

Zadania do kursu R

Jacek Matulewski

Wersja: 5 czerwca 2023

Zadania najlepiej wykonywać w notebooku R Studio.

Część 1

- Oblicz wartość wyrażenia $3^{15}\sqrt{2}/^{7517}\sqrt{2}$.
- Oblicz wartość wyrażenia $e^{i\pi}$ (i to [jednostka urojona](#)) ([tożsamość Eulera](#) łącząca cztery najważniejsze stałe matematyczne).
- Napisz funkcję pobierającą wagę i wzrost, wyświetlającą te wartości, a także obliczającą i wyświetlającą [Indeks masy ciała \(BMI\)](#). Przygotuj zestaw danych testowych i oparte na nich testy funkcji.
- Napisz funkcję, która pobiera wagę i wzrost, oblicza BMI korzystając z funkcji z poprzedniego zadania i klasyfikuje wynik (od wygłodzenia do otyłości III stopnia – kryteria znajdziesz na wcześniej podanej stronie Wikipedii). Klasyfikację przeprowadź "sprytnie" tzn. postaraj się uniknąć serii instrukcji warunkowych.
- Napisz kilka funkcji pobierających wektor liczb i zwracających poniższe parametry statystyczne. Nie używaj gotowych funkcji. Przygotuj kilka zestawów danych testowych i przetestuj wynik korzystając z gotowych funkcji.
Parametry statystyczne:
 - minimalną i maksymalną wartość w zbiorze,
 - średnią oraz odchylenie standardowe,
 - medianę,
 - zliczającą wystąpienie poszczególnych wartości.

Część 2

- Napisz funkcję usuwającą z macierzy wiersze i kolumny, które zawierają NA
Dane testowe: `mym = matrix(c(3, NA, NA, 4, 5, 6, 7, 9), nrow = 2)`
Źródło: <http://r-tutorials.com/r-exercises-beginners-easy-functions/>
- Napisz funkcję przyjmującą dowolną liczbę wektorów i tworzącą z nich `data.frame` ułożonych wierszami.
- Napisz i przetestuj funkcję o nazwie `silnia` obliczającą silnię liczby całkowitej podanej w argumencie korzystając z pętli `for`. Funkcja powinna zgłosić błąd (za pomocą `stopifnot`) jeżeli argument nie jest nieujemną liczbą całkowitą.
- Napisz i przetestuj funkcję obliczającą i zapisującą do wektora wskazaną liczbę wyrazów ciągu Fibonacciego.
- Bazując na poprzednim zadaniu, zapisz do wektora ciąg liczb będących ilorazami kolejnych liczb Fibonacciego (tj. kolejnych elementów ciągu Fibonacciego). Wykreśl go. Sprawdź czy jest zbieżny i jeżeli tak, do jakiej zbiega liczby (wynikiem powinna być *złota liczba* równa $(1 + \sqrt{5})/2$).
- Napisz i przetestuj funkcję opartą na pętli `for` wyświetlającą dzielniki liczby całkowitej. Postaraj się ją zoptymalizować tak, aby zmniejszyć liczbę iteracji pętli.

- Napisz i przetestuj dwie funkcje przyjmujące dwa wektory (należy sprawdzić, czy są równej długości) i zwracające 1) liczbę będącą iloczynem pierwszego wektora i transpozycji drugiego oraz 2) macierz będącą wynikiem iloczynu wektora pierwszego transponowanego i drugiego.
- Napisz i przetestuj funkcję obliczającą sumę wskazanej w argumencie liczby wyrazów poniższego szeregu (zaproponowana przez Leibniza metoda obliczania [liczby \$\pi\$](#)):

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

Porównaj uzyskaną wartość z liczbą π i sprawdź, jak zmienia się dokładność estymacji w zależności od liczby zsumowanych wyrazów szeregu. Narysuj wykres błędu estymacji w zależności od liczby wyrazów.

- Napisz i przetestuj funkcję obliczającą sumę szeregu $1/n!$. Dla odpowiednio dużej liczby zsumowanych wyrazów powinieneś uzyskać wartość e .

- Napisz funkcję, która mnoży podany wektor dwuelementowy przez macierz obrotu:

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

gdzie α jest parametrem funkcji (kątem obrotu).

- Przetestuj funkcję z poprzedniego punktu korzystając z wektora $(1, 0)$. Sprawdź, czy zachowana jest długość wektora dla dowolnego obrotu.
- Sprawdź również, czy obrót nastąpił o kąt wybrany kąt obliczając [iloczyn skalarny](#) wektora oryginalnego i obróconego (sumę iloczynów poszczególnych składowych obu wektorów) i dzieląc go przez iloczyn długości obu wektorów. Tak uzyskana wartość powinna być równa kosinusowi kąta między tymi wektorami (sam kąt równy jest arcusowi kosinusa tej wartości).

Części 3-5

Dane *Iris*:



Źródło: Wikipedia (petal – płatek, sepal – działka kielicha)

- Analiza statystyczna
 - Wczytaj dane z pliku `iris.csv` do `data.frame`
<https://gist.github.com/curran/a08a1080b88344b0c8a7>
 - Wstępny przegląd danych – funkcje `dim`, `names`, `str`, `attributes`, `head`, `summary`.
 - Wyświetl histogramy długości działek kielicha i płatków.
 - Przygotuj wykres gęstości długości działek kielicha `plot(density(...))`.

- Przygotuj wykres kołowy („tort”) pokazujący liczbę kwiatów (rekordów danych) należących do poszczególnych gatunków irysów.
- Podziel dane na trzy osobne `data.frame` przechowujące rekordy dla poszczególnych gatunków irysów.
- Sprawdź normalność rozkładów długości i szerokości płatków i działek kielicha poszczególnych gatunków.
- Porównaj te długości i szerokości korzystając z odpowiednich testów statystycznych i rozstrzygnij, które są istotnie różne u różnych gatunków.
- Oblicz korelację Spearmana między długościami i szerokościami płatków i działek kielicha dla poszczególnych gatunków i dla całego zbioru danych.
- **Uczenie maszynowe**
 - Oryginalny zbiór *Iris* **podziel** na dane treningowe i testowe w stosunku 4:1 (`sample.split`).
 - Przeprowadź **klasyfikację** za pomocą drzew decyzyjnych (`ctree` → model, `predict, multiclass.roc` → print).
 - Przygotuj wykres drzewa decyzyjnego `plot(model)`.
 - Przeprowadź **klasyfikację** metodą *random forest* (`randomForest` → model, `predict, multiclass.roc` → print).
 - Z oryginalnych danych *Iris* usuń nazwy gatunków (kolumna *Species*).
 - Przeprowadź **klasteryzację** metodą *k-means* (`kmeans`).
 - Przygotuj rysunek poszczególnych przyporządkowań (oś x – długość działek, oś y – szerokość działek).
 - Powtórz **klasteryzację** metodą *k-mediod* (`pamk`).

Źródła:

http://rstudio-pubs-static.s3.amazonaws.com/450733_9a472ce9632f4ffbb2d6175aaeee5be6.html
<https://www.kaggle.com/mokosan/practice-from-r-and-data-mining-iris-dataset>

Zob. też Analiza głównych składowych (PCA): <https://www.rpubs.com/ann01/526159>

- Powyższe analizy wykonane zostały dla znanego wcześniej zbioru danych *Iris*. Przeprowadź analogiczne analizy dla nowego zbioru danych dot. pingwinów (`penguins`). Instalacja: `install.packages("palmerpenguins")`. Źródło: <https://github.com/allisonhorst/palmerpenguins>

Inne podobne zbiory danych do ćwiczeń: <https://www.meganstodel.com/posts/no-to-iris/> (tu także opisany został powód rezygnacji niektórych osób z danych *Iris*) oraz <https://medium.com/towards-artificial-intelligence/best-datasets-for-machine-learning-data-science-computer-vision-nlp-ai-c9541058cf4f> (wiele różnorodnych pakietów szczególnie dla uczenia maszynowego)

- Przygotuj 1000 elementowy wektor z wartościami funkcji $\sin^{10}(x)$ dla x z przedziału od 0.0 do 10.0. Następnie korzystając ze [średniej kroczącej](#) jako filtra wygładzającego przygotuj wektor z danymi wygładzonymi. Narysuj wartości oryginalne i wygładzone na wspólnym wykresie.

Prawo Benforda

- Zapisz do wektora 1000 kolejnych wyrazów ciągu geometrycznego np. kolejne potęgi dowolnej liczby q . Dla $q = 2$ będą to 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, itd. Następnie przygotuj funkcję, która przyjmie taki wektor liczb jako argument i obliczy częstość występowania liczb zaczynających się od kolejnych cyfr od 1 do 9. Zrób wykres słupkowy rozkładu tych częstości. Czy rozkład częstości pierwszych cyfr jest jednorodny?
- Użyj funkcji z poprzedniego zadania na zbiorze liczb Fibonacciego. Czy w tym przypadku rozkład jest jednorodny?
- Stwórz wektor zawierający powierzchnię i liczbę ludności w Państwach świata (źródło danych np. https://pl.wikipedia.org/wiki/Pa%C5%84stwa_%C5%9Bwiata). Sprawdź jak w przypadku tych danych wygląda rozkład częstości pierwszych cyfr.
- Powtórz analizę dla liczby ludności miast Polski (źródło danych np. https://pl.wikipedia.org/wiki/Dane_statystyczne_o_miastach_w_Polsce lub Baza Danych Lokalnych GUS <https://bdl.stat.gov.pl/BDL/start>).
- Powtórz analizę dla dowolnego ciągu arytmetycznego. Jaki rozkład jest w tym przypadku?
- Napisz aplikację *shiny*, która pozwala na wczytanie danych z pliku CSV, a następnie sprawdza, czy zachowany jest rozkład Benforda we wskazanej kolumnie.

Więcej zadań: <https://docs.ufpr.br/~marianakleina/rexercises-1-R-basic.pdf>