PoissonDistribution**

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

In[54]:= **poissonpdf = PDF[PoissonDistribution[4], m]**

Out[54]= $\frac{4^m}{E^4\, m!}$

In[55]:= **Plot[poissonpdf, {m, 0, 8}]**



Out[55]= - Graphics -

In[57]:= **N[Exp[-4]]**

Out[57]= 0.0183156

■ Nonlinear Function Fitting

In[60]:= **<< Statistics`NonlinearFit`**

In[61]:= **?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.

In[62]:= **data = Table[{n, 3 Cos[2 n]}, {n, 0., 1.5, .5}]**

Out[62]= {{0., 3.}, {0.5, 1.62091}, {1., -1.24844}, {1.5, -2.96998}}

In[63]:= **NonlinearFit[data, b Cos[a x], x, {a, b}]**

Out[63]= 2.76709 Cos[2.08175 x]

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