1. The sum of 100.1, 234.9 and 12.01
2. The square root of 256
3. Calculate the 10-based logarithm of 100, and multiply the result with the cosine of p. Hint: see  
   ?log and ?pi.
4. Calculate the cumulative sum (’running total’) of the numbers 2,3,4,5,6.
5. Calculate the cumulative sum of those numbers, but in reverse order. *Hint:* use the rev function
6. Find 10 random numbers between 0 and 100, rounded to the nearest whole number (*Hint:* you  
   can use either sample or a combination of round and runif)
7. Type the following code, which assigns numbers to objects x and y.  
   x <- 10  
   y <- 20
8. Calculate the product of x and y
9. Store the result in a new object called z
10. Inspect your workspace by typing ls(), and by clicking the Environment tab in Rstudio, and find  
    the three objects you created.
11. Make a vector of the objects x, y and z. Use this command,  
    myvec <- c(x,y,z)
12. Find the minimum, maximum, length, and variance of myvec.
13. Remove the myvec object from your workspace.
14. The numbers below are the first ten days of rainfall amounts in 1996. Read them into a vector  
    using the c() function (recall Section **??** on p. **??**).  
    0.1 0.6 33.8 1.9 9.6 4.3 33.7 0.3 0.0 0.1
15. What was the mean rainfall, how about the standard deviation?
16. Calculate the cumulative rainfall (’running total’) over these ten days. Confirm that the last value  
    of the vector that this produces is equal to the total sum of the rainfall.
17. Which day saw the highest rainfall (write code to get the answer)?
18. Run the following code, which makes two numeric objects.  
    one <- 1  
    two <- 2
19. Run the following two lines of code, and look at the resulting two vectors. The first line makes  
    a character vector, the second line a numeric vector by recalling the objects you just constructed.  
    Make sure you understand the difference.  
    vector1 <- c("one","two")  
    vector2 <- c(one, two)
20. The following lines of code contain some common errors that prevent them from being evaluated properly or result in error messages. Look at the code without running it and see if you can  
    identify the errors and correct them all. Also execute the faulty code by copying and pasting the  
    text into the console (not typing it, R studio will attempt to avoid these errors by default) so you  
    get to know some common error messages (but not all of these result in errors!).  
    vector1 <- c('one', 'two', 'three', 'four, 'five', 'seven')  
    vec.var <- var(c(1, 3, 5, 3, 5, 1)  
    vec.mean <- mean(c(1, 3, 5, 3, 5, 1))  
    vec.Min <- Min(c(1, 3, 5, 3, 5, 1))  
    Vector2 <- c('a', 'b', 'f', 'g')  
    vector2
21. You have measured five cylinders, their lengths are:  
    2.1, 3.4, 2.5, 2.7, 2.9  
    and the diameters are :  
    0.3, 0.5, 0.6, 0.9, 1.1  
    Read these data into two vectors (give the vectors appropriate names).
22. Calculate the correlation between lengths and diameters (use the cor function).
23. Calculate the volume of each cylinder (V = length \* pi \* (diameter / 2)^2).
24. Calculate the mean, standard deviation, and coefficient of variation of the volumes.
25. Assume your measurements are in centimetres. Recalculate the volumes so that their units are  
    in cubic millimetres. Calculate the mean, standard deviation, and coefficient of variation of these  
    new volumes.
26. You have measured the same five cylinders, but this time were distracted and wrote one of the  
    measurements down twice:  
    2.1, 3.4, 2.5, 2.7, 2.9  
    and the diameters are :  
    0.3, 0.5, 0.6, 0.6, 0.9, 1.1  
    Read these data into two vectors (give the vectors appropriate names). As above, calculate the  
    correlation between the vectors and store in a new vector. Also generate a vector of volumes  
    based on these vectors and then calculate the mean and standard deviations of the volumes.  
    Note that some steps result in errors, others in warnings, and some run perfectly fine. Why were  
    some vectors created and others were not?
27. For the second question, you need to know that the 26 letters of the Roman alphabet are conveniently accessible in R via letters and LETTERS. These are not functions, but vectors that are always loaded.
28. Read in a vector that contains "A", "B", "C" and "D" (use the c() function). Using rep, produce  
    this:  
    "A" "A" "A" "B" "B" "B" "C" "C" "C" "D" "D" "D"  
    and this:  
    "A" "B" "C" "D" "A" "B" "C" "D" "A" "B" "C" "D"
29. Draw 10 random letters from the lowercase alphabet, and sort them alphabetically (*Hint:* use  
    sample and sort). The solution can be one line of code.
30. Draw 5 random letters from each of the lowercase and uppercase alphabets, incorporating  
    both into a single vector, and sort it alphabetically.
31. Repeat the above exercise but sort the vector alphabetically in descending order.
32. Inspect the help page union, and note the useful functions union, setdiff and intersect. These can  
    be used to compare and combine two vectors. Make two vectors :  
    x <- c(1,2,5,9,11)  
    y <- c(2,5,1,0,23)  
    Experiment with the three functions to find solutions to these questions.
33. Find values that are contained in both x and y
34. Find values that are in x but not y (and vice versa)
35. Construct a vector that contains all values contained in either x or y, and compare this vector  
    to c(x,y).
36. Construct a matrix with 10 columns and 10 rows, all filled with random numbers between 0 and  
    1 (see Section **??** on p. **??**)
37. Calculate the row means of this matrix (*Hint:* use rowMeans). Also calculate the standard deviation across the row means (now also use sd).
38. Now remake the above matrix with 100 columns, and 10 rows. Then calculate the column  
    means (using, of course, colMeans), and plot a frequency diagram (a ’histogram’) using hist. We  
    will see this function in more detail in a later chapter, but it is easy enough to use as you just  
    do hist(myvector), where myvector is any numeric vector (like the column means). What sort  
    10 of shape of the histogram do you expect? Now repeat the above with more rows, and more  
    columns.