Computing Exercises

1. Calculate the remainder after dividing 31 079 into 170 166 719.
2. Calculate the sum $s_n = \sum_{i=1}^{n} r^i$ for $r = 1.08$ and compare it to $(r^{n+1} - 1)/(r - 1) - 1$ for $n = 10, 20, 30, 40$. Use the quick formula to compute the sum for all values of n between 1 and 100, and store the results in a vector.
3. Calculate the sum $s_n = \sum_{i=1}^{n} i$ and compare with $n(n + 1)/2$ for $n = 100, 200, 400, 800$. Use the quick formula to compute the sum for all values of n between 1 and 100, and store the results in a vector.
4. Calculate the sum $s_n = \sum_{i=1}^{n} i^2$ and compare with $n(n+1)(2n+1)/6$ for $n = 200, 400, 600, 800$. Use the quick formula to compute the sum for all values of n between 1 and 100, and store the results in a vector.
5. Calculate the sum $s_n = \sum_{i=1}^{n} 1/i$, and compare with $\log(n)$ for $n = 500, 1000, 2000, 4000, 8000$. Do you know the limit of $s_n - \log(n)$ for $n \to \infty$?
6. Using `seq()` and `rep()` as needed, create the vectors

   ```
   0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4
   ```

   and

   ```
   1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
   ```

7. Using `seq()` and `rep()` as needed, create the vector

   ```
   1 2 3 4 5 2 3 4 5 6 3 4 5 6 7 4 5 6 7 8 5 6 7 8 9
   ```

8. Write a function which, for a given natural number n, returns a sequence where each $i \leq n$ is repeated i times, in ascending order. E.g., for $n = 4$ the function should return `c(1, 2, 2, 3, 3, 3, 4, 4, 4, 4)`.
9. Create a vector called `numbers` which contains

   ```
   3 5 8 10 12
   ```

   Dump `numbers` to a file called 'numbers.R' and delete `numbers` using `rm()`. Using `ls()`, confirm that `numbers` has been deleted. Now, use `source()` to retrieve the vector `numbers`.
10. Write a function which saves the 26 upper-case letters of the Roman alphabet to a given file.
11. Plot the graph of the function

$$f(x) = \begin{cases} 3x + 2, & x \leq 3 \\ 2x - 0.5x^2, & x > 3 \end{cases}$$

    on the interval $[0, 6]$. (Do not worry about the discontinuity at $x = 3$.)
12. Write a function which outputs whether a given number is positive or negative.
13. Write a function which outputs whether a given number exceeds π in absolute value.

14. Write a function which returns the geometric (or the harmonic) mean of two given numbers.

15. Write a function which outputs which of two given character strings is shorter. (Hint: `?"nchar"`.)

16. Write a function which outputs a given character string once for each character in the string. (Hint: `?"for"`.)

17. Write a function which for a given natural number n outputs all natural numbers $\leq n$ in reverse order, each on a separate line ("countdown").

18. For two natural numbers `x` and `y`, `x %/% y` and `x %% y` give the result and remainder, respectively, of the integer division of `x` by `y`. Write a function which outputs whether a given `x` is divisible by a given `y` (i.e., whether the remainder is zero, or equivalently, whether `x` is a multiple of `y`).

19. Write a function which computes the sum of the digits of a given natural number. Hint: you can use `%%` and `%/%` to extract and eliminate the last digit, respectively.

20. Write a function which computes the total number of digits (in the decimal representation) in a given sequence of natural numbers.

21. Write a function which takes three numbers as arguments and returns the sum of the squares of the two larger numbers.

22. Any cubic equation can be put into the form

$$x^3 + ax^2 + bx + c = 0.$$

Assuming the coefficients are real, such an equation has at least one real root, which can be computed in closed form as follows. Make the substitution $y = x + a/3$. The equation then reduces to

$$y^3 + py + q = 0, \qquad p = \frac{3b - a^2}{3}, \quad q = \frac{2a^3}{27} - \frac{ab}{3} + c.$$

If the discriminant $D = (p/3)^3 + (q/2)^2$ is positive, then there is only one real root, and it is given by

$$x = -\frac{a}{3} + \sqrt[3]{-\frac{q}{2} + \sqrt{D}} + \sqrt[3]{-\frac{q}{2} - \sqrt{D}}.$$

Write a function which uses this method (in real arithmetic) to compute the unique real root of a cubic equation.

23. Write a function `factorial` to compute $n! = \prod_{i=1}^{n} i$ using `prod()`.

    (a) Compute 10!, 50!, 100!, and 1000!.

    (b) Use `factorial` to write a function which computes the binomial coefficient $\binom{n}{m}$. Compute $\binom{4}{2}$, $\binom{50}{20}$, and $\binom{5000}{2000}$.

    (c) The `sum()` function can used to sum up all elements of its arguments, while the `log()` and `exp()` functions take the natural logarithm and exponential of all elements in their arguments. Use these functions to create an improved function to compute the binomial coefficients. Compute $\binom{4}{2}$, $\binom{50}{20}$, and $\binom{5000}{2000}$ using the improved version.

24. Write a function to compute

$$\rho_n = \frac{\Gamma((n-1)/2)}{\Gamma(1/2)\Gamma((n-2)/2)}.$$

Note that a straightforward implementation using `gamma` to compute the Gamma function will fail for large n. Why? (Try $n = 2000$ for example). Instead, compute ρ_n using the logarithm of the Gamma function (provided by `lgamma`). Can you guess the limit of $\rho_n/\sqrt{n}$ for $n \to \infty$?

25. The function logstar gives the smallest n such that the n-th iterate of the natural logarithm results in a value less than one. For example, logstar(123) = 3, because the second iterate $\log(\log(123)) = 1.571151 \geq 1$ and the third iterate $\log(\log(\log(123))) = 0.4518085 < 1$. Implement logstar.

26. The greatest common divisor (GCD) $\gcd(a, b)$ can be obtained using Euclid's algorithm which is based on the following two observations:

    (a) If b is a divisor of a, then $\gcd(a, b) = b$.

    (b) If r is the remainder in the integer division of a by b, then $\gcd(a, b) = \gcd(b, r)$.

    Write a function `gcd` which computes the GCD using Euclid's algorithm. See also `http://www.cut-the-knot.org/blue/Euclid.shtml`.

27. Write a function which, for a given natural number n, returns a logical vector containing for each $i \leq n$ in reverse order, a sequence of length i indicating whether i is even or not. E.g., for $n = 3$ the function should return `c(FALSE, FALSE, FALSE, TRUE, TRUE, FALSE)`.

28. Write a function which computes the number of odd numbers in a given sequence of natural numbers. (Hint: use `sum` to count the number of `TRUE`s in a logical vector.)

29. Write a function which returns (a logical indicating) whether two given natural numbers are relatively prime (i.e., have GCD 1).

30. Write a function `is_prime` returning (a logical indicating) whether a given natural number is a prime number or not.

31. Numbers of the form $M_n = 2^n - 1$ are called *Mersenne* numbers, and are popular "candidates" for prime numbers. Determine the first 7 Mersenne primes by repeatedly investigating the next candidate until the first 7 are found. (See `http://en.wikipedia.org/wiki/Mersenne_prime` for more information.)

32. A perfect number is a positive integer which equals the sum of its proper divisors. The smallest perfect number is $6 = 1 + 2 + 3$. Write a function which tests whether a given positive integer is a perfect number, and determine the first 4 perfect numbers via simple iteration. (See `http://en.wikipedia.org/wiki/Perfect_number` for more information.)

33. A highly composite number is a positive integer that has more divisors (i.e. positive integers that divide into this number) than any smaller positive integer has (`https://en.wikipedia.org/wiki/Highly_composite_number`). Write a function which returns as a logical value if a given number n is highly composite or not and find the first 20 highly composite numbers.

34. Consider the following sequence: start with any positive integer n_0, and, subsequently, obtain every term n_i for $i \geq 1$ from the previous one as follows:

$$n_i = \begin{cases} n_{i-1}/2 & \text{if } n_{i-1} \text{ is even,} \\ 3n_{i-1} + 1 & \text{otherwise.} \end{cases}$$

    According to the (still unproven) Collatz conjecture (`https://en.wikipedia.org/wiki/Collatz_conjecture`), this sequence will always reach 1, no matter what initial value of n_0. Write a function to verify the Collatz conjecture for $1 \leq n_0 \leq 100,000$. Which starting value n_0 in this range requires the greatest number of steps to reach 1? How many steps does it need?

35. Write a function which computes the binary representation (as a sequence of 0s and 1s) of a given natural number (e.g., for $x = 6$ the function should return `c(1, 1, 0)`). Also implement the inverse function.

36. Write a function `lcm` which computes the least common multiple (LCM) of two given natural numbers. Hints: derive a relation between the LCM, the GCD and the numbers, and try avoiding integer overflows when computing the LCD.

37. The two-argument Ackermann-Péter function is defined for non-negative integers m and n as

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0, \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0, \\ A(m-1, A(m,n-1)) & \text{otherwise} \end{cases}$$

(see `http://en.wikipedia.org/wiki/Ackermann_function`). Implement A, and compute $A(2,3)$ and $A(3,2)$.

38. Pascal's triangle is a triangular array of the binomial coefficients. Implement a function which returns the k-th row of the triangle starting at 0 (so that form example for $k = 4$ we get `c(1, 4, 6, 4, 1)`).

39. The Binomial model is a popular and easy model for stock prices. The starting value of an asset S_0 changes in each time step either by a factor U (up) or by a factor D (down). At timestep N we therefore have $N + 1$ different possibilites $S_{N,i} = S_0 \cdot U^N \cdot D^{(N-i)}$, where $i = 0, \ldots, N$.

    Write a function that returns the values of the asset at time n as list, i.e. a list S, where the n-th element contains the elements $S_{n,i}$ for all $i = 0, \ldots, n$,

    (a) using only "for" loops,

    (b) using the apply function family (i.e., `apply`, `lapply`, `sapply`,...).

    (c) What are the differences of the above methods, which one works better?

40. Have a look at the following double sum:

$$\sum_{r=1}^{n} \sum_{c=1}^{r} \frac{c^2}{5 + 10r^3}$$

    Write a function which takes $n \in \mathbb{N}$ as input and outputs the above sum

    (a) using "for" loops

    (b) using the functions `col()` and `row()` (for details use `?col`/`?row`)

41. How many different ways can we make change of a USD 1 dollar bill, given half-dollars, quarters, dimes, nickels and pennies? (I.e., coins in denominations of 50, 25, 10, 5 and 1 cents.) Implement a function which computes the number of ways to change a given amount a using k kinds of coins with denominations $d_1, \ldots, d_k$ using the observation that this is number is the sum of the number of ways to change a without using the first kind of coin and the number of ways to change $a - d$ using all kinds of coins. (Convince yourself to count one way if the amount is 0, but zero ways if the amount is negative or there are no kinds of coins left.)

    How many different ways can we make change of EUR 1 using Euro coins?

42. The remainder of the (integer) division of x by m is also called (the remainder of) x modulo m. Verify that the remainder of x times y modulo m can be obtained as the remainder of the product of the remainders of x and y modulo m modulo m. Use this to implement a recursive function `expmod` which computes the remainder of a^x modulo m without exponentation. (Distinguish the cases where x is zero, even or odd.)

43. What are the smallest and largest 32-bit integers representable by the biased scheme?

44. Using the two's complement representation for signed integers, what are the numbers corresponding to the k-bit sequences $1\cdots1$ and $10\cdots01$, respectively? (See `http://en.wikipedia.org/wiki/Two%27s_complement`.)

45. Write a function which computes the k-bit (default: $k = 32$) two's complement representation of a given integer. Also implement the inverse function.

46. *Rational numbers* are fractions of the form p/q with integers p and $q > 0$ satisfying $\gcd(p, q) = 1$. We can represent such numbers as sequences `c(p, q)` of length two. Write a function `make_rat` which for two given numbers `p` and `q` returns the corresponding rational number as a sequence of length 2 (e.g., `make_rat(19, -95)` $\Rightarrow$ `c(-1, 5)`.) Also write functions `numer` and `denom` for extracting the numerator and denominator of such a rational number.

47. Write functions `add_rat` and `mul_rat` for the rational addition and multiplication of rational numbers.

48. Consider a floating-point number system with base b, precision p, and exponent range $[e_{\min}, e_{\max}]$.

    (a) How many different non-zero normalized numbers are there?

    (b) What are the smallest (normalized) and largest positive numbers?

49. Consider a floating-point number system with base b, precision p, and exponent range $[e_{\min}, e_{\max}]$. If $b = 10$, what are the smallest values of p and $e_{\min}$ and largest value of $e_{\max}$ such that both 2365.27 and 0.0000512 can be represented exactly as normalized floating-point numbers?

50. In a floating-point number system with precision $p = 6$ decimal digits, let $x = 1.23456$ and $y = 1.23579$.

    (a) How many significant digits does the difference $y - x$ contain?

    (b) What is the minimum exponent range for which x, y and $y - x$ are all exactly representable as normalized numbers?

51. In single precision four bytes are used to represent a floating point number: 1 bit is used for the sign, 8 for the exponent, and 23 for the fraction. What are the largest and smallest non-zero positive numbers in single precision (including denormalized numbers)?

52. Let x and y be adjacent normalized non-zero floating-point numbers in a floating-point number system with base b, precision p, and exponent range $[e_{\min}, e_{\max}]$. What are the minimum and maximum possible spacing between x and y?

53. In many floating-point systems, a quick approximation to the unit roundoff error can be obtained by the approximation

    $$\epsilon_{\text{mach}} \approx |\texttt{3 * (4 / 3 - 1) - 1}|.$$

    What is the idea behind this "trick"? Does it work when using IEEE 754 double precision?

54. Can you explain these two results?

```
R> x <- c(0, 7, 8)
R> x[0.9999999999999999]
numeric(0)
R> x[0.99999999999999999]
[1] 0
```

55. Write functions which use a starting value of 1 and repeated division by 2 to find the smallest numbers ϵ such that $1 + \epsilon \neq 1$ and $1 - \epsilon \neq 1$, respectively. What are these numbers?

56. Consider the IEEE 754 double-precision floating-point system. What are the smallest positive normalized and the largest positive denormalized numbers? What is the difference between them?

57. How many normalized machine numbers are there in a single-precision IEEE floating-point system (which used 23 bits for the fraction and 8 bits for the exponent)? How many additional machine numbers are gained if subnormals are allowed?

58. Find the largest positive floating-point number representable by the IEEE 754 64-bit standard. (See `http://en.wikipedia.org/wiki/IEEE_754` and `docs.sun.com/source/806-3568/ncg_goldberg.html`.)

59. Find the smallest normalized positive IEEE 754 64-bit floating-point number ϵ for which $1 + \epsilon \neq 1$.

60. Find the smallest normalized positive IEEE 754 64-bit floating-point number ϵ for which $1 - \epsilon \neq 1$.

61. Write a function computing the coefficients $\alpha_i \in \{0, 1\}$ of the "best" dyadic rational approximation $\sum_{i=1}^{k} \alpha_i 2^{-i}$ of length k to a given number x in $[0, 1]$. E.g., for $k = 4$ and $x = 3/8 = 0/2 + 1/4 + 1/8 + 0/16$ the function should return `c(0, 1, 1, 0)`.

62. Suppose x and y can be represented without error in double precision. Can the same be said for x^2 and y^2? Which would be more accurate, $x^2 - y^2$ or $(x + y)(x - y)$? For what values of x and y, relative to each other, is there a substantial difference in the accuracy of the two expressions?

63. Find three floating-point numbers a, b and c for which addition is not associative, i.e., for which $(a + b) + c \neq a + (b + c)$.

64. The natural logarithm and exponential functions are inverses of each other, so that mathematically $\log(\exp(x)) = \exp(\log(x)) = x$. Show by example that this property does not hold exactly in computer arithmetic. Does it hold "approximately"?

65. Write a program to compute e, the base of natural logarithms, from the definition $e = \lim_{n\to\infty}(1 + 1/n)^n$. Specifically, compute $(1 + 1/n)^n$ for $n = 10^k$, $k = 1, \ldots, 20$. Determine the error in your successive approximations by comparing them with the value of $\exp(1)$. Does the error always decrease as n increases? Explain your results.

66. Euler's number e is defined as:

$$e = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n = \lim_{h\to 0}(1 + h)^{\frac{1}{h}}$$

Write a function, which, for a given n (or h), outputs an approximation of e using either one of those two definitions. Up to how many decimal digits of precision can you approximate e in this way?

67. According to Wilson's Theorem, a fail-safe test of whether a given number n is prime or not is to check if n divides $(n - 1)! + 1$. If it does, it is sure to be prime, if not, it is a composite number. Write a function that implements a primality test based on Wilson's Theorem and test all the integers from 2 up to 25 for their primality using your function. Is this a good way to test for primality using a computer? For which number does it start to fail and why?

68. Consider the function $f(x) = (e^x - 1)/x$.

(a) Use l'Hôpital's rule to show that $\lim_{x\to 0} f(x) = 1$.

(b) Check the result empirically by writing a program to compute $f(x)$ for $x = 10^{-k}$, $k = 1, \ldots, 15$. Do your results agree with theoretical expectations? Explain why.

(c) Perform the experiment in part (b) again, this time using the mathematically equivalent formulation $f(x) = (e^x - 1)/\log(e^x)$, evaluated as indicated with no simplifications. If this works any better, can you explain why?

69. Write a program to compute an approximate value for the derivative of a function using the finite-difference formula

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

Test your program using the function $\tan(x)$ for $x = 1$ (remember that $\tan'(x) = 1 + \tan(x)^2$).

70. As every schoolchild learns, the two solutions of the quadratic equation $ax^2 + bx + c = 0$ are given by the formula

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

For $|b| \approx \sqrt{b^2 - 4ac}$ it is better to compute one of the roots using the alternative formula

$$\frac{2c}{-b \mp \sqrt{b^2 - 4ac}}.$$

Why?

71. Consider the expression

$$\frac{1}{1-x} - \frac{1}{1+x}$$

assuming $x \neq \pm 1$.

(a) For what range of values x is it difficult to compute the expression accurately in floating-point arithmetic?

(b) Give a re-arrangement of the terms such that, for the range of values from part (a), the computation is more accurate in floating-point arithmetic.

72. If $x \approx y$, we might expect some cancellation in computing $\log(x) - \log(y)$. On the other hand, $\log(x) - \log(y) = \log(x/y)$, and the latter involves no cancellation. Does this mean that computing $\log(x/y)$ is likely to give a better result? (*Hint*: For what value is the log function sensitive (highly elastic)?)

73. Consider functions of the form:

$$\phi_j(x) := \frac{1}{(j-1)!} \int_0^1 e^{(1-\theta)x} \theta^{j-1}\, d\theta, \quad j \geq 1, x \in \mathbb{R}.$$

(a) Prove the following recursion:

$$\phi_j(x) = \frac{\phi_{j-1}(x) - \phi_{j-1}(0)}{x}, \quad j > 1, x \neq 0.$$

(b) What happens in this recursion for small x? What is the problem? How could you avoid/bypass this problem?

74. The Euclidean norm of an n-dimensional vector x with components $x_1, \ldots, x_n$ is defined by

$$\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}.$$

How would you avoid overflow and harmful underflow in this computation?

75. For computing the midpoint m of an interval $[a, b]$, which of the following two formulas is preferable in floating-point arithmetic? Why? When?

(a) $m = (a + b)/2.0$

(b) $m = a + (b - a)/2.0$

Hint: Devise examples for which the "midpoint" given by the formula lies *outside* the interval $[a, b]$.

76. Write a program that sums n random floating-point numbers uniformly distributed on the interval $[0, 1]$ (obtained by `runif(n)`). Sum the numbers in each of the following ways:

(a) Use a loop to sum the numbers in the order in which they were generated.

(b) Use the following *compensated summation* algorithm due to Kahan to sum the numbers in the order in which they were generated:

```
s := 0
c := 0
for i := 1 to n
   y := x_i - c
   t := s + y
   c := (t - s) - y
   s := t
end
```

(c) Use a loop to sum the numbers in order of increasing magnitude.

(d) Use a loop to sum the numbers in order of decreasing magnitude.

Run your program for various values of n and compare the results. You may need a fairly large value of n to see substantial differences. How do the methods rank in terms of accuracy, and why? Can you explain why the algorithm in part (b) works?

77. To calculate $\log(x)$ we use the expansion

$$\log(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$$

Truncating to n terms, the error is no greater in magnitude than the last term in the sum.

How many terms in the expansion are required to calculate $\log(1.5)$ with an error of at most 10^{-16}? How many terms are required to calculate $\log(2)$ with an error of at most 10^{-16}? Use the fact that $\log(2) = 2\log(\sqrt{2})$ to suggest a better way of calculating $\log(2)$.

78. The sine of an angle (specified in radians) can be computed by making use of the approximation $\sin(x) \approx x$ for x sufficiently close to zero, and the trigonometric identity

$$\sin(x) = 3\sin(x/3) - 4(\sin(x/3))^3$$

to reduce the size of the argument. Use these idea to implement a function which approximates sin, taking x "sufficiently close to zero" if its absolute value is less than 0.1 radians. What is the order of the number of different x values and steps needed for computing the approximation?

79. Determine $\sum_{n=1}^{\infty} 1/10^n$ as accurately as possible.

80. Determine $1 - 1/2 + 1/3 - 1/4 + \cdots = \sum_{n=0}^{\infty} (-1)^n/(n+1)$ as accurately as possible.

81. Determine

$$\frac{1}{1 \cdot 3} + \frac{1}{5 \cdot 7} + \frac{1}{9 \cdot 11} + \cdots$$

as accurately as possible.

82. The Wallis product for π states that

$$\frac{\pi}{2} = \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdot 8 \cdots}{1 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 7 \cdot 7 \cdots}$$

(`http://en.wikipedia.org/wiki/Wallis_product`). Use this to compute approximations to π.

83. Write a function to compute how many elements of a given character vector contain more than one word. The standard data set `state.name` gives the names of the 50 US states: how many of these consist of several words?

84. How many US states have names with more vowels than consonants?

85. How many US states have names with more than three 's'?

86. How many US states have names with a double vowel?

87. Function `colors()` gives the names of all available colors in R. Compute the number of different reds available.

88. Write a function which finds the most frequent character in a given file (e.g., `http://statmath.wu.ac.at/~hornik/index.html`). (Hint: use `readLines()` for reading lines of text from a given file or URI, and `table()` for creating frequency tables.)

89. Write a function `wc` which computes the number of different words in a given file (note the existence of punctuation characters).

90. Write a function which counts how often a given word occurs in a given file, ignoring case.

91. Write a function which computes the binary representation of a given natural number x, as a binary string. (I.e., for $x = 6$ the result should be the string `"110"`.)

92. Use the `state.name` data set to write a function that takes some letters as input and randomly selects a state of the data set and outputs a logical indicating whether all of the letters are contained in that state name. (Hint: use `?sample`.)

93. The `"%a"` conversion of `sprintf` allows to print double precision values in the form `0x`f`p`$\pm d$ (optionally prefixed with a minus sign) as a binary fraction expressed in hex multiplied by a decimal power of 2. Use this to write a function which computes the 52-bit fraction and 11-bit exponent in the IEEE 754 64-bit floating-point representation of a given normalized floating point number.

94. Over the complex plane, every polynomial

$$p(z) = z^n + \beta_n z^{n-1} + \cdots + \beta_2 z + \beta_1$$

has n roots $z_1, \ldots z_n$ and can be written as $p(z) = \prod_{k=1}^{n} (z - z_k)$. Write a function which computes the coefficients $(\beta_1, \ldots, \beta_n)$ from the roots.

95. Over the complex plane, the roots of the equation

$$z^n = 1$$

are given by n-th roots of unity $z_k = e^{2\pi ik/n}$, $k = 0, \ldots, n-1$. Compare the "exact" solutions to those obtained by `polyroot()` (note how this parametrizes the coefficients of the polynomial). (Hint: use `order()` to sort roots lexicographically: first by real, then by imaginary part.)

Write a function computing the maximal modulus of the differences for given n.

96. Try recovering the coefficients of $z^n - 1$ from its roots. Write a function computing the maximal modulus of the difference between true and recovered coefficients for given n.

97. Represent polynomials $p(z) = \beta_1 + \beta_2 z + \cdots + \beta_n z^{n-1}$ by their coefficient vectors $(\beta_1, \ldots, \beta_n)$. Write a function `polyderiv` which computes the coefficient vector of the derivative p' of p. Use this function in combination with `polyroot` to compute the inflection points of a given polynomial.

98. Write a function which, for a given natural number $n \leq 26$ returns a list whose i-th element is the sequence of the first i lower case letters of the Roman alphabet.

99. Write a function `count_chars_in_table` which, for a given character vector `x` and a vector of single characters `table`, returns a vector giving the number of characters in the respective elements of `x` contained in `table`. E.g., `count_chars_in_table("Wien", letters[1:13])` $\Rightarrow$ 2, as only 'e' and 'i' are in the first 13 lower-case letters of the Roman alphabet. (Hint: use `match` to simplify matters.)

100. Write a function which, for a given character vector `x`, returns a vector with the number of upper case letters in the elements of `x`.

101. Write a function `strrev` which reverses a given character string (so that e.g. `strrev("ABC")` $\Rightarrow$ `"CBA"`).

102. Write a function which, for each element of a given character vector, counts the number of words in the element, as obtained by splitting according to white space. (Hint: use `strsplit()` and `lapply()`.)

103. In each of the following, determine the final value of `answer`. Check your result by running the code in R.

(a)
```
answer <- 0; for(j in 1 : 5) answer <- answer + j
```

(b)
```
answer <- NULL; for(j in 1 : 5) answer <- c(answer, j)
```

(c)
```
answer <- 0; for(j in 1 : 5) answer <- c(answer, j)
```

(d)
```
answer <- 1; for(j in 1 : 5) answer <- answer * j
```

(e)
```
answer <- 3
for (j in 1 : 15) answer <- c(answer, 7 * answer[j]) %% 31)
```

Inspect the last sequence of numbers. If you did not know the rule used to determine this sequence, would you be able to predict successive elements?

104. A (simple) *continued fraction* is an expression of the form

$$\text{cfrac}(a_0, \ldots, a_n) = a_0 + 1/(a_1 + 1/(a_2 + \cdots + 1/(a_{n-1} + 1/a_n)))$$

with suitable integers $a_0, \ldots, a_n$. Implement cfrac as a function which takes a given sequence of integers and computes the value of the corresponding continued fraction as a floating-point number. Use the infinite continued fraction expansion

$$(3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, \ldots)$$

of π to obtain suitable approximations of π, and compare with the value of π. (See e.g. `http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/cfINTRO.html` and `http://en.wikipedia.org/wiki/Continued_fraction`.)

105. Write a function which computes continued fractions in rational arithmetic, i.e., returns the numerator and denominator of the rational number represented by the continued fraction. Which rational approximations does this yield for the continued fraction expansion $(3, 7, 15, 1, 292, \ldots)$ of π?

106. What is the relative error in approximating π by $22/7$? What about $355/113$?

107. To obtain the (possibly infinite) continued fraction representation of a number $x > 0$, one iteratively computes the integer part $\text{floor}(x)$ of x and if this is different from x, continues with the reciprocal value of $x - \text{floor}(x)$ (see `http://en.wikipedia.org/wiki/Continued_fraction`). Write a function `num2cfrac(x, n)` which determines coefficients $a_0, \ldots, a_n$ of the n-th continued fraction approximation to x (if the process terminates at $m < n$, then the function should only give $a_0, \ldots, a_m$, of course). Use `num2cfrac` to determine the coefficients for x taking the values $\sqrt{2}$, $\sqrt{3}$, e, π, and the golden ratio $(1 + \sqrt{5})/2$ and $n = 12$.

108. Write a function `rat2cfrac(p, q)` which computes the (exact) continued fraction representation of a rational number p/q using integer arithmetic. What does this give for $415/93$, $22/7$ and $355/113$?

109. A finite generalized continued fraction is an expression of the form

$$\text{gcfrac}(a, b) = a_0 + b_1/(a_1 + b_2/(a_2 + \cdots + b_n/a_n))$$

Write a function to compute the value of such expressions, and use it to implement an approximation to the tangent function at given x using k terms of the Lambert continued fraction representation

$$\tan(x) = x/(1 - x^2/(3 - x^2/(5 - x^2/(7 - \cdots))))$$

What does the approximation give for $x = \pi/4$ and $k = 5$?

110. Fermat's Little Theorem states that if n is a prime number and a any positive integer less than n, then the remainder of a^n modulo n equals a. This can be used to develop the probabilistic Fermat test for primality: given n, randomly pick k distinct positive integers $a_1, \ldots, a_k$ less than n, and test whether for all i the remainder of a_i^n modulo n equals a_i. Implement this test using `expmod` for efficiently computing a_i^n modulo n without exponentiation. (Use `sample()` to sample without replacement.)

111. Show that 561 fools the Fermat primality test, and is the smallest such number known as Carmichael numbers (`http://en.wikipedia.org/wiki/Carmichael_number`).

112. Every natural number n has a prime factorization of the form $n = \prod_k p_k^{e_k}$, where the primes p_k divide n exactly e_k times. Write a function `factorize` which returns the sequence of prime factors p_k with the corresponding multiplicities e_k for a given n. (E.g., `factorize(120)` $\Rightarrow$ `c(2, 2, 2, 3, 5)`.)

113. A twin prime is a pair of primes (x, y) such that $y = x + 2$. Construct a list of all twin primes below 1000.

114. Let f_n denote the n-th Fibonacci number.

    (a) Construct a sequence of ratios of the form f_{n+1}/f_n, $n = 1, 2, \ldots, 30$. Does the sequence appear to be converging?

    (b) Compute the golden ratio $(1 + \sqrt{5})/2$. Is the sequence converging to this ratio? Can you prove this?

115. Determine the number of Fibonacci numbers less than 1 000 000.

116. The Fibonacci recursion can be seen as the application of the transformation $x = (a, b) \mapsto Tx = (a + b, a)$ with initial value $x = (1, 0)$ (using the modern convention to begin with $F_0 = 0$) giving $(F_n, F_{n-1}) = T^n(F_1, F_0) = T^n(1, 0)$.

    Consider T to be the special case of $p = 0$ and $q = 1$ in the family of transformations T_{pq} defined by

    $$T_{pq}(a, b) = ((p + q)a + qb, qa + pb).$$

    Show that applying such a transformation T_{pq} twice has the same effect as applying a transformation $T_{p'q'}$ of the same form, and determine p' and q' in terms of p and q. Use this to implement a function which computes $T^n_{pq}x$ in a logarithmic number of steps, by using $T^0_{pq}x = x$, $T^{2k}_{pq}x = T^k_{p'q'}x$ if $n = 2k$ is even and $T^n_{pq}x = T_{pq}(T^{n-1}_{pq}x)$ otherwise.

    Use this function to provide a function which computes the n-th Fibonacci number in a logarithmic number of steps.

117. Implement a function which computes, for a given positive integer n, all pairs (i, j) with $1 \leq i < j \leq n$.

118. Write a function which for given positive integers n and k (with $k \leq n$ computes all k-tuples $(i_1, \ldots, i_k)$ with $1 \leq i_1 < \cdots < i_k \leq n$. (Do not use function `combn()`.)

119. In the Josephus problem, n soldiers are standing in a circle being surrounded by members of the enemy army, so they come up with a system to execute each other to avoid being captured. The first soldier kills the second, the third kills the fourth and so on until the end of the circle is reached (at the end of the circle, either the n-th soldier kills the first or the $(n-1)$-th kills the n-th, depending on whether n is even or odd). This procedure is repeated with the remaining soldiers, starting with the next soldier and going in the same direction until only one person remains. Let's suppose soldier Josephus secretly prefers capture to being killed by a fellow soldier. In which position in the circle does he have to stand in order to be the last one standing?

    Write a function that takes n as an input and outputs the corresponding position in the circle $W(n)$ that will make sure he is the last soldier remaining. Can you come up with an explicit formula for $W(n)$?

120. Write a function `directpoly` which will evaluate polynomials of the form

    $$p(x) = a_n x^{n-1} + a_{n-1}x^{n-2} + \cdots + a_2 x + a_1.$$

    Your function should take `x` and the vector of polynomial coefficients as arguments and return the value of the evaluated polynomial. Ensure that your function returns an appropriate vector of values when `x` is a vector.

121. Write a function `hornerpoly` to evaluate polynomials of the form

    $$p(x) = a_n x^{n-1} + a_{n-1}x^{n-2} + \cdots + a_2 x + a_1.$$

    using Horner's scheme:

    (a) Set $v_n \leftarrow a_n$.

    (b) For $i = n - 1, \ldots, 1$ set $v_i \leftarrow v_{i+1}x + a_i$.

    (c) Return v_1.

    Ensure that your function returns an appropriate vector of values when `x` is a vector.

122. Do some timings to compare the algorithms implemented by functions `directpoly()` and `hornerpoly()`.

   (a) In particular, compare the `system.time()` measurements for `directpoly(x, a)` and `hornerpoly(x, a)` with `x` values `seq(-10, 10, length.out = 5000000)` and `a` coefficients `c(1, -2, 2, 3, 4, 6, 7)`.

   (b) What happens to the comparison when the number of polynomial coefficients is smaller? Try the polynomial

   $$p(x) = 2x^2 + 17x - 3.$$

123. In computational science an important procedure is the polynomial interpolation. The famous Lagrange basis polynomials are given as

   $$L_i(x) = \prod_{j=0, j\neq i}^{n-1} \frac{x - x_j}{x_i - x_j}$$

   where n is the number of points to interpolate.

   (a) Write a function which outputs the Lagrange basis polynomial for different numbers of i, and predefined interpolation points, where the number of points is n. Plot the functions L_0,L_1,L_2 together in one plot (as interpolation points use e.g.: $x_0 = -1$, $x_1 = 2$, $x_2 = 3$).

   (b) The interpolation polynomial (at the interpolation points) in Lagrange form is given as

   $$L(x) = \sum_{i=0}^{n-1} f(x_i) L_i(x)$$

   Have a look at the function $f(x) = x^5$ at the points $x_0 = -1$, $x_1 = 0$, $x_2 = 1$, $x_3 = 2$. Determine the Lagrange basis polynomials at the given points and the interpolation polynomial of f. Display them both in one plot.

124. Write a (recursive) function `count(pat, str)` which counts how many times string `pat` occurs in string `str`. E.g., `count("ab", "abab cd cdab")` $\Rightarrow 3$. (Hint: `?regexpr`.)

125. Write a recursive function for generating all combinations of k out of n objects in the form of 0/1 "incidence" strings (i.e., strings consisting of k `"1"` and $n-k$ `"0"` characters; the i-th character indicates whether object i occurs in the respective combination (`"1"`) or not (`"0"`). E.g., for $n = 3$ and $k = 2$, the function should return `c("110", "101", "011")`.

126. Write a recursive function which computes the octal representation (as a sequence of digits 0 to 7) of a given natural number.

127. The composition $f * g$ ("f after g") of two unary functions f and g is defined by $(f * g)(x) = f(g(x))$. Implement a function `Compose` which takes two unary functions and return their composition.

128. The k-th iterate f^k of a unary function f is the function which repeatedly applies f k times, so that f^0 is the identity and $f^{k+1}(x) = f(f^k(x))$, or equivalently, $f^{k+1} = f * f^k$. Implement a function `Iterate` which takes a unary function f and a non-negative integer k and returns the k-th iterate of f.

129. Use a fixed-point iteration to determine the solution (in $[0, 1]$) of the equation $\cos(x) = x$. Use a starting value of 0.5. How many iterations does it take before you have an answer which is accurate in the first two, three or four digits? What happens if you change the starting value to 0.7 or to 0.0?

130. A bank offers a guaranteed investment certificate (GIC) which pays an annual interest rate of 4% (compounded annually) if the term is 3 years or less, or 5% if the term is more than 3 years. Write a function which takes the initial investment amount P, and the number of interest periods (i.e., years) as arguments, and returns the amount of interest earned over the term of the GIC.

131. A car dealer promotes two options for the purchase of a new USD 20k car. The first option is for the customer to pay up front and receive a USD 1k rebate. The second option is "0%-interest financing" where the customer makes 20 monthly payments of USD 1k beginning on one month's time. The monthly interest rate i employed by the car dealer can be obtained by equating present values, and is found to satisfy the equation

$$i = \frac{1 - (1+i)^{-20}}{19}.$$

Use a fixed-point iteration with a starting guess of $i = 0.006$ to calculate i. Stop the calculation when two successive values are less than 0.000 001 apart. What happens if you try other starting guesses?

132. The equation

$$x^7 + 10\,000x^6 + 1.06x^5 + 10\,600x^4 + 0.0605x^3 + 605x^2 + 0.0005x + 5 = 0$$

has exactly one real solution. How many iterations of Newton's method are required to find this root if the initial guess is $x = 0$?

133. Using a starting value of 2.9, find the time that it takes for Newton's method to find the zero (to within 7 digits of accuracy) of

(a) $(x-3)e^{-x}$,

(b) $(x^2 - 6x + 9)e^{-x}$.

134. Use Newton's method to find a zero of

$$f(x) = x^4 + 3x^3 - 2x^2 - 7$$

using an initial guess of $x = 1$.

135. Use Newton's method to find a zero of

$$f(x) = \cos(x) + e^x$$

using an initial guess of $x = -1.5$.

136. Newton's method is sometimes used to implement the built-in square root function on a computer, with the initial guess supplied by a lookup table.

(a) What is the Newton iteration for computing the square root of a positive number a (i.e., for solving the equation $f(x) = x^2 - a = 0$ for given a)?

(b) Show that

$$x_{k+1}^2 - a = \left(\frac{x_k^2 - a}{2x_k}\right)^2$$

and conclude that $x_k > \sqrt{a}$ for all $k > 0$.

(c) Show that the x_k are decreasing for $k > 0$.

(d) Let $e_k = \sqrt{a} - x_k$ and $r_k = e_k/\sqrt{a}$ be the absolute and relative errors of the approximations. Show that

$$e_{k+1} = -\frac{e_k^2}{2x_k}, \qquad r_{k+1} = -\frac{\sqrt{a}}{2x_k} r_k^2.$$

(e) Derive an upper bound for $|r_4|$ assuming that $x_0 \geq \sqrt{a}$ and $|r_0| \leq 0.1$.

137. On a computer with no functional unit for floating-point division, one might instead use multiplication by the reciprocal of the divisor. Apply Newton's method to produce an iterative scheme for approximating the reciprocal of a number $y > 0$ (i.e., to solve the equation $f(x) = (1/x) - y = 0$ for given y). Considering the intended application, your formula should contain no divisions.

138. Show that the iterative method

$$x_{k+1} = \frac{x_{k-1} f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

is mathematically equivalent to the secant method

$$x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

for solving a scalar non-linear equation $f(x) = 0$.

When implemented in finite-precision floating-point arithmetic, what advantages does the new formula have compared with the standard formula for the secant method?

139. Consider the system of equations

$$\begin{aligned} x_1 - 1 &= 0, \\ x_1 x_2 - 1 &= 0. \end{aligned}$$

For which starting point or points, if any, will Newton's method for solving this system fail? Why?

140. How many zeros does the function $f(x) = \sin(10x) - x$ have? Find all these using a root solver of your choice. *Hint*: inspecting the graph of the function will be very helpful, and you will need a different starting point or initial bracketing interval for each root.

141. Consider the function

$$f(x) = (((x - 0.5) + x) - 0.5) + x$$

evaluated as indicated (i.e., without any simplification). On your computer, is there any floating-point value of x such that $f(x)$ is exactly zero? If you use a root finder on this function, what result is returned, and what is the value of f at this argument? How are the results obtained influenced by error tolerances?

142. How does Newton's method behave when you apply it to find a solution to the non-linear equation $f(x) = x^5 - x^3 - 4x = 0$ with $x_0 = 1$ as starting point? What are the real roots of this equation? Is there anything pathological about them?

143. Find a minimizer of the function

$$f(x) = (x - 3)^4 + 7(x - 2)^2 + x.$$

144. How many zeros does the function

$$f(x) = \frac{5x-3}{x-1}$$

have? What are they? Describe the behavior of Newton's method applied to this function if the initial guess is 0.5, 0.75, 0.2, or 1.24, respectively.

145. How many zeros does the function

$$f(x) = (x^2 - 6x + 9)e^{-x}$$

have? What are they? Describe the behavior of Newton's method applied to this function if the initial guess is 3, 3.2, 2.99, or 3.01, respectively.

146. A car dealer promotes two options for the purchase of a new USD 20k car. The first option is for the customer to pay up front and receive a USD 1k rebate. The second option is "0%-interest financing" where the customer makes 20 monthly payments of USD 1k beginning on one month's time. The monthly interest rate i employed by the car dealer can be obtained by equating present values, and is found to satisfy the equation

$$i = \frac{1-(1+i)^{-20}}{19}.$$

Use Newton's method with a starting guess of $i = 0.006$ to calculate i. How many iterations are required for two successive values of i to be within 0.000 001 of each other?

147. The equations

$$\begin{aligned} f(x,y) &= 0 \\ g(x,y) &= 0 \end{aligned}$$

can be solved numerically using a form of Newton's method. Assign initial guesses to each of x_0 and y_0. Then perform the following iteration for $n = 1, 2, \ldots$:

$$\begin{aligned} x_{n+1} &= x_n - (g_{y,n} f_n - f_{y,n} g_n)/d_n \\ y_{n+1} &= y_n - (f_{x,n} g_n - g_{x,n} f_n)/d_n \end{aligned}$$

where $f_n = f(x_n, y_n)$, $g_n = g(x_n, y_n)$,

$$f_{x,n} = \frac{\partial f}{\partial x}(x_n, y_n), \quad f_{y,n} = \frac{\partial f}{\partial y}(x_n, y_n), \quad g_{x,n} = \frac{\partial g}{\partial x}(x_n, y_n), \quad g_{y,n} = \frac{\partial g}{\partial y}(x_n, y_n),$$

and $d_n = f_{x,n} g_{y,n} - f_{y,n} g_{x,n}$. The iteration is terminated when the function values are close enough to 0.

(a) Write a function which will perform this iteration.

(b) Apply the function to the system

$$\begin{aligned} x + y &= 0 \\ x^2 + 2y^2 - 2 &= 0. \end{aligned}$$

Find the two solutions to this system analytically as a check on your numerical result.

148. The *half-interval method* is a simple yet powerful method for finding the root of an equation $f(x) = 0$ for continuous f. If we have a and b with $f(a) < 0 < f(b)$, then f must have at least one zero between a and b. To locate a zero, compute the mid-point x of a and b. If $f(x) < 0$, continue searching on between x and b; otherwise, continue between a and x. The iteration is stopped (for example) when the length of the interval containing a zero is smaller than a given tolerance τ. Implement the half-interval method, and use it to find the zero of $f = \sin$ between $a = 2$ and $b = 4$.

149. For a real-valued function f, the Newton-Raphson algorithm uses a sequence of linear approximations to f to find a root. What happens if we use quadratic approximations instead?

Suppose that x_n is our current approximation to the root, then a quadratic approximation to f at x_n is given by the second order Taylor expansion

$$f(x) \approx g_n(x) = f(x_n) + (x - x_n)f'(x_n) + \frac{1}{2}(x - x_n)^2 f''(x_n).$$

Let x_{n+1} be the solution of $g_n(x) = 0$ that is closest to x_n, assuming a solution exists. If there is no solution, then let x_{n+1} be the point at which g_n attains either its minimum or maximum.

Implement this algorithm in R and use it to find the roots of the following functions:

(a) $\cos(x) - x$ using $x_0 = 1, 3, 6$.

(b) $x^3 - x - 3$ using $x_0 = 0$.

(c) $x^3 - 7x^2 + 14x - 8$ using $x_0 = 1.1, 1.2, \ldots, 1.9$.

For your implementation, assuming that you are given a function `f(x)` which returns the vector $(f(x), f'(x), f''(x))$. Given x_n, if you rewrite g_n as $g_n(x) = Ax^2 + Bx + C$ then you can use the formula $(-B \pm \sqrt{B^2 - 4AC})/(2A)$ to find the roots of g_n and thus x_{n+1}. If g_n has no roots then the min/max occurs at the point where $g_n'(x) = 0$.

How does this algorithm compare to the Newton-Raphson algorithm?

150. How do we know $\pi = 3.1415926$ (to 7 decimal places)? One way of finding π is to solve $\sin(x) = 0$, which has the solutions $k\pi$ for $k = 0, \pm 1, \pm 2, \ldots$, so the root closest to 3 should be π.

(a) Use a root-finding algorithm, such as the Newton-Raphson algorithm, to find the root of $\sin(x)$ near 3. How close can you get to π?

The function $\sin(x)$ is *transcendental*, which means that it cannot be written as a rational function of x. Instead, we can write it as an infinite sum

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

(This is the Taylor series expansion of $\sin(x)$ about 0.) In practice, to calculate $\sin(x)$ numerically we have to truncate this sum (so any numerical calculation of $\sin(x)$ is an approximation).

(b) Put

$$f_n(x) = \sum_{k=0}^{n} (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

Write a function to calculate $f_n(x)$. Plot f_n over the range $[0, 7]$ for a number of values of n, and verify that it looks like sin for large n.

(c) Choose a large value of n, the find an approximation to π by solving $f_n(x) = 0$ near 3. Can you get an approximation that is correct up to 6 decimal places? Can you think of a better way of calculating π?

151. Apply the golden section minimization technique to the following functions:

(a) $f(x) = |x - 3.5| + |x - 2| + |x - 1|$,

(b) $f(x) = |x - 3.2| + |x - 3.5| + |x - 2| + |x - 1|$.

For the second function, check the graph to see that the minimizer is not unique. Show that the minimizer found by `golden()` depends on the initial interval supplied to the function.

152. For an odd number of data values $x_1, \ldots, x_n$, the minimizer of the function

$$f(x) = \sum_{i=1}^{n} |x - x_i|$$

is the sample median of the data values. Verify this result for the following data:

(a) 3, 7, 9, 12, 15.
(b) 3, 7, 9, 12, 15, 18, 21.

Describe, in words, what happens when the number of observations is even.

153. Write a function that would find the maximizer of a function using the golden section search.

154. Use the golden-section search algorithm to find all local minima of the function

$$f(x) = \begin{cases} 0, & x = 0 \\ |x| \log(|x|/2) e^{-|x|}, & \text{otherwise} \end{cases}$$

within the interval $[-10, 10]$. (Hint: plotting the function first will give you a good idea where to look.)

155. Use the `optimize()` function to minimize the following functions:

(a) $f(x) = |x - 3.5| + |x - 2| + |x - 1|$,
(b) $f(x) = |x - 3.2| + |x - 3.5| + |x - 2| + |x - 1|$.

156. Use `nlm()` and `optim()` to minimize the function

$$f(a, b) = (a - 1) + 3.2/b + 3\log(\Gamma(a)) + 3a\log(b).$$

157. Use `nlminb()` to minimize the function

$$f(a, b) = (a - 1) + 3.2/b + 3\log(\Gamma(a)) + 3a\log(b),$$

noting that a and b should be restricted to being non-negative.

158. The Rosenbrock function is a commonly used test function, given by

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2.$$

It has a single global minimum at $(1, 1)$.

Use a contour plot to inspect the function in the region $[-2, 2] \times [-2, 5]$, and try to find the minimum of f using a starting value of $(0, 3)$.

159. A simple way of using local search strategies to find a global maximum is to consider several different starting points, and hope that for one of them its local maximum is in fact the global maximum. If you have no idea where to start, then randomization can be used to choose the starting point.

Consider the function

$$\begin{aligned} f(x, y) \quad = \quad & -(x^2 + y^2 - 2)(x^2 + y^2 - 1)(x^2 + y^2)(x^2 + y^2 + 1)(x^2 + y^2 + 2) \\ & * (2 - \sin(x^2 - y^2)\cos(y - \exp(y))). \end{aligned}$$

It has several local maxima in the region $[-1.5, 1.5] \times [-1.5, 1.5]$. Try creating several random starting points in this region using `runif(2, -1.5, 1.5)` to find the global maximum.

160. The function $f(x) = x^2 - 2x + 2$ has a minimum at $x^* = 1$. On your computer, for what range of values of x near x^* is $f(x) = f(x^*)$? Can you explain this phenomenon? What are the implications regarding the accuracy with which a minimum can be computed?

161. The function $f(x) = 0.5 - xe^{-x^2}$ has a minimum at $x^* = 1/\sqrt{2}$. On your computer, for what range of values of x near x^* is $f(x) = f(x^*)$? Can you explain this phenomenon? What are the implications regarding the accuracy with which a minimum can be computed?

162. Consider the function f defined by

$$f(x) = \begin{cases} 0.5 & \text{if } x = 0 \\ (1 - \cos(x))/x^2 & \text{otherwise.} \end{cases}$$

Use l'Hôpital's rule and differentiation to show that f is continuous at $x = 0$ and has a local maximum there.

Use R to find a maximum of f on the interval $[-2\pi, 2\pi]$ on which $-f$ is unimodal. Experiment with the error tolerance to determine how accurately the program can determine the known solution at $x = 0$.

163. Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(x, y) = 2x^3 - 3x^2 - 6xy(x - y - 1).$$

(a) Determine all critical points of f analytically (i.e., without using a computer).

(b) Classify each critical point found as a minimum, maximum, or a saddle point, again working analytically.

(c) Verify your analysis graphically by creating a contour plot of f over the region $-2 \leq x \leq 2$, $-2 \leq y \leq 2$.

(d) Use R to find the minima and maxima of f. Experiment with various starting points to see how the program gets around other types of critical points to find minima and maxima.

164. Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(x, y) = 2x^2 - 1.05x^4 + x^6/6 + xy + y^2.$$

How many critical points can you find for this function? Classify each such point you can find as a local minimum, local maximum or saddle point of the function. What is the global minimum of this function?

165. Find non-negative x_1, x_2, x_3 and x_4 to minimize

$$C(x) = x_1 + 3x_2 + 4x_3 + x_4$$

subject to the constraints

$$x_1 - 2x_2 \geq 9, \qquad 3x_2 + x_3 \geq 9, \qquad x_2 + x_4 \geq 10.$$

Will the solution change if there is a requirement that any of the variables should be integers? Explain.

Suppose that the objective function is changed to

$$C(x) = x_1 - 3x_2 + 4x_3 + x_4.$$

What happens to the solution now?

166. Find non-negative x_1, x_2, x_3 and x_4 to maximize

$$C(x) = x_1 + 3x_2 + 4x_3 + x_4$$

subject to the constraints

$$x_1 - 2x_2 \leq 9, \qquad 3x_2 + x_3 \leq 9, \qquad x_2 + x_4 \leq 10.$$

167. Find the portfolio allocations maximizing

$$Q(x) = m'x - \frac{k}{2}x'Vx$$

with means and covariances of the daily returns given by

$$m = \begin{bmatrix} 0.002 \\ 0.005 \\ 0.010 \end{bmatrix}, \qquad V = \begin{bmatrix} 0.010 & 0.002 & 0.002 \\ 0.002 & 0.010 & 0.002 \\ 0.002 & 0.002 & 0.010 \end{bmatrix},$$

and $k = 4$ and $k = 1$, respectively (shorting is not allowed). How does being less risk-averse affect the investor's behavior?

168. Often, there are upper bounds on the proportion that can be invested in a particular stock. Re-do the portfolio allocation problem with $k = 4$ and the requirement that no more than 50% of the investor's fortune can be tied up in any one stock.

169. Duncan's Donut's Inc. (DDI) and John's Jeans Ltd. (JJL) are two stocks with mean daily returns of 0.005 and 0.010, respectively. What is the optimal portfolio for a completely risk-loving investor (i.e., risk tolerance constant $k = 0$)? (Hint: answering this question does not require computations.)

170. Suppose the daily returns for DDI and JJL are independent, but $\sigma^2_{\text{DDI}} = 0.01$ and $\sigma^2_{\text{JJL}} = 0.04$. What is the optimal allocation for an investor with risk tolerance constants $k = 1$ and $k = 2$, respectively? What are the optimal allocations if the stocks have a covariance of 0.01?

171. Use `matrix()`, `seq()` and `rep()` to construct the following 5×5 *Hankel* matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

Convert the code into a function which can be used to construct matrices of dimension $n \times n$ which have the same pattern. Use the function to output 10×10 and 12×12 Hankel matrices.

172. Construct the following stochastic matrix (all entries non-negative):

```
       sunny rainy
sunny    0.2   0.8
rainy    0.3   0.7
```

173. Construct the two vectors of heights (in cm) and weights (in kg) for 5 individuals:

```
R> height <- c(172, 168, 167, 175, 180)
R> weight <- c(62, 64, 51, 71, 69)
```

Bind these vectors into a matrix, and modify the result to obtain

```
          height weight
Neil         172     62
Cindy        168     64
Pardeep      167     51
Deepak       175     71
Hao          180     69
```

Pardeep's height is really 162 cm, Hao's height is really 181 cm and his weight is really 68 kg. Correct the matrix accordingly.