Pyladies - matplotlib - exercises

April 24, 2017

In [ ]:
```
# matplotlib
# installation
# lepiej pracowac w pliku i wywolywac plik po dokonaniu zmian (chyba ze w jupyter noteb
# import do podstawowych wykresow (tzw. MATLAB interface)
import matplotlib.pyplot as plt
# inne submoduly np. do animacji matplotlib.animation,
# do pozyskania roznych ksztaltow matplotlib.patches,
# do pelniejszej kontroli nad lokalizacja i rozmiarem elementow matplotlib.offsetbox
%matplotlib inline
```

In [ ]:
```
# do latwej pracy na danych tabelarycznych
import pandas as pd
import numpy as np

# przykladowe zestawy danych (z R)
from sklearn import datasets
```

In [ ]:
```
# prosty wykres liniowy
x = np.arange(-5, 6)
x
```

In [ ]:
```
y = x ** 2
y
```

In [ ]:
```
plt.plot(x, y) # plt.show()
```

In [ ]:
```
# dodanie opisu osi
plt.xlabel('x') #plt.show()
```

In [ ]:
```
plt.plot(x, y)
plt.xlabel('x') #plt.show()
```

In [ ]:
```
plt.plot(x, y, c='green', label='data1') # c - kolor
plt.xlabel('x') # os x
plt.ylabel('y') # os y
plt.xlim([-5,5]) # zakres osi x
plt.axvline(x=0, c = 'red') # pionowa linia na x =0
plt.title('Tytul') # tytul
plt.legend() #legenda - label! plt.legend(loc=0/1/2/3/4) - lokalizacja
```

```
In [ ]: # kolory i style
        # grubosc linii
        plt.plot(x, x+1, color="red", linewidth=0.25)
        plt.plot(x, x+2, color="red", linewidth=0.50)
        plt.plot(x, x+3, color="red", linewidth=1.00)
        plt.plot(x, x+4, color="red", linewidth=2.00)

        # styl linii
        plt.plot(x, x+5, color="green", lw=3, linestyle='-')
        plt.plot(x, x+6, color="green", lw=3, ls='-.')
        plt.plot(x, x+7, color="green", lw=3, ls=':')

        # znaczniki na linii
        plt.plot(x, x+ 9, color="blue", lw=3, ls='-', marker='+')
        plt.plot(x, x+10, color="blue", lw=3, ls='--', marker='o')
        plt.plot(x, x+11, color="blue", lw=3, ls='-', marker='s')

        # rozmiar i kolor znacznika
        plt.plot(x, x+13, color="purple", lw=1, ls='-', marker='o', markersize=2)
        plt.plot(x, x+14, color="purple", lw=1, ls='-', marker='o', markersize=4)
        plt.plot(x, x+15, color="purple", lw=1, ls='-', marker='o', markersize=8, markerfacecolo
        plt.plot(x, x+16, color="purple", lw=1, ls='-', marker='s', markersize=8,
                markerfacecolor="yellow", markeredgewidth=3, markeredgecolor="green");
```

1 cwiczenie 1

stworz wykres liniowy, gdzie zakresy x [-100,100], y [-10,10]
- y = sin(x)
- wykres czerwony
- legenda, gdzie wykres opisany jest jako sin(x)
- tytu - Sin
- opis osi x - x
- opis osi y - sin(x)
- linia pionowa czarna w x = 0

```
In [ ]: # kilka wykresow osobno
        # plt.subplot(nrows, ncols, plot_number)
        plt.subplot(2,2,1)
        plt.plot(x, y, 'r')
        plt.subplot(2,2,2)
        plt.plot(y, x, 'g')
        plt.subplot(2,2,3)
        plt.plot(x, y, 'b')
        plt.subplot(2,2,4)
        plt.plot(y, x, 'y')
```

```
In [ ]: #kilka wykresow razem
        # kilka wykresow osobno
```

```
        # plt.subplot(nrows, ncols, plot_number)

        plt.plot(x, y + 10, 'r')
        plt.plot(y + 10, x, 'g')
        plt.plot(x, y, 'b')
        plt.plot(y, x, 'y')
```

```
In [ ]: # zeby bylo latwiej - podejscie obiektowe
        fig = plt.figure() # figura zaczyna si w punkcie 0,0 i ma 1 wysokosci i 1 szerokosci
        #miejsce dla wykresu
        axes = fig.add_axes([0, 0, 0.5, 0.5]) # (lewe ograniczenie, dolne ograniczenie, szerokos
        axes.plot(x, x ** 2, 'b')
        axes.set_xlabel('x') # wszedzie metody zaczynaja sie od set_
        axes.set_ylabel('y')
        axes.set_title('tytul')
```

```
In [ ]: fig = plt.figure()
        # wykres1
        axes = fig.add_axes([0, 0, 0.5, 0.5])
        axes.plot(x, x ** 2, 'b')
        axes.set_xlabel('x')
        axes.set_ylabel('y')
        axes.set_title('tytul')
        # wykres2
        axes2 = fig.add_axes([0.5, 0.5, 0.5, 0.5])
        axes2.plot(x, x ** 2, 'b')
        axes2.set_xlabel('x')
        axes2.set_ylabel('y')
        axes2.set_title('tytul')
```

```
In [ ]: fig = plt.figure()
        # wykres1
        axes = fig.add_axes([0, 0, 0.4, 0.4])
        axes.plot(x, x ** 2, 'b')
        axes.set_xlabel('x')
        axes.set_ylabel('y')
        axes.set_title('tytul')
        # wykres2
        axes2 = fig.add_axes([0.5, 0.5, 0.4, 0.4])
        axes2.plot(x, x ** 2, 'b')
        axes2.set_xlabel('x')
        axes2.set_ylabel('y')
        axes2.set_title('tytul')
```

```
In [ ]: # wykresy moga byc 'w sobie'
        # canvas
        fig = plt.figure()

        ax1 = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # zewnetrzna
```

```
        ax2 = fig.add_axes([0.2, 0.5, 0.2, 0.3]) # wewnetrzna

        # Larger Figure Axes 1
        ax1.plot(x, x, 'r')
        ax1.set_xlabel('x1')
        ax1.set_ylabel('y1')
        ax1.set_title('tytul1')

        # Insert Figure Axes 2
        ax2.plot(x, y, 'g')
        ax2.set_xlabel('x2')
        ax2.set_ylabel('y2')
        ax2.set_title('tytul2');
```

```
In [ ]: #subplots to ulatwia
        # 2 subwykresy
        fig, axs = plt.subplots(nrows=1, ncols=2)
        for ax in axs:
            ax.plot(x, y, 'r')
            ax.set_xlabel('x')
            ax.set_ylabel('y')
            ax.set_title('tytul')
```

```
In [ ]: #nachodzace sie subwykresy?
        fig, axs = plt.subplots(nrows=1, ncols=2)
        for ax in axs:
            ax.plot(x, y, 'r')
            ax.set_xlabel('x')
            ax.set_ylabel('y')
            ax.set_title('tytul')
        plt.tight_layout()
```

```
In [ ]: #rozmiar wykresu
        fig = plt.figure(figsize=(8,4), dpi=100)
        ax1 = fig.add_axes([0.1, 0.1, 0.8, 0.8])
        ax1.plot(x, y, 'r')
        ax1.set_xlabel('x')
        ax1.set_ylabel('y')
        ax1.set_title('tytul')
```

```
In [ ]: # zapisywanie do pliku
        fig.savefig("filename.png", dpi=200)
        # PNG, JPG, EPS, SVG, PGF and PDF
```

2 cwiczenie 2

stworz figure skladajaca sie z 4 wykresow o rownym rozmiarze
na wykresach przestaw x ** 2, x ** 3, x ** 4, x ** 5 dla x [-5, 5]

pamietaj zeby dopasowac os y jezeli zostanie zle dopasowana.
nadaj kazdemu wykresowi tytul, niech kazdy wykres ma inny kolor (lub styl linii)
uwazaj zeby wykresy nie nachodzily na siebie

```
In [ ]: # inne typy wykresow
        # scatterplot
        plt.scatter(x, x**2)
```

```
In [ ]: from random import sample
        data = sample(range(1, 101), 50)
        plt.hist(data)
```

```
In [ ]: plt.boxplot(x,vert=True,patch_artist=True)
```

```
In [ ]: # jakis wykres ktory cos pokazuje - dodatek
        # dane o kwiatkach
        iris = datasets.load_iris()
        df = pd.DataFrame(iris.data, columns = iris.feature_names)
        df['type'] = iris.target
        df.head()
```

```
In [ ]: plot1 = plt.scatter(df['sepal length (cm)'], df['petal length (cm)'])
```

```
In [ ]: colors = df['type'].replace(0, 'red').replace(1, 'blue').replace(2, 'green')
```

```
In [ ]: plot1 = plt.scatter(df['sepal length (cm)'], df['petal length (cm)'], c = colors)
```

```
In [ ]: plot1 = plt.scatter(df['sepal length (cm)'], df['petal length (cm)'], c = colors)
        plt.xlabel('sepal length (cm)')
        plt.ylabel('petal length (cm)')
```

```
In [ ]: plot1 = plt.scatter(df['sepal length (cm)'], df['petal length (cm)'], c = colors)
        plt.xlabel('sepal length (cm)')
        plt.ylabel('petal length (cm)')
        plt.legend()
```

```
In [ ]: df['type'] == 0
```

```
In [ ]: plot1 = plt.scatter(df[df['type'] == 0]['sepal length (cm)'], df[df['type'] == 0]['petal
        plot2 = plt.scatter(df[df['type'] == 1]['sepal length (cm)'], df[df['type'] == 1]['petal
        plot3 = plt.scatter(df[df['type'] == 2]['sepal length (cm)'], df[df['type'] == 2]['petal
        plt.xlabel('sepal length (cm)')
        plt.ylabel('petal length (cm)')
        plt.legend()
```

```
In [ ]: plot1 = plt.scatter(df[df['type'] == 0]['sepal length (cm)'], df[df['type'] == 0]['petal
        plot2 = plt.scatter(df[df['type'] == 1]['sepal length (cm)'], df[df['type'] == 1]['petal
        plot3 = plt.scatter(df[df['type'] == 2]['sepal length (cm)'], df[df['type'] == 2]['petal
        plt.xlabel('sepal length (cm)')
        plt.ylabel('petal length (cm)')
        plt.legend([plot1, plot2, plot3], ['setosa', 'versicolor', 'virginica'])
```

```
In [ ]: # calosc kodu
        import matplotlib.pyplot as plt
        import pandas as pd
        from sklearn import datasets
        iris = datasets.load_iris()
        frame = pd.DataFrame(iris.data, columns = iris.feature_names)
        frame['type'] = iris.target
        plot1 = plt.scatter(frame[frame['type'] == 0]['sepal length (cm)'], frame[frame['type']
        plot2 = plt.scatter(frame[frame['type'] == 1]['sepal length (cm)'], frame[frame['type']
        plot3 = plt.scatter(frame[frame['type'] == 2]['sepal length (cm)'], frame[frame['type']
        plt.xlabel('sepal length (cm)')
        plt.ylabel('petal length (cm)')
        plt.legend([plot1, plot2, plot3], ['setosa', 'versicolor', 'virginica'])
        plt.title('Iris sepal/petal length')
        plt.show()
```

3 zasady projektowania wykresow

Oszukiwanie w graficznej reprezentacji danych:

1) pokazywanie zbyt wielu informacji na pojedynczym wykresie (szczegolnie informacji zbednych).

2) pokazywanie nie wstarczajcej informacji, zeby ocenic problem

3) zniekształcanie danych (np. poprzez zle dobranie wykresu, skali itd)

Cairo, A. (2015). Graphics lies, misleading visuals. In New Challenges for Data Design (pp. 103-116). Springer London.

4 zadanie domowe

znalezc zestaw danych, wymyslic pytanie na ktore wykres ma nam pomoc odpowiedziec i stworzyc odpowiednio dobrany wykres :)

5 jeszcze jedne zajecia

1) na podstawie Waszych zada domowych, wchodzimy glebiej w matplotlib i je upiekszamy, poprawiamy itd.

2) wykresy interaktywne/wykresy 3D

3) inne biblioteki do wizualizacji danych

4) hands on data, praca na zestawie danych od postaw (laczenie danych z roznych plikow, latanie dziur itd.), analiza i wizualizacja

5) krotki wstep do machine learningu w Pythonie (wiecej teorii mniej kodu)

```
In [ ]:
```