Solutions Manual to accompany Quantitative Methods

An Introduction for Business Management

Provisional version of May 23, 2011

Paolo Brandimarte

A Wiley-Interscience Publication

JOHN WILEY & SONS, INC.

New York / Chichester / Weinheim / Brisbane / Singapore / Toronto

Contents

Preface

This solutions manual contains

- worked-out solutions to end-of-chapter problems in the book
- additional problems (solved)
- computational supplements illustrating the application of the following tools:
  - Microsoft Excel
  - R
  - MATLAB
  - AMPL

Some software tools are introduced in the appendices, where I am giving you a few hints and clues about how they can be used to apply the methods described in the book. Some of these tools are free, some have free student demos, some can be obtained at a reduced price. Anyway, they are all widely available and I encourage you to try them. To the very least, they can provide us with quantiles from probability distributions, and are much more handy and precise than using old-style statistical tables.

The manual is work-in-progress, so be sure to check back every now and then whether a new version has been posted.

This version is dated May 23, 2011

As usual, for comments, suggestions, and criticisms, my e-mail address is:

PAOLO BRANDIMARTE
paolo.brandimarte@polito.it

1

Quantitative Methods: Should We Bother?

1.1 SOLUTIONS

Problem 1.1 We consider the strategy of trying Plan A first, and then Plan B; a more complete solution approach should reply on the decision tree framework of Chapter 13. Say that the first move has been a success; if we try Plan B we invest 4000 now, and at the end of the second year we will make 6600 with probability $0.5 + \alpha$ and 0 with probability $1 - (0.5 + \alpha) = 0.5 - \alpha$. The expected NPV for this part of the strategy is

$$\frac{6600}{1.1} \times (0.5 + \alpha) - 4000 = 6000 \times \alpha - 1000.$$

We go on with Plan B only if this is positive, i.e.

$$\alpha \geq \frac{1000}{6000} = 0.1667$$

On the other branch of the tree, the first movie has been a flop. The second movie, if we adopt plan B, has expected profit

$$\frac{6600}{1.1} \times (0.5 - \alpha) - 4000 = -6000 \times \alpha - 5000.$$

which will always be negative for $\alpha \geq 0$ (this makes sense; with 50–50 probabilities, the bet is not quite promising, and the situation does not improve after a flop).
If we step back at the root node, where we apply Plan A, we see the expected NPV:

$$\begin{aligned} & -2500 + 0.5 \times \frac{4400 - 4000}{1.1} + 0.5 \times \frac{6000 \times \alpha - 1000}{(1.1)^2} \\ & = -2731.405 + 1239.6694\alpha \end{aligned}$$

which is positive if

$$\alpha \geq \frac{1239.6694}{2731.405} = 0.4539.$$

We know that the strategy is worthwhile if $\alpha = 0.5$, but we see that, with these fictional numbers, a small decrease will make us change our mind.

Problem 1.2 Rather than extending the little numerical example of Chapter 1, let us state the model in general form (also see Chapter 12):

$$\begin{aligned}
\max \quad & \sum_{i=1}^{N} (p_i - c_i) x_i \\
\text{s.t.} \quad & \sum_{i=1}^{N} r_{im} x_i \le R_m && m = 1, \ldots, M \\
& 0 \le x_i \le d_i && i = 1, \ldots, N
\end{aligned}$$

where

- items are indexed by $i = 1, \ldots, N$
- resources are indexed by $m = 1, \ldots, M$
- d_i, p_i, and c_i are the demand, selling price, and production cost, respectively, for item i
- r_{im} is the requirement of resource m for item i, and R_m is the total availability of resource m

In this model, we have a single decision variable, x_i, representing what we produce *and* sell. If we introduce the possibility of third-party production, we can no longer identify production and sales. We need to change decision variables as follows:

- x_i is what we produce
- y_i is what we buy
- we could also introduce a variable z_i to denote what we sell, but since $z_i = x_i + y_i$, we can avoid that[1]

Let us denote by g_i the cost of purchasing item i from the supplier. The model is now

$$\begin{aligned}
\max \quad & \sum_{i=1}^{N} \left[(p_i - c_i) x_i + (p_i - g_i) y_i \right] \\
\text{s.t.} \quad & \sum_{i=1}^{N} r_{im} x_i \le R_m && m = 1, \ldots, M \\
& x_i + y_i \le d_i && i = 1, \ldots, N \\
& x_i, y_i \ge 0 && i = 1, \ldots, N
\end{aligned}$$

[1]However, in multiperiod problems involving inventory holding we do need such a variable; see Chapter 12.

If we allow for overtime work, we change the first model by introducing the amount of overtime O_m on resource m, with cost q_m:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{N}(p_i - c_i)x_i - \sum_{m=1}^{M} q_m O_m \\
\text{s.t.} \quad & \sum_{i=1}^{N} r_{im}x_i \leq R_m + O_m && m = 1, \ldots, M \\
& 0 \leq x_i \leq d_i && i = 1, \ldots, N \\
& O_m \geq 0 && m = 1, \ldots, M
\end{aligned}$$

These are just naive models used for introductory purposes. In practice, we should (to the very least) account for limitations on overtime work, as well as fixed charges associated with purchasing activities.

1.2 COMPUTATIONAL SUPPLEMENTS

1.2.1 Optimal mix problem

In this section we show how different software tools can be used to solve the optimal mix problem of Section 1.1.2.
A first alternative is using MATLAB, as shown in Section B.5:

```
>> m = [45, 60];
>> reqs = [ 15 10; 15 35; 15 5; 25 15];
>> res = 2400*ones(4,1);
>> d = [100, 50];
>> x = linprog(-m, reqs, res, [], [], zeros(2, 1), d)
Optimization terminated.
x =
   73.8462
   36.9231
```

Unfortunately, with MATLAB we cannot solve integer programming problems. To this aim, we may use AMPL (see Appendix C). We need a model and a data file, illustrated in Fig. 1.1.[2] Using the CPLEX solver with AMPL, we find

```
ampl: model ProdMix.mod;
ampl: data ProdMix.dat;
ampl: solve;
CPLEX 11.1.0: optimal integer solution; objective 5505
2 MIP simplex iterations
0 branch-and-bound nodes
1 Gomory cut
ampl: display x;
x [*] :=
1  73
2  37
;
```

[2]The code is provided in the book Web sites.

```
# ProdMix.mod
param NumItems > 0;
param NumResources > 0;
param ProfitContribution{1..NumItems};
param MaxDemand{1..NumItems};
param ResReqs{1..NumItems, 1..NumResources};
param ResAvail{1..NumResources};

var x{i in 1..NumItems} >= 0, <= MaxDemand[i], integer;

maximize profit:
      sum {i in 1..NumItems} ProfitContribution[i] * x[i];

subject to Capacity {j in 1..NumResources}:
      sum {i in 1..NumItems} ResReqs[i,j] * x[i] <= ResAvail[j];
```

```
param NumItems := 2;
param NumResources := 4;
param: ProfitContribution MaxDemand :=
    1 45 100
    2 60 50;
param ResAvail := default 2400;
param ResReqs:
     1  2  3  4 :=
1   15 15 15 25
2   10 35  5 15;
```

Fig. 1.1 AMPL model (`ProdMix.mod`) and data (`ProdMix.dat`) files for product mix optimization.

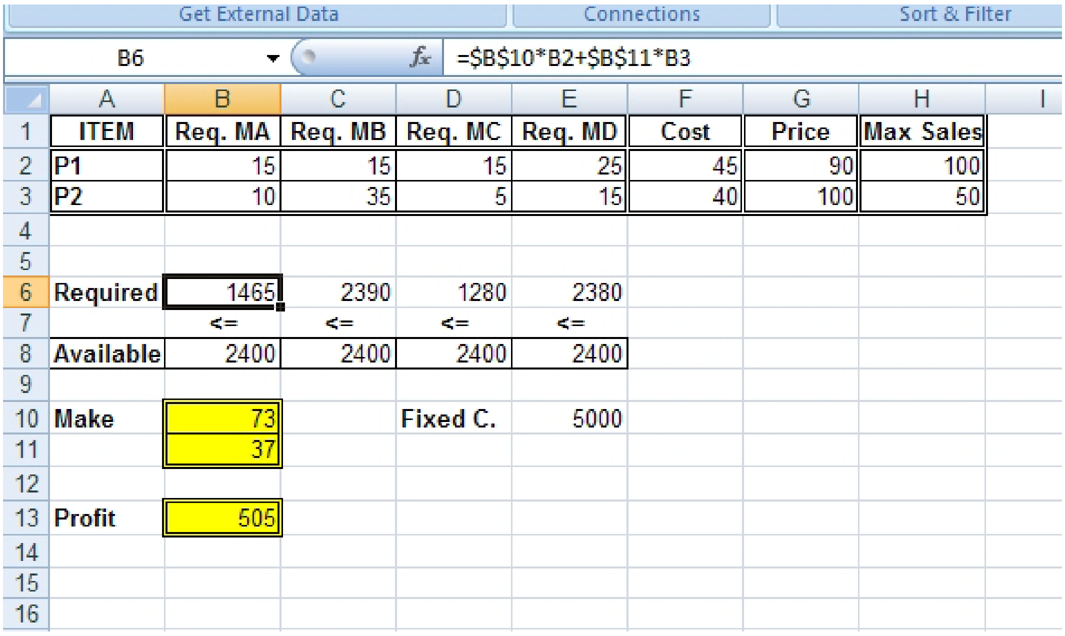
If we omit the `integer` keyword in the definition of decision variable `x`, we obtain the same solution as MATLAB.

Another widespread tool that can be used to solve LP and MILP problems is Microsoft Excel, which is equipped by a Solver directly interfaced with the spreadsheet.[3] This is a double-edged sword, since it means that the model must be expressed in a two-dimensional array of cells, where data, constraints, and the objective function must be related by formulas.

The product mix problem can be represented as in the `ProdMix.xls` workbook, as shown in Fig. 1.2. The cell `Profit` contain the profit contribution and the cells `Required` are used to calculate the resource requirements as a function of the amounts produced, which are the contents of the cells `Make`. It is important to name ranges of cells to include them in the model.

The model is described by opening the Solver window and specifying decision variables, constraints, and the objective cell as illustrated in Fig. fig:ProdMixExcel2. As you see, reference is made to named cell ranges; the cells containing `<=` in the worksheet have no meaning, actually, and are only included to clarify the model structure.

[3]You should make sure that the Solver was included in your Excel installation; sometimes, it is not included to save space on disk.



	A	B	C	D	E	F	G	H	I
1	ITEM	Req. MA	Req. MB	Req. MC	Req. MD	Cost	Price	Max Sales	
2	P1	15	15	15	25	45	90	100	
3	P2	10	35	5	15	40	100	50	
4									
5									
6	Required	1465	2390	1280	2380				
7		<=	<=	<=	<=				
8	Available	2400	2400	2400	2400				
9									
10	Make	73		Fixed C.	5000				
11		37							
12									
13	Profit	505							
14									
15									
16									

Fig. 1.2 The `ProdMix.xls` workbook to solve the optimal product mix problem.

As one can imagine, describing and maintaining a large-scale model in this form may quickly turn into nightmare (not taking into account the fact that state-of-the-art solvers are needed to solve large problem instances). Nevertheless, Excel can be used to solve small scale models and is in fact the tool of choice of many Management Science books. We should mention, however, that the true power of Excel is its integration with VBA (Visual Basic for Application), a powerful programming language which can be used to develop remarkable applications. A clever strategy is to use Excel as a familiar and user-friendly interface, and VBA to build a link it with state-of-the-art software libraries.

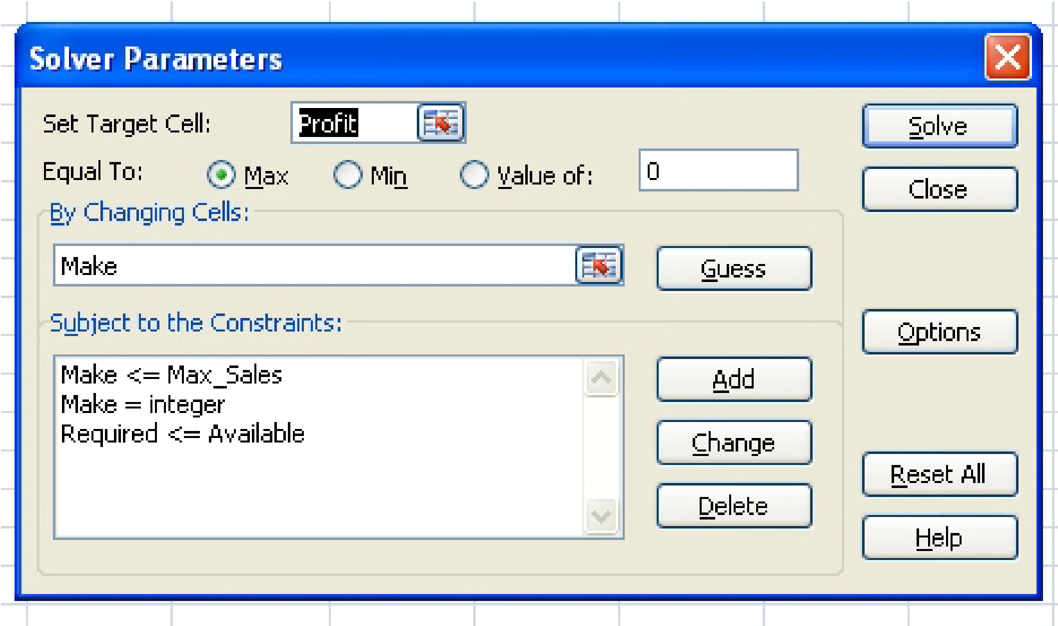


Fig. 1.3 Caption for ProdMixExcel2

2

Calculus

2.1 SOLUTIONS

Problem 2.1 A first requirement for function $f(x)$ is that the argument of the square root is positive:

$$1 - x^2 \geq 0 \quad \Rightarrow \quad -1 \leq x \leq 1$$

Then, the denominator of the ratio cannot be zero:

$$\sqrt{1 - x^2} \neq 1 \quad \Rightarrow \quad x \neq 0$$

Then, the domain of f is $[-1, 0) \cup (0, 1]$.
The square root in function $g(x)$ is not an issue, as $x^2 + 1 \neq 0$. We also observe that the denominator is never zero, since

$$\sqrt{x^2 + 1} = x \quad \Rightarrow \quad x^2 + 1 = x^2 \quad \Rightarrow \quad 1 = 0$$

which is false. Then, the domain of g is the whole real line.

Problem 2.2 The first line is easy to find using the form $y = mx + q$

$$y = -3x + 10$$

For the second one, we use the form $y - y_0 = m(x - x_0)$

$$y - 4 = 5(x + 2) \quad \Rightarrow \quad y = 5x + 14$$

For the third line, we observe that its slope is

$$m = \frac{3 - (-5)}{1 - 3} = -4$$

Then we have

$$y - 3 = -4(x - 1) \quad \Rightarrow \quad y = -4x + 7$$

Alternatively, we might also consider its parametric form

$$\begin{cases} y = \lambda y_a + (1-\lambda)y_b = 3\lambda - 5(1-\lambda) \\ x = \lambda x_a + (1-\lambda)x_b = \lambda + 3(1-\lambda) \end{cases}$$

and eliminate λ between the two equations. This approach is less handy, but it stresses the idea of a line as the set of *affine combinations* of two vectors. An affine combination of vectors is a linear combination whose weights add up to one (see Chapter 3).

Problem 2.3

$$f_1'(x) = \frac{3\cdot(x^2+1) - 3x\cdot 2x}{(x^2+1)^2} = \frac{3(3x^2+1)}{(x^2+1)^2}$$

$$f_2'(x) = (3x^2 - 2x + 5)e^{x^3 - x^2 + 5x - 3}$$

$$f_3'(x) = \frac{1}{2\sqrt{\exp\left(\frac{x+2}{x-1}\right)}} \cdot \exp\left(\frac{x+2}{x-1}\right) \cdot \frac{-3}{(x-1)^2}$$

Problem 2.4 Let us start with $f(x) = x^3 - x$. We observe the following:

- Limits:
$$\lim_{x\to-\infty} f(x) = -\infty, \qquad \lim_{x\to+\infty} f(x) = +\infty$$
- Roots: we have $f(x) = 0$ for
$$x(x^2-1) = 0 \Rightarrow x = 0,\ x = \pm 1$$
- The first order derivative $f'(x) = 3x^2 - 1$ is zero for $x = \pm 1/\sqrt{3} \approx \pm 0.5774$, positive for $x < -1/\sqrt{3}$ and $x > 1/\sqrt{3}$, negative otherwise. Hence, the function is increasing (from $-\infty$) for $x < -1/\sqrt{3}$, decreasing for $-1/\sqrt{3} < x < 1/\sqrt{3}$, and then it increases to $+\infty$.
- The second order derivative $f''(x) = 6x$ is negative for negative x and positive for positive x; hence, the function is concave for $x < 0$ (with a maximum at $x = -1/\sqrt{3}$) and convex for $x > 0$ (with a minimum at $x = 1/\sqrt{3}$).

For function $g(x) = x^3 + x$ the analysis is similar, but now the first-order derivative $f'(x) = 3x^2 + 1$ is always positive and the function has a unique root at $x = 0$ (and neither minima nor maxima).
See the plots in Fig. 2.1.

Problem 2.5

1. For function $f_1(x)$, we observe that the function is continuous at $x = 0$, as
$$f_1(0_-) = 0 = 0 = f_1(0_+)$$
but not differentiable, as
$$f_1'(0_-) = -1 \neq 0 = f_1'(0_+)$$
2. For function $f_1(x)$, we observe that the function is not continuous at $x = 0$, as
$$f_2(0_-) = 1 \neq 0 = f_2(0_+)$$

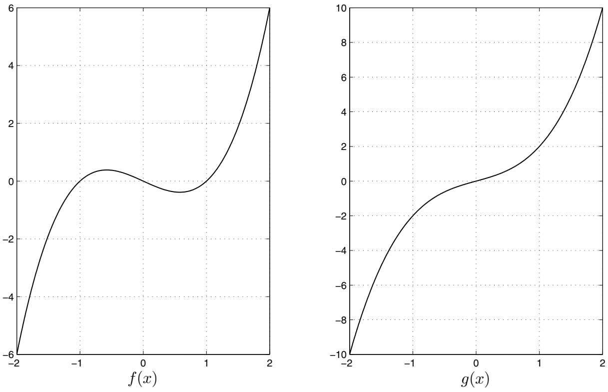


Fig. 2.1 Plots of functions in Problem 2.4.

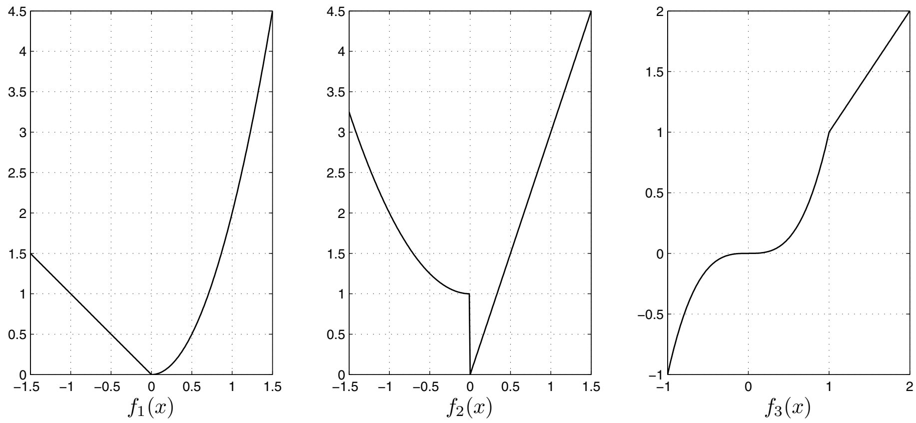


Fig. 2.2 Plots of functions in Problem 2.5.

Then, the function cannot be differentiable.

3. For function $f_3(x)$, we observe that the function is continuous at $x = 1$, as

$$f_3(1_-) = 1 = 1 = f_3(1_+)$$

but not differentiable, as

$$f_3'(1_-) = 3 \neq 1 = f_3'(1_+)$$

See the plots in Fig. 2.2.

Problem 2.6 Consider function

$$f(x) = \exp\left(-\frac{1}{1+x^2}\right)$$

and find linear (first-order) and quadratic (second-order) approximations around points $x_0 = 0$ and $x_0 = 10$. Check the quality of approximations around these points.

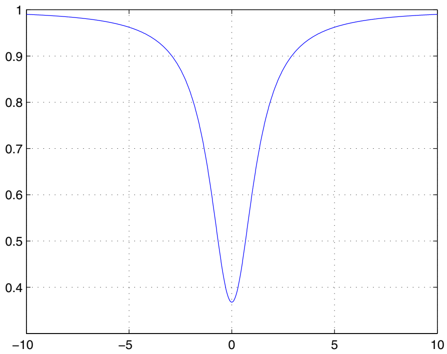


Fig. 2.3 Plot of function in Problem 2.6.

We have

$$f(0) = 0.3678794411714 42,\ f(10) = 0.990147863338053$$

The function is plotted in Fig. 2.3.
Let us find and evaluate the first-order derivative

$$f'(x) = \exp\left(-\frac{1}{1+x^2}\right)\frac{2x}{(1+x^2)^2},$$
$$f'(0) = 0,\ f'(10) = 0.001941276077518$$

Then the second-order derivative

$$\begin{aligned} f''(x) &= \exp\left(-\frac{1}{1+x^2}\right)\left[\frac{2x}{(1+x^2)^2}\right]^2 + \exp\left(-\frac{1}{1+x^2}\right)\frac{2\cdot(1+x^2)^2 - 2x\cdot 2(1+x^2)\cdot 2x}{(1+x^2)^4} \\ &= \exp\left(-\frac{1}{1+x^2}\right)\left[\frac{4x^2}{(1+x^2)^4} + \frac{2-6x^2}{(1+x^2)^3}\right] \\ &= \exp\left(-\frac{1}{1+x^2}\right)\frac{2-6x^4}{(1+x^2)^4} \\ f''(0) &= 0.735758882342885,\ f''(10) = -5.708885506270199\cdot 10^{-4} \end{aligned}$$

The Taylor expansions $p_{n,x_0}(x)$, where n is the order and x_0 is where the approximation is built, are:

$$\begin{aligned} p_{1,0}(x) &= 0.367879441171442 \\ p_{2,0}(x) &= 0.367879441171442 + \frac{1}{2}\cdot 0.735758882342885\cdot x^2 \\ p_{1,10}(x) &= 0.990147863338053 + 0.001941276077518\cdot(x-10) \\ p_{2,10}(x) &= 0.990147863338053 + 0.001941276077518\cdot(x-10) \\ &\quad - \frac{1}{2}\cdot 0.0005708885506270199\cdot(x-10)^2 \end{aligned}$$

The four approximations are plotted in Figs. 2.4(a), (b), (c) and (d), respectively, where the function plot is the dashed line.

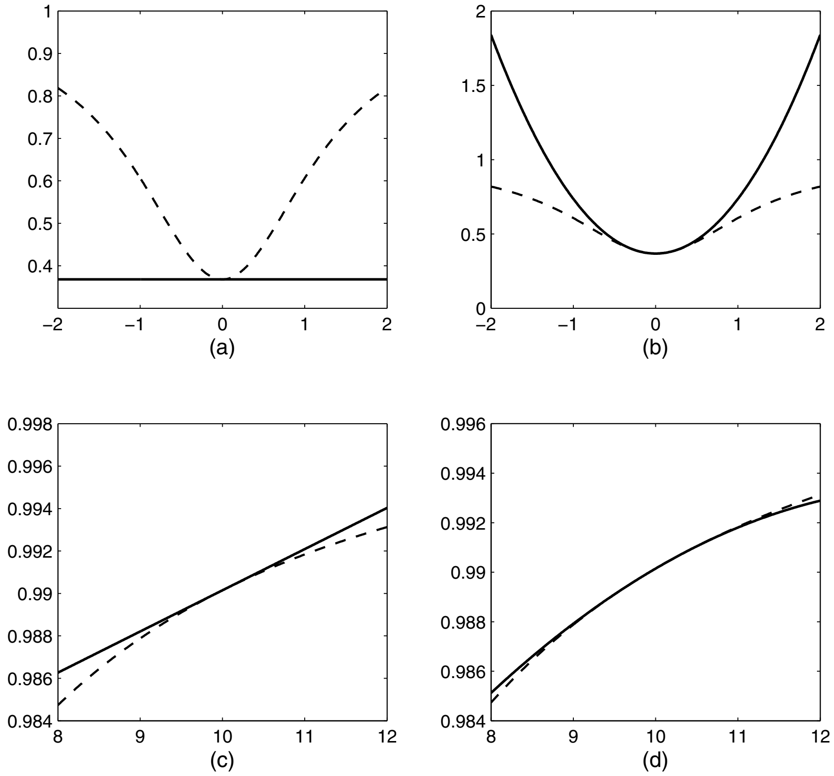


Fig. 2.4 Plot of approximations of Problem 2.6.

Problem 2.7 Let us express the bond price in terms of the continuous-time yield y_c:

$$P(y_c) = \sum_{t=1}^{T-1} Ce^{-y_c t} + (C+F)e^{-y_c T}$$

and take its first-order derivative

$$P'(y_c) = -\sum_{t=1}^{T-1} Cte^{-y_c t} - (C+F)Te^{-y_c T} = -\sum_{t=1}^{T} td_t$$

where d_t is the discounted cash flow at time t, i.e., $d_t = Ce^{-y_c t}$ for $t = 1, \ldots, T-1$ and $d_T = (C+F)e^{-y_c T}$. We see that this expression, unlike the case with discrete-time compounding, does not contain any extra-term involving yield. Hence, we may write

$$P'(y_c) = \frac{dP}{dy_c} = -\sum_{t=1}^{T} td_t \cdot \frac{\sum_{t=1}^{T} d_t}{\sum_{t=1}^{T} d_t} = -D \cdot P$$

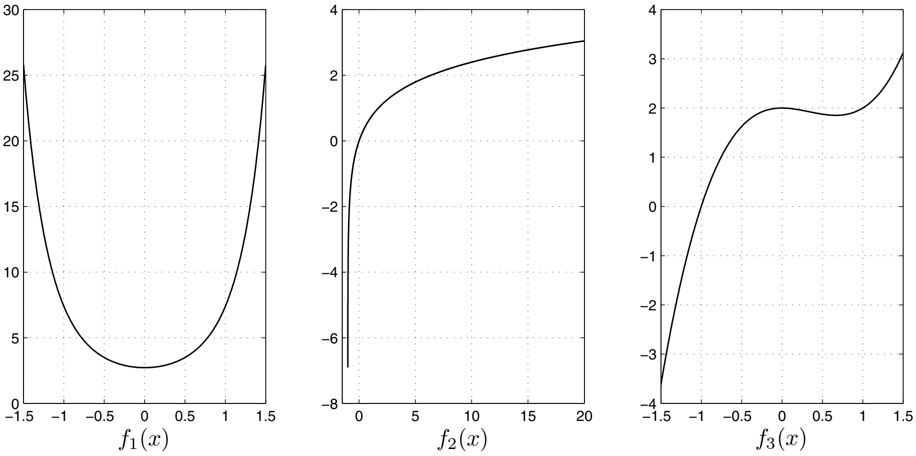


Fig. 2.5 Plots of functions in Problem 2.8.

where

$$D \equiv \frac{\sum_{t=1}^{T} t d_t}{\sum_{t=1}^{T} d_t}$$

is duration. Then, we may express relative price variations for small changes in yield as

$$\frac{\Delta P}{P} = -D \cdot \Delta y_c$$

where we use duration, rather than modified duration.

Problem 2.8 For function f_1 we have

$$f_1'(x) = 2xe^{x^2+1}, \quad f_1''(x) = 2e^{x^2+1} + 4x^2e^{x^2+1} = 2e^{x^2+1}(1+x^2) > 0$$

Then, the function is convex on the real line.
For function f_2, with domain $x > -1$, we have

$$f_2'(x) = \frac{1}{x+1}, \quad f_2''(x) = \frac{-1}{(x+1)^2} > 0$$

Then, the function is concave on its domain.
For function f_3 we have

$$f_3'(x) = 3x^2 - 2x, \quad f_3''(x) = 6x - 2$$

Since the second order derivative changes its sign at $x = 1/3$, the function neither convex nor concave.
See Fig. 2.5.

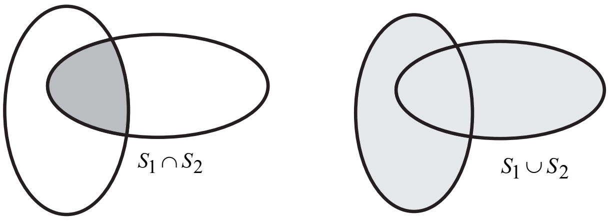


Fig. 2.6 Intersection and union of two convex sets.

Problem 2.9 We must prove that if $\mathbf{x}_a, \mathbf{x}_b \in S_1 \cap S_2$, then $\mathbf{x}_\lambda = \lambda \mathbf{x}_a + (1-\lambda)\mathbf{x}_b \in S_1 \cap S_2$, for any $\lambda \in [0, 1]$.
Now consider two elements $\mathbf{x}_a, \mathbf{x}_b \in S_1 \cap S_2$. Since S_1 and S_2 are both convex, we know that, for any $\lambda \in [0, 1]$,

$$\lambda \mathbf{x}_a + (1-\lambda)\mathbf{x}_b \in S_1$$
$$\lambda \mathbf{x}_a + (1-\lambda)\mathbf{x}_b \in S_2$$

But this shows $\mathbf{x}_\lambda \in S_1 \cap S_2$.
This property is visualized in Fig. 2.6; note that the union of convex sets need not be convex.

Problem 2.10 From Section 2.12 we know that the present value V_0, at time $t = 0$, of a stream of constant cash flows $C_t = C$, $t = 1, \ldots, T$, is

$$V_0 = \frac{C}{r}\left[1 - \frac{1}{(1+r)^T}\right]$$

To get the future value V_T, at time $t = T$, we just multiply by a factor $(1+r)^T$, which yields

$$V_T = \frac{C}{r}\left[(1+r)^T - 1\right]$$

Problem 2.11 You work for $T_s = 40$ years saving S per year; then you live $T_c = 20$ years, consuming $C = 20000$ per year. The cumulated wealth when you retire is

$$\frac{S}{r}\left[(1+r)^{T_s} - 1\right]$$

and the present value of the consumption stream is

$$\frac{C}{r}\left[1 - \frac{1}{(1+r)^{T_c}}\right]$$

Equating these two expressions we find

$$S = \frac{C\left[1 - \frac{1}{(1+r)^{T_c}}\right]}{(1+r)^{T_s} - 1} = \frac{20000 \times \left[1 - \frac{1}{1.05^{20}}\right]}{(1.05)^{40} - 1} = 2063.28$$

If $T_c = 10$, S is reduced to 1278.44

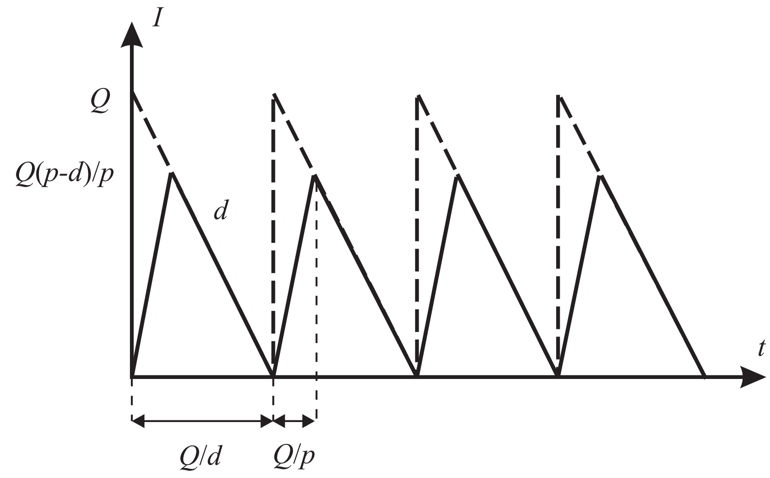


Fig. 2.7 Inventory with finite rate of replenishment.

Problem 2.12 The finite replenishment rate p is the number of items delivered per unit of time. When the inventory level reaches zero, it does not immediately increase by Q units but increases progressively at rate $p-d$, as shown in Fig. 2.7; this rate is the difference between the rates of item inflow and outflow. It takes Q/p time units to complete the production lot Q; hence, when the lot is completed, the inventory level has reached a level

$$(p-d)\,Q/p$$

which is the height of the triangle corresponding to one cycle. Then, inventory decreases between $(p-d)\,Q/p$ and 0 at rate d. With respect to the EOQ model, there is a difference in the average inventory level, which is now

$$\frac{(p-d)Q}{2p}$$

Then, the total cost function is

$$\frac{Ad}{Q} + h \cdot \frac{(p-d)Q}{2p}$$

and using the same drill as the EOQ case (set first-order derivative to 0) we find

$$Q^* = \sqrt{\frac{2Ad}{h} \cdot \frac{p}{p-d}}$$

It is interesting to note that when $p \to +\infty$ we get back to the EOQ formula.

3

Linear Algebra

3.1 SOLUTIONS

Problem 3.1 Gaussian elimination works as follows:

$$\left[\begin{array}{rrr|r} 1 & 2 & -1 & 3 \\ 1 & 0 & 4 & 9 \\ 0 & 2 & 1 & 0 \end{array}\right] \overset{E_2 \leftarrow E_2 - E_1}{\Longrightarrow} \left[\begin{array}{rrr|r} 1 & 2 & -1 & 3 \\ 0 & -2 & 5 & 12 \\ 0 & 2 & 1 & 0 \end{array}\right] \overset{E_3 \leftarrow E_3 + E_2}{\Longrightarrow} \left[\begin{array}{rrr|r} 1 & 2 & -1 & 3 \\ 0 & -2 & 5 & 12 \\ 0 & 0 & 6 & 12 \end{array}\right]$$

Using backsubstitution:

$$\begin{aligned} 6x_3 &= 12 && \rightarrow && x_3 = 2 \\ -2x_2 + 5x_3 &= 12 && \rightarrow && x_2 = \frac{12 - 5x_3}{-2} = -1 \\ x_1 + 2x_2 - x_3 &= 3 && \rightarrow && x_1 = 3 - 2x_2 + x_3 = 1 \end{aligned}$$

To apply Cramer's rule we compute the determinant of the matrix

$$\Delta = \left|\begin{array}{rrr} 1 & 2 & -1 \\ 1 & 0 & 4 \\ 0 & 2 & 1 \end{array}\right| = 1 \times \left|\begin{array}{rr} 0 & 4 \\ 2 & 1 \end{array}\right| - 2 \times \left|\begin{array}{rr} 1 & 0 \\ 4 & 1 \end{array}\right| - 1 \times \left|\begin{array}{rr} 1 & 0 \\ 0 & 2 \end{array}\right|$$

$$= 1 \times (-8) - 2 \times 1 - 2 \times 2 = -12$$

By a similar token

$$\Delta_1 = \left|\begin{array}{rrr} -3 & 2 & -1 \\ 9 & 0 & 4 \\ 0 & 2 & 1 \end{array}\right| = -12$$

$$\Delta_2 = \left|\begin{array}{rrr} 1 & -3 & -1 \\ 1 & 9 & 4 \\ 0 & 0 & 1 \end{array}\right| = 12$$

$$\Delta_3 = \left|\begin{array}{rrr} 1 & 2 & -3 \\ 1 & 0 & 9 \\ 0 & 2 & 0 \end{array}\right| = -24$$

By the way, it is sometimes convenient to develop the determinant not using the first row, but any row or column with few nonzero entries; for instance

$$\Delta_2 = 1 \times \begin{vmatrix} 1 & -3 \\ 1 & 9 \end{vmatrix}$$
$$\Delta_3 = -2 \times \begin{vmatrix} 1 & -3 \\ 1 & 9 \end{vmatrix}$$

Then, we find

$$x_1 = \frac{\Delta_1}{\Delta} = 1, \quad x_2 = \frac{\Delta_2}{\Delta} = -1, \quad x_3 = \frac{\Delta_3}{\Delta} = 2$$

Problem 3.2 Let us consider polynomials of degree up to n

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

Such a polynomial may be expressed as a vector $\mathbf{a} \in \mathbb{R}^{n+1}$

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

where we associate each monomial x^k, $k = 0, 1, 2, \ldots, n$ with a unit vector:

$$\mathbf{e}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{e}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots \mathbf{e}_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

We know that

$$(x^k)' = kx^{k-1}$$

Therefore, the mapping from a monomial to its derivative may be represented as follows

$$\mathbf{e}_k = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \mathbf{e}'_k = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ k \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

If we align such vectors, we obtain the following matrix

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & n-1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & n \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

For instance, consider the polynomial

$$p(x) = 3 + 5x + 2x^2 - x^3 + 2x^4 \quad \Rightarrow \quad p'(x) = 5 + 4x - 3x^2 + 8x^3$$

The mapping is

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \\ 2 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ -3 \\ 8 \\ 0 \end{bmatrix}$$

We may also observe that the matrix is not invertible, which makes sense, since two polynomials that differ only in the constant term have the same derivative.

Problem 3.3 Prove that the representation of a vector using a basis is unique.
Let us assume that, contrary to the statement, we have two representations of the same vector using the same basis (which is a set of linearly independent vectors:

$$\mathbf{v} = \alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \cdots + \alpha_n \mathbf{e}_n$$
$$\mathbf{v} = \beta_1 \mathbf{e}_1 + \beta_2 \mathbf{e}_2 + \cdots + \beta_n \mathbf{e}_n$$

Taking the difference, we have

$$\mathbf{0} = (\alpha_1 - \beta_1)\mathbf{e}_1 + (\alpha_2 - \beta_2)\mathbf{e}_2 + \cdots + (\alpha_n - \beta_n)\mathbf{e}_n$$

However, since the vectors $\mathbf{e}_k$, $k = 1, \ldots, n$, are a basis, there is no way to find a linear combination of them yielding the null vector, unless all of the coefficients in the linear combination are all zero, which implies

$$\alpha_k = \beta_k, \quad k = 1, \ldots, n$$

Problem 3.4 Let $\mathbf{B} = \mathbf{AD}$, where $\mathbf{B} \in \mathbb{R}^{m,n}$ and we denote its generic element b_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$. To obtain b_{ij}, we multiply elements of row i of $\mathbf{A}$ by elements of column j of $\mathbf{D}$:

$$b_{ij} = \sum_{k=1}^{n} a_{ik} d_{kj}$$

We may think of this element as the inner product between the row vector $\mathbf{a}_i^T$ and the column vector $\mathbf{d}_j$. However, we have $d_{kj} = 0$ for $j \neq k$, so

$$b_{ij} = a_{ik} d_{kk}$$

i.e., element k of row i of matrix $\mathbf{A}$ is multiplied by the corresponding element d_{kk} on the diagonal of $\mathbf{D}$:

$$\mathbf{AD} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 6 & 4 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 7 \end{bmatrix} = \begin{bmatrix} 2 & -9 & 35 \\ 4 & -18 & 28 \end{bmatrix}$$

Problem 3.5 Multiplying the matrices, we find:

$$\mathbf{AX} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} 6 & 5 & 7 \\ 2 & 2 & 4 \\ 3 & 3 & 6 \end{bmatrix} = \begin{bmatrix} 12 & 11 & 19 \\ 5 & 5 & 10 \\ 18 & 16 & 26 \end{bmatrix}$$

$$\mathbf{BX} = \begin{bmatrix} 1 & 3 & 0 \\ 0 & 4 & -1 \\ 2 & 3 & 0 \end{bmatrix} \begin{bmatrix} 6 & 5 & 7 \\ 2 & 2 & 4 \\ 3 & 3 & 6 \end{bmatrix} = \begin{bmatrix} 12 & 11 & 19 \\ 5 & 5 & 10 \\ 18 & 16 & 26 \end{bmatrix}$$

This implies that, unlike with the scalar case, we cannot simplify an equation $\mathbf{AX} = \mathbf{BX}$ to $\mathbf{A} = \mathbf{B}$. If matrix $\mathbf{X}$ is invertible, then we may postmultiply by its inverse $\mathbf{X}^{-1}$ and simplify. But in the example $\mathbf{X}$ is singular; to see this, observe that its second and third row are linearly dependent, as they are obtained multiplying the vector $[1, 1, 2]$ by 2 and 3, respectively.

Problem 3.6 Given the matrix $\mathbf{H} = \mathbf{I} - 2\mathbf{hh}^T$, we may check orthogonality directly:

$$\begin{aligned} \mathbf{H}^T\mathbf{H} &= (\mathbf{I} - 2\mathbf{hh}^T)(\mathbf{I} - 2\mathbf{hh}^T) \\ &= \mathbf{I} - 2\mathbf{hh}^T - 2\mathbf{hh}^T + 4\mathbf{h}(\mathbf{h}^T\mathbf{h})\mathbf{h}^T \\ &= \mathbf{I} - 4\mathbf{hh}^T + 4\mathbf{hh}^T \\ &= \mathbf{I} \end{aligned}$$

where we exploit the symmetry of $\mathbf{H}$ and the condition $\mathbf{h}^T\mathbf{h} = 1$.
This transformation is actually a reflection of a generic vector with respect to a hyperplane passing through the origin and characterized by an orthogonal vector $\mathbf{h}$. To see this, consider the application of $\mathbf{H}$ to vector $\mathbf{v}$:

$$\mathbf{Hv} = \mathbf{Iv} - 2\mathbf{hh}^T\mathbf{v} = \mathbf{v} - 2\alpha\mathbf{h}$$

where $\alpha = \mathbf{h}^T\mathbf{v}$ is the length of the projection of $\mathbf{v}$ on the unit vector $\mathbf{h}$. To illustrate in the plane, consider $\mathbf{v} = [3, 1]^T$ and $\mathbf{h} = [1, 0]^T$:

$$\mathbf{Hv} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} - 2 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left([1 \quad 0] \begin{bmatrix} 3 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 3 \\ 1 \end{bmatrix} - 2 \times 3 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} -3 \\ 1 \end{bmatrix}$$

The resulting vector is indeed the reflection of $\mathbf{v}$ with respect to the horizontal axis, which is a line going through the origin and orthogonal to vector $\mathbf{h}$.
This implies that $\mathbf{H}$ is a rotation matrix, and we know that rotation matrices are orthogonal.

Problem 3.7 To prove these results, it is convenient to regard a matrix $\mathbf{J}_n$ whose elements are all 1 as the product

$$\mathbf{1}_n\mathbf{1}_n^T$$

where $\mathbf{1}_n = [1, 1, 1, \ldots, 1]^T \in \mathbb{R}^n$ is a column vector of ones. Then

$$\begin{aligned}
\mathbf{x}^T\mathbf{C} &= \mathbf{x}^T\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right) \\
&= \mathbf{x}^T\mathbf{I}_n - \frac{1}{n}\mathbf{x}^T\mathbf{1}_n\mathbf{1}_n^T \\
&= \mathbf{x}^T - \left(\frac{1}{n}\sum_{k=1}^{n} x_k\right)\mathbf{1}_n^T \\
&= [x_1, x_2, x_3, \ldots, x_n]^T - [\bar{x}, \bar{x}, \bar{x}, \ldots, \bar{x}]^T \\
&= [x_1 - \bar{x}, x_2 - \bar{x}, x_3 - \bar{x}, \ldots, x_n - \bar{x}]^T
\end{aligned}$$

where we exploit the fact

$$\mathbf{x}^T\mathbf{1}_n = \sum_{k=1}^{n} x_k = n\bar{x}$$

As to the second statement, we recall that

$$\sum_{k=1}^{n}(x_k - \bar{x})^2 = \sum_{k=1}^{n} x_k^2 - n\bar{x}^2$$

Moreover

$$\begin{aligned}
\mathbf{x}^T\mathbf{C}\mathbf{x} &= \mathbf{x}^T\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\mathbf{x} \\
&= \mathbf{x}^T\mathbf{x} - \frac{1}{n}(\mathbf{x}^T\mathbf{1}_n)(\mathbf{1}_n^T\mathbf{x}) \\
&= \sum_{k=1}^{n} x_k^2 - \frac{1}{n}\left(\sum_{k=1}^{n} x_k\right)\left(\sum_{k=1}^{n} x_k\right) \\
&= \sum_{k=1}^{n} x_k^2 - n\bar{x}^2
\end{aligned}$$

Problem 3.8 The determinant of a diagonal matrix $\mathbf{D}$ may be developed by rows; since only one element in the first row is nonzero, we have

$$\det(\mathbf{D}) = \begin{vmatrix} d_1 & & & & \\ & d_2 & 0 & & \\ & & d_3 & & \\ & & & \ddots & \\ & & & & d_n \end{vmatrix} = d_1 \cdot \begin{vmatrix} d_2 & 0 & & \\ & d_3 & & \\ & & \ddots & \\ & & & d_n \end{vmatrix}$$

Repeating the scheme recursively, we have

$$\det(\mathbf{D}) = \prod_{j=1}^{n} d_j$$

The reasoning is the same for a lower triangular matrix, whereas for an upper triangular triangular it is convenient to start from the last row:

$$\det(\mathbf{U}) = \begin{vmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1,n-1} & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2,n-1} & u_{2n} \\ & & u_{33} & \cdots & u_{3,n-1} & u_{3n} \\ & & & \ddots & \vdots & \vdots \\ & & & & u_{n-1,n-1} & u_{n-1,n} \\ & & & & & u_{nn} \end{vmatrix} = u_{nn} \cdot \begin{vmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1,n-1} \\ & u_{22} & u_{23} & \cdots & u_{2n-1} \\ & & u_{33} & \cdots & u_{3,n-1} \\ & & & \ddots & \vdots \\ & & & & u_{n-1,n-1} \end{vmatrix}$$

Going on recursively, we find

$$\det(\mathbf{U}) = \prod_{j=1}^{n} u_{jj}$$

Problem 3.9 Of course we may apply the standard approach based on minors, but sometimes shortcuts are possible. For instance, the case of a diagonal matrix is easy:

$$\mathbf{A}_1^{-1} = \begin{bmatrix} \frac{1}{6} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & -\frac{1}{5} \end{bmatrix}$$

To deal with the second case, observe that, if denote the generic element of $\mathbf{A}_2^{-1}$ by b_{ij}, we must have $\mathbf{A}_2\mathbf{A}_2^{-1} = \mathbf{I}$, i.e.,

$$\begin{bmatrix} 0 & 0 & 5 \\ 0 & 2 & 0 \\ 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} 5b_{31} & 5b_{32} & 5b_{33} \\ 2b_{21} & 2b_{22} & 2b_{23} \\ 3b_{11} & 3b_{12} & 3b_{13} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If we equate element by element, we find

$$5b_{31} = 1, \quad 2b_{22} = 1, \quad 3b_{13} = 1$$

whereas all the remaining elements of the inverse are zero. Hence

$$\mathbf{A}_2^{-1} = \begin{bmatrix} 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ \frac{1}{5} & 0 & 0 \end{bmatrix}$$

With respect to the diagonal case, we have a permutation of elements.
For the last case, let us apply the standard procedure. The first step is easy:

$$\det(\mathbf{A}_3) = 2$$

Then we must find the adjoint matrix $\tilde{\mathbf{A}}$ of $\mathbf{A}_3$. Since its element $\tilde{a}_{ij}$ is the cofactor C_{ji}, we may transpose the matrix and find a sequence of 2×2 determinants:

$$\mathbf{B} = \mathbf{A}_3^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Now, to find $\tilde{a}_{11}$ we just eliminate the first row and the first column of $\mathbf{B}$, obtaining a 2×2 determinant:

$$\tilde{a}_{11} = (-1)^{1+1} \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix} = 1$$

By a similar token:

$$\tilde{a}_{12} = (-1)^{1+2} \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = -1$$

Here we cross the first row and the second column, and change the sign to the resulting determinant. In the second row of $\tilde{\mathbf{A}}$ there is a different pattern, as the first element has a change in sign, and so on. This yields

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix}$$

and

$$\mathbf{A}_3^{-1} = \frac{1}{2}\tilde{\mathbf{A}} = \begin{bmatrix} 0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & 0.5 \end{bmatrix}$$

Problem 3.10 Let us interpret $\mathbf{Ax}$ as a linear combination of columns $\mathbf{A}_j$ of $\mathbf{A}$ with weights x_j:

$$\mathbf{Ax} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_n \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{A}_1 + x_2\mathbf{A}_2 + \cdots + x_n\mathbf{A}_n$$

If this linear combination yields the null vector $\mathbf{0}$, but $\mathbf{x} \neq \mathbf{0}$, then the columns of $\mathbf{A}$ are not linearly independent; hence, the matrix is singular.
We say that $\mathbf{x}$ is in the *null space* of $\mathbf{A}$; note that this implies that any vector $\lambda\mathbf{x}$, for any real number λ is in this null space, too. Now consider a vector $\mathbf{z}$ such that $\mathbf{Az} = \mathbf{b}$, and imagine that you wish to invert the mapping. It is easy to see that this is impossible, since

$$\mathbf{A}(\mathbf{z} + \lambda\mathbf{x}) = \mathbf{b}$$

as well, for any λ. The mapping cannot be inverted, matrix $\mathbf{A}$ cannot be inverted, and it is singular.

Problem 3.11 We should find a vector $\mathbf{x} \neq \mathbf{0}$ such that

$$(\mathbf{hh}^T - \mathbf{h}^T\mathbf{hI})\mathbf{x} = \mathbf{h}(\mathbf{h}^T\mathbf{x}) - (\mathbf{h}^T\mathbf{h})\mathbf{x} = \mathbf{0}$$

(see Problem 3.10). Note that the terms between parentheses in the above expression are actually scalars. We have

$$\mathbf{h}^T\mathbf{x} = \sum_{i=1}^n h_i x_i, \qquad \mathbf{h}(\mathbf{h}^T\mathbf{x}) = \begin{bmatrix} h_1\left(\sum_{i=1}^n h_i x_i\right) \\ h_2\left(\sum_{i=1}^n h_i x_i\right) \\ \vdots \\ h_n\left(\sum_{i=1}^n h_i x_i\right) \end{bmatrix}$$

$$\mathbf{h}^T\mathbf{h} = \sum_{i=1}^n h_i^2, \qquad (\mathbf{h}^T\mathbf{h})\mathbf{x} = \begin{bmatrix} x_1\left(\sum_{i=1}^n h_i^2\right) \\ x_2\left(\sum_{i=1}^n h_i^2\right) \\ \vdots \\ x_n\left(\sum_{i=1}^n h_i^2\right) \end{bmatrix}$$

Therefore, we have a system of linear equations with the following form:

$$h_j \left(\sum_{i=1}^{n} h_i x_i \right) - x_j \left(\sum_{i=1}^{n} h_i^2 \right), \quad j = 1, \ldots, n$$

Looking at this expression, we see that the non-zero vector

$$x_j = \frac{h_j}{\sum_{i=1}^{n} h_i^2}, \quad j = 1, \ldots, n$$

is in fact a solution of the system. Hence, the matrix is singular.

Problem 3.12 Consider two nonnull orthogonal vectors $\mathbf{x}$ and $\mathbf{y}$. If they are linearly dependent, then we may write $\mathbf{x} = \alpha \mathbf{y}$, for some real number $\alpha \neq 0$. Then, orthogonality implies

$$\mathbf{x}^T \mathbf{y} = \alpha \mathbf{y}^T \mathbf{y} = \alpha \parallel \mathbf{y} \parallel^2 = 0$$

But, given the properties of vector norms, this is possible only if $\mathbf{y} = \mathbf{0}$, which contradicts the hypotheses. Since we have a contradiction, we conclude that $\mathbf{x}$ and $\mathbf{y}$ are linearly independent.

Problem 3.13 Show that if λ is an eigenvalue of $\mathbf{A}$, then $1/(1+\lambda)$ is an eigenvalue of $(\mathbf{I} + \mathbf{A})^{-1}$.
We may prove the result in two steps:

1. If λ is an eigenvalue of $\mathbf{A}$, then $1 + \lambda$ is an eigenvalue of $\mathbf{I} + \mathbf{A}$.
2. If λ is an eigenvalue of $\mathbf{A}$, then $1/\lambda$ is an eigenvalue of $\mathbf{A}^{-1}$ (assuming that $\mathbf{A}$ is invertible, which implies that there is no eigenvalue $\lambda = 0$).

If λ is an eigenvalue of $\mathbf{A}$, it is a solution of the characteristic equation for matrix $\mathbf{A}$

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

Now consider the characteristic equation for matrix $\mathbf{I} + \mathbf{A}$

$$\det(\mathbf{A} + \mathbf{I} - \mu \mathbf{I}) = 0$$

Say that μ solves the equation, which we may rewrite as follows

$$\det(\mathbf{A} + (1 - \mu)\mathbf{I}) = 0$$

which implies that $1 - \mu = \lambda$ is an eigenvalue of $\mathbf{A}$ or, in other words, that $\mu = 1 + \lambda$ is an eigenvalue of $\mathbf{I} + \mathbf{A}$.
By a similar token, let us consider the characteristic equation for $\mathbf{A}^{-1}$:

$$\det(\mathbf{A}^{-1} - \mu \mathbf{I}) = 0$$

We know that $\det(AB) = \det(A)\det(B)$, for square matrices $\mathbf{A}$ and $\mathbf{B}$. Then, let us multiply the last equation by $\det(\mathbf{A})/\mu$:

$$\frac{1}{\mu} \det(\mathbf{A}) \det(\mathbf{A}^{-1} - \mu \mathbf{I}) = \det \left(\frac{1}{\mu} \mathbf{A}\mathbf{A}^{-1} - \frac{\mu}{\mu} \mathbf{A}\mathbf{I} \right) = \det \left(\frac{1}{\mu} \mathbf{I} - \mathbf{A} \right) = 0$$

We see that $\lambda = 1/\mu$ is an eigenvalue of $\mathbf{A}$, which also implies that $\mu = 1/\lambda$ is an eigenvalue of $\mathbf{A}^{-1}$.
Putting the two results together, we find that $1/(1+\lambda)$ is an eigenvalue of $(\mathbf{I} + \mathbf{A})^{-1}$.

Problem 3.14 In the book[1] we show that a symmetric matrix $\mathbf{A}$ can be factored as

$$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$$

where $\mathbf{P}$ is an orthogonal matrix ($\mathbf{P}^T = \mathbf{P}^{-1}$) whose columns are normalized eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix consisting of (real) eigenvalues. It is also easy to see that

$$\mathbf{A}^{-1} = \mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T$$

To see this, recall that of $\mathbf{B}$ and $\mathbf{B}$ are square and invertible, $(\mathbf{BC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}$, which implies

$$\mathbf{A}^{-1} = (\mathbf{P}\mathbf{\Lambda}\mathbf{P}^T)^{-1} = (\mathbf{P}^T)^{-1}\mathbf{\Lambda}^{-1}\mathbf{P}^{-1} = \mathbf{A}^{-1} = \mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T$$

Then we have

$$\mathbf{A} + \mathbf{A}^{-1} = \mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T + \mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T = \mathbf{P}(\mathbf{\Lambda} + \mathbf{\Lambda}^{-1})\mathbf{P}^T$$

from which we see that the eigenvalues of $\mathbf{A} + \mathbf{A}^{-1}$ are given by $\lambda + 1/\lambda$, where λ is an eigenvalue of $\mathbf{A}$. Now take the minimum of this expression

$$\min \lambda + \frac{1}{\lambda} \quad \Rightarrow \quad 1 - \frac{1}{\lambda^2} = 0 \quad \Rightarrow \quad \lambda^* = 1 \quad \Rightarrow \quad \lambda^* + \frac{1}{\lambda^*} = 2$$

where we use the assumption that $\lambda > 0$.
For the general case, we take

$$\mathbf{AU} = \mathbf{U\Lambda}$$

where the columns of $\mathbf{U}$ are eigenvectors. If this matrix is invertible, we have

$$\mathbf{A} = \mathbf{U\Lambda U}^{-1}$$

and we may repeat the argument. However, we must assume that the matrix $\mathbf{U}^{-1}$ is in fact invertible, which amounts to saying that $\mathbf{A}$ is *diagonable*. Furthermore, in general, eigenvalues might be complex, which introduces further complication outside the scope of the book.

Problem 3.15 Prove that, for a symmetric matrix $\mathbf{A}$, we have

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2 = \sum_{k=1}^{n} \lambda_k^2$$

where λ_k, $k = 1, \ldots, n$, are the eigenvalues of $\mathbf{A}$.
This may a rather challenging problem, which is considerably simplified if we use a rather concept, the *trace* of a square matrix. The trace of the matrix is just the sum of the elements on its diagonal:

$$\mathrm{tr}(\mathbf{A}) = \sum_{k=1}^{n} a_{kk}$$

Two important properties of the trace are:

[1]Problems 3.14 and 3.15 are taken from the book by Searle, see end of chapter references.

1. The trace is equal to the sum of the eigenvalues:

$$\mathrm{tr}(\mathbf{A}) = \sum_{k=1}^{n} \lambda_k$$

2. If we have symmetric matrices $\mathbf{B}$ and $\mathbf{C}$

$$\mathrm{tr}(\mathbf{BC}) = \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij} c_{ij}$$

   i.e., the trace of $\mathbf{BC}$ is the sum of the elementwise product of $\mathbf{B}$ and $\mathbf{C}$.

Since we have $\mathbf{A}^T = \mathbf{A}$

$$\mathrm{tr}(\mathbf{A}^T\mathbf{A}) = \mathrm{tr}(\mathbf{A}^2) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} a_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2 \tag{3.1}$$

Furthermore, the eigenvalues of $\mathbf{A}^2$ are the squared eigenvalues of $\mathbf{A}$. Indeed, if $\mathbf{u}_i$ is an eigenvector corresponding to eigenvalue λ_i of $\mathbf{A}$, we have

$$\mathbf{A}^2\mathbf{u}_i = \mathbf{A}(\mathbf{A}\mathbf{u}_i) = \lambda_i \mathbf{A}\mathbf{u}_i = \lambda_i^2 \mathbf{u}_i$$

But the trace of $\mathbf{A}^2$ is the sum of its eigenvalues

$$\mathrm{tr}(\mathbf{A}^2) = \sum_{i=1}^{n} \lambda_i^2 \tag{3.2}$$

Putting Eqs. (3.1) and (3.2) together we obtain the result.

4

Descriptive Statistics: On the Way to Elementary Probability

4.1 SOLUTIONS

Problem 4.1 You are carrying out a research about how many pizzas are consumed by teenagers, in the age range from 13 to 17. A sample of 20 boys/girls in that age range is taken, and the number of pizzas eaten per month is given in the following table:
If we work on the raw data

4	12	7	11	9	7	8	13	16	11
4	7	5	7	11	7	7	41	9	14

we obtain the mean as follows:

$$\overline{X} = \frac{4 + 12 + 7 + \cdots + 14}{20} = 10.5$$

We may also sort data

4	4	5	7	7	7	7	7	7	8
9	9	11	11	11	12	13	14	16	41

and find frequencies

X_i	4	5	7	8	9	11	12	13	14	16	41
f_i	2	1	6	1	2	3	1	1	1	1	1

and compute

$$\overline{X} = \frac{2 \times 4 + 1 \times 5 + 6 \times 7 + \cdots + 1 \times 16 + 1 \times 41}{20} = 10.5$$

Sorting data is useful to immediately spot the median:

$$m = \frac{8 + 9}{2} = 8.5$$

If we get rid of the largest observation (41), we obtain

$$\overline{X} = 8.8947, \quad m = 8$$

As we see, the median is less sensitive to outliers.
To find standard deviation:

$$\sum_{i=1}^{20} X_i^2 = 4^2 + 12^2 + \cdots + 14^2 = 3386$$
$$\overline{X}^2 = 110.25$$
$$S^2 = \frac{1}{19}(3386 - 20 \times 110.25) = 62.1579$$
$$S = \sqrt{62.1579} = 7.8840$$

Problem 4.2 The following table shows a set of observed values and their frequencies:

Value	1	2	3	4	5	6	7	8
Frequency	5	4	7	10	13	8	3	1

- Compute mean, variance, and standard deviation.
- Find the cumulated relative frequencies.

We have

$$\sum_{k=1}^{8} f_k = 5 + 4 + \cdots + 3 + 1 = 51$$

observations; the relative frequencies p_i are

0.0980, 0.0784, 0.1373, 0.1961, 0.2549, 0.1569, 0.0588, 0.0196

from which we immediately find the cumulative relative frequencies $P_k = \sum_{j=1}^{k} p_j$

0.0980, 0.1765, 0.3137, 0.5098, 0.7647, 0.9216, 0.9804, 1.0000

To find mean, variance, and standard deviation:

$$\overline{X} = \sum_{k=1}^{8} p_k X_k = 0.0980 \times 1 + 0.0784 \times 2 + \cdots + 0.0196 \times 8 = 4.2353$$
$$S^2 = \frac{1}{50}\left(\sum_{k=1}^{8} f_k X_k^2 - 51\overline{X}^2\right) = \frac{1}{50}\left(5 \times 1 + 4 \times 2^2 + \cdots + 1 \times 8^2 - 51 \times 4.2353^2\right) = 3.0635$$
$$S = \sqrt{3.0635} = 1.7503$$

Problem 4.3 First we find frequencies, relative frequencies, and cumulative frequencies:

X_k	f_k	p_k	P_k
2	3	0.1875	0.1875
3	2	0.125	0.3125
4	5	0.3125	0.625
5	2	0.125	0.75
6	3	0.1875	0.9375
8	1	0.0625	1

The mean is

$$\overline{X} = 3 \times 2 + 2 \times 3 + 5 \times 4 + \cdots + 1 \times 8 = 4.25$$

The largest frequency, 5, is associated to the value 4, which is the mode.
To compute median and quartiles, it may be convenient to just sort the data:

X_k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
f_k	2	2	2	3	3	4	4	4	4	4	5	5	6	6	6	8

The median (which is also the second quartile) is the average of data in positions 8 and 9 (i.e., 4). The first quartile is the average of data in positions 4 and 5 (i.e., 3). The third quartile is the average of data in positions 12 and 13 (i.e., 5.5).

Problem 4.4 Let us sort the observations

13.60, 14.30, 15.00, 15.20, 15.60, 16.10, 19.20, 20.10, 21.00, 21.10, 21.30, 22.20

The descriptive statistics are

$$\overline{X} = 17.89, \quad S = 3.19, \quad m = \frac{16.10 + 19.20}{2} = 17.65$$

The mean and the median are not too different, so data are not much skewed.
We have $X_{(8)} = 20.1$, i.e., that person is in position 8 out of 12. The percentile rank can be calculated as

$$\frac{b+e}{n} = \frac{7+1}{12} = 67\%$$

where b is the number of observations "before" in the rank and e is the number of "equal" observations. We may find a different value if we take another definition

$$\frac{b}{b+a} = \frac{7}{7+4} = 63.63\%$$

where a is the number of observations "after" in the rank.
The quartiles are

$$Q_1 = \frac{15.00 + 15.10}{2} = 15.05, \quad Q_2 \equiv m = 17.65, \quad Q_3 = \frac{21.00 + 21.50}{2} = 21.05$$

To check the definition of the first quartile Q_1, we observe that

1. There are at least $\frac{25 \times 12}{100} = 3$ observations less than or equal to $Q_1 = 15.05$ (i.e, 13.60 , 14.30, 15.00)
2. There are at least $\frac{(100-25)\times 12}{100} = 9$ observations larger than or equal to $Q_1 = 15.05$ (i.e, 15.20 , 15.60 , 16.10 , 19.20 , 20.10 , 21.00, 21.10 , 21.30, 22.20)

We observe that the above statements apply to both $X_(3) = 15.00$ and $X_{(4)} = 15.20$, and we take their average.
If we use linear interpolations, the order statistic $X_{(i)}$ is associated with the percentile

$$100 \times \frac{(i - 0.5)}{12}, \qquad i = 1, \ldots, 12$$

For instance, the percentile corresponding to $X_{(1)} = 13.60$ is

$$100 \times \frac{0.5}{12} = 4.1667$$

By the same token, the next percentiles are

$$
\begin{aligned}
X_{(2)} &\rightarrow 12.5000 \\
X_{(3)} &\rightarrow 20.8333 \\
X_{(4)} &\rightarrow 29.1667 \\
X_{(5)} &\rightarrow 37.5000 \\
X_{(6)} &\rightarrow 45.8333 \\
X_{(7)} &\rightarrow 54.1667 \\
X_{(8)} &\rightarrow 62.5000 \\
X_{(9)} &\rightarrow 70.8333 \\
X_{(10)} &\rightarrow 79.1667 \\
X_{(11)} &\rightarrow 87.5000 \\
X_{(12)} &\rightarrow 95.8333
\end{aligned}
$$

To find the 90% percentile, we must interpolate between $X_{(11)} = 21.30$ and $X_{(12)} = 22.20$ as follows:

$$21.30 + \frac{90.0000 - 87.500}{95.8333 - 87.5000} \times (22.20 - 21.30) = 21.57$$

Problem 4.5 The first answer is obtained by taking the ratio between how many female professors who had a hangover twice or more and the total number of female professors:

$$\frac{36}{66 + 25 + 36} = 28.35\%$$

The second answer is obtained by taking the ratio between the number of male professors who had a hangover once or less and the total number of professors (male or female) who had a hangover once or less:

$$\frac{61 + 23}{61 + 23 + 66 + 25} = 48\%$$

In Chapter 5, where we introduce the language of probability theory, we learn how to express these questions in terms of conditional probabilities.

1. The first answer can also be written as

$$\mathrm{P}\{(\geq 2)\,|\,F\} = \frac{\mathrm{P}\{(\geq 2) \cap F\}}{\mathrm{P}(F)}$$

   where (≥ 2) is the event "the professor had twice or more hangovers" and F is the event "the professor is female." Since

$$\mathrm{P}\{(\geq 2) \cap F\} = \frac{36}{251}, \quad \mathrm{P}(F) = \frac{66 + 25 + 36}{251}$$

   where 251 is the total number of professors, we obtain the above result.

2. By the same token, the second answer can also be written as

$$\mathrm{P}\{M\,|\,(\leq 1)\} = \frac{\mathrm{P}\{M \cap (\leq 1)\}}{\mathrm{P}(M)}$$

   where $(\leq 1) = (= 0) \cup (= 1)$ is the event "the professor had one or less hangover," which is the union of the event "the professor had no hangover" and "the professor had one hangover."

5

Probability Theories

5.1 SOLUTIONS

Problem 5.1 Consider two events E and G, such that $E \subseteq G$. Then prove that $\mathrm{P}(E) \leq \mathrm{P}(G)$.
We may express G as the union of E and the part of G that does not intersect with E:

$$G = E \cup (G \backslash E)$$

Then, of course the two components are disjoint:

$$E \cap (G \backslash E) = \emptyset$$

and we may apply additive probability

$$\mathrm{P}(G) = \mathrm{P}(E) + \mathrm{P}(G \backslash E) \geq \mathrm{P}(E)$$

since $\mathrm{P}(G \backslash E) \geq 0$.

Problem 5.2 From Bayes' theorem we have

$$\mathrm{P}(A \mid E) = \frac{\mathrm{P}(E \mid A)\mathrm{P}(A)}{\mathrm{P}(E)}, \quad \mathrm{P}(B \mid E) = \frac{\mathrm{P}(E \mid B)\mathrm{P}(B)}{\mathrm{P}(E)}$$

Taking ratios

$$\frac{\mathrm{P}(A \mid E)}{\mathrm{P}(B \mid E)} = \frac{\mathrm{P}(E \mid A)\mathrm{P}(A)}{\mathrm{P}(E)} \cdot \frac{\mathrm{P}(E)}{\mathrm{P}(E \mid B)\mathrm{P}(B)} = \frac{\mathrm{P}(E \mid A)}{\mathrm{P}(E \mid B)}$$

under the assumption that $\mathrm{P}(A) = \mathrm{P}(B)$.
This is an inversion formula in the following sense:

- The ratio $\frac{\mathrm{P}(E|A)\mathrm{P}(A)}{\mathrm{P}(E)}$ gives the relative likelihood of A and B given the occurrence of E.
- If we invert conditioning, we consider the probability of E given A or B, which is what we need when we may observe A or B, but not E.
- The formula states that when $\mathrm{P}(A) = \mathrm{P}(B)$ the second ratio is equal to the first one.

Problem 5.3 In this case, it is necessary to lay down all of the events and the pieces of information we have. We know that:

1. the event $\mathsf{chooseA}$ occurred, i.e., the participant selected box A
2. the event opC occurred, i.e., the presenter opened box C
3. the event notC occurred, the prize is not in box C (otherwise, the game would have stopped immediately, and the participant could not switch from box A to box B)

Hence, we need the conditional probability $\mathrm{P}(\mathsf{A} \,|\, \mathsf{chooseA} \cap \mathsf{opC} \cap \mathsf{notC})$. Using the definition of conditional probability:

$$\mathrm{P}(\mathsf{A} \,|\, \mathsf{chooseA} \cap \mathsf{opC} \cap \mathsf{notC}) = \frac{\mathrm{P}(\mathsf{A} \cap \mathsf{chooseA} \cap \mathsf{opC} \cap \mathsf{notC})}{\mathrm{P}(\mathsf{opC} \cap \mathsf{chooseA} \cap \mathsf{notC})}$$

However, the event $\mathsf{opC} \cap \mathsf{chooseA}$ is independent on the other events, as neither the participant nor the presenter has any clue, and so their behavior is not influenced by knowledge of the box containing the prize:

$$\mathrm{P}(\mathsf{A} \cap \mathsf{chooseA} \cap \mathsf{opC} \cap \mathsf{notC}) = \mathrm{P}(\mathsf{chooseA} \cap \mathsf{opC}) \cdot \mathrm{P}(\mathsf{A} \cap \mathsf{notC})$$
$$\mathrm{P}(\mathsf{opC} \cap \mathsf{chooseA} \cap \mathsf{notC}) = \mathrm{P}(\mathsf{chooseA} \cap \mathsf{opC}) \cdot \mathrm{P}(\mathsf{notC})$$

Therefore

$$\mathrm{P}(\mathsf{A} \,|\, \mathsf{chooseA} \cap \mathsf{opC} \cap \mathsf{notC}) = \frac{\mathrm{P}(\mathsf{A} \cap \mathsf{notC})}{\mathrm{P}(\mathsf{notC})} = \frac{\mathrm{P}(\mathsf{A})}{\mathrm{P}(\mathsf{notC})} = \frac{1/3}{2/3} = \frac{1}{2}$$

where we use the fact $\mathsf{A} \subseteq \mathsf{notC}$. Hence, the participant has no incentive to switch, as he cannot squeeze any useful information out of the presenter's behavior.

5.2 ADDITIONAL PROBLEMS

Problem 5.4 Each of two cabinets identical in appearance has two drawers. Cabinet A contains a silver coin in each drawer; cabinet B contains a silver coin in one of its drawers and a gold coin in the other. A cabinet is randomly selected, one of its drawers is opened, and a silver coin is found. What is the probability that there is a silver coin in the other drawer?

Problem 5.5 Disgustorama is a brand new food company producing cakes. Cakes may suffer from two defects: incomplete leavening (which affects rising) and/or excessive baking. The two defects are the result of two operations carried out at different stages, so they can be considered independent. Quality is checked only before packaging. We know that the first defect (bad leavening) occurs with probability 7% and the second one with probability 3%. Find:

- The probability that a cake is both dead flat and burned
- The probability that a cake is defective (i.e., it has at least one defect)
- The probability that a cake is burned, if we know that it is defective

5.3 SOLUTIONS OF ADDITIONAL PROBLEMS

Problem 5.4 One possible approach is based on Bayes' theorem. Let A be the event "we have picked cabinet A"; B is the event "we have picked cabinet B". Since we select the cabinet purely at random, a priori $\mathrm{P}(\mathsf{A}) = \mathrm{P}(\mathsf{B}) = 0.5$.
Now we have some additional information, and we should revise our belief by finding the conditional probabilities $\mathrm{P}(\mathsf{A} \mid \mathsf{S}_1)$ and $\mathrm{P}(\mathsf{B} \mid \mathsf{S}_1)$, where S_1 is the event "the first coin is silver".
Bayes' theorem yields:

$$\mathrm{P}(\mathsf{A}\,|\,\mathsf{S}_1) = \frac{\mathrm{P}(\mathsf{S}_1\,|\,\mathsf{A})\mathrm{P}(\mathsf{A})}{\mathrm{P}(\mathsf{S}_1)} = \frac{\mathrm{P}(\mathsf{S}\,|\,\mathsf{A})\mathrm{P}(\mathsf{A})}{\mathrm{P}(\mathsf{S}_1\,|\,\mathsf{A})\mathrm{P}(\mathsf{A}) + \mathrm{P}(\mathsf{S}_1\,|\,\mathsf{B})\mathrm{P}(\mathsf{B})} = \frac{1 \times 0.5}{1 \times 0.5 + 0.5 \times 0.5} = \frac{2}{3}$$

By the way, it may be useful to observe that $\mathrm{P}(\mathsf{S}_1) = 0.75$, which may sense, as there are 3 silver coins and 1 gold coin. Then

$$\mathrm{P}(\mathsf{B}\,|\,\mathsf{S}_1) = 1 - \mathrm{P}(\mathsf{A}\,|\,\mathsf{S}_1) = \tfrac{1}{3}.$$

Now we may calculate

$$\mathrm{P}(\mathsf{S}_2\,|\,\mathsf{S}_1) = \mathrm{P}(\mathsf{S}_2\,|\,\mathsf{A}) \cdot \mathrm{P}(\mathsf{A}\,|\,\mathsf{S}_1) + \mathrm{P}(\mathsf{S}_2\,|\,\mathsf{B}) \cdot \mathrm{P}(\mathsf{B}\,|\,\mathsf{S}_1) = 1 \times \tfrac{2}{3} + 0 \times \tfrac{1}{3} = \tfrac{2}{3}$$

The above solution is unnecessarily contrived, but it is a good illustration of a general framework based on the revision of unconditional probabilities. In our case, we know that the second coin is silver only if we picked cabinet A, so

$$\mathrm{P}(\mathsf{S}_2\,|\,\mathsf{S}_1) = \mathrm{P}(\mathsf{A}\,|\,\mathsf{S}_1) = \tfrac{2}{3}$$

A quite straightforward solution is found by applying the definition of conditional probability:

$$\mathrm{P}(\mathsf{S}_2\,|\,\mathsf{S}_1) = \frac{\mathrm{P}(\mathsf{S}_2 \cap \mathsf{S}_1)}{\mathrm{P}(\mathsf{S}_1)} = \frac{\mathrm{P}(\mathsf{A})}{\mathrm{P}(\mathsf{S}_1)} = \frac{0.5}{0.75} = \frac{2}{3}$$

This is a smarter solution, even though it works only in this peculiar case and misses the more general style of reasoning.

Problem 5.5 Let A_1 be the event "incomplete leavening" and let Let A_2 be the event "excessive baking":

$$\mathrm{P}(A_1) = 0.07, \quad \mathrm{P}(A_2) = 0.03$$

By assumption, these events are independent.

1. We want $\mathrm{P}(A_1 \cap A_2)$. Because of independence,

$$\mathrm{P}(A_1 \cap A_2) = \mathrm{P}(A_1) \cdot \mathrm{P}(A_2) = 0.07 \times 0.03 = 0.0021$$

2. We want $\mathrm{P}(A_1 \cup A_2)$. Note that the two events are not disjoint (if they were, they could *not* be independent!). Hence

$$\mathrm{P}(A_1 \cup A_2) = \mathrm{P}(A_1) + \mathrm{P}(A_2) - \mathrm{P}(A_1 \cap A_2) = 0.07 + 0.03 - 0.0021 = 0.0979$$

3. We apply Bayes' theorem to find the conditional probability

$$\mathrm{P}(A_2\,|\,A_1 \cup A_2) = \frac{\mathrm{P}(A_1 \cup A_2\,|\,A_2) \cdot \mathrm{P}(A_2)}{\mathrm{P}(A_1 \cup A_2)} = \frac{\mathrm{P}(A_2)}{\mathrm{P}(A_1 \cup A_2)} = \frac{0.03}{0.0979} = 0.3064$$

After all, this solution is rather intuitive if we interpret probabilities as relative frequencies; however, we prefer to use a more general and sound reasoning. Also note that, in this case, we do not apply the usual theorem of total probability to find the denominator of the ratio, as A_1 and S_2 are not independent.

6

Discrete Random Variables

6.1 SOLUTIONS

Problem 6.1 Clearly, variance is minimized when $p = 0$ or $p = 1$, i.e., when there is no variability at all. In Chapter 6, we show that when $x_1 = 1$ and $x_2 = 0$, i.e., when we deal with a standard Bernoulli variable, its variance is maximized for $p = 0.5$. Here we wonder whether this applies to general values x_1 and x_2 as well.
One possible approach is to repeat the drill, write variance explicitely, and maximize it with respect to p:

$$\begin{aligned}
\mathrm{E}[X] &= px_1 + (1-p)x_2 = p(x_1 - x_2) + x_2 \\
\mathrm{E}^2[X] &= p^2(x_1 - x_2)^2 + 2p(x_1 - x_2)x_2 + x_2^2 \\
\mathrm{E}[X^2] &= px_1^2 + (1-p)x_2^2 = p(x_1^2 - x_2^2) + x_2^2 \\
\mathrm{Var}(X) &= p(x_1^2 - x_2^2) + x_2^2 - p^2(x_1 - x_2)^2 - 2p(x_1 - x_2)x_2 - x_2^2 \equiv \sigma^2(p)
\end{aligned}$$

Applying the first-order condition

$$\frac{d\sigma^2(p)}{dp} = x_1^2 - x_2^2 - 2p(x_1 - x_2)^2 - 2(x_1 - x_2)x_2 = 0$$

we find

$$p = \frac{x_1^2 - x_2^2 - 2x_1x_2 + 2x_2^2}{2p(x_1 - x_2)^2} = \frac{(x_1 - x_2)^2}{2p(x_1 - x_2)^2} = 0.5$$

which is the same result as the standard Bernoulli variable. This is not surprising after all and can be obtained by a more straightforward approach. Let us consider random variable

$$Y = \frac{X - x_2}{x_1 - x_2}$$

We see that when $X = x_1$, $Y = 1$, and when $X = x_2$, $Y = 0$. Hence, Y is the standard Bernoulli variable, whose variance is maximized for $p = 0.5$. But by recalling the properties of variance we see that

$$\mathrm{Var}(Y) = \frac{\mathrm{Var}(X)}{(x_1 - x_2)^2}$$

Since the two variances differ by a positive constant, they are both maximized for the same value of p.

Problem 6.2 A much easier and insightful proof will be given in Chapter 8, based on conditioning, but if we observe the expected value of the geometric random variable

$$\mathrm{E}[X] = \sum_{i=1}^{\infty} i(1-p)^{i-1}p = pS_1$$

we see that the sum

$$S_1 \equiv \sum_{i=1}^{\infty} i(1-p)^{i-1}$$

looks like the derivative of a geometric series. More precisely, if we let

$$S(p) \equiv \sum_{i=0}^{\infty} (1-p)^i = \frac{1}{1-(1-p)} = \frac{1}{p}$$

term-by-term differentiation yields

$$S'(p) = -\sum_{i=0}^{\infty} i(1-p)^{i-1} = -\sum_{i=1}^{\infty} i(1-p)^{i-1} = -S_1$$

Then

$$S_1 = -\frac{d}{dp}\left(\frac{1}{p}\right) = \frac{1}{p^2}$$

and

$$\mathrm{E}[X] = p\frac{1}{p^2} = \frac{1}{p}$$

Problem 6.3 The binomial expansion formula is

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{n-k} a^{n-k}b^k$$

and we want to prove that

$$\sum_{k=1}^{n} p_k = 1$$

where

$$p_k \equiv \mathrm{P}(X=k) = \binom{n}{k} p^k(1-p)^{n-k}$$

Now it is easy to see that

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} = \frac{n!}{(n-k)![n-(n-k)]!} = \binom{n}{n-k}$$

Hence

$$\sum_{k=1}^{n} p_k = \sum_{k=0}^{n} \binom{n}{n-k} p^k(1-p)^{n-k} = (1-p+p)^n = 1$$

Problem 6.4 Measuring profit in \$ millions, if the probability of success is p, expected profit is

$$p \times 16 - (1 - p) \times 5 = p \times 21 - 5$$

If we launch the product immediately, then $p = 0.65$ and expected profit is \$8.65 million. If we delay, expected profit is, taking the additional cost and the time discount into account

$$-1 + \frac{p \times 21 - 5}{1.03}$$

We are indifferent between the two options if

$$-1 + \frac{p \times 21 - 5}{1.03} = 8.65 \quad \Rightarrow \quad p = \frac{(1 + 8.65) \times 1.03 + 5}{21} = 0.7114$$

Then, the minimal improvement in success probability is

$$\Delta = 0.7114 - 0.65 \approx 6.14\%$$

Problem 6.5 If we assume that the persons we test are independent (i.e., we do not consider groups of heavy drinkers...), we are dealing with a binomial random variable with parameters $p = 0.4$ and $n = 25$:

$$\begin{aligned}
\mathrm{P}(X \geq 4) &= 1 - \mathrm{P}(X \leq 3) \\
&= 1 - [\mathrm{P}(X = 0) + \mathrm{P}(X = 1) + \mathrm{P}(X = 2) + \mathrm{P}(X = 3)] \\
\mathrm{P}(X = 0) &= 0.6^{25} = 2.8430 \cdot 10^{-6} \\
\mathrm{P}(X = 1) &= \binom{25}{1} 0.4^1 \times 0.6^{24} = 25 \times 0.4^1 \times 0.6^{24} = 4.7384 \cdot 10^{-5} \\
\mathrm{P}(X = 2) &= \binom{25}{2} 0.4^2 \times 0.6^{23} = \frac{25 \times 24}{2} \times 0.4^2 \times 0.6^{23} = 3.7907 \cdot 10^{-4} \\
\mathrm{P}(X = 3) &= \binom{25}{3} 0.4^3 \times 0.6^{22} = \frac{25 \times 24 \times 23}{2 \times 3} \times 0.4^3 \times 0.6^{22} = 0.001937
\end{aligned}$$

which yields

$$\mathrm{P}(X \geq 4) = 0.99763$$

Problem 6.6 Assuming that batteries in a package are independent (which is in fact a debatable assumption), we are dealing with a binomial random variable with parameters $p = 0.02$ and $n = 8$. The probability that a package is returned is

$$P_{\mathsf{bad}} = 1 - \mathrm{P}(X = 0) - \mathrm{P}(X = 1) = 1 - 0.98^8 - 8 \times 0.02 \times 0.08^7 = 0.010337$$

If the consumer buys three packages, the number of returned packages is binomial with parameters $p = P_{\mathsf{bad}}$ and $n = 3$:

$$\mathrm{P}(Y = 1) = 3 \times P_{\mathsf{bad}} \times (1 - P_{\mathsf{bad}})^2 = 0.030379$$

7

Continuous Random Variables

7.1 SOLUTIONS

Problem 7.1 We want to find $\mathrm{P}(X > 200)$, where $X \sim \mathcal{N}(250, 40^2)$. Using standardization:

$$\begin{aligned}
\mathrm{P}(X > 200) &= \mathrm{P}\left(\frac{X-250}{40} > \frac{200-250}{40}\right) \\
&= \mathrm{P}\left(Z > -1.25\right)
\end{aligned}$$

where Z is standard normal. Depending on the tool you have at hand, there are different ways of evaluating $\mathrm{P}\left(Z > -1.25\right)$:

- If you have software for evaluating the CDF $\Phi(z) = \mathrm{P}(Z \le z)$

$$\mathrm{P}\left(Z > -1.25\right) = 1 - \Phi(-1.25) = 1 - 0.1056 = 0.8944$$

- If you have statistical tables, you will typically find $\Phi(z)$ only for nonnegative values of z; in such a case, take advantage of the symmetry of the PDF of the standard normal:

$$\mathrm{P}\left(Z > -1.25\right) = \mathrm{P}\left(Z \ge 1.25\right) = \Phi(1.25) = 0.8944$$

Whatever you do, check the sensibility of your result! Since 200 is smaller than the expected value 250, we must find a probability that is larger than 0.5.

Problem 7.2 Using standardization again:

$$\begin{aligned}
\mathrm{P}(230 \le X \le 260) &= \mathrm{P}\left(\frac{230-250}{20} \le \frac{X-250}{20} \le \frac{260-250}{20}\right) \\
&= \mathrm{P}\left(-1 \le Z \le 0.5\right) \\
&= \Phi(0.5) - \Phi(-1) \\
&= 0.6915 - 0.1587 = 0.5328
\end{aligned}$$

The considerations we stated for problem 1.1 apply here as well.

Problem 7.3 We should set the reorder point R for an item, whose demand during lead time is uncertain. We have a very rough model of uncertainty – the lead time demand is uniformly distributed between 5000 and 20000 pieces. Set the reorder point in such a way that the service level is 95%.
In this case, we must find a quantile from the uniform distribution on $[5000, 20000]$. Such a quantile can be easily obtained by finding a value "covering" 95% of the interval:

$$R = 5000 + (20000 - 5000) \times 0.95 = 19250$$

Problem 7.4 You are working in your office, and you would like to take a very short nap, say, 10 minutes. However, every now and then, your colleagues come to your office to ask you for some information; the interarrival time of your colleagues is exponentially distributed with expected value 15 minutes. What is the probability that you will not be caught asleep and reported to you boss?
The time until the next visit by a colleague is an exponentially distributed random variable X with rate $\lambda = 1/15$. The required probability is

$$\mathrm{P}(X \le 10)$$

that may be obtained by recalling the CDF of the exponential distribution, $F_X(x) = 1 - e^{-\lambda x}$. In our case

$$\mathrm{P}(X \le 10) = 1 - e^{-\frac{10}{15}} = 0.4866$$

As an equivalent approach, we may integrate the PDF $f_X(x) = \lambda e^{-\lambda x}$:

$$\mathrm{P}(X \le 10) = \int_0^{10} \tfrac{1}{15} e^{-\frac{x}{15}} dx = -e^{-\frac{x}{15}}\Big|_0^{10} = -e^{-\frac{10}{15}} - (-e^{-\frac{0}{15}}) = 0.4866$$

Problem 7.5 We know that virtually all of the realizations of a normal variable are in the interval $(\mu - 3\sigma, \mu + 3\sigma)$. Since we are representing demand (which is supposed to be positive, unless you are *very bad* with marketing), we should make sure that the probability of a negative demand is a negligible modeling error.
In our case, $12000 - 3 \times 7000 = -9000$; hence, there is a nonnegligible probability of negative demand, according to the normal model. Indeed:

$$\mathrm{P}(D \le 0) = \mathrm{P}\left(Z \le \frac{0 - 12000}{7000}\right) = \Phi(-1.7143) = 4.32\%$$

Therefore, the model does not look quite reasonable. Please note that it is quite plausible that demand has expected value 12000 and standard deviation 7000; we are just observing that it cannot be normally distributed; for instance, it could well be skewed to the right.

Problem 7.6 Let X be a normal random variable with expected value $\mu = 6$ and standard deviation $\sigma = 1$. Consider random variable $W = 3X^2$. Find the expected value $\mathrm{E}[W]$ and the probability $\mathrm{P}(W > 120)$.
We recall that $\mathrm{Var}(X) = \mathrm{E}[X^2] - \mathrm{E}^2[X]$. Then

$$\mathrm{E}[W] = \mathrm{E}[3X^2] = 3(\mathrm{Var}(X) + \mathrm{E}^2[X]) = 3 \times (1^2 + 6^2) = 111$$

The probability is a little trickier because of a potential trap: It is *tempting* to write

$$\mathrm{P}(W > 120) = \mathrm{P}(3X^2 > 120) = \mathrm{P}(X > \sqrt{40})$$

where we take the square root of both sides of inequality. but this is not quite correct, as we are not considering the possibility of large negative values of X. Indeed, $\sqrt{X^2} = \mid X \mid$. Hence

$$\mathrm{P}(W > 120) = \mathrm{P}(X > \sqrt{40}) + \mathrm{P}(X < -\sqrt{40}).$$

Using the familiar drill:

$$\mathrm{P}(X > \sqrt{40}) = 1 - \Phi\left(\frac{6.3246 - 6}{1}\right) = 0.3728$$
$$\mathrm{P}(X < -\sqrt{40}) = \Phi\left(\frac{-6.3246 - 6}{1}\right) = \Phi(-12.3246) \approx 0$$

Hence, $\mathrm{P}(W > 120) = 37.28\%$. In this *lucky* case, the above error would be inconsequential.

Problem 7.7 Let us denote the inventory level after replenishment by I, and the demand of customer i by D_i. We know that $I \sim \mathcal{N}(400, 40^2)$, $\mathrm{E}[D_i] = 3$, and $\mathrm{Var}(D_i) = 0.3^2$. The total demand, if we receive N customer orders, is

$$D = \sum_{i=1}^{N} D_i$$

If we assume that customer demands are independent, and N is large enough, the central limit theorem states that

$$D \sim \mathcal{N}(3N, 0.3^2 N)$$

The probability of a stockout is given by

$$\mathrm{P}\left(D > I\right)$$

and we should find N such that

$$\mathrm{P}\left(\sum_{i=1}^{N} D_i > I\right) > 0.1$$

or, equivalently

$$\mathrm{P}\left(Y_N \leq 0\right) < 0.9$$

where $Y_N = \sum_{i=1}^{N} D_i - I$.
We know that Y_N is normal, and we need its parameters:

$$\mu_N = 3N - 400$$
$$\sigma_N = \sqrt{0.3^2 N + 40^2}$$

To find N, which is integer, let us allow for real values and find n such that

$$\mathrm{P}\left(Y_n \leq 0\right) = 0.9$$

which means that the 90% quantile of Y_n should be 0. Since $z_{0.9} = 1.2816$, we must solve the equation:

$$(3n - 400) + 1.2816\sqrt{0.3^2 n + 40^2} = 0$$

which is equivalent to

$$8.8522n^2 - 2400n + 157372.20 = 0$$

whose solutions are

$$n_1 = 116.19, \quad n_2 = 150.49$$

Hence, taking the smaller root and rounding it up, we find $N = 117$.

Problem 7.8 We know that a chi-square variable with n degrees of freedom is the sum of n independent standard normals squared:

$$X = \sum_{i=1}^{n} Z_i^2$$

Then

$$\mathrm{E}[X] = \sum_{i=1}^{n} \mathrm{E}[Z_i^2] = n\mathrm{E}[Z^2] = n(\mathrm{Var}(Z) + \mathrm{E}^2[Z]) = n,$$

since the expected value of the standard normal Z is 0 and its variance is 1.

Problem 7.9 This is a little variation on the classical newsvendor's problem. Usually, we are given the economics (profit margin m and cost of unsold items c_u), from which we obtain the optimal service level β, and then the order quantity Q. Here we go the other way around:

$$Q = \mu + z_\beta \sigma \quad \Rightarrow \quad z_\beta = \frac{Q - \mu}{\sigma} = \frac{15000 - 10000}{2500} = 2$$

From the CDF of the standard normal we find

$$\beta = \Phi(2) = 0.9772 = \frac{m}{m + c}$$

In our case $m = 14 - 10 = 4$ and $c_u = 10 - b$, where b is the buyback price. Hence

$$\frac{4}{4 + (10 - b)} = 0.9772 \quad \Rightarrow \quad b = 14 - \frac{4}{0.9772} = 9.9069$$

We see that the manufacturer should bear the whole risk, which is not surprising since she requires a very high service level (97.72%).

Problem 7.10

- If the probability that the competitor enters the market is assumed to be 50%, how many items should you order to maximize expected profit? (Let us assume that selling prices are the same in both scenarios.)
- What if this probability is 20%? Does purchased quantity increase or decrease?

The service level must be

$$\beta = \frac{m}{m + c_u} = \frac{16 - 10}{(16 - 10) + (10 - 7)} = \frac{2}{3}$$

The only difficulty in this problem is figuring out the PDF of the demand, which is given in Fig. 7.1.
Roughly speaking, we have two scenarios:

1. If the competition is strong, the PDF is the uniform distribution on the left
2. If the competition is weak, the PDF is the uniform distribution on the right

Since the probability of strong competition is 50%, the two areas are the same, which implies that the value of the density is 1/2000. Note that the two pieces do not overlap, which makes

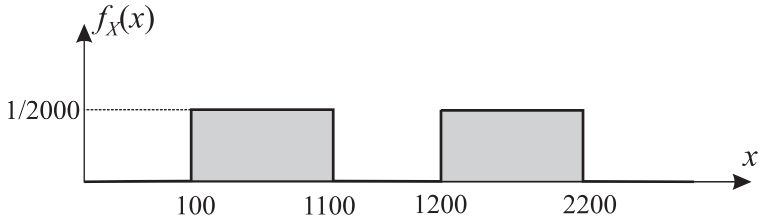


Fig. 7.1 PDF of demand in Problem 7.10.

the reasoning easy. Since $\beta > 0.5$, we must take a quantile in the second range, between 1200 and 2200: the area within the second rectangle must be

$$\tfrac{2}{3} - 0.5 = \tfrac{1}{6}$$

and we must be careful, as the area in this second rectangle is only 0.5 (the PDF is 1/2000 and not 1/1000):

$$Q = 1200 + \tfrac{1}{6} \times (2200 - 1200) \times \tfrac{1}{0.5} \approx 1533$$

In the second case, the reasoning is quite similar, but now the rectangle on the left has "weight" 0.2, rather than 0.8:

$$Q = 1200 + (\tfrac{2}{3} - 0.2) \times (2200 - 1200) \times \tfrac{1}{0.8} \approx 1783$$

A couple of observations are in order:

1. A common mistake is finding the optimal order quantities in the two scenarios and then taking their average. This is conceptually wrong, as it amounts to

   - finding the optimal solution *assuming that we know what the competition is going to do*
   - taking the average of the two optimal solutions for the two alternative scenarios

   This is not correct, as we have to make a decision *before* we discover what competitors choose.

2. We have found the PDF of the demand a bit informally, which worked well as the two intervals are disjoint. A sounder reasoning is based on finding the CDF first, by applying the total probability theorem and conditioning with respect to the competition level (strong or weak):

   $$F_X(x) \equiv \mathrm{P}(D \le x) = 0.5 \times \mathrm{P}(D \le x \,|\, \mathsf{strong}) + 0.5 \times \mathrm{P}(D \le x \,|\, \mathsf{weak})$$

   But

   $$\mathrm{P}(D \le x \,|\, \mathsf{strong}) = \begin{cases} 0 & x < 100 \\ \frac{x-100}{1000} & 0 \le x \le 1100 \\ 1 & x > 1100 \end{cases}$$

   $$\mathrm{P}(D \le x \,|\, \mathsf{weak}) = \begin{cases} 0 & x < 1200 \\ \frac{x-1200}{1000} & 1200 \le x \le 2200 \\ 1 & x > 2200 \end{cases}$$

   Adding everything up, we obtain the CDF in Fig. 7.2. Taking its derivative, we find the PDF above.

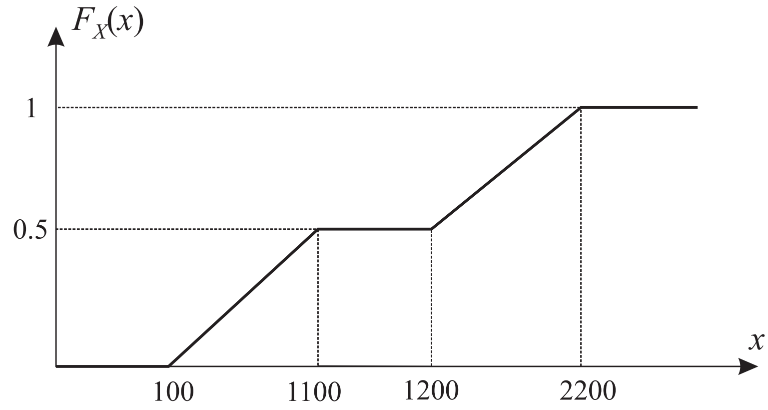


Fig. 7.2 CDF of demand in Problem 7.10.

Problem 7.11 Let us write the CDF $F_X(x)$:

$$\begin{aligned} F_X(x) &\equiv \mathrm{P}(X \le x) \\ &= \mathrm{P}(\max\{U_1, U_2, \ldots, U_n\} \le x) \\ &= \mathrm{P}(U_1 \le x, U_2 \le x, \ldots, U_n \le x) \\ &= \mathrm{P}(U_1 \le x) \cdot \mathrm{P}(U_2 \le x) \cdots \mathrm{P}(U_n \le x) \end{aligned}$$

where we take advantage of independency among the random variables U_i, $i = 1, \ldots, n$. We also know that all of them are uniform on the unit interval, and $\mathrm{P}(U \le x) = x$, for $x \in [0, 1]$. Therefore

$$F_X(x) = [\mathrm{P}(U \le x)]^n = x^n$$

Observation: As you see, the statement of the problem is not quite correct, as $F_X(x) = x^n$ for $x \in [0, 1]$, whereas $F_X(x) = 1$ for $x > 1$ and $F_X(x) = 0$ for $x < 0$.

8

Dependence, Correlation, and Conditional Expectation

8.1 SOLUTIONS

Problem 8.1 To solve the problem we need to characterize the aggregate demand we see at the central warehouse:

$$D_C = D_1 + D_2$$

where D_1 and D_2 are not independent, since they are affected by a common risk factor X. Rather than computing their covariance, since we have independent factors X, ϵ_1, and ϵ_2, it is much easier to rewrite aggregate demand as

$$D_C = (100 + 120)X + \epsilon_1 + \epsilon_2$$

This demand is normal, with expected value

$$\mu_C = 220\mu_X + \mu_1 + \mu_2 = 220 \times 28 + 200 + 300 = 6660$$

and standard deviation

$$\sigma_C = \sqrt{220^2\sigma_X^2 + \sigma_1^2 + \sigma_2^2} = \sqrt{220^2 \times 16 + 100 + 150} = 880.14$$

The inventory level should be chosen as

$$Q = \mu_C + z_{0.95}\sigma_C = 6660 + 1.6449 \times 880.14 \approx 8108$$

If the two specific factors are positively correlated, standard deviation is larger:

$$\sigma_C = \sqrt{220^2\sigma_X^2 + \sigma_1^2 + \sigma_2^2 + 2\rho_{1,2}\sigma_1\sigma_2}$$

This implies a larger stocking level, needed to hedge against some more uncertainty; however, with these numbers, the effect would be negligible, as the most variability comes from the common factor X.

Problem 8.2 We just need to find the distribution of the demand D_C for component C, which depends on demands D_1 and D_2 for end items P_1 and P_2, respectively:

$$D_C = 2D_1 + 3D_2$$

Since D_1 and D_2 are normal, so is D_C, and its expected value and standard deviation are:

$$\begin{aligned}\mu_C &= \mathrm{E}[D_C] = 2\mathrm{E}[D_1] + 3\mathrm{E}[D_2] = 2 \times 1000 + 3 \times 700 = 4100\\ \sigma_C &= \sqrt{\mathrm{Var}(D_C)} = \sqrt{4\mathrm{Var}(D_1) + 9\mathrm{Var}(D_2)}\\ &= \sqrt{4 \times 250^2 + 9 \times 180^2} = 735.9348\end{aligned}$$

The inventory level is the 92% quantile of this distribution:

$$Q = \mu_C + z_{0.92}\sigma_C = 4100 + 1.4051 \times 735.9348 = 5134.04$$

Problem 8.3 Let R_{IFM} and R_{PM} be the rates of return from the two stocks, over one day. Loss is

$$L = -(10000 \times R_{\mathrm{IFM}} + 20000 \times R_{\mathrm{PM}})$$

where the sign is inconsequential as we assume that expected return over one day is 0% and the normal distribution is symmetric. We need the standard deviation of the distribution of loss:

$$\begin{aligned}\sigma_L &= \sqrt{(10000\sigma_{\mathrm{IFM}})^2 + (20000\sigma_{\mathrm{PM}})^2 + 2\rho \times 10000\sigma_{\mathrm{IFM}} \times 20000\sigma_{\mathrm{PM}}}\\ &= \sqrt{(10000 \times 0.02)^2 + (20000 \times 0.04)^2 + 2 \times 0.68 \times 10000 \times 0.02 \times 20000 \times 0.04}\\ &= 947.42\end{aligned}$$

Since $z_0.95 = 1.6449$:

$$\mathrm{VaR}_{0.95} = 1.6449 \times 947.42 = \$1558.36$$

A little discussion. Note that for each individual position we have:

$$\begin{aligned}\mathrm{VaR}_{0.95,\mathrm{IFM}} &= 1.6449 \times 10000 \times 0.02 = \$328.97\\ \mathrm{VaR}_{0.95,\mathrm{PM}} &= 1.6449 \times 20000 \times 0.04 = \$1315.88\end{aligned}$$

We see that

$$\mathrm{VaR}_{0.95,\mathrm{IFM}} + \mathrm{VaR}_{0.95,\mathrm{PM}} = 328.97 + 1315.88 = 1644.85 > 1558.36$$

The sum of the two individual risks does exceed the joint risk, but not by much, since there is a fairly strong positive correlation. In real life, correlations do change dramatically when markets crash, going either to +1 or to -1. So, the power of diversification must not be overstated when managing tail risk, not to mention the fact that normal distribution do not feature the negative skewness and excess kurtosis that we observe in actual data.

Problem 8.4 **Note:** There is an error in the statement of the problem, as the two variables X and Y must be *identically distributed* but not necessarily independent.
One way of proving the claim is by taking advantage of the distributive property of covariance:

$$\begin{aligned}\mathrm{Cov}(X-Y, X+Y) &= \mathrm{Cov}(X, X+Y) - \mathrm{Cov}(Y, X+Y)\\ &= \mathrm{Cov}(X,X) + \mathrm{Cov}(X,Y) - \mathrm{Cov}(Y,X) - \mathrm{Cov}(Y,Y)\\ &= \mathrm{Var}(X) - \mathrm{Var}(Y) = 0\end{aligned}$$

since the two variances are the same.
Alternatively, we may rewrite covariance as follows:

$$\begin{aligned}\mathrm{Cov}(X-Y, X+Y) &= \mathrm{E}[(X-Y)(X+Y)] - \mathrm{E}[X-Y]\mathrm{E}[X+Y] \\ &= \mathrm{E}[X^2-Y^2] - \mathrm{E}[X-Y]\mathrm{E}[X+Y].\end{aligned}$$

But, since X and Y are identically distributed,

$$\mathrm{E}[X^2-Y^2] = 0, \quad \mathrm{E}[X-Y] = 0,$$

and the claim follows.

Appendix A
R – A software tool for statistics

R is a statistical computing which can be downloaded for free.[1] To install the software, you just have to donwload the installer from the web site and follow the instructions.
There is a wide and growing set of libraries implementing an array of quite sophisticated methods, but a minimal application is finding quantiles of normal distributions, which is obtained by the function `qnorm`:

```
> qnorm(0.95)
[1] 1.644854
> qnorm(0.95,20,10)
[1] 36.44854
```

In this snapshot you see the R prompt (¿) which is displayed in the command window when you start the software. The first command returns $z_{0.95}$, i.e., the 95% quantile for the standard normal distribution. In the second case, we provide additional parameters corresponding to $\mu = 20$ and $\sigma = 10$. If you need the CDF, use `pnorm`:

```
> pnorm(0)
[1] 0.5
> pnorm(3)
[1] 0.9986501
> pnorm(20,15,10)
[1] 0.6914625
```

[1]R Development Core Team (2010). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org`

Generally speaking, given a distribution name, such as `norm`, the prefix `p` selects the CDF, and the prefix `q` selects the quantile function. The same applies to the `t` distribution, in which case we need to specifies the degrees of freedom:

```
> qt(0.95,5)
[1] 2.015048
> qt(0.95,500)
[1] 1.647907
```

If needed, we may also generate random samples, with prefix `r`:

```
> rnorm(5,20,30)
[1]  26.67444  18.84493 -24.30770  26.03461  35.93114
```

Finally, to get quantiles from the chi-square distribution:

```
> qchisq(0.95,4)
[1] 9.487729
```

or the F distribution:

```
> qf(0.95,3,5)
[1] 5.409451
```

Appendix B
Introduction to MATLAB

MATLAB[1] is a powerful environment for numerical computing, originally developed as an outgrowth of a package for linear algebra. This explains why MATLAB was built around vectors and matrices, even though now it has much more sophisticated data structures. For the purposes of the book, a grasp of the basics is more than enough.

B.1 WORKING WITH VECTORS AND MATRICES IN THE MATLAB ENVIRONMENT

- MATLAB is an interactive computing environment. You may enter expressions and obtain an immediate evaluation:

  ```
  >> rho = 1+sqrt(5)/2
  rho =
      2.1180
  ```

  By entering a command like this, you also define a variable `rho` which is added to the current environment and may be referred to in any other expression.

- There is a rich set of predefined functions. Try typing `help elfun`, `help elmat`, and `help ops` to get information on elementary mathematical functions, matrix manipulation, and operators, respectively. For each predefined function there is an online help:

  ```
  >> help sqrt
  ```

[1]See `http://www.mathworks.com`.

```
 SQRT   Square root.
    SQRT(X) is the square root of the elements of X. Complex
    results are produced if X is not positive.

    See also sqrtm, realsqrt, hypot.

    Reference page in Help browser
       doc sqrt
```

The `help` command should be used when you know the name of the function you are interested in, but you need additional information. Otherwise, `lookfor` may be tried:

```
>> lookfor sqrt
REALSQRT Real square root.
SQRT   Square root.
SQRTM     Matrix square root.
```

We see that `lookfor` searches for functions whose online help documentation includes a given string. Recent MATLAB releases include an extensive online documentation which can be accessed by the command `doc`.

- MATLAB is case sensitive (`Pi` and `pi` are different).

```
>> pi
ans =
    3.1416
>> Pi
??? Undefined function or variable 'Pi'.
```

- MATLAB is a matrix-oriented environment and programming language. Vectors and matrices are the basic data structures, and more complex ones have been introduced in the more recent MATLAB versions. Functions and operators are available to deal with vectors and matrices directly. You may enter row and column vectors as follows:

```
>> V1=[22, 5, 3]
V1 =
    22     5     3

>> V2 = [33; 7; 1]
V2 =
    33
     7
     1
```

We may note the difference between comma and semicolon; the latter is used to terminate a row. In the example above, commas are optional, as we could enter the same vector by typing `V1=[22 5 3]`.

- The `who` and `whos` commands may be used to check the user defined variables in the current environment, which can be cleared by the `clear command`.

```
>> who
```

```
Your variables are:
V1        V2
>> whos
  Name      Size                    Bytes  Class
  V1        1x3                        24  double array
  V2        3x1                        24  double array
Grand total is 6 elements using 48 bytes
>> clear V1
>> whos
  Name      Size                    Bytes  Class
  V2        3x1                        24  double array
Grand total is 3 elements using 24 bytes
>> clear
>> whos
>>
```

- You may also use the semicolon to suppress output from the evaluation of an expression:

```
>> V1=[22, 5, 3];
>> V2 = [33; 7; 1];
>>
```

Using semicolon to suppress output is important when we deal with large matrices (and in MATLAB programming as well).

- You may also enter matrices (note again the difference between '`;`' and '`,`'):

```
>> A=[1 2 3; 4 5 6]
A =
     1     2     3
     4     5     6
>> B=[V2 , V2]
B =
    33    33
     7     7
     1     1
>> C=[V2 ; V2]
C =
    33
     7
     1
    33
     7
     1
```

Also note the effect of the following commands:

```
>> M1=zeros(2,2)
M1 =
     0     0
     0     0
>> M1=rho
```

```
M1 =
    2.1180
>> M1=zeros(2,2);
>> M1(:,:)=rho
M1 =
    2.1180    2.1180
    2.1180    2.1180
```

- The colon (:) is used to spot subranges of an index in a matrix.

```
>> M1=zeros(2,3)
M1 =
     0     0     0
     0     0     0
>> M1(2,:)=4
M1 =
     0     0     0
     4     4     4
>> M1(1,2:3)=6
M1 =
     0     6     6
     4     4     4
```

- The dots (...) may be used to write multiline commands.

```
>> M=ones(2,
??? M=ones(2,

Missing variable or function.
>> M=ones(2,...
2)
M =
     1     1
     1     1
```

- The `zeros` and `ones` commands are useful to initialize and preallocate matrices. This is recommended for efficiency. In fact, matrices are resized automatically by MATLAB whenever you assign a value to an element beyond the current row or column range, but this may be time consuming and should be avoided when possible.

```
>> M = [1 2; 3 4];
>> M(3,3) = 5
M =
     1     2     0
     3     4     0
     0     0     5
```

  It should be noted that this flexible management of memory is a double-edged sword: It may increase flexibility, but it may make debugging difficult.

- `[]` is the empty vector. You may also use it to delete submatrices:

```
>> M1
```

```
M1 =
     0     6     6
     4     4     4
>> M1(:,2)=[]
M1 =
     0     6
     4     4
```

- Another use of the empty vector is to pass default values to MATLAB functions. Unlike other programming languages, MATLAB is rather flexible in its processing of input arguments to functions. Suppose we have a function `f` taking three input parameters. The standard call would be something like `f(x1, x2, x3)`. If we call the function with one input arguments, `f(x1)`, the missing ones are given default values. Of course this does not happen automatically; the function must be programmed that way, and the reader is urged to see how this is accomplished by opening predefined MATLAB functions with the editor.

  Now suppose that we want to pass only the first and the third argument. We obviously cannot simply call the function like `f(x1, x3)`, since `x3` would be assigned to the second input argument of the function. To obtain what we want, we should use the empty vector: `f(x1, [], x3)`.

- Matrices can be transposed and multiplied easily (if dimensions fit):

```
>> M1'
ans =
     0     4
     6     4
>> M2=rand(2,3)
M2 =
    0.9501    0.6068    0.8913
    0.2311    0.4860    0.7621
>> M1*M2
ans =
    1.3868    2.9159    4.5726
    4.7251    4.3713    6.6136
>> M1+1
ans =
     1     7
     5     5
```

  The `rand` command yields a matrix with random entries, uniformly distributed in the (0,1) interval.

- Note the use of the dot `.` to operate element by element on a matrix:

```
>> A=0.5*ones(2,2)
A =
    0.5000    0.5000
    0.5000    0.5000
>> M1
M1 =
```

```
     0     6
     4     4
>> M1*A
ans =
     3     3
     4     4
>> M1.*A
ans =
     0     3
     2     2

>> I=[1 2; 3 4]
I =
     1     2
     3     4
>> I^2
ans =
     7    10
    15    22
>> I.^2
ans =
     1     4
     9    16
```

- Subranges may be used to build vectors. For instance, to compute the factorial:

```
>> 1:10
ans =
     1     2     3     4     5     6     7     8     9    10
>> prod(1:10)
ans =
     3628800
>> sum(1:10)
ans =
    55
```

  You may also specify an optional increment step in these expressions:

```
>> 1:0.8:4
ans =
    1.0000    1.8000    2.6000    3.4000
```

  The step can be negative too:

```
>> 5:-1:0
ans =
     5     4     3     2     1     0
```

- One more use of the colon operator is to make sure that a vector is a column vector:

```
>> V1 = 1:3
V1 =
     1     2     3
```

```
>> V2 = (1:3)'
V2 =
     1
     2
     3
>> V1(:)
ans =
     1
     2
     3
>> V2(:)
ans =
     1
     2
     3
```

The same effect cannot be obtained by transposition, unless one writes code using the function `size` to check matrix dimensions:

```
>> [m,n] = size(V2)
m =
     3
n =
     1
```

- Note the use of the special quantities `Inf` (infinity) and `NaN` (not a number):

```
>> l=1/0
Warning: Divide by zero.
l =
   Inf
>> l
l =
   Inf
>> prod(1:200)
ans =
   Inf
>> 1/0 - prod(1:200)
Warning: Divide by zero.
ans =
   NaN
```

- Useful functions to operate on matrices are: `eye`, `inv`, `eig`, `det`, `rank`, and `diag`:

```
>> eye(3)
ans =
     1     0     0
     0     1     0
     0     0     1
>> K=eye(3)*[1 2 3]'
K =
     1
```

```
     2
     3
>> K=inv(K)
K =
    1.0000         0         0
         0    0.5000         0
         0         0    0.3333
>> eig(K)
ans =
    1.0000
    0.5000
    0.3333
>> rank(K)
ans =
     3
>> det(K)
ans =
    0.1667
>> K=diag([1 2 3])
K =
     1     0     0
     0     2     0
     0     0     3
```

We should note a sort of dual nature in `diag`. If it receives a vector, it builds a matrix; if it receives a matrix, it returns a vector:

```
>> A = [1:3 ; 4:6 ; 7:9];
>> diag(A)
ans =
     1
     5
     9
```

- Some functions operate on matrices columnwise:

```
>> A = [1 3 5 ; 2 4 6 ];
>> sum(A)
ans =
     3     7    11
>> mean(A)
ans =
    1.5000    3.5000    5.5000
```

The last example may help to understand the rationale behind this choice. If the matrix contains samples from multiple random variables, and we want to compute the sample mean, we should arrange data in such a way that variables corresponds to columns, and joint realizations corresponds to rows. However, it is possible to specify the dimension along which these functions should work:

```
>> sum(A,2)
ans =
```

```
     9
    12
>> mean(A,2)
ans =
     3
     4
```

Another useful function in this vein computes cumulative sums:

```
>> cumsum(1:5)
ans =
     1     3     6    10    15
```

B.2 MATLAB GRAPHICS

Most plots in the book have been obtained using the following MATLAB commands.

- Plotting a function of a single variable is easy. Try the following commands:

```
>> x = 0:0.01:2*pi;
>> plot(x,sin(x))
>> axis([0 2*pi -1 1])
```

  The `axis` command may be used to resize plot axes at will. There is also a rich set of ways to annotate a plot.

- Different types of plots may be obtained by using optional parameters of the `plot` command. Try with

```
>> plot(0:20, rand(1,21), 'o')
>> plot(0:20, rand(1,21), 'o-')
```

- To obtain a tridimensional surface, the `surf` command may be used.

```
>> f = @(x,y) exp(-3*(x.^2 + y.^2)).*(sin(5*pi*x)+ cos(10*pi*y));
>> [X Y] = meshgrid(-1:0.01:1 , -1:0.01:1);
>> surf(X,Y,f(X,Y))
```

  Some explanation is in order here. The function `surf` must receive three matrices, corresponding to the x and y coordinates in the plane, and to the function value (the 'z' coordinate). A first requirement is that the function we want to draw should be encoded in such a way that it can receive matrix inputs; use of the dot operator is essential: Without the dots '`.`', input matrices would be multiplied row by column, as in linear algebra, rather than element by element. To build the two matrices of coordinates, `meshgrid` is used. To understand what this function accomplishes, let us consider a small scale example:

```
>> [X,Y] = meshgrid(1:4,1:4)
X =
     1     2     3     4
```

```
     1     2     3     4
     1     2     3     4
     1     2     3     4
Y =
     1     1     1     1
     2     2     2     2
     3     3     3     3
     4     4     4     4
```

We see that, for each point in the plane, we obtain matrices containing each coordinate.

B.3 SOLVING EQUATIONS AND COMPUTING INTEGRALS

- Systems of linear equations are easily solved:

```
>> A = [3 5 -1; 9 2 4;  4 -2 -9];
>> b = (1:3)';
>> X = A\b
X =
    0.3119
   -0.0249
   -0.1892
>> A*X
ans =
    1.0000
    2.0000
    3.0000
```

- To solve a nonlinear equation, we must write a piece of code evaluating the function. This can be done by writing a full-fledged program in the MATLAB programming language. However, when the function is a relatively simple expression it may be preferable to define functions in a more direct way, based on the function handle operator `@`:

```
>> f = @(x,y) exp(2*x).*sin(y)
f =
    @(x,y) exp(2*x).*sin(y)
```

We see that the operator is used to “abstract” a function from an expression. The `@` operator is also useful to define anonymous functions which may be passed to higher-order functions, i.e., functions which receive functions as inputs (e.g., to compute integrals or to solve non-linear equations).

We may also fix some input parameters to obtain function of the remaining arguments:

```
>> g = @(y) f(2,y)
g =
    @(y) f(2,y)
>> g(3)
ans =
```

```
    7.7049
```

- As an example, let us solve the equation

$$x^3 - x = 0$$

To this aim, we may use the `fzero` function, which needs to input arguments: the function f defining the equation $f(x) = 0$; a starting point x_0. From the following snapshot, we see that the function returns a zero close to the starting point:

```
>> f = @(x) x^3 - x
f =
    @(x)x^3-x
>> fzero(f, 2)
ans =
     1
>> fzero(f, -2)
ans =
    -1
>> fzero(f, 0.4)
ans =
  1.0646e-017
```

In general, finding all the roots of a nonlinear equation is a difficult problem. Polynomial equations are an exception. In MATLAB, we may solve a polynomial equation by representing a polynomial with a vector collecting its coefficients and passing it to the function `roots`:

```
>> p = [1 0 -1 0]
p =
     1     0    -1     0
>> roots(p)
ans =
     0
    -1
     1
```

- The `quad` function can be used for numerical quadrature, i.e., the numerical approximation of integrals. Consider the integral

$$I = \int_0^{2\pi} e^{-x} \sin(10x)\, dx$$

This integral can be calculated as follows

$$I = -\frac{1}{101} e^{-x} \left[\sin(10x) + 10\cos(10x)\right]\Big|_0^{2\pi} \approx 0.0988$$

but let us pretend we do not know it. To use `quad`, we have to define the function using the anonymous handle trick:

```
>> f=@(x) exp(-x).*sin(10*x)
```

```
f =
    @(x) exp(-x).*sin(10*x)
>> quad(f,0,2*pi)
ans =
    0.0987
```

Precision may be improved by specifying a tolerance parameter:

```
>> quad(f,0,2*pi, 10e-6)
ans =
    0.0987
>> quad(f,0,2*pi, 10e-8)
ans =
    0.0988
```

B.4 STATISTICS IN MATLAB

MATLAB, like R, can be used to carry out common tasks in statistics, such as generating pseudorandom variates, calculating descriptive statistics, and finding quantiles.
The following snapshot shows how to generate a column vector of 10 observations from a normal distribution with expected value 10 and standard deviation 20; then, we compute sample mean, sample variance, and sample deviation:[2]

```
>> X = normrnd(10,20,10,1)
X =
   20.7533
   46.6777
  -35.1769
   27.2435
   16.3753
  -16.1538
    1.3282
   16.8525
   81.5679
   65.3887
>> mean(X)
ans =
   22.4856
>> var(X)
ans =
  1.2530e+003
>> std(X)
ans =
   35.3977
```

We may also estimate the covariance matrix for a joint distribution:

```
>> mu = [10, 20, -5]
mu =
```

[2]Given the nature of random number generators, you will find different results.

```
    10    20    -5
>> rho = [1 0.9 -0.4
0.9 1 -0.2
-0.4 -0.2 1]
rho =
    1.0000    0.9000   -0.4000
    0.9000    1.0000   -0.2000
   -0.4000   -0.2000    1.0000
>> sigma = [20 30 9]
sigma =
    20    30     9
>> Sigma = corr2cov(sigma,rho)
Sigma =
   400   540   -72
   540   900   -54
   -72   -54    81
>> X = mvnrnd(mu, Sigma, 1000);
>> mean(X)
ans =
    9.2523   19.5316   -4.5473
>> cov(X)
ans =
  389.4356  522.0261  -71.6224
  522.0261  868.4043  -49.5950
  -71.6224  -49.5950   84.0933
```

In this snapshot, we have given the correlation matrix `rho` and the vector of standard deviations `sigma`, which have been transformed into the covariance matrix `Sigma` by the function `corr2cov`; the function `mvnrnd` generates a sample from a multivariate normal. If we need quantiles of the normal distribution, we use `norminv`:

```
>> norminv(0.95)
ans =
    1.6449
>> norminv(0.95, 20, 10)
ans =
   36.4485
```

Just like with R, by default we find quantiles of the standard normal distribution; providing MATLAB with additional parameters, we may specify expected value and standard deviation. If we need the normal CDF, we use `normcdf`:

```
>> normcdf(0)
ans =
    0.5000
>> normcdf(3)
ans =
    0.9987
>> normcdf(20,15,10)
ans =
    0.6915
```

Using `inv`, `chi2inv`, and `finv` we find quantiles of the t, chi-square, and F distribution:

```
>> tinv(0.95,5)
```

```
ans =
    2.0150
>> chi2inv(0.95,4)
ans =
    9.4877
>> finv(0.95,3,5)
ans =
    5.4095
```

The first argument is always the probability level, and the remaining ones specify the parameters of each distribution.

B.5 USING MATLAB TO SOLVE LINEAR AND QUADRATIC PROGRAMMING PROBLEMS

The Optimization toolbox includes a function, `linprog`, which solves LP problems of the form

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}} \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned}$$

The call to the function is `x = linprog(f,A,b,Aeq,beq,lb,ub)`. t As an example, let us solve the problem

$$\begin{aligned} \max \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1 + 3x_2 \leq 100 \\ & 2x_1 + x_2 \leq 80 \\ & x_1 \geq 0,\ 0 \leq x_2 \leq 40 \end{aligned}$$

```
>> c = [-1, -1];
>> A = [1 3; 2 1];
>> b = [100; 80];
>> lb = zeros(2,1);
>> ub = [inf, 40];
>> x = linprog(c, A, b, [], [], lb, ub)
Optimization terminated.
x =
   28.0000
   24.0000
```

Note the use of "infinity" to specify the upper bound on x_1 and the empty vector `[]` as an empty placeholder for the arguments associated with equality constraints; since `linprog` solves a minimization problem, we have to change the sign of the coefficients of the objective function.

To solve quadratic programming problems, such as

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{f}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}} \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u} \end{aligned}$$

we may use `x = quadrog(H,f,A,b,Aeq,beq,lb,ub)`.
As an example, let us find the minimum variance portfolio consisting of 3 assets with expected return and covariance matrix given by[3]

$$\boldsymbol{\mu} = \begin{bmatrix} 0.15 \\ 0.20 \\ 0.08 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.200 & 0.050 & -0.010 \\ 0.050 & 0.300 & 0.015 \\ -0.010 & 0.015 & 0.100 \end{bmatrix}$$

Let $r_T = 0.10$ be the target return:

```
>> Sigma =  [0.200  0.050 -0.010
  0.050  0.300  0.015
  -0.010  0.015  0.100 ]
Sigma =
    0.2000    0.0500   -0.0100
    0.0500    0.3000    0.0150
   -0.0100    0.0150    0.1000
>> mu = [ 0.15; 0.20; 0.08];
>> rt = 0.10;
>> Aeq = [ones(1,3); mu'];
>> beq = [1; rt];
>> w = quadprog(Sigma, [], [], [], Aeq, beq, zeros(3,1))
Optimization terminated.
w =
    0.2610
    0.0144
    0.7246
```

MATLAB cannot be used to solve mixed-integer programming problems; an excellent solver for this purpose is CPLEX, which can be invoked from AMPL (see Appendix C).

[3]See Example 12.5 in the book; also see Section C.2 for a solution using AMPL.

Appendix C
Introduction to AMPL

In this brief appendix, we want to introduce the basic syntax of AMPL. Since its syntax is almost self explanatory, we will just describe a few basic examples, so that the reader can get a grasp of the basic language elements.[1] AMPL is not a language to write procedures; there is a part of the language which is aimed at writing scripts, which behave like any program based on a sequence of control statements and instructions. But the core of AMPL is a *declarative* syntax to describe a mathematical programming model and the data to instantiate it. The optimization solver is separate: You can write a model in AMPL, and solve it with different solvers, possibly implementing different algorithms. Actually, AMPL interfaces have been built for many different solvers; in fact, AMPL is more of a language standard which has been implemented and is sold by a variety of providers.
A demo version is currently available on the web site `http://www.ampl.com`. The reader with no access to a commercial implementation can get the student demo and install it following the instructions. This student demo comes with two solvers: MINOS and CPLEX. MINOS is a solver for linear and nonlinear programming models with continuous variables, developed at Stanford University. CPLEX is a solver for linear and mixed-integer programming models. Originally, CPLEX was am academic product, but it is now developed and distributed by IBM. Recent CPLEX versions are able to cope with quadratic programming models, both continuous and mixed-integer. All the examples in this book have been solved using CPLEX.

[1]For more information, the reader is referred to the original reference for more information: R. Fourer, D.M. Gay, B.W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming* (2nd ed.), Duxbury Press, 2002.

C.1 RUNNING OPTIMIZATION MODELS IN AMPL

Typically, optimization models in AMPL are written using two separate files.

- A **model** file, with standard extension `*.mod`, contains the description of parameters (data), decision variables, constraints, and the objective function.
- A separate **data** file, with standard extension `*.dat`, contains data values for a specific model instance. These data must match the description provided in the model file.

Both files are normal ASCII files which can be created using any text editor, including MATLAB editor (if you are using word processors, be sure you are creating plain text files, with no hidden control characters for formatting). It is also possible to describe a model in one file, but separating structure and data is a good practice, enabling to solve multiple instances of the same model easily.
When you start AMPL, you get a DOS-like window[2] with a prompt like:

```
ampl:
```

To load a model file, you must enter a command like:

```
ampl: model mymodel.mod;
```

where the semicolon must not be forgotten, as it marks the end of a command (otherwise AMPL waits for more input by issuing a prompt like `ampl?`).[3] To load a data file, the command is

```
ampl: data mymodel.dat;
```

Then we may solve the model by issuing the command:

```
ampl: solve;
```

To change data without loading a new model, you should do something like:

```
ampl: reset data;
ampl: data mymodel.dat;
```

Using `reset;` unloads the model too, and it must be used if you want to load and solve a different model. This is also important if you get error messages because of syntax errors in the model description. If you just correct the model file and load the new version, you will get a lot of error messages about duplicate definitions.
The solver can be select using the `option` command. For instance, you may choose

```
ampl: option solver minos;
```

or

```
ampl: option solver cplex;
```

Many more options are actually available, as well as ways to display the solution and to save output to files. We will cover only the essential in the following. We should also mention that the commercial AMPL versions include a powerful script language, which can be used to write complex applications in which several optimization models are dealt with, whereby one model provides input to another one.

[2]The exact look of the window and the way you start AMPL depend on the AMPL version you use.
[3]Here we are assuming that the model and data files are in the same directory as the AMPL executable, which is not good practice. It is much better to place AMPL on the DOS path and to launch it from the directory where the files are stored. See the manuals for details.

```
param NAssets > 0;
param ExpRet{1..NAssets};
param CovMat{1..NAssets, 1..NAssets};
param TargetRet;

var W{1..NAssets} >= 0;

minimize Risk:
      sum {i in 1..NAssets, j in 1..NAssets} W[i]*CovMat[i,j]*W[j];

subject to SumToOne:
      sum {i in 1..NAssets} W[i] = 1;

subject to MinReturn:
      sum {i in 1..NAssets} ExpRet[i]*W[i] = TargetRet;
```

```
param NAssets := 3;
param ExpRet :=
    1 0.15
    2 0.2
    3 0.08;
param CovMat:
     1          2          3        :=
1    0.2000     0.0500    -0.0100
2    0.0500     0.3000     0.0150
3   -0.0100     0.0150     0.1000;

param TargetRet := 0.1;
```

Fig. C.1 AMPL model (`MeanVar.mod`) and data (`MeanVar.dat`) files for mean-variance efficient portfolios.

C.2 MEAN-VARIANCE EFFICIENT PORTFOLIOS IN AMPL

To get acquainted with AMPL syntax, we represent the mean-variance portfolio optimization problem (see Example 12.5 in the book):

$$\begin{aligned} \min \quad & \mathbf{w}'\mathbf{\Sigma}\mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}'\bar{\mathbf{r}} = \bar{r}_T \\ & \sum_{i=1}^{n} w_i = 1 \\ & w_i \geq 0. \end{aligned}$$

AMPL syntax for this model is given in figure C.1. First we define model parameters: the number of assets `NAssets`, the vector of expected return (one per asset), the covariance matrix, and the target return. Note that each declaration must be terminated by a semicolon, as AMPL does not consider end of line characters. The restriction `NAssets > 0` is *not* a constraint of the model: It is an optional consistency check that is carried out when data are loaded, *before* issuing the `solve` command. Catching data inconsistencies as early as possible may be very helpful. Also note that in AMPL it is typical (but not required) to assign long names to parameters and variables, which are more meaningful than the terse names we use in mathematical models.

Then the decision variable `W` is declared; this variable must be non-negative to prevent short-selling, and this bound is associated to the variable, rather than being declared as a constraint. Finally, the objective function and the two constraints are declared. In both cases we use the `sum` operator, with a fairly natural syntax. We should note that braces ({}) are used when declaring vectors and matrices, whereas squares brackets (`[]`) are used to access elements. Objectives and constraints are always given a name, so that later we can access information such as the objective value and dual variables. Expressions for constraints and objective can be entered freely. There is no natural order in the declarations: We may interleave any type of model elements, provided what is used has already been declared.

In the second part of figure C.1 we show the data file. The syntax is fairly natural, but you should notice its basic features:

- Blank and newline characters do not play any role: We must assign vector data by giving both the index and the value; this may look a bit involved, but it allows quite general indexing.

- Each declaration must be closed by a semicolon.

- To assign a matrix, a syntax has been devised that allows to write data as a table, with rows and columns arranged in a visually clear way.

Now we are ready to load and solve the model, and to display the solution:

```
ampl: model MeanVar.mod;
ampl: data MeanVar.dat;
ampl: solve;
    CPLEX 9.1.0: optimal solution; objective 0.06309598494
    18 QP barrier iterations; no basis.
ampl: display W;
    W [*] :=
       1  0.260978
       2  0.0144292
       3  0.724592
    ;
```

We can also evaluate expressions based on the output from the optimization models, as well as checking the shadow prices (Lagrange multipliers) associated with the constraints:

```
ampl: display Risk;
  Risk = 0.063096
ampl: display sqrt(Risk);
  sqrt(Risk) = 0.251189
ampl: display MinReturn.dual;
  MinReturn.dual = -0.69699
ampl: display sum {k in 1..NAssets} W[k]*ExpRet[k];
  sum{k in 1 .. NAssets} W[k]*ExpRet[k] = 0.1
```

```
param NItems > 0;
param Value{1..NItems} >= 0;
param Cost{1..NItems} >= 0;
param Budget >= 0;

var x{1..NItems} binary;

maximize TotalValue:
      sum {i in 1..NItems} Value[i]*x[i];

subject to AvailableBudget:
      sum {i in 1..NItems} Cost[i]*x[i] <= Budget;
```

```
param NItems = 4;

param: Value Cost :=
   1      10    2
   2       7    1
   3      25    6
   4      24    5;

param Budget := 7;
```

Fig. C.2 AMPL model (`Knapsack.mod`) and data (`Knapsack.dat`) files for the knapsack model.

C.3 THE KNAPSACK MODEL IN AMPL

As another example, we consider the knapsack problem (see Section 12.4.1):

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} R_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^{N} C_i x_i \le W \\ & x_i \in \{0, 1\}. \end{aligned}$$

The corresponding AMPL model is displayed in figure C.2. Again, the syntax is fairly natural, and we should just note a couple of points:

- The decision variables are declared as `binary`.
- In the data file, the two vectors of parameters are assigned at the same time to save on writing; you should compare carefully the syntax used here against the syntax used to assign a matrix (see the covariance matrix in the previous example).

Now we may solve the model and check the solution (we must use `reset` to unload the previous model):

```
ampl: reset;
```

```
ampl: model Knapsack.mod;
ampl: data Knapsack.dat;
ampl: solve;
  CPLEX 9.1.0: optimal integer solution; objective 34
  3 MIP simplex iterations
  0 branch-and-bound nodes
ampl: display x;
x [*] :=
1  1
2  0
3  0
4  1
;
```

In this case, branch and bound is invoked (see Chapter 12). In fact, if you are using the student demo, you cannot solve this model with MINOS; CPLEX must be selected using

```
ampl: option solver cplex;
```

If you use MINOS, you will get the solution for the continuous relaxation of the model above, i.e., a model in which the binary decision variables are relaxed: $x \in [0, 1]$, instead of $x \in \{0, 1\}$. The same can be achieved in ILOG AMPL/CPLEX by issuing appropriate commands:

```
ampl: option cplex_options 'relax';
ampl: solve;
  CPLEX 9.1.0: relax
  Ignoring integrality of 4 variables.
  CPLEX 9.1.0: optimal solution; objective 36.2
  1 dual simplex iterations (0 in phase I)
ampl: display x;
  x [*] :=
  1  1
  2  1
  3  0
  4  0.8
;
```

Here we have used the `relax` option to solve the relaxed model. We may also use other options to gain some insights on the solution process:

```
ampl: option cplex_options 'mipdisplay 2';
ampl: solve;
CPLEX 9.1.0: mipdisplay 2
MIP start values provide initial solution with objective 34.0000.
Clique table members: 2
MIP emphasis: balance optimality and feasibility
Root relaxation solution time =    0.00 sec.

     Nodes                                        Cuts/
Node  Left   Objective  IInf  Best Integer    Best Node  ItCnt   Gap

   0     0     36.2000     1       34.0000     36.2000      1  6.47%
                cutoff             34.0000    Cuts:  2      2  0.00%
```

```
Cover cuts applied:  1
CPLEX 9.1.0: optimal integer solution; objective 34
2 MIP simplex iterations
0 branch-and-bound nodes
```

To interpret this output, the reader should have a look at Section 12.6.2., where the branch and bound method is explained.