

Kako fitovati i plotovati sa pythonom.

lazar_zivadinovic@hotmail.com ver: 1.01

ssh -X polaznik@titan.aob.rs # komanda za logovanje na titan

scp polaznik@titan.aob.rs:./vezbovni/ime/fajl.png fajl.png

```
import numpy as np #Uvodimo modul za rad sa nizovima i matricama
from scipy.optimize import curve_fit #Uvodimo funkciju za fitovanje
import matplotlib.pyplot as plt #Uvodimo modul za plotovanje

xdata, ydata, greske_po_y = np.loadtxt('ime_fajla.txt', unpack=True)
#ucitava vase podatke, po kolonama, u nizove (mozete nizove nazvati kako hocete)

def func(x, a, b):
    return x*a + b

#iznad definisemo funkciju kojom zelimo da fitujemo
#uvom slucaju je to linearna funkcija koja zavisi od dva parametra
#mozemo definisati i razne druge funkcije ali tako da im prosledjujemo argumente
#sledecim redom
#(x, a, b, c, d, e), gde je x promenljiva, a slova parametri
#ako zelimo da fitujemo kvadratnom funkcijom, nju definisemo na sledeci nacin
'''
def kvad(x,a,b,c):
    return a*x**2 + b*x + c
'''
#obratite paznju na to da ce nam onda funkcija curve_fit vratiti 3 parametra

parametri, cov_m = curve_fit(func, xdata, ydata, sigma=greske_po_y)
#ovako se poziva funkcija za fitovanje
#parametri koje prosledjujemo funkciji su sledeci:
#ime nase funkcije (func, ili kvad ili nesto trece)
#nasa merenja po x osi
#nasa merenja po y osi
#nase greske. Ovo sigma=neki_niz nam kaze da ce za greske uzeti niz koji prosledimo
#ovo je opcioni parametar i kada nemamo greske (onda nemamo pojma sta radimo)
#onda ne moramo proslediti nista

greske_parametara = np.sqrt(np.diag(cov_m))
#cov_m je zapravo matrica greske (covariance matrix)
#clanovi na dijagonali te matrice su nase greske, ali kvadrirane
#zato formulom iznad kazemo sledece:greske_p = koren iz dijagonalnih clanova matrice

print('Parametar a je: ', parametri[0], '+/-', greske_parametara[0])
print('Parametar b je: ', parametri[1], '+/-', greske_parametara[1])

plt.plot(modelirano_x, modelirano_y, '-r', label='fit')
#modelirano_x vam je opseg na kom zelite da prikazete fit
#modelirano_y su vam vrednosti tog fita
#modelirano_x moramo da generisemo nekom metodom, i treba da nam bude
#u opsegu koji je malo veci od nasih merenja. Da bi bilo lepo prikazano.
#modelirano_y definisemo kao nasa funkcija(modelirano_y, parametri[0], parametri[1])

plt.errorbar(xdata, ydata, yerr=greske_po_y, fmt='o') #plotuje greske vasih merenja
#prosledjujete vase MERENE podatke i vase greske!!!
plt.legend(loc='best') #lokacija legende
plt.xlabel('X-OSA') #Naziv X ose
plt.ylabel('Y-OSA') #Naziv Y ose
plt.xlim([x1,x2]) #Podesavamo opseg x ose
plt.ylim([y1,y2]) #Podesavamo opseg y ose
plt.title('NAZIV') #Naziv grafika
plt.savefig('ime_plota.png') #cuva plot u nasem folderu
plt.show() #Prikazuje nam plot
```