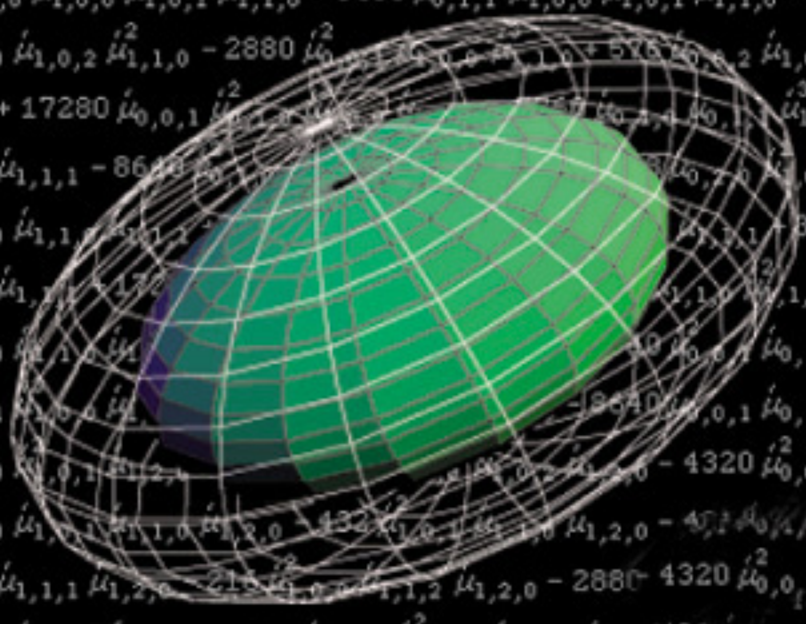
SPRINGER TEXTS IN STATISTICS

MATHEMATICAL STATISTICS *with* *Mathematica*®



COLIN ROSE
MURRAY D. SMITH

Mathematical Statistics with *Mathematica*

Chapter 1 – Introduction

1.1 Mathematical Statistics with *Mathematica* 1
- A A New Approach 1
- B Design Philosophy 1
- C If You Are New to *Mathematica* 2

1.2 Installation, Registration and Password 3
- A Installation, Registration and Password 3
- B Loading mathStatica 5
- C Help 5

1.3 Core Functions 6
- A Getting Started 6
- B Working with Parameters 8
- C Discrete Random Variables 9
- D Multivariate Random Variables 11
- E Piecewise Distributions 13

1.4 Some Specialised Functions 15

1.5 Notation and Conventions 24
- A Introduction 24
- B Statistics Notation 25
- C *Mathematica* Notation 27

Please reference this 2002 edition as:

Rose, C. and Smith, M.D. (2002)
Mathematical Statistics with Mathematica, Springer-Verlag, New York.

Latest edition

For the latest up-to-date edition, please visit: www.mathStatica.com

Chapter 1

Introduction

1.1 Mathematical Statistics with *Mathematica*

1.1 A A New Approach

The use of computer software in statistics is far from new. Indeed, hundreds of statistical computer programs exist. Yet, underlying existing programs is almost always a numerical / graphical view of the world. *Mathematica* can easily handle the numerical and graphical sides, but it offers in addition an extremely powerful and flexible symbolic computer algebra system. The **mathStatica** software package that accompanies this book builds upon that symbolic engine to create a sophisticated toolset specially designed for doing mathematical statistics.

While the subject matter of this text is similar to a traditional mathematical statistics text, this is not a traditional text. The reader will find few proofs and comparatively few theorems. After all, the theorem / proof text is already well served by many excellent volumes on mathematical statistics. Nor is this a cookbook of numerical recipes bundled into a computer package, for there is limited virtue in applying *Mathematica* as a mere numerical tool. Instead, this text strives to bring mathematical statistics to life. We hope it will make an exciting and substantial contribution to the way mathematical statistics is both practised and taught.

1.1 B Design Philosophy

mathStatica has been designed with two central goals: it sets out to be **general**, and it strives to be **delightfully simple**.

By **general**, we mean that it should *not* be limited to a set of special or well-known textbook distributions. It should *not* operate like a textbook appendix with prepared 'crib sheet' answers. Rather, it should know how to solve problems from first principles. It should seamlessly handle: univariate and multivariate distributions, continuous and discrete random variables, and smooth and kinked densities — all with and without parameters. It should be able to handle mixtures, truncated distributions, reflected distributions, folded distributions, and distributions of functions of random variables, as well as distributions no-one has ever thought of before.

By **delightfully simple**, we mean both (i) easy to use, and (ii) able to solve problems that seem difficult, but which are formally quite simple. Consider, for instance, playing a devilish game of chess against a strong chess computer: in the middle of the game, after a short pause, the computer announces, “Mate in 16 moves”. The problem it has solved might seem fantastically difficult, but it is really just a ‘delightfully simple’ finite problem that is conceptually no different than looking just two moves ahead. The salient point is that as soon as one has a tool for solving such problems, the notion of what is difficult changes completely. A pocket calculator is certainly a delightfully simple device: it is easy to use, and it can solve tricky problems that were previously thought to be difficult. But today, few people bother to ponder at the marvel of a calculator any more, and we now generally spend our time either using such tools or trying to solve higher-order conceptual problems — and so, we are certain, it will be with mathematical statistics too.

In fact, while much of the material traditionally studied in mathematical statistics courses may appear difficult, such material is often really just delightfully simple. Normally, all we want is an expectation, or a probability, or a transformation. But once we are armed with say a computerised expectation operator, we can find any kind of expectation including the mean, variance, skewness, kurtosis, mean deviation, moment generating function, characteristic function, raw moments, central moments, cumulants, probability generating function, factorial moment generating function, entropy, and so on. Normally, many of these calculations are not attempted in undergraduate texts, because the mechanics are deemed too hard. And yet, underlying all of them is just the delightfully simple expectation operator.

1.1 C If You Are New to *Mathematica*

For those readers who do not own a copy of *Mathematica*, this book comes bundled with a free trial copy of *Mathematica* Version 4. This will enable you to use **mathStatica**, and try out and evaluate all the examples in this book.

If you have never used *Mathematica* before, we recommend that you first read the opening pages of Wolfram (1999) and run through some examples. This will give you a good feel for *Mathematica*. Second, new users should learn how to enter formulae into *Mathematica*. This can be done via palettes, see

File Menu ▹ Palettes ▹ BasicInput,

or via the keyboard (see §1.5 below), or just by copy and pasting examples from this book. Third, both new and experienced readers may benefit from browsing Appendices A.1 to A.7 of this book, which cover a plethora of tips and tricks.

Before proceeding further, please ensure that *Mathematica* Version 4 (or later) is installed on your computer.

1.2 Installation, Registration and Password

1.2 A Installation, Registration and Password

Before starting, please make sure you have a working copy of *Mathematica* Version 4 (or later) installed on your computer.

Installing **mathStatica** is an easy 4-step process, irrespective of whether you use a Macintosh, Windows, or a flavour of UNIX.

Step 1: Insert the **mathStatica** CD-ROM into your computer.

Step 2: Copy the following files:

(i) `mathStatica.m` (file)
(ii) `mathStatica` (folder / directory)

from the **mathStatica** CD-ROM *into* the

`Mathematica ▷ AddOns ▷ Applications`

folder on your computer's hard drive. The installation should look something like Fig. 1.

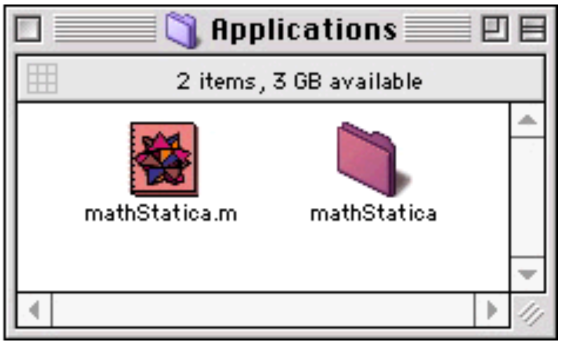


Fig. 1: Typical installation of **mathStatica**

Step 3: Get a password

To use **mathStatica**, you will need a password. To get a password, you will need to register your copy of **mathStatica** at the following web site:

`www.mathstatica.com`

mathStatica is available in two versions: Basic and Gold. The differences are summarised in Table 1; for full details, see the web site.

class	*description*
Basic	• Fully functional **mathStatica** package code
	• Included on the CD-ROM
	• FREE to buyers of this book
	• Single-use license
Gold	• All the benefits of Basic, *plus* …
	• *Continuous* and *Discrete* Distribution Palettes
	• Detailed interactive HELP system
	• Upgrades
	• Technical support
	• and more …

Table 1: mathStatica — Basic and Gold

Once you have registered your copy, you will be sent a password file called: `pass.txt`. Put this file into the `Mathematica` ▷ `AddOns` ▷ `Applications` ▷ `mathStatica` ▷ `Password` directory, as shown in Fig. 2.

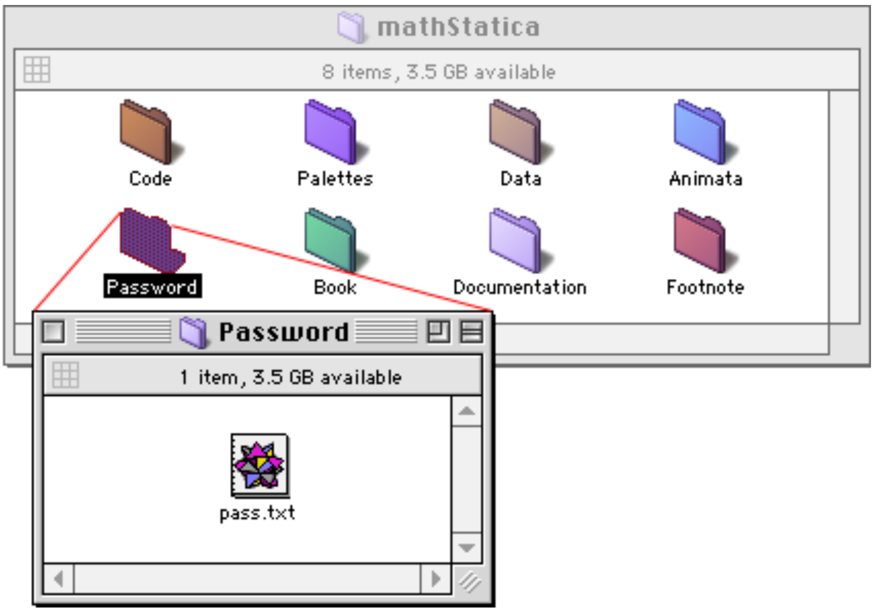


Fig. 2: Once you have received `"pass.txt"`, put it into the `Password` folder

Step 4: Run *Mathematica*, go to its HELP menu, and select: "Rebuild Help Index"

That's it — all done. Your installation is now complete.

1.2 B Loading mathStatica

If everything is installed correctly, first start up *Mathematica* Version 4 or later. Then **mathStatica** can be loaded by evaluating:

```
<< mathStatica.m
```

or by clicking on a button such as this one: **Start mathStatica**

The **Book** palette should then appear, as shown in Fig. 3 (right panel). The **Book** palette provides a quick and easy way to access the electronic version of this book, including the live hyperlinked index. If you have purchased the Gold version of **mathStatica**, then the **mathStatica** palette will also appear, as shown in Fig. 3 (left panel). This provides the **Continuous** and **Discrete** distribution palettes (covering 37 distributions), as well as the detailed **mathStatica Help** system (complete with hundreds of extra examples).

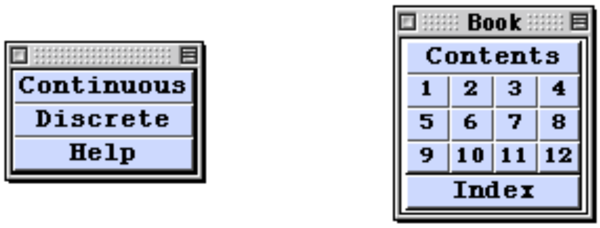


Fig. 3: The **mathStatica** palette (left) and the **Book** palette (right)

WARNING: To avoid so-called 'context' problems, **mathStatica** should always be loaded from a fresh *Mathematica* kernel. If you have already done some calculations in *Mathematica*, you can get a fresh kernel by either typing `Quit` in an Input cell, or by selecting Kernel Menu ▹ Quit Kernel.

1.2 C Help

Both Basic Help and Detailed Help are available for any **mathStatica** function:

(i) Basic Help is shown in Table 2.

function	*description*
? Name	show information on Name

Table 2: Basic Help on function names

For example, to get Basic Help on the **mathStatica** function `CentralToRaw`, enter:

```
? CentralToRaw
```

CentralToRaw[r] expresses the rth central moment μ_r in terms of raw moments $\acute{\mu}_i$. To obtain a multivariate conversion, let r be a list of integers.

(ii) Detailed Help (Gold version only) is available via the **mathStatica** palette (Fig. 3).

1.3 Core Functions

1.3 A Getting Started

mathStatica adds about 100 new functions to *Mathematica*. But most of the time, we can get by with just four of them:

function	*description*	
`PlotDensity[f]`	Plotting (automated)	
`Expect[x, f]`	Expectation operator	$E[X]$
`Prob[x, f]`	Probability	$P(X \leq x)$
`Transform[eqn, f]`	Transformations	

Table 3: Core functions for a random variable X with density $f(x)$

This ability to handle plotting, expectations, probability, and transformations, with just four functions, makes the **mathStatica** system very easy to use, even for those not familiar with *Mathematica*.

To illustrate, let us suppose the continuous random variable X has probability density function (pdf)

$$f(x) = \frac{1}{\pi \sqrt{1-x} \sqrt{x}}, \quad \text{for } x \in (0, 1).$$

In *Mathematica*, we enter this as:

```
f = 1/(π Sqrt[1 - x] Sqrt[x]);        domain[f] = {x, 0, 1};
```

This is known as the Arc–Sine distribution. Here is a plot of $f(x)$:

```
PlotDensity[f];
```

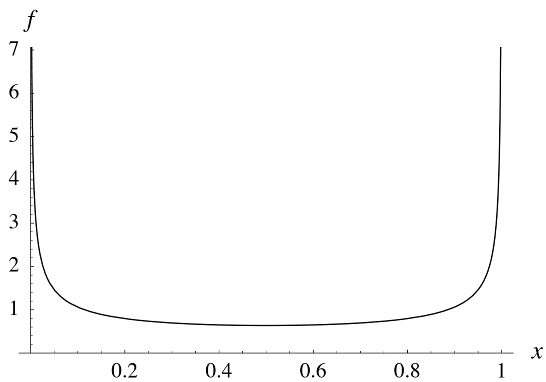


Fig. 4: The Arc–Sine pdf

Here is the cumulative distribution function (cdf), $P(X \leq x)$, which also provides the clue to the naming of this distribution:

```
Prob[x, f]
```

$$\frac{2\,\text{ArcSin}\left[\sqrt{x}\right]}{\pi}$$

The mean, $E[X]$, is:

```
Expect[x, f]
```

$$\frac{1}{2}$$

while the variance of X is:

```
Var[x, f]
```

$$\frac{1}{8}$$

The r^{th} moment of X is $E[X^r]$:

```
Expect[x^r, f]
```

— This further assumes that: $\{r > -\frac{1}{2}\}$

$$\frac{\Gamma[\frac{1}{2} + r]}{\sqrt{\pi}\,\Gamma[1 + r]}$$

Now consider the transformation to a new random variable Y such that $Y = \sqrt{X}$. By using the `Transform` and `TransformExtremum` functions, the pdf of Y, say $g(y)$, and the domain of its support can be found:

```
g = Transform[y == √x, f]
```

$$\frac{2\,y}{\pi\sqrt{y^2 - y^4}}$$

```
domain[g] = TransformExtremum[y == √x, f]
```

{y, 0, 1}

So, we have started out with a quite arbitrary pdf $f(x)$, transformed it to a new one $g(y)$, and since both density g and its `domain` have been entered into *Mathematica*, we can also apply the **mathStatica** tool set to density g. For example, use `PlotDensity[g]` to plot the pdf of $Y = \sqrt{X}$.

1.3 B Working with Parameters (Assumptions technology ☼)

mathStatica has been designed to seamlessly support parameters. It does so by taking full advantage of the new *Assumptions technology* introduced in Version 4 of *Mathematica,* which enables us to make assumptions about parameters. To illustrate, let us consider the familiar Normal distribution with mean μ and variance σ^2. That is, let $X \sim N(\mu, \sigma^2)$, where $\mu \in \mathbb{R}$ and $\sigma > 0$. We enter the pdf $f(x)$ in the standard way, but this time we have some extra information about the parameters μ and σ. We use the `And` function, `&&`, to add these assumptions to the end of the `domain[f]` statement:

$$f = \frac{1}{\sigma\sqrt{2\pi}} \text{Exp}\left[-\frac{(x-\mu)^2}{2\sigma^2}\right];$$

```
domain[f] = {x, -∞, ∞} && {μ ∈ Reals, σ > 0};
```

From now on, the assumptions about μ and σ will be 'attached' to density f, so that whenever we operate on density f with a **mathStatica** function, these assumptions will be applied automatically in the background. With this new technology, **mathStatica** can usually produce remarkably crisp textbook-style answers, even when working with very complicated distributions.

The **mathStatica** function, `PlotDensity`, makes it easy to examine the effect of changing parameter values. The following input reads: "Plot density $f(x)$ when μ is 0, and σ is 1, 2 and 3". For more detail on using the `/.` operator, see Wolfram (1999, Section 2.4.1).

```
PlotDensity[f /. {μ → 0, σ → {1, 2, 3}}];
```

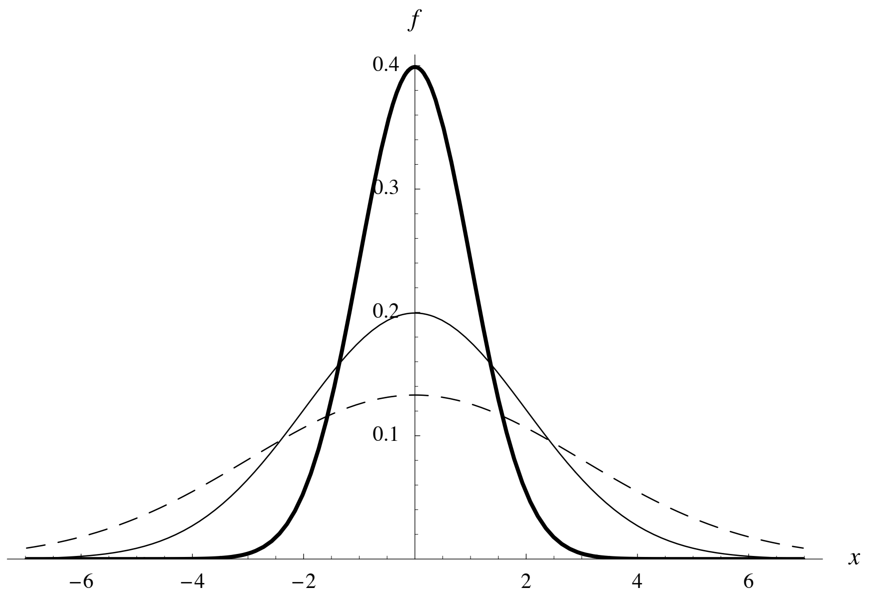


Fig. 5: The pdf of a Normal random variable, when $\mu = 0$ and $\sigma = 1$(**——**), 2(—), 3(– – –)

It is well known that $E[X] = \mu$ and $\mathrm{Var}(X) = \sigma^2$, as we can easily verify:

```
Expect[x, f]
```

μ

```
Var[x, f]
```

σ^2

Because **mathStatica** is general in its design, we can just as easily solve problems that are both less well-known and more 'difficult', such as finding $\mathrm{Var}(X^2)$:

Var[x², f]

$2\ (2\ \mu^2\ \sigma^2 + \sigma^4)$

Assumptions technology is a very important addition to *Mathematica*. In order for it to work, one should enter as much information about parameters as possible. The resulting answer will be much neater, it may also be obtained faster, and it may make it possible to solve problems that could not otherwise be solved. Here is an example of some Assumptions statements:

{α > 1, β ∈ Integers, -∞ < γ < π, δ ∈ Reals, θ > 0}

mathStatica implements Assumptions technology in a *distribution*-specific manner. This means the assumptions are attached to the density $f(x;\theta)$ and not to the parameter θ. What if we have two distributions, both using the same parameter? No problem … suppose the two pdf's are

(i) $f(x;\theta)$ $\theta > 0$

(ii) $g(x;\theta)$ $\theta < 0$

Then, when we work with density f, **mathStatica** will assume $\theta > 0$; when we work with density g, it will assume $\theta < 0$. For example,

(i) `Expect[`x, f`]` will assume $\theta > 0$

(ii) `Prob[`x, g`]` will assume $\theta < 0$

It is important to realise that the assumptions will only be automatically invoked when using the suite of **mathStatica** functions. By contrast, *Mathematica*'s built-in functions, such as the derivative function, `D[`f, x`]`, will not automatically assume that $\theta > 0$.

1.3 C Discrete Random Variables

mathStatica automatically handles discrete random variables in the same way. The only difference is that, when we define the density, we add a flag to tell *Mathematica* that the random variable is `{Discrete}`. To illustrate, let the discrete random variable X have probability mass function (pmf)

$$f(x) = P(X = x) = \binom{r+x-1}{x} p^r (1-p)^x, \qquad \text{for } x \in \{0, 1, 2, \ldots\}.$$

Here, parameter p is the probability of success, while parameter r is a positive integer. In *Mathematica*, we enter this as:

$$f = \text{Binomial}[r + x - 1, x]\, p^r\, (1 - p)^x;$$

```
domain[f] = {x, 0, ∞} && {Discrete} &&
                    {0 < p < 1, r > 0, r ∈ Integers} ;
```

This is known as the Pascal distribution. Here is a plot of $f(x)$:

$$\text{PlotDensity}\left[f \,/.\, \left\{p \to \frac{1}{2},\ r \to 10\right\}\right];$$

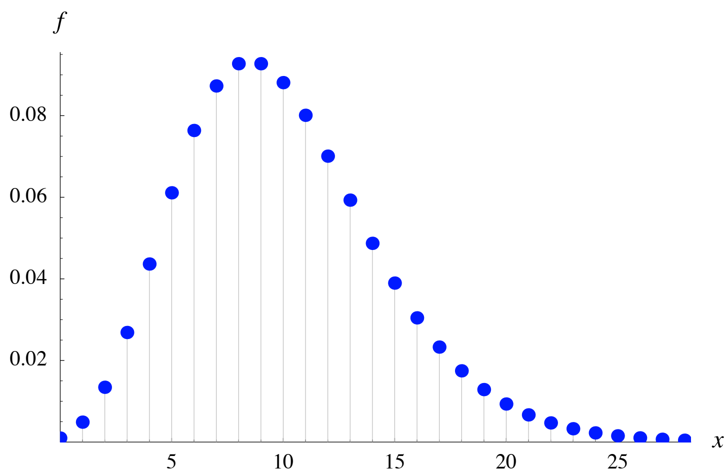


Fig. 6: The pmf of a Pascal discrete random variable

Here is the cdf, equal to $P(X \le x)$:

```
Prob[x, f]
```

$$1 - \frac{1}{\Gamma[r]\ \Gamma[2 + \text{Floor}[x]]} \left((1-p)^{1+\text{Floor}[x]}\ p^r\ \Gamma[1 + r + \text{Floor}[x]]\ \text{Hypergeometric2F1}[1,\ 1 + r + \text{Floor}[x],\ 2 + \text{Floor}[x],\ 1 - p]\right)$$

The mean $E[X]$ and variance of X are given by:

```
Expect[x, f]
```

$$\left(-1 + \frac{1}{p}\right) r$$

```
Var[x, f]
```

$$\frac{r - p\, r}{p^2}$$

The probability generating function (pgf) is $E[t^X]$:

```
Expect[t^x, f]
```

```
p^r (1 + (-1 + p) t)^-r
```

For more detail on discrete random variables, see Chapter 3.

1.3 D Multivariate Random Variables

mathStatica extends naturally to a multivariate setting. To illustrate, let us suppose that X and Y have joint pdf $f(x, y)$ with support $x > 0$, $y > 0$:

```
f = ⅇ^(-2 (x+y)) (ⅇ^(x+y) + α (ⅇ^x - 2) (ⅇ^y - 2));
domain[f] = {{x, 0, ∞}, {y, 0, ∞}} && {-1 < α < 1};
```

where parameter α is such that $-1 < \alpha < 1$. This is known as a Gumbel bivariate Exponential distribution. Here is a plot of $f(x, y)$. To display the code that generates this plot, simply click on the ▹ adjacent to Fig. 7 in the electronic version of this chapter. Clicking the 'View Animation' button in the electronic notebook brings up an animation of $f(x, y)$, allowing parameter α to vary from -1 to 0 in step sizes of $1/20$. This provides a rather neat way to visualise how the shape of the joint pdf changes with α. In the printed text, the symbol 🎬 indicates that an animation is available.

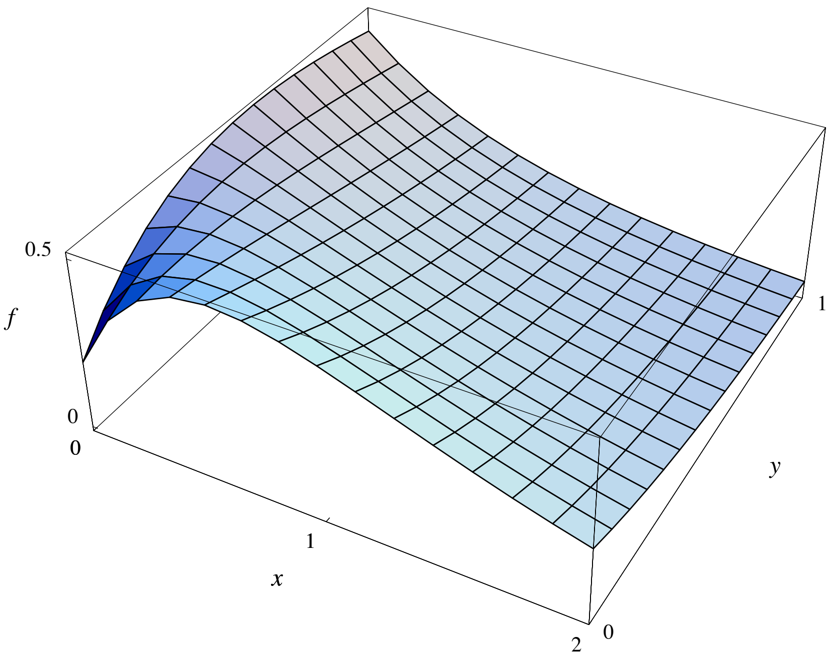


Fig. 7: A Gumbel bivariate Exponential pdf when $\alpha = -0.8$ 🎬

Here is the cdf, namely $P(X \leq x, Y \leq y)$:

```
Prob[{x, y}, f]
```

$e^{-2(x+y)} (-1 + e^x) (-1 + e^y) (e^{x+y} + \alpha)$

Here is Cov(X, Y), the covariance between X and Y:

```
Cov[{x, y}, f]
```

$\frac{\alpha}{4}$

More generally, here is the variance-covariance matrix:

```
Varcov[f]
```

$\begin{pmatrix} 1 & \frac{\alpha}{4} \\ \frac{\alpha}{4} & 1 \end{pmatrix}$

Here is the marginal pdf of X:

```
Marginal[x, f]
```

e^{-x}

Here is the conditional pdf of Y, given $X = x$:

```
Conditional[y, f]
```

– Here is the conditional pdf $f(y \mid x)$:

$e^{x-2(x+y)} (e^{x+y} + (-2 + e^x) (-2 + e^y) \alpha)$

Here is the bivariate mgf $E[e^{t_1 X + t_2 Y}]$:

```
mgf = Expect[ e^(t₁ x + t₂ y), f]
```

– This further assumes that: $\{t_1 < 1, t_2 < 1\}$

$$\frac{4 - 2 t_2 + t_1 (-2 + (1 + \alpha) t_2)}{(-2 + t_1) (-1 + t_1) (-2 + t_2) (-1 + t_2)}$$

Differentiating the mgf is one way to derive moments. Here is the product moment $E[X^2 Y^2]$:

```
D[mgf, {t₁, 2}, {t₂, 2}] /. t_ → 0 // Simplify
```

$4 + \frac{9\alpha}{4}$

which we could otherwise have found directly with:

```
Expect[x² y², f]
```

$4 + \frac{9\,\alpha}{4}$

Multivariate transformations pose no problem to **mathStatica** either. For instance, let $U = \frac{Y}{1+X}$ and $V = \frac{1}{1+X}$ denote transformations of X and Y. Then our transformation equation is:

```
eqn = {u == y/(1 + x), v == 1/(1 + x)};
```

Using `Transform`, we can find the joint pdf of random variables U and V, denoted $g(u, v)$:

```
g = Transform[eqn, f]
```

$$\frac{e^{\frac{-2-2u+v}{v}}\left(4\,e\,\alpha - 2\,e^{\frac{1}{v}}\,\alpha - 2\,e^{\frac{u+v}{v}}\,\alpha + e^{\frac{1+u}{v}}\,(1+\alpha)\right)}{v^3}$$

while the extremum of the domain of support of the new random variables are:

```
TransformExtremum[eqn, f]
```

{{u, 0, ∞}, {v, 0, 1}}

For more detail on multivariate random variables, see Chapter 6.

1.3 E Piecewise Distributions

Some density functions take a bipartite form. To illustrate, let us suppose X is a continuous random variable, $0 < x < 1$, with pdf

$$f(x) = \begin{cases} 2\left(\frac{c-x}{c}\right) & \text{if } x < c \\ 2\left(\frac{x-c}{1-c}\right) & \text{if } x \geq c \end{cases}$$

where $0 < c < 1$. We enter this as:

```
f = If[x < c , 2 (c - x)/c, 2 (x - c)/(1 - c)];

domain[f] = {x, 0, 1} && {0 < c < 1};
```

This is known as the Inverse Triangular distribution, as is clear from a plot of $f(x)$, as illustrated in Fig. 8.

```
PlotDensity[f /. c → {1/4, 1/2, 3/4}];
```

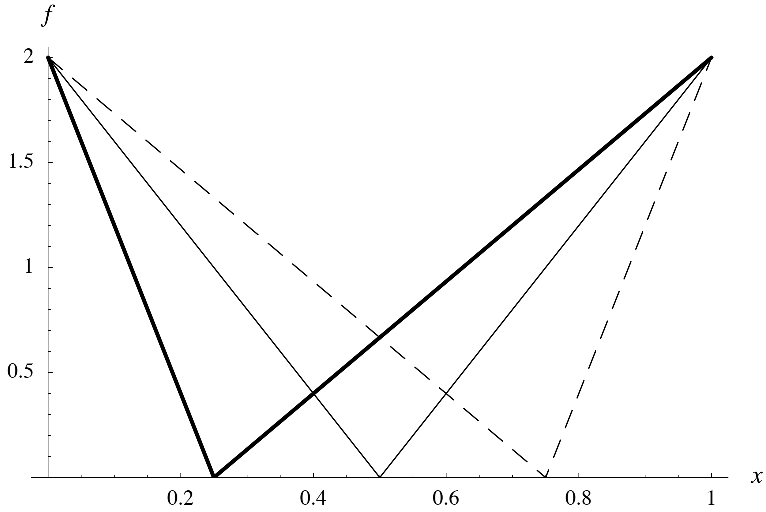


Fig. 8: The Inverse Triangular pdf, when $c = \frac{1}{4}$(——), $\frac{1}{2}$(—), $\frac{3}{4}$(– – –)

Here is the cdf, $P(X \le x)$:

```
Prob[x, f]
```

$$\text{If}\left[x < c,\ x\left(2 - \frac{x}{c}\right),\ \frac{c - 2\,c\,x + x^2}{1 - c}\right]$$

Note that the solution depends on whether $x < c$ or $x \ge c$. Figure 9 plots the cdf at the same three values of c used in Fig. 8.

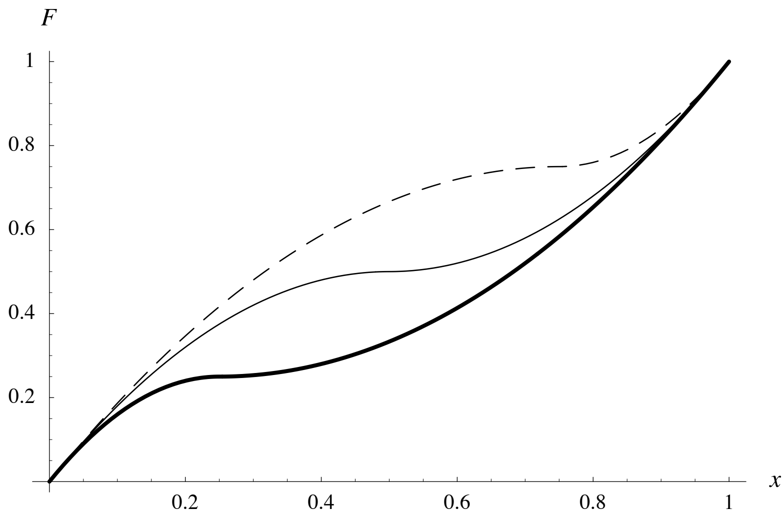


Fig. 9: The Inverse Triangular cdf, when $c = \frac{1}{4}$(——), $\frac{1}{2}$(—), $\frac{3}{4}$(– – –)

mathStatica operates on bipartite distributions in the standard way. For instance, the mean $E[X]$ is given by:

```
Expect[x, f]
```

$$\frac{2-c}{3}$$

while the entropy is given by $E[-\log(f(X))]$:

```
Expect[-Log[f], f]
```

$$\frac{1}{2} - \text{Log}[2]$$

1.4 Some Specialised Functions

⊕ ***Example 1:*** Moment Conversion Functions

mathStatica allows one to express any moment (raw $\acute{\mu}$, central μ, or cumulant κ) in terms of any other moment ($\acute{\mu}$, μ, or κ). For instance, to express the second central moment (the variance) $\mu_2 = E\left[(X - E[X])^2\right]$ in terms of raw moments, we enter:

```
CentralToRaw[2]
```

$$\mu_2 \to -\acute{\mu}_1^2 + \acute{\mu}_2$$

This is just the well-known result that $\mu_2 = E[X^2] - (E[X])^2$. As a further example, here is the sixth cumulant expressed in terms of raw moments:

```
CumulantToRaw[6]
```

$$\kappa_6 \to -120\,\acute{\mu}_1^6 + 360\,\acute{\mu}_1^4\,\acute{\mu}_2 - 270\,\acute{\mu}_1^2\,\acute{\mu}_2^2 + 30\,\acute{\mu}_2^3 - 120\,\acute{\mu}_1^3\,\acute{\mu}_3 + 120\,\acute{\mu}_1\,\acute{\mu}_2\,\acute{\mu}_3 - 10\,\acute{\mu}_3^2 + 30\,\acute{\mu}_1^2\,\acute{\mu}_4 - 15\,\acute{\mu}_2\,\acute{\mu}_4 - 6\,\acute{\mu}_1\,\acute{\mu}_5 + \acute{\mu}_6$$

The moment converter functions are completely general, and extend in the natural manner to a multivariate framework. Here is the bivariate central moment $\mu_{2,3}$ expressed in terms of bivariate cumulants:

```
CentralToCumulant[{2, 3}]
```

$$\mu_{2,3} \to 6\,\kappa_{1,1}\,\kappa_{1,2} + \kappa_{0,3}\,\kappa_{2,0} + 3\,\kappa_{0,2}\,\kappa_{2,1} + \kappa_{2,3}$$

For more detail, see Chapter 2 (univariate) and Chapter 6 (multivariate). ■

⊕ ***Example 2:*** Pseudo-Random Number Generation

Let X be *any* discrete random variable with probability mass function (pmf) $f(x)$. Then, the **mathStatica** function `DiscreteRNG[n, f]` generates n pseudo-random copies of X. To illustrate, let us suppose $X \sim \text{Poisson}(6)$:

```
f = (ⅇ^-λ λ^x)/(x!) /. λ → 6;   domain[f] = {x, 0, ∞} && {Discrete};
```

As usual, `domain[f]` must *always* be entered along with `f`, as it passes important information onto `DiscreteRNG`. Here are 30 copies of X:

```
DiscreteRNG[30, f]
```

```
{10, 4, 8, 3, 5, 6, 3, 2, 9, 6, 3, 5, 6, 5,
 5, 4, 3, 5, 3, 8, 2, 3, 6, 5, 3, 10, 8, 5, 8, 5}
```

Here, in a fraction of a second, are 50000 more copies of X:

```
data = DiscreteRNG[50000, f]; // Timing
```

```
{0.39 Second, Null}
```

`DiscreteRNG` is not only completely general, but it is also very efficient. We now contrast the empirical distribution of `data` with the true distribution of X:

```
FrequencyPlotDiscrete[data, f];
```

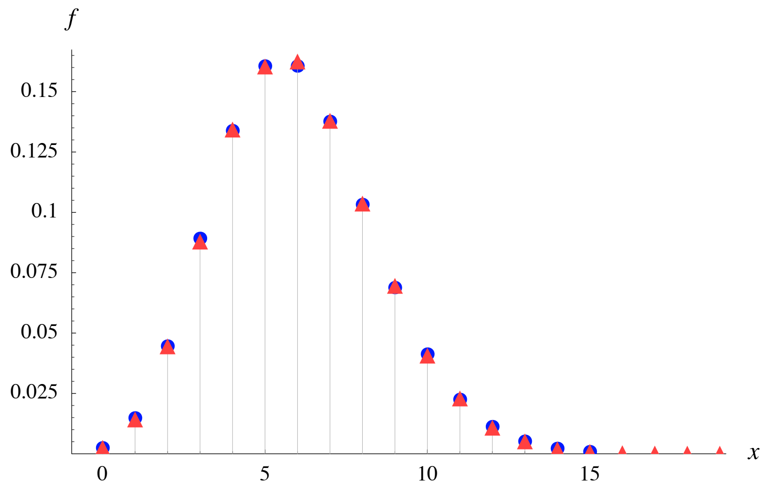


Fig. 10: The empirical pmf (▲) and true pmf (●)

The triangular dots denote the empirical pmf, while the round dots denote the true density $f(x)$. One obtains a superb fit because `DiscreteRNG` is an exact solution. This may make it difficult to distinguish the triangles from the round dots. For more detail, see Chapter 3. ■

⊕ ***Example 3:*** Pearson Fitting

Karl Pearson showed that if we know the first four moments of a distribution, we can construct a density function that is consistent with those moments. This can provide a neat way to build density functions that approximate a given set of data. For instance, for a given data set, let us suppose that:

```
mean = 37.875;
μ̂234 = {191.55, 1888.36, 107703.3};
```

denoting estimates of the mean, and of the second, third and fourth central moments. The Pearson family consists of 7 main *Types*, so our first task is to find out which type this data is consistent with. We do this with the `PearsonPlot` function:

```
PearsonPlot[μ̂234];
```

```
{β1 → 0.507368, β2 → 2.93538}
```

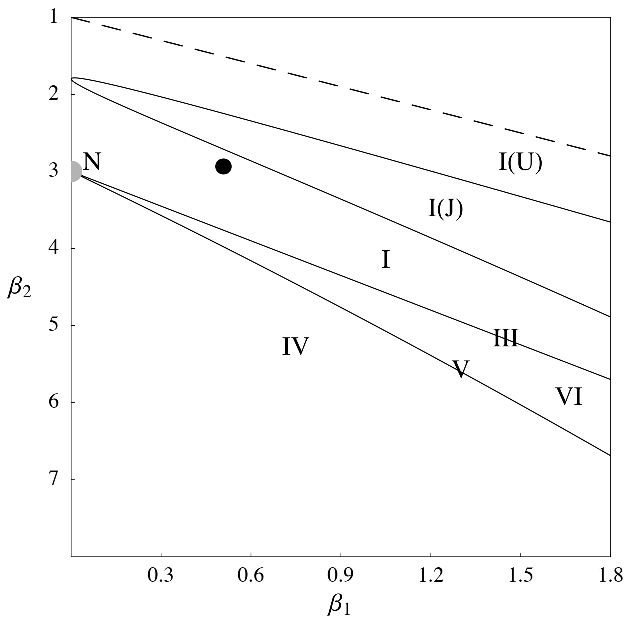


Fig. 11: The β_1, β_2 chart for the Pearson system

The big black dot in Fig. 11 is in the *Type I* zone. Then, the fitted Pearson density $f(x)$ and its domain are immediately given by:

```
{f, domain[f]} = PearsonI[mean, μ̂234, x]
```

```
{9.62522 × 10^-8 (94.3127 - 1. x)^2.7813
  (-16.8709 + 1. x)^0.407265, {x, 16.8709, 94.3127}}
```

The actual data used to create this example is grouped data (see *Example 3* of Chapter 5) depicting the number of sick people (`freq`) at different ages (`X`):

```
X = {17, 22, 27, 32, 37, 42, 47, 52, 57, 62, 67, 72, 77, 82, 87};
freq = {34, 145, 156, 145, 123, 103, 86, 71, 55, 37, 21, 13, 7, 3, 1};
```

We can easily compare the histogram of the empirical data with our fitted Pearson pdf:

```
FrequencyGroupPlot[{X, freq}, f];
```

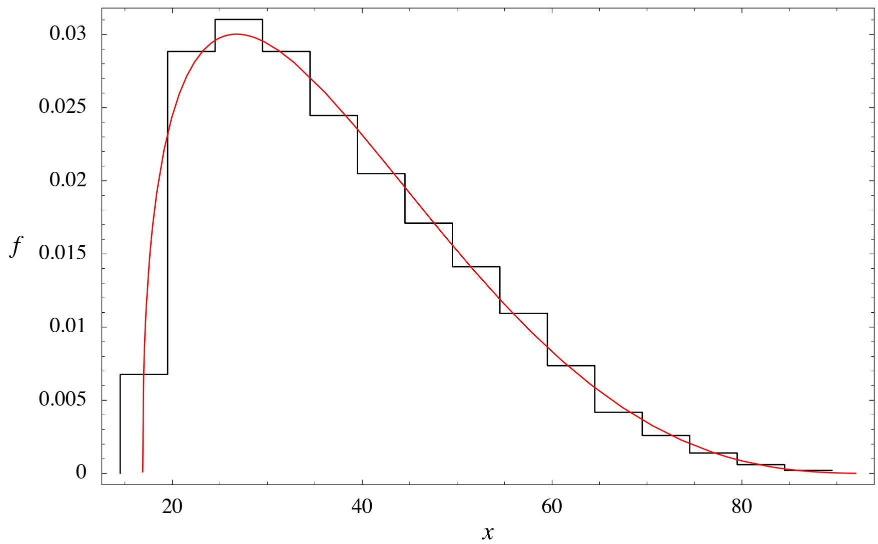


Fig. 12: The data histogram and the fitted Pearson pdf

Related topics include Gram–Charlier expansions, and the Johnson family of distributions. For more detail, see Chapter 5. ■

⊕ ***Example 4:*** Fisher Information

The Fisher Information on a parameter can be constructed from first principles using the `Expect` function. Alternatively, we can use **mathStatica**'s `FisherInformation` function, which automates this calculation. To illustrate, let $X \sim \text{InverseGaussian}(\mu, \lambda)$ with pdf $f(x)$:

$$\textbf{f} = \sqrt{\frac{\lambda}{2\,\pi\,\mathbf{x}^3}}\ \textbf{Exp}\left[-\lambda\ \frac{(\mathbf{x}-\mu)^2}{2\,\mu^2\,\mathbf{x}}\right];$$

```
domain[f] = {x, 0, ∞} && {μ > 0, λ > 0};
```

Then, Fisher's Information on (μ, λ) is the (2×2) matrix:

```
FisherInformation[{μ, λ}, f]
```

$$\begin{pmatrix} \frac{\lambda}{\mu^3} & 0 \\ 0 & \frac{1}{2\,\lambda^2} \end{pmatrix}$$

For more detail on Fisher Information, see Chapter 10. ■

⊕ ***Example 5:*** Non-Parametric Kernel Density Estimation

Here is some raw data measuring the diagonal length of 100 forged Swiss bank notes and 100 real Swiss bank notes (Simonoff, 1996):

```
data = ReadList["sd.dat"];
```

Non-parametric kernel density estimation involves two components: (i) the choice of a kernel, and (ii) the selection of a bandwidth. Here we use a Gaussian kernel f:

$$\mathbf{f} = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}; \qquad \mathbf{domain[f]} = \{x, -\infty, \infty\};$$

Next, we select the bandwidth c. Small values for c produce a rough estimate while large values produce a very smooth estimate. A number of methods exist to automate bandwidth choice; **mathStatica** implements both the Silverman (1986) approach and the more sophisticated Sheather and Jones (1991) method. For the Swiss bank note data set, the Sheather–Jones optimal bandwidth (using the Gaussian kernel f) is:

```
c = Bandwidth[data, f, Method → SheatherJones]
```

```
0.200059
```

We can now plot the smoothed non-parametric kernel density estimate using the `NPKDEPlot[`*data*, *kernel*, *c*`]` function:

```
NPKDEPlot[data, f, c];
```

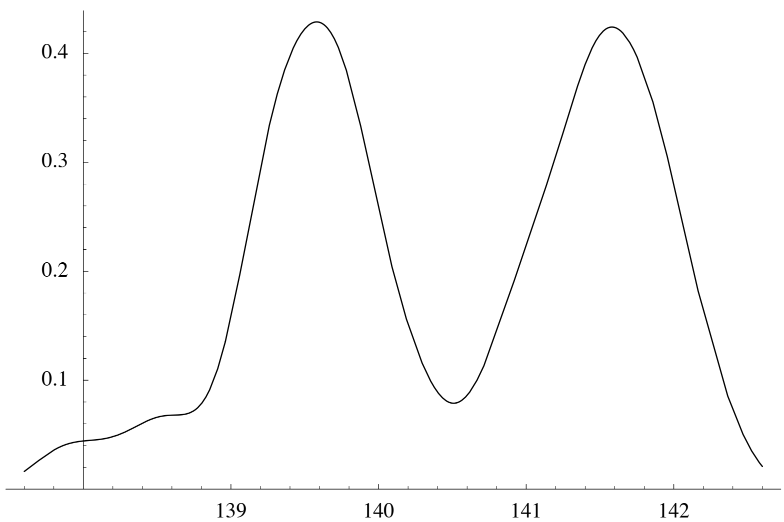


Fig. 13: The smoothed non-parametric kernel density estimate (Swiss bank notes)

For more detail, see Chapter 5. ■

⊕ ***Example 6:*** Unbiased Estimation of Population Moments; Moments of Moments

mathStatica can find unbiased estimators of population moments. For instance, it offers h-statistics (unbiased estimators of population central moments), k-statistics (unbiased estimators of population cumulants), multivariate varieties of the same, polykays (unbiased estimators of products of cumulants) and more. Consider the k-statistic k_r which is an unbiased estimator of the r^{th} cumulant κ_r; that is, $E[k_r] = \kappa_r$, for $r = 1, 2, \ldots$. Here are the 2^{nd} are 3^{rd} k-statistics:

```
k2 = KStatistic[2]
k3 = KStatistic[3]
```

$$k_2 \to \frac{-s_1^2 + n\, s_2}{(-1+n)\, n}$$

$$k_3 \to \frac{2\, s_1^3 - 3\, n\, s_1\, s_2 + n^2\, s_3}{(-2+n)\,(-1+n)\, n}$$

As per convention, the solution is expressed in terms of power sums $s_r = \sum_{i=1}^{n} X_i^r$.

Moments of moments: Because the above expressions (sample moments) are functions of random variables X_i, we might want to calculate population moments of them. With **mathStatica**, we can find any moment (raw, central, or cumulant) of the above expressions. For instance, k_3 is meant to have the property that $E[k_3] = \kappa_3$. We test this by calculating the first raw moment of k_3, and express the answer in terms of cumulants:

```
RawMomentToCumulant[1, k3⟦2⟧]
```

$$\kappa_3$$

In 1928, Fisher published the product cumulants of the k-statistics, which are now listed in reference bibles such as Stuart and Ord (1994). Here is the solution to $\kappa_{2,2}(k_3, k_2)$:

```
CumulantMomentToCumulant[{2, 2}, {k3⟦2⟧, k2⟦2⟧}]
```

$$\frac{288\, n\, \kappa_2^5}{(-2+n)\,(-1+n)^3} + \frac{288\,(-23+10\, n)\, \kappa_2^2\, \kappa_3^2}{(-2+n)\,(-1+n)^3} + \frac{360\,(-7+4\, n)\, \kappa_2^3\, \kappa_4}{(-2+n)\,(-1+n)^3} +$$
$$\frac{36\,(160-155\, n+38\, n^2)\, \kappa_3^2\, \kappa_4}{(-2+n)\,(-1+n)^3\, n} + \frac{36\,(93-103\, n+29\, n^2)\, \kappa_2\, \kappa_4^2}{(-2+n)\,(-1+n)^3\, n} +$$
$$\frac{24\,(202-246\, n+71\, n^2)\, \kappa_2\, \kappa_3\, \kappa_5}{(-2+n)\,(-1+n)^3\, n} + \frac{2\,(113-154\, n+59\, n^2)\, \kappa_5^2}{(-1+n)^3\, n^2} +$$
$$\frac{6\,(-131+67\, n)\, \kappa_2^2\, \kappa_6}{(-2+n)\,(-1+n)^2\, n} + \frac{3\,(117-166\, n+61\, n^2)\, \kappa_4\, \kappa_6}{(-1+n)^3\, n^2} +$$
$$\frac{6\,(-27+17\, n)\, \kappa_3\, \kappa_7}{(-1+n)^2\, n^2} + \frac{37\, \kappa_2\, \kappa_8}{(-1+n)\, n^2} + \frac{\kappa_{10}}{n^3}$$

This is the correct solution. Unfortunately, the solutions given in Stuart and Ord (1994, equation (12.70)) and Fisher (1928) are actually incorrect (see *Example 14* of Chapter 7). ■

⊕ ***Example 7:*** Symbolic Maximum Likelihood Estimation

Although statistical software has long been used for maximum likelihood (ML) estimation, the focus of attention has almost always been on obtaining ML estimates (a *numerical* problem), rather than on deriving ML estimators (a *symbolic* problem). **mathStatica** makes it possible to derive *exact* symbolic ML estimators from first principles with a computer algebra system.

For instance, consider the following simple problem: let $(X_1, \ldots, X_n)$ denote a random sample of size n collected on $X \sim \text{Rayleigh}(\sigma)$, where parameter $\sigma > 0$ is unknown. We wish to find the ML estimator of σ. We begin in the usual way by inputting the likelihood function into *Mathematica*:

$$\mathbf{L} = \prod_{i=1}^{n} \frac{x_i}{\sigma^2}\, \mathbf{Exp}\left[-\frac{x_i^2}{2\,\sigma^2}\right];$$

If we try to evaluate the log-likelihood:

Log[L]

$$\text{Log}\left[\prod_{i=1}^{n} \frac{e^{-\frac{x_i^2}{2\sigma^2}}\, x_i}{\sigma^2}\right]$$

… nothing happens! (*Mathematica* assumes nothing about the symbols that have been entered, so its inaction is perfectly reasonable.) But we can enhance `Log` to do what is wanted here using the **mathStatica** function `SuperLog`. To activate this enhancement, we switch it on:

SuperLog[On]

— SuperLog is now On.

If we now evaluate `Log[L]` again, we obtain a much more useful result:

logL = Log[L]

$$-2\,n\,\text{Log}[\sigma] + \sum_{i=1}^{n} \text{Log}[x_i] - \frac{\sum_{i=1}^{n} x_i^2}{2\,\sigma^2}$$

To derive the first-order conditions for a maximum:

FOC = D[logL, σ]

$$-\frac{2\,n}{\sigma} + \frac{\sum_{i=1}^{n} x_i^2}{\sigma^3}$$

… we solve `FOC==0` using *Mathematica*'s `Solve` function. The ML estimator $\hat{\sigma}$ is given as a replacement rule → for σ:

```
σ̂ = Solve[FOC == 0, σ] [[2]]
```

$$\left\{\sigma \to \frac{\sqrt{\sum_{i=1}^{n} x_i^2}}{\sqrt{2}\ \sqrt{n}}\right\}$$

The second-order conditions (evaluated at the first-order conditions) are always negative, which confirms that $\hat{\sigma}$ is indeed the ML estimator:

```
SOC = D[logL, {σ, 2}] /. σ̂
```

$$-\frac{8\ n^2}{\sum_{i=1}^{n} x_i^2}$$

Finally, let us suppose that an observed random sample is {1, 6, 3, 4}:

```
data = {1, 6, 3, 4};
```

Then the ML estimate of σ is obtained by substituting this data into the ML estimator $\hat{\sigma}$:

```
σ̂ /. {n → 4, x_i_ :> data[[i]]}
```

$$\left\{\sigma \to \frac{\sqrt{31}}{2}\right\}$$

Figure 14 plots the observed likelihood (for the given data) against values of σ, noting the derived exact optimal solution $\hat{\sigma} = \frac{\sqrt{31}}{2}$.

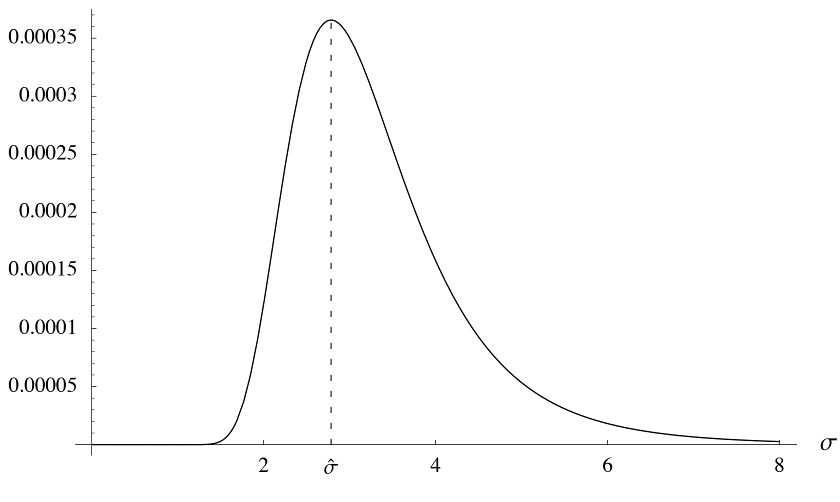


Fig. 14: The observed likelihood and $\hat{\sigma}$

Before continuing, we return `Log` to its default condition:

```
SuperLog[Off]
```

```
— SuperLog is now Off.
```

For more detail, see Chapter 11. ■

⊕ ***Example 8:*** Order Statistics

Let random variable X have a Logistic distribution with pdf $f(x)$:

$$f = \frac{e^{-x}}{(1 + e^{-x})^2}; \qquad \text{domain}[f] = \{x, -\infty, \infty\};$$

Let $(X_1, X_2, \ldots, X_n)$ denote a sample of size n drawn on X, and let $(X_{(1)}, X_{(2)}, \ldots, X_{(n)})$ denote the ordered sample, so that $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$. The pdf of the r^{th} order statistic, $X_{(r)}$, is given by the **mathStatica** function:

```
OrderStat[r, f]
```

$$\frac{(1 + e^{-x})^{-r}\ (1 + e^{x})^{-1-n+r}\ n!}{(n - r)!\ (-1 + r)!}$$

The joint pdf of $X_{(r)}$ and $X_{(s)}$, for $r < s$, is given by:

```
OrderStat[{r, s}, f]
```

$$\frac{e^{x_s}\ (1 + e^{-x_r})^{-r}\ (1 + e^{x_s})^{-1-n+s}\ \left(\frac{1}{1+e^{x_r}} - \frac{1}{1+e^{x_s}}\right)^{-r+s}\ \Gamma[1 + n]}{(-e^{x_r} + e^{x_s})\ \Gamma[r]\ \Gamma[1 + n - s]\ \Gamma[-r + s]}$$

The `OrderStat` function also supports piecewise pdf's. For example, let random variable $X \sim \text{Laplace}(\mu, \sigma)$ with pdf $f(x)$:

$$f = \text{If}\left[x < \mu,\ \frac{e^{\frac{x-\mu}{\sigma}}}{2\,\sigma},\ \frac{e^{-\frac{x-\mu}{\sigma}}}{2\,\sigma}\right];$$

$$\text{domain}[f] = \{x, -\infty, \infty\}\ \&\&\ \{\mu \in \text{Reals},\ \sigma > 0\};$$

Then, the pdf of the r^{th} order statistic, $X_{(r)}$, is:

```
OrderStat[r, f]
```

$$\text{If}\left[x < \mu,\ \frac{2^{-r}\ e^{\frac{r\,(x-\mu)}{\sigma}}\ \left(1 - \frac{1}{2}\ e^{\frac{x-\mu}{\sigma}}\right)^{n-r}\ n!}{\sigma\ (n - r)!\ (-1 + r)!},\ \frac{2^{-1-n+r}\ e^{\frac{(1+n-r)\ (-x+\mu)}{\sigma}}\ \left(1 - \frac{1}{2}\ e^{\frac{-x+\mu}{\sigma}}\right)^{-1+r}\ n!}{\sigma\ (n - r)!\ (-1 + r)!}\right]$$

The textbook reference solution, given in Johnson *et al.* (1995, p. 168), is alas incorrect. For more detail on order statistics, see Chapter 9. ■

1.5 Notation and Conventions

1.5 A Introduction

This book brings together two conceptually different worlds: on the one hand, the *statistics* literature has a set of norms and conventions, while on the other hand *Mathematica* has its own (and different) norms and conventions for symbol entry, typefaces and notation. For instance, Table 4 describes the different conventions for upper and lower case letters, say X and x:

Statistics	X denotes a random variable, x denotes a realisation of that random variable, such as $x = 3$.
Mathematica	Since *Mathematica* is case-specific, X and x are interpreted as completely different symbols, just as different as y is to Z .

Table 4: Upper and lower case conventions

While one could try to artificially fuse these disparate worlds together, the end solution would most likely be a forced, unnatural and ultimately irritating experience. Instead, the approach we have adopted is to keep the two worlds separate, in the obvious way:

- In Text cells: standard statistics notation is used.
- In Input cells: standard *Mathematica* notation is used.

Thus, the Text of this book reads exactly like a standard mathematical statistics text. For instance,

$$\text{“Let } X \text{ have pdf } f(x) = \frac{\text{sech}(x)}{\pi},\ x \in \mathbb{R}. \text{ Find } E[X^2]\text{ .”}$$

By contrast, the computer Input for the same problem follows *Mathematica* conventions, so lower case x is used throughout (no capital X), functions use square brackets (not round ones), and the names of mathematical functions are capitalised so that sech(x) becomes Sech[x]:

$$\mathbf{f} = \frac{\mathbf{Sech[x]}}{\pi}; \quad \mathbf{domain[f]} = \{\mathbf{x}, -\infty, \infty\}; \quad \mathbf{Expect[x^2, f]}$$

$$\frac{\pi^2}{4}$$

If it is necessary to use *Mathematica* notation in the main text, this is indicated by using Courier font. This section summarises these notational conventions in both statistics (Part B) and *Mathematica* (Part C). Related material includes Appendices A.3 to A.8.

1.5 B Statistics Notation

abbreviation	*description*
cdf	cumulative distribution function
cf	characteristic function
cgf	cumulant generating function
$\text{Cov}(X_i, X_j)$	covariance of X_i and X_j
$E[X]$	the expectation of X
iid	independent and identically distributed
mgf	moment generating function
mgfc	central mgf
$M(t)$	mgf : $M(t) = E[e^{tX}]$
MLE	maximum likelihood estimator
MSE	mean square error
$N(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
pdf	probability density function
pgf	probability generating function
pmf	probability mass function
$\Pi(t)$	pgf : $\Pi(t) = E[t^X]$
$P(X \leq x)$	probability
$\text{Var}(X)$	the variance of X
Varcov( )	the variance-covariance matrix

Table 5: Abbreviations

symbol	*description*
$\mathbb{R}$	set of real numbers
$\mathbb{R}^2$	two-dimensional real plane
$\mathbb{R}_+$	set of positive real numbers
$\vec{X}$	$\vec{X} = (X_1, X_2, \ldots, X_m)$
Σ	summation operator
$\prod$	product operator
d	total derivative
∂	partial derivative
$\log(x)$	natural logarithm of x
H^{T}	transpose of matrix H
$\binom{n}{r}$	Binomial coefficient

Table 6: Sets and operators

symbol	*description*
μ	the population mean (same as $\acute{\mu}_1$)
$\acute{\mu}_r$	r^{th} raw moment $\quad \acute{\mu}_r = E[X^r]$
μ_r	r^{th} central moment $\quad \mu_r = E[(X-\mu)^r]$
κ_r	r^{th} cumulant
$\acute{\mu}_{r,s,\ldots}$	multivariate raw moment $\quad \acute{\mu}_{r,s} = E[X_1^r\, X_2^s]$
$\mu_{r,s,\ldots}$	multivariate central moment $\quad \mu_{r,s} = E\left[(X_1 - E[X_1])^r\, (X_2 - E[X_2])^s\right]$
$\kappa_{r,s,\ldots}$	multivariate cumulant
$\acute{\mu}[r]$	r^{th} factorial moment
$\acute{\mu}[r, s]$	multivariate factorial moments
β_1	Pearson skewness measure is $\sqrt{\beta_1}$, where $\beta_1 = \mu_3^2 / \mu_2^3$
β_2	Pearson kurtosis measure $\quad \beta_2 = \mu_4 / \mu_2^2$
p	success probability in Bernoulli trials
ρ or ρ_{ij}	correlation between two random variables
s_r	power sums $\quad s_r = \sum_{i=1}^n X_i^r$
$\acute{m}_r$	sample raw moments $\quad \acute{m}_r = \frac{1}{n} \sum_{i=1}^n X_i^r$
m_r	sample central moments $\quad m_r = \frac{1}{n} \sum_{i=1}^n \left(X_i - \acute{m}_1\right)^r$
S_n	sample sum, for a sample of size n (same as s_1)
$\bar{X}_n$	the sample mean, for a sample of size n (same as $\acute{m}_1$)
θ	population parameter
$\hat{\theta}$	estimate or estimator of θ
h_r	h-statistic : $E[\mathrm{h}_r] = \mu_r$
k_r	k-statistic : $E[\mathrm{k}_r] = \kappa_r$
i_θ	Fisher Information on parameter θ
I_θ	Sample Information on parameter θ
$\sim$	distributed as; $e.g.\ X \sim$ Chi-squared(n)
$\overset{a}{\sim}$	asymptotically distributed
$\overset{d}{\longrightarrow}$	convergence in distribution
$\overset{p}{\longrightarrow}$	convergence in probability

Table 7: Statistics notation

1.5 C *Mathematica* Notation

Common: Table 8 lists common *Mathematica* expressions.

- Note that ⋮ denotes the ESC key.
- *Mathematica* only understands that Γ[x] is equal to Gamma[x] if **mathStatica** has been loaded (see Appendix A.8).

expression	*description*	*short cut*
π	Pi	⋮p⋮
∞	Infinity	⋮inf⋮
i	$\sqrt{-1}$	⋮ii⋮
e	e^x or Exp[x]	⋮ee⋮
Γ	Γ[x] = Gamma[x]	⋮G⋮
∈	Element : {x ∈ Reals}	⋮elem⋮
lis⟦4⟧	Part 4 of lis	⋮[[⋮ or [[
Binomial[n, r]	Binomial coefficient : $\binom{n}{r}$	

Table 8: *Mathematica* notation (common)

Brackets: In *Mathematica,* each kind of bracket has a very specific meaning. Table 9 lists the four main types.

bracket	*description*	*example*		
{ }	Lists	lis = {1, 2, 3, 4}		
[ ]	Functions	y = Exp[x]	not	Exp(x)
( )	Grouping	$(y(x+2)^3)^4$	not	$\{y(x+2)^3\}^4$
lis⟦4⟧	Part 4 of lis	⋮[[⋮ or [[		

Table 9: *Mathematica* bracketing

Replacements: Table 10 lists notation for making replacements; see also Wolfram (1999, Section 2.4.1). Note that :→ is entered as :> and *not* as :->. Example:

3 x² /. x → θ

3 θ²

operator	*description*	*short cut*
/.	ReplaceAll	
→	Rule	⋮->⋮ or ->
:→	RuleDelayed	⋮:>⋮ or :>

Table 10: *Mathematica* replacements

Greek alphabet (common):

letter		*short cut*	*name*
α		⋮a⋮	alpha
β		⋮b⋮	beta
γ,	Γ	⋮g⋮ , ⋮G⋮	gamma
δ,	Δ	⋮d⋮ , ⋮D⋮	delta
ε		⋮ce⋮	epsilon
θ,	Θ	⋮q⋮ , ⋮Q⋮	theta
κ		⋮k⋮	kappa
λ,	Λ	⋮l⋮ , ⋮L⋮	lambda
μ		⋮m⋮	mu
ξ		⋮x⋮	xi
π		⋮p⋮	pi
ρ		⋮r⋮	rho
σ,	Σ	⋮s⋮ , ⋮S⋮	sigma
ϕ,	Φ	⋮f⋮ , ⋮F⋮	phi
χ		⋮c⋮	chi
ψ,	Ψ	⋮y⋮ , ⋮Y⋮	psi
ω,	Ω	⋮w⋮ , ⋮W⋮	omega

Table 11: Greek alphabet (common)

Notation entry: Mathematica's sophisticated typesetting engine makes it possible to use standard statistical notation such as $\hat{x}$ instead of typing `xHAT`, and x_1 instead of `x1` (see Appendix A.5). This makes the transition from paper to computer a much more natural, intuitive and aesthetically pleasing experience. The disadvantage is that we have to learn how to enter expressions like $\hat{x}$. One easy way is to use the BasicTypesetting palette, which is available via File Menu ▷ Palettes ▷ BasicTypesetting. Alternatively, Table 12 lists five essential notation short cuts which are well worth mastering.

notation	*short cut*
$\frac{x}{y}$	x CTRL / y
x^r	x CTRL 6 r
x_1	x CTRL – 1
$\overset{2}{x}$	x CTRL 7 2
$\underset{3}{x}$	x CTRL = 3

Table 12: Five essential notation short cuts

These five notation types

$$\left\{\frac{x}{y},\ x^r,\ x_1,\ \overset{2}{x},\ \underset{3}{x}\right\}$$

can generate almost any expression used in this book. For instance, the expression $\hat{x}$ has the same form as $\overset{2}{x}$ in Table 12, so we can enter $\hat{x}$ with x CTRL 7 ^. If the expression is a well-known notational type, *Mathematica* will represent it internally as a 'special' function. For instance, the internal representation of $\hat{x}$ is actually:

```
x̂ // InputForm

OverHat[x]
```

Table 13 lists these special functions — they provide an alternative way to enter notation. For instance, to enter $\vec{x}$ we could type in OverVector[x], then select the latter with the mouse, and then choose Cell Menu ▹ Convert to StandardForm. This too yields $\vec{x}$.

notation	*short cut*	*function name*
x^+	x CTRL 6 +	SuperPlus[x]
x^-	x CTRL 6 −	SuperMinus[x]
x^*	x CTRL 6 *	SuperStar[x]
$x^\dagger$	x CTRL 6 †	SuperDagger[x]
x_+	x CTRL − +	SubPlus[x]
x_-	x CTRL − −	SubMinus[x]
x_*	x CTRL − *	SubStar[x]
$\overline{x}$	x CTRL 7 _	OverBar[x]
$\vec{x}$	x CTRL 7 ⋮vec⋮	OverVector[x]
$\tilde{x}$	x CTRL 7 ~	OverTilde[x]
$\hat{x}$	x CTRL 7 ^	OverHat[x]
$\dot{x}$	x CTRL 7 .	OverDot[x]
$\underline{x}$	x CTRL = _	UnderBar[x]

Table 13: Special forms

Even more sophisticated structures can be created with `Subsuperscript` and `Underoverscript`, as Table 14 shows.

notation	*function name*
x_1^r	Subsuperscript[x, 1, r]
$\underset{a}{\overset{b}{x}}$	Underoverscript[x, a, b]

Table 14: `Subsuperscript` and `Underoverscript`

Entering $\acute{\mu}_r$: This text uses $\acute{\mu}_r$ to denote the r^{th} raw moment. The prime ′ above μ is entered by typing ESC ' ESC. This is because the keyboard ' is reserved for other purposes by *Mathematica*. Further, notation such as x' (where the prime comes *after* the x, rather than above it) should generally be avoided, as *Mathematica* may interpret the prime as a derivative. This problem does not occur with $\acute{x}$ notation.

```
x́ // InputForm
```

```
Overscript[x, ′]
```

```
x′ // InputForm
```

```
Derivative[1][x]
```

Animations: In the printed text, the symbol 🎬 is used to indicate that an animation is available at the marked point in the electronic version of the chapter.

Magnification: If the on-screen notation seems too small, magnification can be used: Format Menu ▹ Magnification.

Notes: Here is an example of a note.[1] In the electronic version of the text, notes are live links that can be activated with the mouse. In the printed text, notes are listed near the end of the book in the Notes section.

Timings: All timings in this book are based on *Mathematica* Version 4 running on a PC with an 850 MHz Pentium III processor.

Finally, the Appendix provides several tips for both the new and advanced user on the accuracy of symbolic and numeric computation, on working with Lists, on using Subscript notation, on working with matrices and vectors, on changes to default behaviour, and on how to expand the **mathStatica** framework with your own functions.