

# Laboratorium: Python i wizualizacja

Wojciech Jaśkowski

termin: +1 tydzień, waga: 2

## 1 Motywacja

Istotnym elementem sprawnej komunikacji człowiek-komputer jest wizualizacja danych. Python i narzędzia zbudowane wokół dostarczają wygodnej platformy do obróbki, **interaktywnej** analizy i wizualizacji danych.

## 2 Zadanie

Dokonaj wizualizacji danych pochodzących z eksperymentów ewolucyjnych. Napisz w Pythonie program, który:

1. Wczytaj [dane z plików](#). Zawierają one wyniki działania pięciu różnych algorytmów ewolucyjnych<sup>1</sup>. Każdy algorytm został uruchomiony 32 razy. 'Czas' (oś OX) jest mierzony w rozegranych grach (max ok. 500.000) oraz w pokoleniach (max 200).
2. Wygeneruje wykresy wizualizujące:
  - (a) przebiegi w funkcji 'czasu' algorytmów
  - (b) wyniki końcowe algorytmów.

Oczekiwane wyniki wizualizacji (w zależności od poziomu aspiracji) przedstawione są poniżej. Do wizualizacji użyj biblioteki [matplotlib](#).

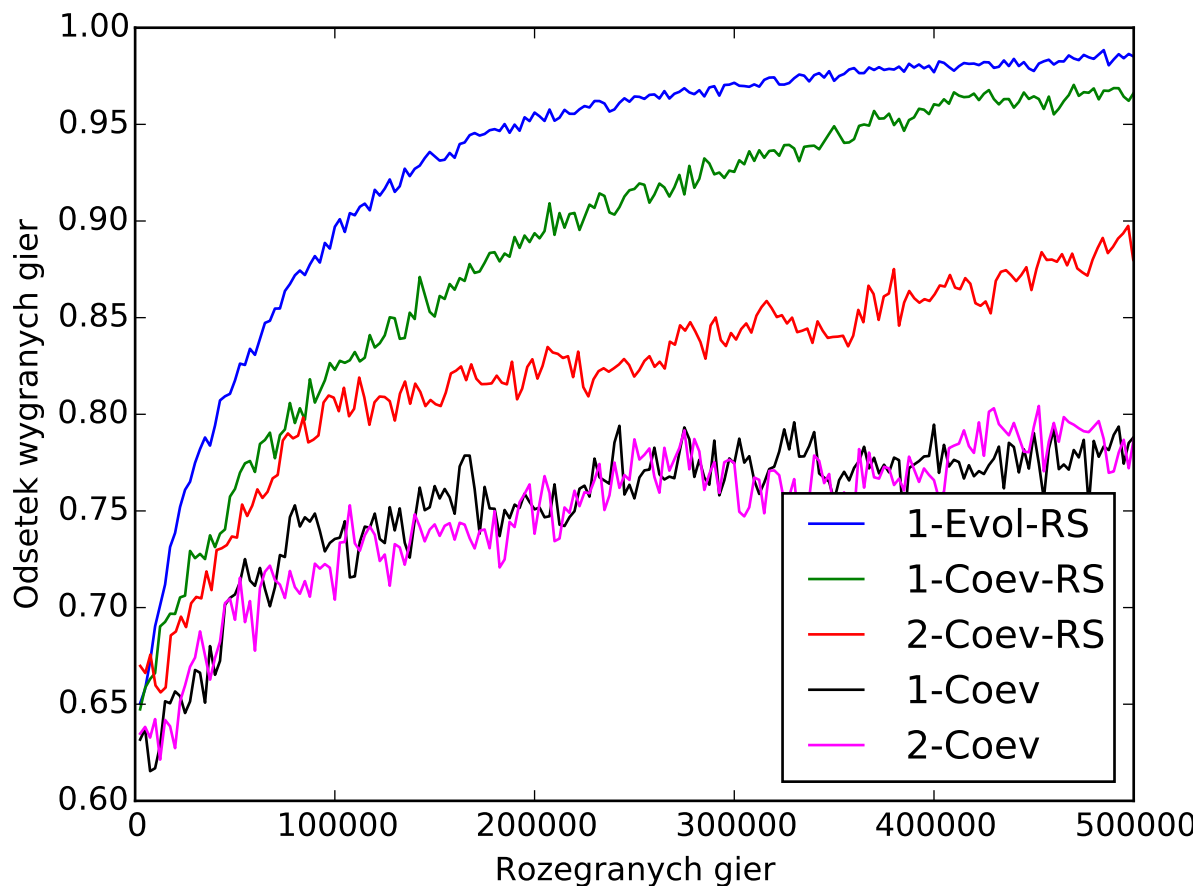
### 2.1 Uwagi

- Graficzne wyjaśnienie co widzimy na wykresie pudełkowym: [http://en.wikipedia.org/wiki/Interquartile\\_range](http://en.wikipedia.org/wiki/Interquartile_range). Czerwona kreska oznacza medianę, a kropka — wartość średnią.

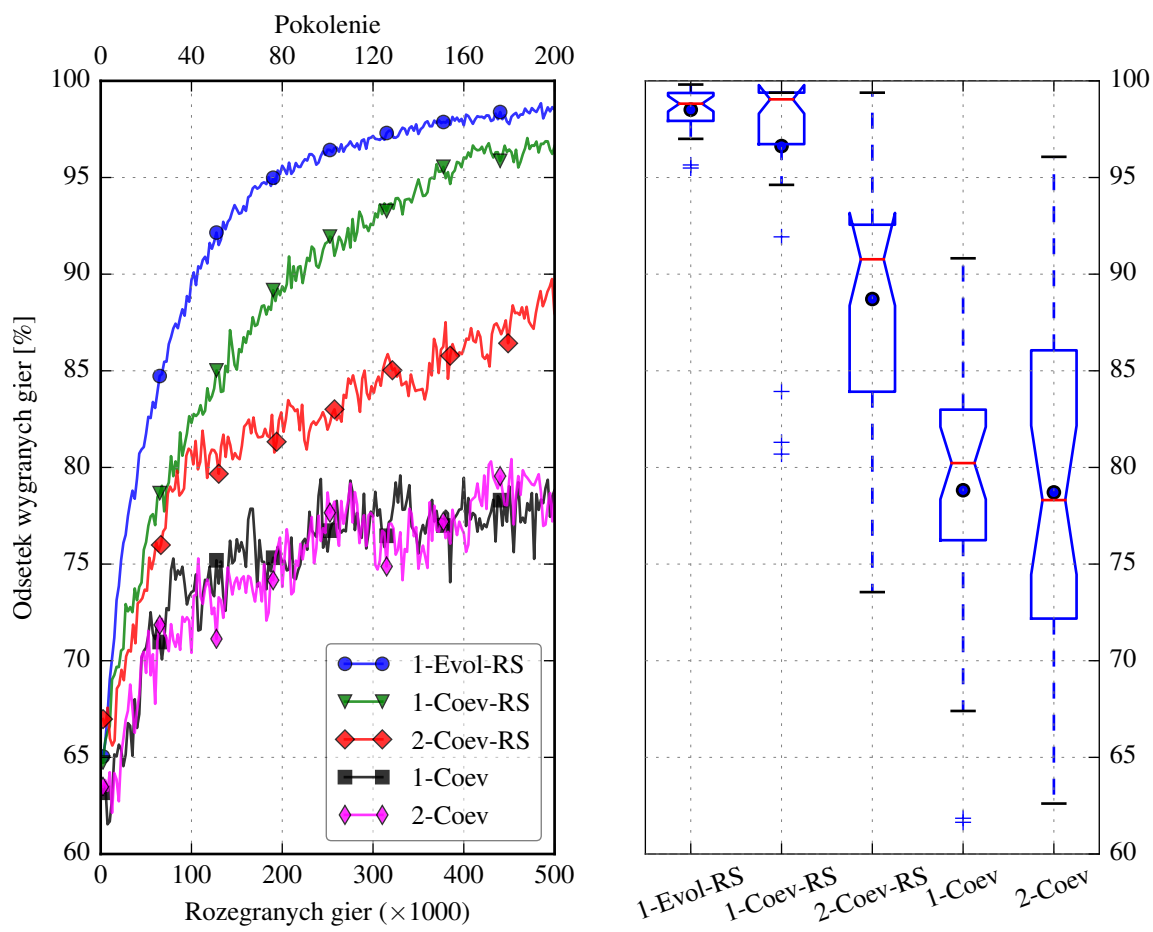
---

<sup>1</sup>Algorytmy te szukały strategii gry dla uogólnionego problemu iterowanego dylematu więźnia. Były to warianty algorytmów ewolucyjnych i koewolucyjnych.

## 2.2 Wizualizacja na 3.0



## 2.3 Wizualizacja na 5.0 (proszę zwrócić uwagę na czcionkę)



## 2.4 Od czego zacząć?

Listing 1: Some simple plot

---

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt

def main():
    plt.figure(figsize=(3, 3))

    plt.plot([100, 200, 300, 400], [0.1, 0.2, 0.8, 0.9])
    plt.savefig('myplot.pdf')
    plt.close()

if __name__ == '__main__':
    main()
```

---

Sugerowane jest zrobienie zadania w notebooku z włączoną flagą `%matplotlib inline`

## 2.5 Podpowiedzi

- Matplotlib.pyplot:  
[http://matplotlib.org/api/pyplot\\_summary.html](http://matplotlib.org/api/pyplot_summary.html)
- Szerokość rysunku to ok. 480 punktów, czyli ok. 6.7 cala.
- Wykres liniowy:  
[http://matplotlib.org/api/pyplot\\_api.html#matplotlib.pyplot.plot](http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)
- Formatowanie osi:  
[http://matplotlib.org/examples/pylab\\_examples/custom\\_ticker1.html](http://matplotlib.org/examples/pylab_examples/custom_ticker1.html)
- Wykres pudełkowy:  
[http://matplotlib.org/api/pyplot\\_api.html#matplotlib.pyplot.boxplot](http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.boxplot)
- Dwa wykresy na jednym obrazku:  
[http://matplotlib.org/api/pyplot\\_api.html#matplotlib.pyplot.subplot](http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.subplot)
- Dodanie kropek najłatwiej osiągnąć za pomocą nałożenia wykresu punktowego:  
[http://matplotlib.org/api/pyplot\\_api.html#matplotlib.pyplot.scatter](http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.scatter)
- Markery co N punktów:  
[http://matplotlib.org/api/artist\\_api.html#matplotlib.lines.Line2D.set\\_markevery](http://matplotlib.org/api/artist_api.html#matplotlib.lines.Line2D.set_markevery)