CHAPTER 22

Student projects

This chapter presents a suite of problems that can be tackled by students. They are less involved than the case studies that were detailed in the preceding chapter, but more substantial than the exercises that we have included in each chapter.

22.1 The level of a dam

In this assignment we will model the changing level (or height) of water in a dam (Figure 22.1). The minimum level is 0 and the maximum is $h_{\max}$. The level increases when rain falls in the catchment area and decreases as a result of evaporation and use. We will ignore any loss due to leaks or seepage.

22.1.1 Height and volume

Volume Let $A(h)$ be the cross-sectional area of the dam at height h.

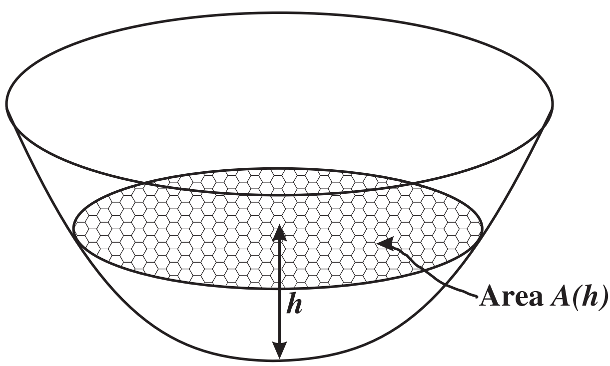

Figure 22.1 *An idealized dam.*

The volume of water contained by the dam when it is filled to level h is

$$V(h) = \int_0^h A(u)\, du.$$

Write a function `volume(h, hmax, ftn)` that returns $V(h)$ for $h \in [0, h_{\max}]$, where `hmax` is $h_{\max}$ and `ftn` is a function of a single variable which is assumed to return $A(h)$. For $h < 0$ your function should return 0, and for $h > h_{\max}$ it should return $V_{\max} = V(h_{\max})$.

Use at least 100 subdivisions when calculating the integral numerically.

Height If the current level of the dam is h and the volume of the dam changes by an amount v, then the level of the dam becomes $u = H(h, v)$ where u satisfies

$$V(u) = V(h) + v.$$

Note that if the right-hand side of this equation is $> V_{\max}$ or < 0, then this equation has no solution. In this case we take $u = h_{\max}$ or $u = 0$, respectively.

Using a root-finding algorithm, write a function `height(h, hmax, v, ftn)` that returns $H(h, v)$, where `hmax` is $h_{\max}$ and `ftn` is a function of a single variable that is assumed to return $A(h)$.

Use a tolerance of `1e-6` in your root-finding algorithm.

Test case Suppose that the dam is bowl-shaped with profile given by the equation $y = \pi x^2$. That is, the dam has the shape obtained by rotating the curve $y = \pi x^2$ about the y-axis (Figure 22.2).

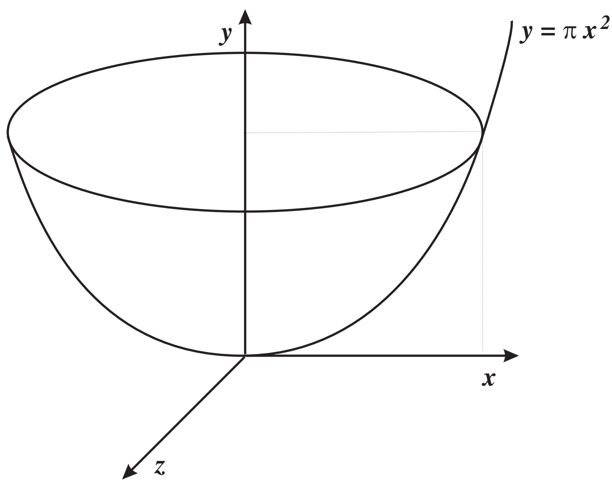

Figure 22.2 *A schematic dam.*

Show that, for $h \in [0, h_{\max}]$ and $v \in [-h^2/2, V_{\max} - h^2/2]$,

$$\begin{aligned} A(h) &= h; \\ V(h) &= h^2/2; \text{ and} \\ H(h, v) &= \sqrt{h^2 + 2v}. \end{aligned}$$

To test that your function `height(h, hmax, v, ftn)` works, define

```
A <- function(h) return(h)
```

then calculate `height(h, hmax = 4, v, ftn = A)` for the following values of h and v:

h	0	2	4	1	1
v	1	1	1	0.1	-0.1

22.1.2 Tracking height over time

Suppose that $h(t)$ is the level of the dam at the start of day t, and that $v(t)$ is the volume of rain falling into the catchment during day t, for $t = 1, \ldots, n$. Also let α be the volume of water taken from the dam for use per day, and let $\beta A(h(t))$ be the volume of water lost due to evaporation during day t. Then the level of water in the dam at the start of day $t + 1$ is given by

$$h(t+1) = H(h(t), v(t) - \alpha - \beta A(h(t))).$$

Further suppose that $h_{\max} = 10$, $\alpha = 1$, $\beta = 0.05$, and $A(h)$ has the form

$$A(h) = \begin{cases} 100h^2 & \text{for } 0 \leq h \leq 2; \\ 400(h-1) & \text{for } 2 \leq h. \end{cases}$$

The file `catchment.txt` (in the `spuRs` archive) gives $v(t)$ for $n = 100$ consecutive days. Write a program that reads this file then, for a given value of $h(1)$, calculates $h(2), \ldots, h(n+1)$. Plot your output for the cases $h(1) = 1$ and $h(1) = 5$, as in Figures 22.3 and 22.4, respectively.

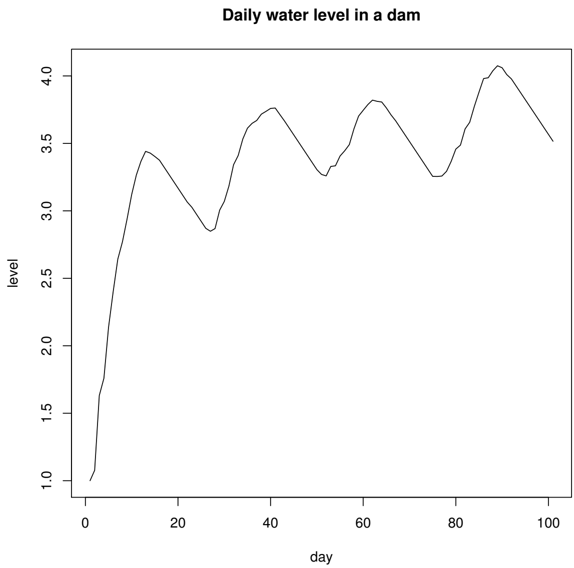

Figure 22.3 *Simulated time trace of water level for dam, $h(1) = 1$.*

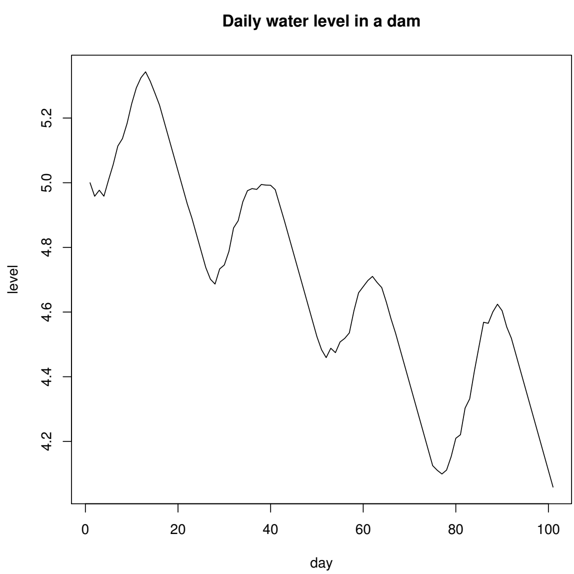

Figure 22.4 *Simulated time trace of water level for dam, $h(1) = 5$.*

22.2 Roulette

At the Crown Casino in Melbourne, Australia, some roulette wheels have 18 slots coloured red, 18 slots coloured black, and 1 slot (numbered 0) coloured green. The red and black slots are also numbered from 1 to 36. (Note that some of the roulette wheels also have a double zero, also coloured green, which nearly doubles the house percentage.)

You can play various 'games' or 'systems' in roulette. Four possible games are:

- A. Betting on Red

  This game involves just one bet. You bet \$1 on red. If the ball lands on red you win \$1, otherwise you lose.

- B. Betting on a Number

  This game involves just one bet. You bet \$1 on a particular number, say 17; if the ball lands on that number you win \$35, otherwise you lose.

- C. Martingale System

  In this game you start by betting \$1 on red. If you lose, you double your previous bet; if you win, you bet \$1 again. You continue to play until you have won \$10, or the bet exceeds \$100.

- D. Labouchere System

  In this game you start with the list of numbers (1, 2, 3, 4). You bet the sum of the first and last numbers on red (initially \$5). If you win you delete the first and last numbers from the list (so if you win your first bet it becomes (2,3)), otherwise you add the sum to the end of your list (so if you lose your first bet it becomes (1, 2, 3, 4, 5)). You repeat this process until your list is empty, or the bet exceeds \$100. If only one number is left on the list, you bet that number.

Different games offer different playing experiences, for example some allow you to win more often than you lose, some let you play longer, some cost more to play, and some risk greater losses. The aim of this assignment is to compare the four games above using the following criteria:

1. The expected winnings per game;
2. The proportion of games you win;
3. The expected playing time per game, measured by the number of bets made;
4. The maximum amount you can lose;
5. The maximum amount you can win.

22.2.1 Simulation

For each game write a function (with no inputs) that plays the game once and returns a vector of length two consisting of the amount won/lost and how many bets were made. Then write a program that estimates 1, 2, and 3, by simulating 100,000 repetitions of each game. Note that a game is won if you make money and lost if you lose money.

22.2.2 Verification

For games A and B, check your estimates for 1 and 2 by calculating the exact answers. What is the percentage error in your estimates for 100,000 repetitions?

For each game, work out the exact answers for 4 and 5. Of course, if this is not close to the answer given by your simulation, then you should suspect that either your calculation or your program is erroneous.

22.2.3 Variation

Repeat the simulation experiment of Part 22.2.1 five times. Report the minimum and maximum values for 1, 2, and 3 in a table as follows:

Game	Exp. winnings min–max	Prop. wins min–max	Exp. play time min–max
A			
B			
C			
D			

Modify your program from Part 22.2.1 so that in addition to estimating the expected winnings, expected proportion of wins, and expected playing time, it also estimates the *standard deviation* of each of these values. (You may use the built-in function `sd(x)` to do this.) For a single run, consisting of 100,000 repetitions of each game, report your results in a table as follows:

Game	Winnings mean, std dev	Prop. wins mean, std dev	Play time mean, std dev
A			
B			
C			
D			

For which game is the amount won most variable?

For which game is the expected playing time most variable?

22.3 Buffon's needle and cross

The following question was first considered by George Louis Leclerc, later Comte de Buffon, in 1733:

> *'If a thin, straight needle of length l is thrown at random onto the middle of a horizontal table ruled with parallel lines a distance $d \geq l$ apart, so that the needle lies entirely on the table, what is the probability that no line will be crossed by the needle?'*

The answer depends on π^{-1} and so simulation of this experiment offers a way of estimating π^{-1}. We will look at the complementary probability that the needle actually intersects with a ruled line on the table; call this a crossing.

22.3.1 Theoretical analysis

We can think of the position of the needle as being determined by two random variables:

Y : the perpendicular distance of the centre of the needle from the nearest line on the table and

X : the angle that the top half of the needle makes with a ray through its centre, parallel to the table lines and extending in a positive direction.

See Figure 22.5 for a sketch.

For the position of the needle to be random, we require Y to be $U(0, d/2)$ and X to be $U(0, \pi)$. We then define the sample space Ω of all possible outcomes or positions of the needle as $\Omega = [0, \pi] \times [0, d/2]$.

1. Identify the inequality that X and Y must satisfy if the needle is to cross a ruled table line. Draw a picture of the sample space Ω and use your inequality to shade that part of it that corresponds to a crossing. We will refer to this region as the crossing region C.
2. As the needle is thrown at random, the probability of falling in any region R in Ω can be calculated as the ratio of the area of R, denoted $|R|$, to the total area of Ω. That is, $2|R|/(\pi d)$. Using integration, find the area of C and hence confirm that the probability of a crossing is $2l/(\pi d)$.

22.3.2 Simulation estimates

Let T_1 be the number of crossings in n tosses of the needle, then $E_1 = T_1 d/(nl)$ is an unbiased estimator of $2/\pi$. Write a program to simulate E_1 using $n =$ 100,000 needle tosses.

Calculate the variance of E_1 and thus suggest the best needle length l to use, subject to the restriction $l \leq d$.

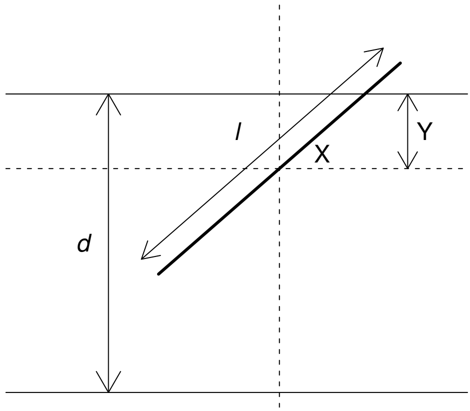

Figure 22.5 *Sketch of Buffon's needle.*

22.3.3 Buffon's cross

An extension of the Buffon needle problem is to think of throwing a cross made up of two equal length needles joined at right angles at their centres. We will assume that the needle lengths $l = d$. The cross can intersect the ruled lines 0, 1, or 2 times.

1. If the position of the first needle (and hence the cross) is specified by (X, Y) as above, show that the second needle crosses the ruled lines if:

$$Y \leq \frac{l}{2}\cos(X), \qquad \text{for } 0 < X < \frac{\pi}{2};$$
$$Y \leq \frac{-l}{2}\cos(X), \qquad \text{for } \frac{\pi}{2} < X < \pi.$$

2. Write a program to estimate the probabilities of 0, 1, or 2 crossings, using n =50,000 simulated tosses of the cross.
3. You can think of the cross as just a convenient way of throwing two needles at once. So if T_2 represents the total crossings in n tosses of the cross, then $E_2 = T_2/2n$ should be another unbiased estimator of $2/\pi$.

   Write E_2 as $\sum_{i=1}^{n} Z_i/n$, where $Z_i \in \{0, 1, 2\}$ is the number of crossings on the i-th toss. We can estimate the variance of E_2 using S_Z^2/n, where $S_Z^2 = \sum_{i=1}^{n}(Z_i - \overline{Z})^2/(n-1)$ is the sample variance. Compare your answer with the theoretical variance of E_1 when n =100,000. Is it smaller, larger, or about the same? (This is an example of antithetic sampling.)

22.4 Insurance risk

This is a simplified version of two common problems faced by insurance companies: calculating the probability that they go bust and estimating how much money they will make.

Suppose that an insurance company has current assets of \$1,000,000. They have $n = 1{,}000$ customers who each pay an annual premium of \$5,500, paid at the start of each year. Based on previous experience, it is estimated that the probability of a customer making a claim is $p = 0.1$ per year, independently of previous claims and other customers. The size X of a claim varies, and is believed to have the following density, with $\alpha = 3$ and $\boldsymbol{\beta} = 100{,}000$,

$$f(x) = \begin{cases} \dfrac{\alpha\boldsymbol{\beta}^\alpha}{(x+\boldsymbol{\beta})^{\alpha+1}} & \text{for } x \geq 0, \\ 0 & \text{for } x < 0. \end{cases}$$

(Such an X is said to have a Pareto distribution, and in the real world is not an uncommon model for the size of an insurance claim.)

We consider the fortunes of the insurance company over a five-year period. Let $Z(t)$ be the company's assets at the end of year t, so

$$\begin{aligned} Z(0) &= 1,000,000, \\ Z(t) &= \begin{cases} \max\{Z(t-1) + \text{premiums} - \text{claims},\ 0\} & \text{if } Z(t-1) > 0, \\ 0 & \text{if } Z(t-1) = 0. \end{cases} \end{aligned}$$

Note that if $Z(t)$ falls below 0 then it stays there. That is, if the company goes bust then it stops trading.

22.4.1 Simulating X

Let X be the size of a typical claim as above. Calculate the cdf F_X, $\mathbb{E}X$, and $\mathrm{Var}\,X$.

Using the inversion method, write a subroutine to simulate X.

Use simulation to estimate the pdf of X and compare your estimate to the true pdf. Your answer should include a plot like Figure 22.6.

22.4.2 Simulating Z

Write a function to simulate the assets of the company over five years, then use it to plot the assets as a graph like Figure 22.7.

Using your function, estimate:

1. The probability that the company goes bust, and
2. The expected assets at the end five years.

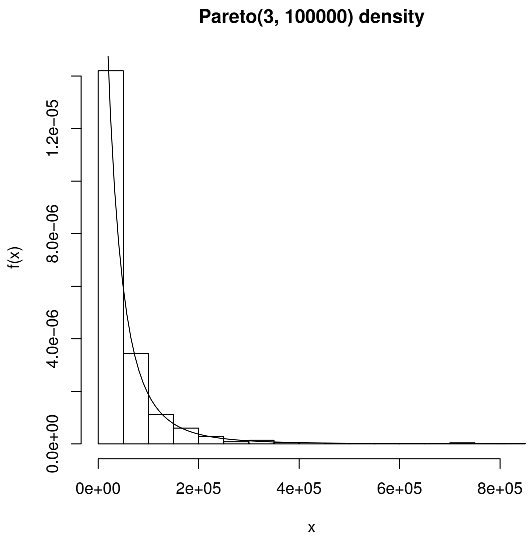

Figure 22.6 *Simulated and true pdf for insurance risk example.*

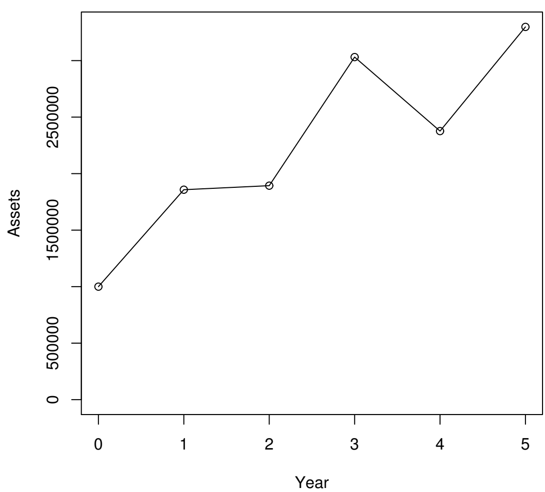

Figure 22.7 *Simulated assets for insurance risk example.*

22.4.3 Profit taking

Suppose now that the company takes profits at the end of each year. That is, if $Z(t) > 1{,}000{,}000$ then $Z(t) - 1{,}000{,}000$ is paid out to the shareholders. If $Z(t) \leq 1{,}000{,}000$ then the shareholders get nothing that year.

Using this new scheme, estimate

1. The probability of going bust.
2. The expected assets at the end of five years, and
3. The expected total profits taken over the five years.

Compare these answers with your answers for Part 22.4.2 and comment.

22.5 Squash

A game of squash is played by two people: player 1 and player 2. The *game* consists of a sequence of *points*. If player i serves and wins the point, then his/her score increases by 1 and he/she retains the serve (for $i = 1$ or 2). If player i serves and loses the point, then the serve is transferred to the other player and the scores stay the same.

The winner is the first person to get 9 points, unless the score reaches 8 all first. If the score reaches 8 all then play continues until one player is 2 points ahead of the other, in which case he/she is the winner.

The object of this assignment is to simulate a game of squash and estimate the probability that player 1 wins. Define

$$\begin{aligned}
a &= \mathbb{P}(\text{ player 1 wins a point} \mid \text{player 1 serves }) \\
b &= \mathbb{P}(\text{ player 1 wins a point} \mid \text{player 2 serves }) \\
x &= \text{ number of points won by player 1} \\
y &= \text{ number of points won by player 2} \\
z &= \begin{cases} 1 & \text{if player 1 has the serve} \\ 2 & \text{if player 2 has the serve} \end{cases}
\end{aligned}$$

We will assume that player 1 serves first.

22.5.1 Status of the game

Write a function `status` that takes inputs x and y and returns one of the following text strings:

`"unfinished"` if the game has not yet finished;

`"player 1 win"` if player 1 has won the game;

`"player 2 win"` if player 2 has won the game;

`"impossible"` if x and y are impossible scores.

You may assume that the inputs x and y are integers.

When you have written your function, load or type the function `status.test` below.

```
# Program spuRs/resources/scripts/status.test.r

status.test <- function(s.ftn) {
  x.vec <- (-1):11
  y.vec <- (-1):11
  plot(x.vec, y.vec, type = "n", xlab = "x", ylab = "y")
  for (x in x.vec) {
    for (y in y.vec) {
```

```
      s <- s.ftn(x, y)
      if (s == "impossible") text(x, y, "X", col = "red")
      else if (s == "unfinished") text(x, y, "?", col = "blue")
      else if (s == "player 1 win") text(x, y, "1", col = "green")
      else if (s == "player 2 win") text(x, y, "2", col = "green")
    }
  }
  return(invisible(NULL))
}
```

Executing the expression `status.test(status)` should give you the output presented in Figure 22.8.

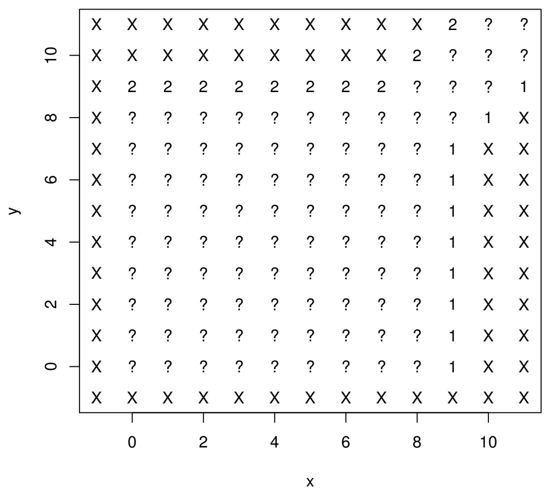

Figure 22.8 *Squash game status.*

22.5.2 Simulating a game

The vector *state* $= (x, y, z)$ describes the current state of the game. Write a function `play_point` that takes inputs *state*, a and b, simulates the play of a single point, then returns an updated vector *state* representing the new state of the game.

Now code up the function `play_game` exactly as follows.

```
# Program spuRs/resources/scripts/play_game.r
```

```
play_game <- function(a, b) {
    state <- c(0, 0, 1)
    while (status(state[1], state[2]) == "unfinished") {
        # show(state)
        state <- play_point(state, a, b)
    }
    if (status(state[1], state[2]) == "player 1 win") {
        return(TRUE)
    } else {
        return(FALSE)
    }
}
```

Provided your functions `status` and `play_point` work properly, function `play_game` simulates a single game of squash and returns `TRUE` if player 1 wins and `FALSE` otherwise.

We define $p(a, b) = \mathbb{P}(\text{ player 1 wins the game} \mid \text{player 1 serves first })$. By simulating n squash games, estimate $p(0.55, 0.45)$ for $n = 2^k$ and $k = 1, 2, \ldots, 12$, then plot the results, as per Figure 22.9.

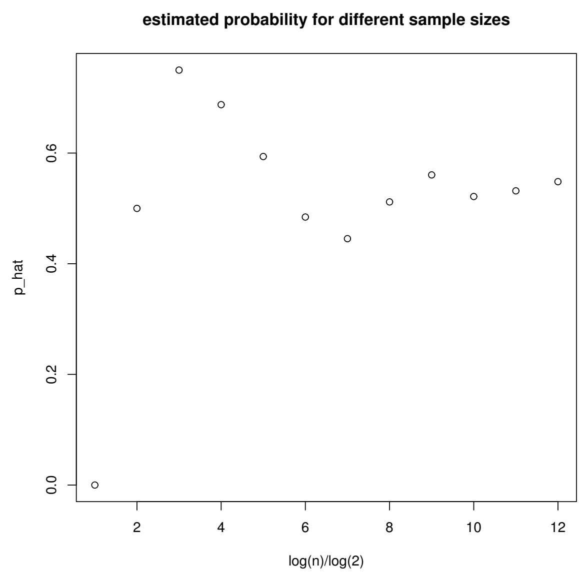

Figure 22.9 *Squash game simulations.*

Is $p(0.55, 0.45) = 0.5$? Explain your answer briefly?

Note that your code should specify a seed for the random number generator, so that you can reproduce your results exactly, if required.

22.5.3 Probability of winning

Let $X_1, \ldots, X_n$ be an iid sample of Bernoulli(p) random variables. We use $\hat{p} = \overline{X}$ to estimate p. Show that $\operatorname{Var} \hat{p} = p(1-p)/n$.

The standard deviation is the square root of the variance. What value of n will guarantee that the standard deviation of $\hat{p}$ is ≤ 0.01 for *any* value of p?

Using the value of n calculated above, reproduce the following table, which estimates $p(a, b)$ for different values of a and b.

`estimated p(a, b) for various a and b`

	b=0.1	b=0.2	b=0.3	b=0.4	b=0.5	b=0.6	b=0.7	b=0.8	b=0.9
a=0.1	0	0	0	0	0	0	0.01	0.05	0.51
a=0.2	0	0	0	0	0.01	0.04	0.16	0.51	0.96
a=0.3	0	0	0.01	0.03	0.09	0.25	0.51	0.85	1
a=0.4	0.01	0.01	0.05	0.12	0.28	0.53	0.79	0.97	1
a=0.5	0.02	0.06	0.15	0.32	0.53	0.76	0.94	0.99	1
a=0.6	0.06	0.19	0.35	0.55	0.76	0.9	0.98	1	1
a=0.7	0.18	0.36	0.56	0.75	0.9	0.97	1	1	1
a=0.8	0.35	0.59	0.79	0.9	0.97	0.99	1	1	1
a=0.9	0.66	0.82	0.92	0.98	0.99	1	1	1	1

Briefly explain the pattern of values you observe. Note that your numbers will be slightly different, as they are simulation estimates.

Make sure that your code specifies a seed for the random number generator, so that you can reproduce your results exactly, if required.

22.5.4 Length of a game

Modify the function `play_game` so that it returns the number of points played in the game (rather than the winning status of player 1).

Using your modified function, reproduce the following table, which estimates the expected number of points played in a game, for different values of a and b. Use the same value for n as above.

`average length of game for various a and b`

	b=0.1	b=0.2	b=0.3	b=0.4	b=0.5	b=0.6	b=0.7	b=0.8	b=0.9
a=0.1	12.23	14.87	18.29	22.90	28.81	38.71	54.38	85.37	151.84
a=0.2	12.54	15.38	19.06	23.69	30.52	40.21	54.54	74.18	84.84
a=0.3	12.82	15.92	19.63	24.86	31.46	39.78	48.48	53.86	53.11
a=0.4	13.26	16.44	20.43	25.37	30.52	35.53	39.41	39.18	37.49
a=0.5	13.83	17.22	20.94	24.58	28.09	30.28	29.91	29.12	27.79
a=0.6	14.46	17.51	20.48	22.79	23.87	24.07	23.54	22.59	21.57

```
a=0.7 |  14.81  17.06  18.91  19.46  19.88  19.10  18.48  17.72  17.04
a=0.8 |  14.36  15.63  16.14  15.91  15.32  14.93  14.50  14.00  13.87
a=0.9 |  12.39  12.90  12.61  12.23  11.94  11.70  11.34  11.22  11.09
```

Briefly explain the pattern of values you observe. Note that your numbers will be slightly different, as they are simulation estimates.

Make sure that your code specifies a seed for the random number generator, so that you can reproduce your results exactly, if required.

22.6 Stock prices

A popular model for stock prices is *Geometric Brownian Motion*. Let $S(i)$ be the stock price at the close of trading on day i (we take today as day 0), then using a Geometric Brownian Motion model we assume that

$$S(i+1) = S(i)\exp(\mu - \tfrac{1}{2}\sigma^2 + \sqrt{\sigma^2}Z(i+1))$$

where $Z(1), Z(2), \ldots$, are iid $N(0,1)$ random variables. The parameter μ is called the drift and σ^2 is known as the volatility.

In practice both μ and σ^2 have to be estimated from the previous behaviour of the stock price.

22.6.1 Simulating S

Write a program that takes as input μ, σ^2, $S(0)$, and t, then simulates $S(1), \ldots, S(t)$ and plots them as a graph.

In your report include sample plots for at least two values of μ and two values of σ^2, and describe qualitatively what happens as μ increases/decreases and as σ^2 increases/decreases.

22.6.2 Estimating $\mathbb{E}S(t)$

Fix $S(0) = 1$ then show that $\log S(t) \sim N(\alpha, \boldsymbol{\beta}^2)$ for some α and $\boldsymbol{\beta}^2$, and find α and $\boldsymbol{\beta}^2$.

Unfortunately, $\mathbb{E}S(t) = \mathbb{E}\exp(\log S(t)) \neq \exp(\mathbb{E}\log S(t)) = \exp(\alpha)$. It turns out that $\mathbb{E}S(t)$ can be calculated exactly (the answer is $\exp(\mu t)$), but the calculation is rather difficult. Instead we will estimate $\mathbb{E}S(t)$ using simulation.

Write a program that takes as input μ, σ^2, and t, simulates $S(\mathrm{t})$ a number of times (at least 10,000) and then estimates $\mathbb{E}S(\mathrm{t})$ and $\mathbb{P}(S(\mathrm{t}) > S(0))$ and gives a 95% confidence interval for each estimate.

Use your program to complete the following table

μ σ^2	0.05 0.0025	0.01 0.0025	0.01 0.01
Estimate of $\mathbb{E}S(100)$ 95% CI for $\mathbb{E}S(100)$ Estimate of $\mathbb{P}(S(100) > S(0))$ 95% CI for $\mathbb{P}(S(100) > S(0))$			

22.6.3 Down-and-out call option

A *Down-and-Out Call Option* is a financial instrument that is sold alongside shares in our stock of interest. The option is determined by its *strike price* K, *time to maturity* t, and *barrier price* B. A single option gives you the right to buy a single share at time t for price K, provided the share price stayed above B.

Let $V(t)$ be the value of our option at maturity, then

$$V(t) = \begin{cases} S(t) - K & \text{if } S(t) \geq K \text{ and } \min_{0 \leq i \leq t} S(i) > B, \\ 0 & \text{if } S(t) < K \text{ or } \min_{0 \leq i \leq t} S(i) \leq B. \end{cases}$$

Options are used by companies to reduce the risks caused by changing prices. For example, a steel producer knows that it will need large quantities of iron ore 12 months in the future. Rather than buy the iron ore now it can buy options, which give a guaranteed price at which to buy the ore in the future.

Assuming $S(0) = 1$, write a program that asks for μ, σ^2, K, t, and B, then simulates $V(t)$ a number of times (at least 10,000) and estimates the cumulative distribution function of $V(t)$. Note that the cdf of $V(t)$ will have a jump at 0, but be continuous otherwise (it is an example of a mixed distribution).

Hence or otherwise, for $\mu = 0.01$, $\sigma \in \{0.0025, 0.005, 0.01\}$, $K = 2$, $t = 100$, and $B = 0.2$, estimate $P(V(100) > 0)$. What can you say about the distribution of $V(t)$ as σ^2 increases/decreases?

An important question (but one you do not have to answer) is what should we pay for the option now? Merton and Scholes won the 1997 Nobel Prize in Economics for their answer to this question (in the special case $B = 0$).