

Jeden wykres wart jest tysiąca słów, czyli o wizualizacji danych w Pythonie

Marcin Kawka

10 grudnia 2019



PyLightwaw



O mnie

- pracuję w Instytucie Ochrony Środowiska - PIB
- a konkretniej w Zespole Modelowania Atmosfery i Klimatu
- Na co dzień: **NumPy**, **Pandas**, **netCDF**, SciPy, scikit-learn, pyqgis
- poza pythonem: R, bash, PostgreSQL, FORTRAN

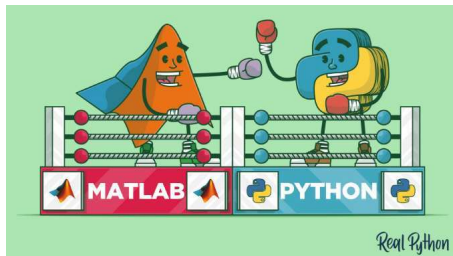
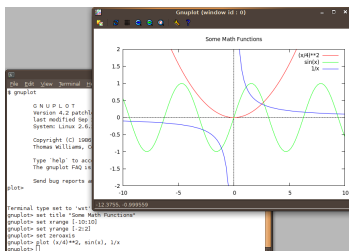
Dziś poznamy

- 1 Matplotlib
- 2 Seaborn
- 3 Dash
- 4 Cartopy

Prehistoria

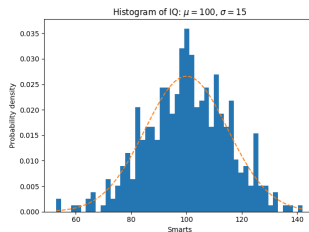
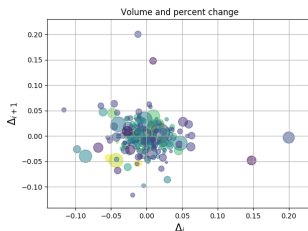
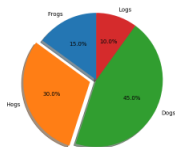
Zanim *datascience* stał się popularny

- MATLAB (1984) i pochodne: Octave (1988), Scilab (1990), ...
- GNU Plot (1986)
- R (1993) wraz z ggplot (2007)
- narzędzia dedykowane do wąskich zastosowań: NCL(+), SAS (1976), IDL (1977)



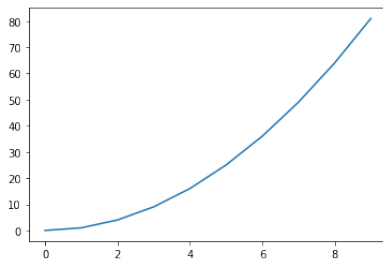
Matplotlib

- Pierwsze wydanie w 2003 roku
- Cel: zbliżenie IPython do MATLABa
- Wada: surowy wygląd wykresów (zwłaszcza domyślny)
- Zaleta: możliwość nadbudowy przez *toolkits*

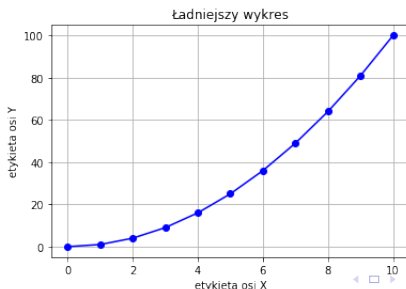


#Prosty przykład

```
import matplotlib.pyplot as plt  
x=[i for i in range(10)]  
y=[i**2 for i in range(10)]  
plt.plot(x,y)
```

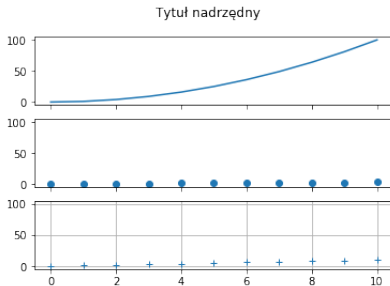


```
import matplotlib.pyplot as plt
import numpy as np
x=np.linspace(0,10,11)
plt.plot(x,x**2, 'b-o')
plt.grid(True)
plt.xlabel('etykieta osi X')
plt.ylabel('etykieta osi Y')
plt.title('Ładniejszy wykres')
```



Dwa główne obiekty matplotlib to *figure* oraz *axis*

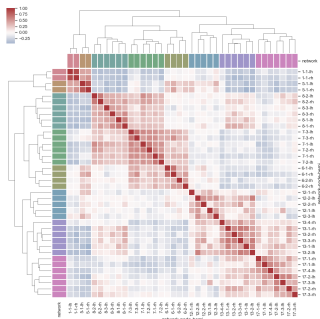
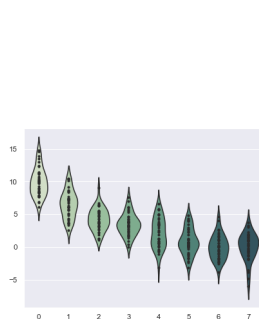
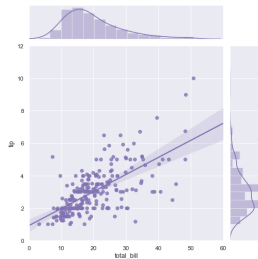
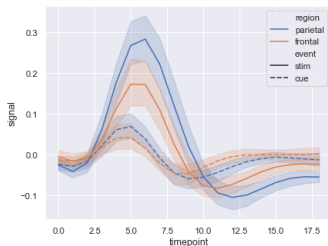
```
fig, axs = plt.subplots(3, sharex=True, sharey=True)
fig.suptitle('Tytuł nadrzędny')
axs[0].plot(x, x ** 2)
axs[1].plot(x, 0.3 * x, 'o')
axs[2].plot(x, x, '+')
axs[2].grid(True)
```



Seaborn

- Pierwsze wydanie w 2012 roku (obecnie wersja 0.9)
- Opracowany jako nadbudowa na matplotlib
- Silnie zintegrowany z pandasem
- Mocno inspirowany R
- Zaleta: domślnie ładny wygląd wykresów
- Więcej nietypowych wykresów

Przykład wykresów z Seaborn



Dash

Zbudowany z użyciem *Flask*, *React.js*, framework do budowy interaktywnych aplikacji webowych. W skład dasha wchodzi następujące komponenty:

- *Dash Core Components* - wykresy, suwaki, listy
- *Dash HTML Components* - elementy HTML: DIV, H1, itp.
- *Dash DataTable* - tabelki do raportowania (listopad 2018)
- *Dash Cytoscape* - wizualizacja grafów (luty 2019)
- *Dash Bio Components* - wizualizacje bioinformatyczne (kwiecień 2019)

Cartopy

- Toolkit zbudowany na bazie matplotlib
- Zamienia współrzędne przestrzenne np. długość i szerokość geograficzną na współrzędne płaskie XY na obszarze wykresu matplotlib

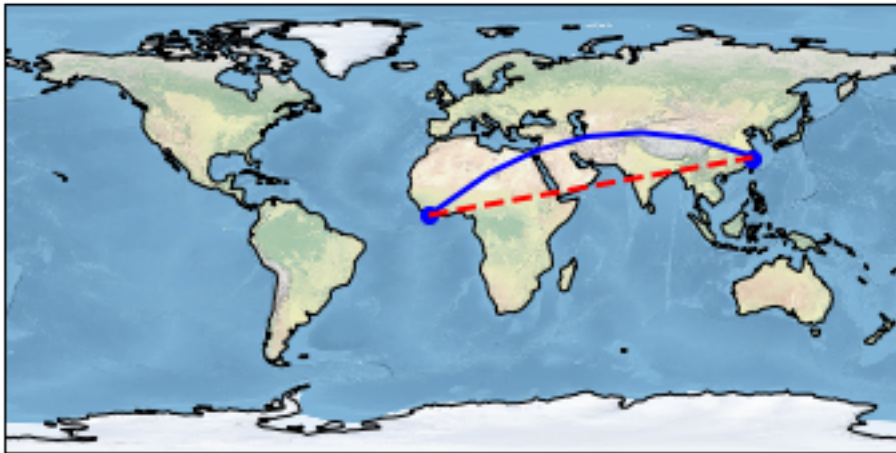
Dodatkowo

- wyświetlanie danych wektorowych (punkty, linie, wektory), jak i rastrowych (pola temperatury, stężeń itp.)
- możliwość dodawania map podkładowych (np. OpenStreetMap)

Cartopy - przykład

```
import cartopy.crs as ccrs
import matplotlib.pyplot as plt
ax = plt.axes(projection=ccrs.PlateCarree())
ax.coastlines()
ax.stock_img()
liberia_lat, liberia_lon = 5.106076, -9.068785
wenling_lat, wenling_lon = 28.371992, 121.385069
#czerwona linia prosta w matplotlib
plt.plot([liberia_lon, wenling_lon], [liberia_lat, wenling_lat],
         color='r', linewidth=2, linestyle='—')
#niebieska linia prosta w rzeczywistosci
plt.plot([liberia_lon, wenling_lon], [liberia_lat, wenling_lat],
         color='b', linewidth=2, marker='o',
         transform=ccrs.Geodetic())
plt.show()
```

Cartopy - przykład



Na zakończenie - linki

Materiały z dzisiejszej prezentacji:

- <https://github.com/marcinkawka/pylight>

Galerie z przykładowymi wykresami wraz z kodem:

- <https://matplotlib.org/3.1.0/gallery/index.html>
- <https://seaborn.pydata.org/>
- <https://dash-gallery.plotly.host/Portal/>
- <https://scitools.org.uk/cartopy/docs/latest/gallery/index.html>

Przykłady *naszych* wizualizacji

- <http://ios.edu.pl/jakosc-powietrza/>
- <http://klimada2.ios.gov.pl/klimat-scenariusze/>