
Laboratorium 1

Konfiguracja środowiska, Python i pierwsze programy

Zainstaluj Pythona i środowisko programistyczne (na komputerze domowym, w pracowni już jest)

- a) Pythona należy zainstalować korzystając ze strony <https://www.python.org/downloads/> .
Taka czysta wersja Pythona już wystarcza do programowania, ale my chcemy sobie ułatwić sprawę i zainstalować przyjemne środowisko. Po udanym zainstalowaniu Pythona, przejdź do następnego podpunktu.
- b) W Pythonie wygodnie programuje się w specjalnych środowiskach programistycznych np.
- Visual Studio Code <https://code.visualstudio.com/> (środowisko dla wielu języków programowania, wg wielu rankingów to najlepszy program do kodowania, możliwość instalacji wielu rozszerzeń)
 - PyCharm <https://www.jetbrains.com/pycharm/> (wygodne środowisko w wersji darmowej i płatnej)
 - Spyder <https://www.spyder-ide.org/> (alternatywna dla PyCharm)
 - Anaconda <https://www.anaconda.com/products/individual> (wygodne zarządzanie paczkami i środowiskami)
 - Jupyter Online <https://jupyter.org/> (programowanie za pomocą notebooków pythonowych, online)
 - Google Colab https://colab.research.google.com/?utm_source=scs-index (notebook pythonowe z użyciem Googla, online)

Przejrzyj linki, wybierz najwygodniejsze dla Ciebie środowisko.

Poczytaj o Pythonie

Zapoznaj się z tutorialami dotyczącymi programowania w Pythonie:

- Fajny tutorial o programowaniu w Pythonie: <https://www.w3schools.com/python/default.asp>
Oferuje nawet możliwość testowania swoich programów online.
- Oficjalny tutorial Pythona po polsku: <https://docs.python.org/pl/3/tutorial/index.html>

- Jak ładnie pisać w Pythonie. Poradnik stylu PEP8:
<https://www.python.org/dev/peps/pep-0008/>
- Fajna książka z zadaniami do Pythona
[https://www.brianheinold.net/python/A Practical Introduction to Python Programming Heinold.pdf](https://www.brianheinold.net/python/A_Practical_Introduction_to_Python_Programming_Heinold.pdf)

Zadanie 1

- a) Stwórz funkcję `prime(n)`, która będzie sprawdzała, czy podana liczba `n` jest liczbą pierwszą. Jeśli tak, to zwróci `True`, w przeciwnym wypadku zwróci `false`.
Przetestuj czy funkcja działa dobrze np.

```
prime(3)
Wynik: True
prime(4)
Wynik: False
prime(49)
Wynik: False
```

- b) Stwórz drugą funkcję `select_primes(x)`, która dostanie listę `x` liczb naturalnych, odfiltruje z niej wszystkie liczby pierwsze i zwróci listę liczb pierwszych.
W środku funkcji `select_primes` można użyć funkcji `prime` z podpunktu a.

Przykład działania:

```
select_primes([3, 6, 11, 25, 19])
Wynik: [3, 11, 19]
```

Zadanie 2

W tym zadaniu wykonamy kilka komend matematyczno-statystycznych:

- Wczytaj dwa wektory z liczbami `[3, 8, 9, 10, 12]` oraz `[8, 7, 7, 5, 6]` (jako pythonowe listy). Następnie zwróć sumę tych wektorów oraz iloczyn (po współrzędnych) tych wektorów.
- Dla powyższych wektorów podaj iloczyn skalarny.
- Dla powyższych wektorów podaj ich długości euklidesowe (długość wektora jako strzałki w przestrzeni).
- Stwórz wektor 50 losowych liczb z zakresu od 1 do 100.
- Dla wektora z punktu (d) policz średnią z wszystkich jego liczb, min, max oraz odchylenie standardowe.
- Dokonaj normalizacji wektora z podpunktu (d) (ściskamy wszystkie liczby do przedziału `[0,1]`) za pomocą poniższego wzoru (x_i to liczba w starym wektorze na pozycji i , a z_i to liczba w nowym wektorze na pozycji i)

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

W oryginalnym wektorze jakie było max? Na której pozycji stało? Jaka liczba stoi na tej pozycji w nowym wektorze?

- g) Dokonaj standaryzacji wektora z podpunktu (d). Wzór na standaryzację wykorzystuje średnią i odchylenie standardowe:

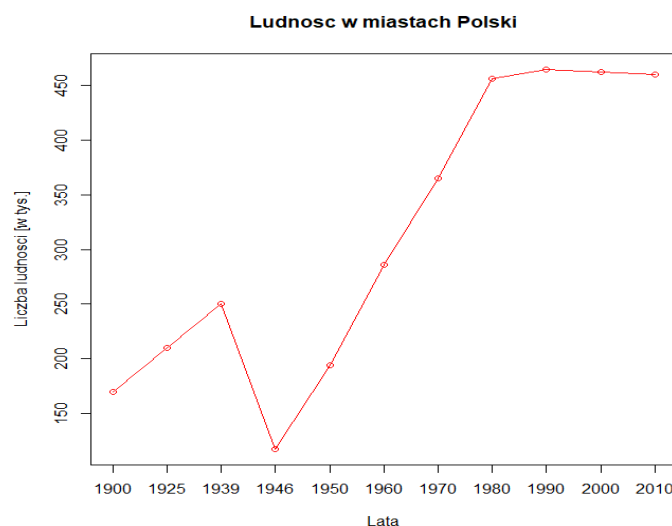
$$z_i = \frac{x_i - \text{mean}(x)}{\sigma(x)}$$

Jaką średnią i odchylenie standardowe ma nowy wektor z?

Zadanie 3

W tym zadaniu zrobimy parę operacji na bazie danych i stworzymy wykres. Możesz wykorzystać paczkę `pandas` do obsługi baz danych i `matplotlib` do robienia wykresu¹.

- Załaduj plik `miasta.csv` do programu w Pythonie i zapisz go pod nazwą `miasta`. Wyświetl tabelę z danymi: `miasta`. Sprawdź jak wyglądają czyste wartości tej tabeli tzn. `miasta.values`
- Dodaj za pomocą odpowiedniej instrukcji wiersz do tabeli z ludnością w 2010 roku: 2010, 460, 555, 405
- Stwórz wykres dla ludności Gdańska (skorzystaj z paczki `matplotlib`). Dodaj odpowiednie oznaczenia osi, tytuły. Wykres ma być liniowy z punktami i w kolorze czerwonym. Powinno wyjść coś następującego:



- Stwórz dodatkowo wykres, który będzie zestawiał zmiany ludności wszystkich miast na jednym wykresie w różnych kolorach. Dodaj legendę.

Dla chętnych

Poniższe taski nie są obowiązkowe, ale mogą ułatwić Ci pracę w przyszłości:

- Zintegruj PyCharm z Githubem, tak aby Twoje zadania łatwo można było wyeksportować do repozytorium online. Jak to zrobić?
- Sprawdź możliwość tworzenia notebooków Jupyterowych w PyCharmie.

¹ Paczkę najłatwiej instaluje się z terminala (z reguły dostępny w środowisku), gdzie wpisujemy komendę `pip install nazwapaczki`