Exercises for R Course

Jacek Matulewski
Version: March 23, 2022

I suggest using the R Studio notebook.

- Calculate the value of the expression $3\sqrt[15]{2}/\sqrt[7517]{2}$.
- Calculate the value of the expression $e^{i\pi}$ (where i is imaginary unit) (Euler's identity links four most important mathematical constants).

- Write and test a function that takes weight and height, displays these values, and calculates and displays the body mass index (BMI). Prepare a set of test data and function tests based on them.
- Write and test a function that takes weight and height, calculates the BMI using the function from the previous task and classifies the result (from starvation to obesity grade III - the criteria can be found on the previously mentioned Wikipedia page). Perform the classification "cleverly", avoiding a series of conditional statements.

- Write some functions that take a vector of numbers as argument and return the following statistical parameters. Do not use ready-made functions. Prepare a few sets of test data and verify the result using ready-made R functions.
  Statistical parameters:
  - minimum and maximum value in the set,
  - mean and standard deviation,
  - median,
  - counting the occurrence of individual values.

- Write a function that removes from the matrix the rows and columns that contain NA
  Test data: `mym = matrix(c(3, NA, NA, 4, 5, 6, 7, 9), nrow = 2)`
  Source: *http://r-tutorials.com/r-exercises-beginners-easy-functions/*

- Write and test a function that takes any number of vectors and creates a `data.frame` with vectors elements in columns.
- Write and test a function called `factorial` that calculates the factorial of the integer given in the argument using a `for` loop. The function should raise an error (using `stopifnot`) if the argument is not a non-negative integer.
- Write and test a function that calculates and writes to a vector a given number of Fibonacci sequence terms.
- Based on the previous exercise, write to the vector a sequence of numbers that are the quotients of successive Fibonacci numbers (i.e. of successive elements of the Fibonacci sequence). Plot values from the vector. Check whether it converges and, if so, to what number (the result should be a *golden number* equal to $(1+\sqrt{5})/2$).
- Write and test a function based on a `for` loop that displays integer divisors.
  Try to optimize it to reduce the number of iterations of the loop.

- Write and test two functions that take two vectors each (make sure both are of equal length) and return 1) a number that is the product of the first vector and the transposition of the second, and 2) the matrix that is the product of the first vector and the second.
- Write and test a function that calculates the sum of the indicated in the argument number of terms of the following series (the proposed by Leibniz method of calculating the number π):

$$4\sum_{n=0}^{N}\frac{(-1)^n}{2n+1}$$

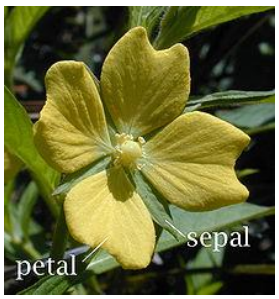
  Compare the obtained value with the number π and check how the accuracy of the estimation changes depending on the number of summed terms of the series. Draw a plot of the error of the estimation against the number of words.
- Write and test a function that calculates the sum of the series $1/n!$. For a sufficiently large number of summed terms, it should close to the value of *e*.

- Write a function that multiplies the given two-element vector by the rotation matrix:

$$\begin{pmatrix}\cos(\alpha) & \sin(\alpha)\\ -\sin(\alpha) & \cos(\alpha)\end{pmatrix},$$

  where α is the parameter of the function (rotation angle).
- Test the function from the previous point using the vector (1, 0). Check whether the vector length for any rotation is kept.
- Check if the rotation took place by the correct angle by calculating the dot product of the original and the rotated vector (the sum of the products of individual components of both vectors) and dividing it by the product of the lengths of both vectors. The value obtained in this way should equal the cosine of the angle between these vectors (the angle itself is equal to the arc cosine of this value).

***Iris* data set:**



Source: Wikipedia

- Statistical analysis
  - Load data from *iris.csv* file to `data.frame` https://gist.github.com/curran/a08a1080b88344b0c8a7
  - Preliminary data review – functions `dim`, `names`, `str`, `attributes`, `head`, `summary`.
  - Display sepal and petal length histograms
  - Prepare a plot of sepal lengths density `plot(density(…))`.
  - Prepare a pie chart showing the number of flowers (data records) belonging to each iris species.
  - Split the data into three separate `data.frame` storing records for each iris species.

    - Check if the distributions of each iris species' petals and sepals' length and width are normal.
    - Compare these lengths and widths using appropriate statistical tests and determine which are significantly different between species.
    - Calculate the Spearman correlation between the lengths and widths of sepals and petals by species and for the entire data set.
- Machine learning
    - **Split** the original Iris collection into training and test data in a 4:1 ratio (`sample.split`).
    - **Classify** using decision trees (`ctree` → model, `predict`, `multiclass.roc` → `print`).
    - Prepare a plot for a decision tree `plot(`model).
    - Perform *random forest* **classification** (`randomForest` → model, `predict`, `multiclass.roc` → `print`).
    - Remove the species names (labels) from the original *Iris* data set (*Species* column).
    - Perform *k-means* **clustering** (`kmeans`).
    - Prepare a drawing of individual assignments (x-axis – sepal length, y-axis – sepal width).
    - Repeat the **clustering** using the *k-mediod* method (`pamk`).

Sources:
http://rstudio-pubs-static.s3.amazonaws.com/450733_9a472ce9632f4ffbb2d6175aaaee5be6.html
https://www.kaggle.com/mokosan/practice-from-r-and-data-mining-iris-dataset

See also: Principal Components Analysis (PCA):
http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

- The above analyzes were performed for the *Iris* data set discussed already during the classes. Perform the same analysis for the `penguins` data set.
  Installation: `install.packages("palmerpenguins")`.
  Source: https://github.com/allisonhorst/palmerpenguins

Other similar datasets for exercises: https://www.meganstodel.com/posts/no-to-iris/
(It is also described here the reason why some people opted out of Iris data.)
and https://medium.com/towards-artificial-intelligence/best-datasets-for-machine-learning-data-science-computer-vision-nlp-ai-c9541058cf4f (many various packages, esp. for machine learning)

- Prepare a thousand-element vector with the values of $\sin^{10}(x)$ for x between 0.0 and 10.0. Then, using the moving average as a smoothing filter, prepare the vector with the smoothed data. Plot the original and smoothed values in a common plot.

Benford's Law

- Write to a vector a 1000 successive terms of a geometric progression sequence e.g. successive powers of any number q. For $q = 2$ it will be: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, etc. Then prepare a function that takes this vector of numbers as an argument and calculates the frequency of numbers starting with the consecutive digits from 1 to 9. Make a bar graph of the distribution of these frequencies. Is it homogeneous?
- Use the function from the last task for the set of Fibonacci numbers.
  Is the distribution homogeneous in this case?
- Create a vector containing the area and population of the countries of the world
  (data source e.g. https://pl.wikipedia.org/wiki/Pa%C5%84stwa_%C5%9Bwiata).
  Check what the frequency distribution of the first digits looks like in the case of this data.
- Repeat the analysis for the population of Polish cities
  (data source np. https://pl.wikipedia.org/wiki/Dane_statystyczne_o_miastach_w_Polsce or Database of Central Statistical Office of Poland https://bdl.stat.gov.pl/BDL/start).
- Repeat the analysis for any arithmetic sequence. What is the distribution in this case?
- Write *shiny* app, that allows one to load data from a CSV file and then checks whether the Benford distribution is preserved in the indicated column.

More exercises: https://docs.ufpr.br/~marianakleina/rexercises-1-R-basic.pdf