Python, III deo:
PyLab i Jupyter

Šta je PyLab?

- environment koji cine
  - NumPy
  - SciPy
  - Matplotlib
  - IPython
- kako se instalira pylab?
- Ubuntu: Software Center ili Synaptic
- win: https://www.anaconda.com/products/individual

IDE? Spyder!

- potrebno?
- kako kome, kako za šta ...
- IPython3 meni sasvim dobar ...
- ... postoji i Spyder
- https://pypi.python.org/pypi/spyder
- ako nema dovucite iz repository ...
- ima i pod win, Anaconda

NumPy

- http://www.numpy.org/
- osnovna biblioteka za numerički zahtevne Python primene, sadrži:
  1. N-dimensional array object
  2. array slicing methods
  3. array reshaping methods

  i module za:
  1. basic linear algebra functions
  2. basic Fourier transforms
  3. advanced random number capabilities
- pokrenete IDLE
- `import numpy`
- `dir(numpy)`
- `help(numpy)`
- `del numpy`

NumPy

- obradićemo, sve, naravno, imamo vremena, ...
- evolutivno nastao, „haotično“ iz Numeric i NumArray
- nema načina da se zapamti i nije „zauvek“
- potreban nov način učenja i snalaženja
- izbeći reinventing
- masovno korišćenje help-a i primera
- Matlab-Python-R

SciPy

- http://www.scipy.org/
- scientific lib za Python, zavisi od NumPy
- nekoliko modula in a single package, kao i NumPy
- moduli za:
  1. statistics
  2. optimization
  3. numerical integration
  4. linear algebra
  5. Fourier transforms
  6. signal processing
  7. image processing
  8. ODE solvers
  9. special functions

SciPy

- i dalje u `IDLE`
- `import scipy`
- `dir(scipy)`
- `help(scipy)`
- `del scipy`
- dobro razmislite pre nego što krenete u reinvent!
- ja ovo ne mogu da zapamtim, sto zapamtim zaboravim
- pomenuh li nov način učenja?
- kako organizovati informacije, previše ih je (i kratko traju)
- “Numerical Recipes”

matplotlib

- http://matplotlib.sourceforge.net/, sjajan sajt
- package sa dugim nizom modula
- jako dobro se vidi struktuiran namespace, package.modul
- ono što ima sjajno je, a sada ima skoro sve
- galerija i primeri
- uputstvo, 3.2.1, pdf, 2598 strana, 19.03.2020.
- sintaksa vrlo liči na Matlab i Octave

IPython i Jupyter

- http://ipython.org/
- https://jupyter.org/
- interaktivni Python envirinment
- autocompletion by tab
- doteruje komande da budu shvaćene
- od mnogo mogućnosti: store, history, logging, ...
- Jupyter, isto samo u web browser okruženju!
- web server, može da se izvršava na nekoj drugoj mašini

PyLab

- sve prethodno zgodno spakovano
- sređen namespace da ne mora puno dot notation
- sintaksa jako liči na Octave/Matlab
- počinjemo, komandna linija, terminal
- `ipython3 --pylab`

PyLab, IPython, osnovno

```
help()
?
%quickref
a = 3
b = 'string'
lista = [1, 2, 3]
li<tab>
whos
```

PyLab, IPython, store

```
store a
store b
store lista
del a
del b
a
b
store -r
a
b
store -z
store
```

Pylab, matematika, konstante

```
pi
e
j
1j
exp(1j * pi)
math.exp(1j * pi)
help(exp)
help(math.exp)
exp?
math.exp?
e**(1j * pi) + 1
```

Pylab, matrice 1

```
a = array([[1, 2], [3, 4]])
a
a.size
a.shape
a.ndim
a.dtype
a.dtype.name
a.itemsize
a.transpose()
transpose(a)
a.T
```

Pylab, matrice 2

```
det(a)
eig(a)
b = eig(a)
type(b)
len(b)
type(b[0])
type(b[1])
c, d = b
c.size
c.shape
c.ndim
c.dtype.name
c.itemsize
```

Pylab, matrice 3, inverzija i množenje

```
aa = inv(a)
aa
aa.dtype.name
aa * a
a * aa
dot(a, aa)
dot(aa, a)
x = arange(10)
x
print(x)
dot(x, x)
```

Sve operacije su **elementwise**!!!
Velika razlika u odnosu na **Octave**!!!

Pylab, gde je dot, tu je cross

```
i = array([1, 0, 0])
j = array([0, 1, 0])
print(dot(i, j), dot(j, i))
print(cross(i, j))
print(cross(j, i))
cross?
source(cross)
```

Poznato od nekud? Rekoh da već ima, . . .
Teško je naći potpuno nov primer . . .
Reinventing problem! Google pomaže puno!

Pylab, rang!!!

```
a = array([[1, 1], [1, 1]])
rank(a)    # ???
help(rank)    # sad jasno?
linalg.matrix_rank(a)
```

Pazite!!!
Nažalost, lično iskustvo, ne tuđe!

Pylab, inicijalizacija, neke posebne matrice

```
I = eye(3)
print(I)
nule = zeros(4)
print(nule)
zeros?
zeros((4, 2))
zeros(4, 2)     # pazite!
zeros((4, 2), dtype = int)
zeros((4, 2), dtype = complex)
ones([3, 4])
empty([6, 4])
empty?
```

Pylab, još elementwise operacija

```
a + 1
a * a
a**3
a / 2
a / 2.
```

Pazite na / kod Python 2!

Pylab, reshape

```
help(reshape)
print(a)
a.reshape(1, 4)
a.reshape(4, 1)
reshape(a, (1, 4))
reshape(a, (4, 1))
a.reshape(2, 3)
a.reshape(4)
a.reshape((4, ))
reshape(a, 4)
reshape(a, (4, ))
```

Pylab, in place array operations

```
xx = arange(20)
print(xx)
xx += 1
print(xx)
xx = xx.reshape(4, 5)
print(xx)
xx -= 1
print(xx)
xx *= 2
print(xx)
```

Pylab, transpose, revisited

```
help(transpose)
xx = arange(5) + 1
print(xx)
xx.shape
xx.ndim
xx = xx.transpose()
xx.shape
print(xx)
xx = xx.reshape(1, 5)
xx.ndim
print(xx)
xx.shape
xx = xx.transpose()
xx.shape
print(xx)
```

Pylab, eig, revisited

```
a = eye(2)
print(a)
b = eig(a)
type(b)
len(b)
c, d = b
print(c)
print(d)
a[0, 1] = 1
print(a)
b = eig(a)
c, d = b
print(c)
print(d)
```

Pylab, plot 1

```
x = arange(10) + 0.5
y = x**2
plot(x, y)
plot(x, y, 'r+')
xlabel('x')
ylabel('y')
title('y = x**2')
```

pretty matlabeće, ali nema `hold on`
save kako hocete, odaberete format
u IDLE za ovo je bilo potrebno ion() (zaglavljivanje!)
pogledati `show()` i `ion()` help

`close()` ili `close('all')`

Pylab, plot 2

```
close('all')
phi = linspace(0, 2 * pi, 1000)
x = 2 * cos(phi)
y = 2 * sin(phi)
plot(x, y)
axis([-3, 3, -3, 3])
axis('equal')
axis([-3, 3, -3, 3])
axis([-4, 4, -3, 3])
title('kruznica')

axis?
```

Pylab, plot 3

```
close('all')

help(figure)

figure(figsize = (6, 6))
plot(x, y)
axis('equal')
axis([-3, 3, -3, 3])
title('kruznica')
```

Pazite na redosled `'equal'` i `[-3, 3, -3, 3]`!!!

Pylab, plot 4, histograms

```
x = rand(10000)
plot(x)
close()
rand?

hist(x, 100)
close()

hist?
```

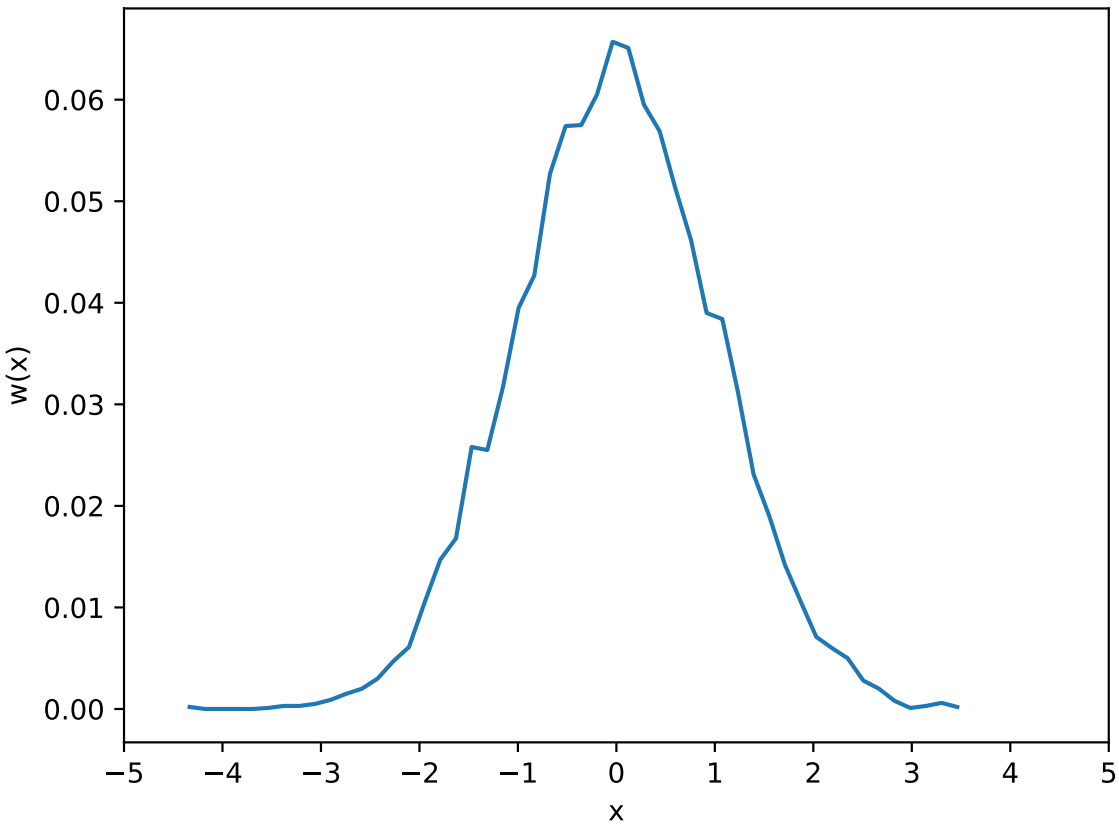
Pylab, plot 5, normal distribution

```
x = randn(10000)
plot(x)
close()

randn?

hist(x, 100)
close()

t = hist(x, 50)
type(t)
len(t)
len(t[0])
len(t[1])
```

Pylab, plot 6

```
y = t[0] / 10000.0   # pazite kod /!
x = t[1]
len(y)
len(x)
xx = (x[0 : len(x)-1] + x[1 : len(x)]) / 2
len(xx)
close()
plot(xx, y)
xlabel('x')
ylabel('w(x)')
xlim(-5, 5)
xticks(linspace(-5, 5, 11))
sum(y)
help(savefig)
savefig('slika') # potrazite slika.png
savefig('slika.pdf') # potrazite slika.pdf
```

slika.pdf



Pylab, jedan script, dat.py

```
from pylab import *

deg = linspace(0, 4*360, 4*360 + 1)
wt = radians(deg)

f = exp(- wt / 2 / pi * 0.5) * sin(wt)
fp = exp(- wt / 2 / pi * 0.5)
fm = -fp

dat = array([deg, wt, f, fp, fm]).transpose()

np.save('deg.npy', deg)
np.save('f.npy', f)
np.save('fp.npy', fp)

np.savetxt('dat.txt', dat, fmt='%.4f')
```

Pylab, run, #1

na komandnoj liniji:

```
python3 dat.py
more dat.txt
less dat.txt
ls *.npy
ipython3 --pylab
```

Pylab, run, #2 i #3

inside IPython:

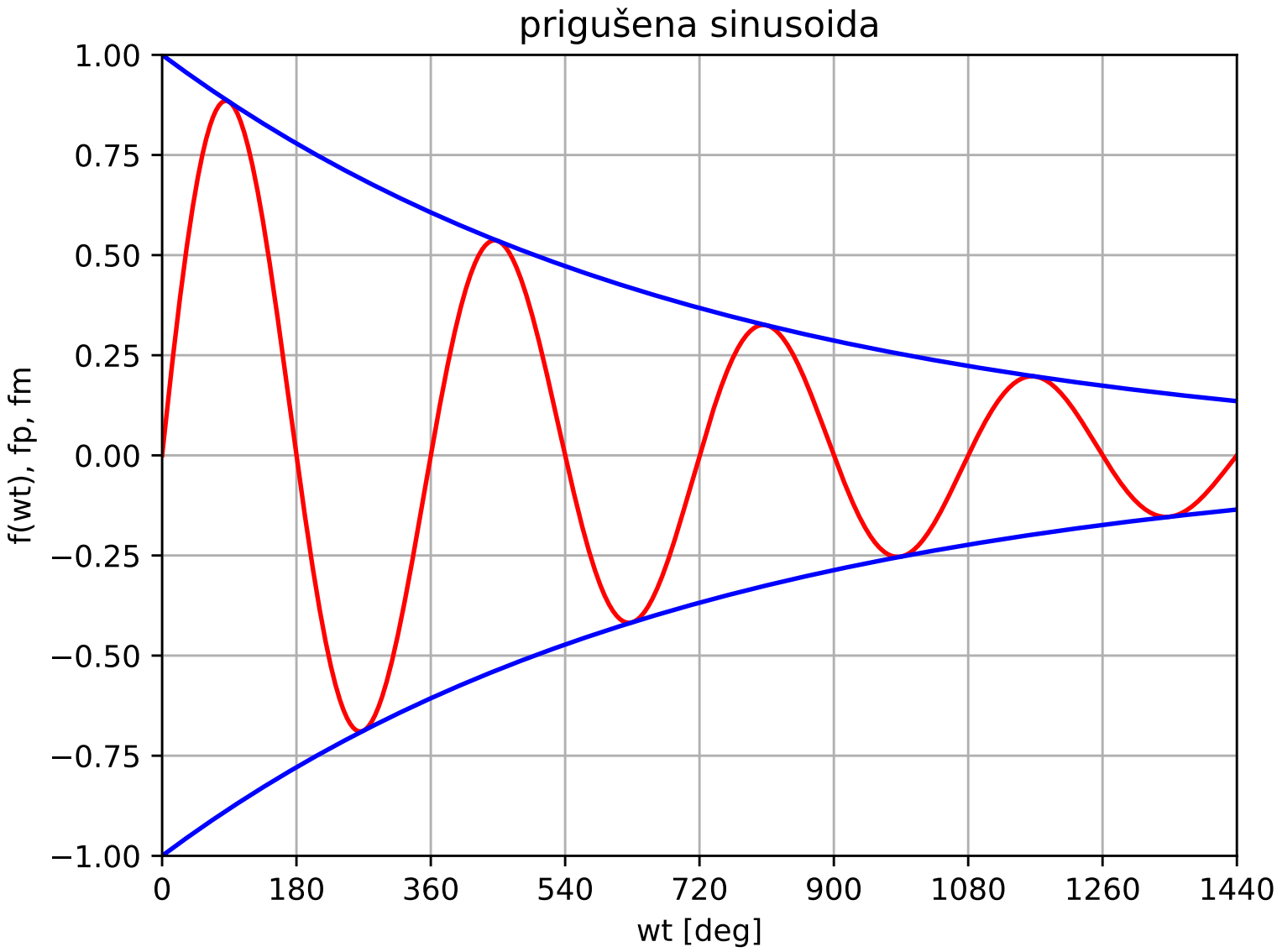
```
run dat.py
exec(open('dat.py').read())
```

`execfile` radi i kod IDLE

Pylab, after the run

```
np.save?
np.savetxt?
np.load?
deg = np.load('deg.npy')
f = np.load('f.npy')
fp = np.load('fp.npy')
fm = -fp
plot(deg, f, 'r')
plot(deg, fp, 'b')
plot(deg, fm, 'b')
xlim(0, 360 * 4)
xticks(arange(0, 360 * 4 + 1, 180))
xlabel('wt [deg]')
ylabel('f(wt)')
grid(True)
savefig('datslik.pdf')
close()
```

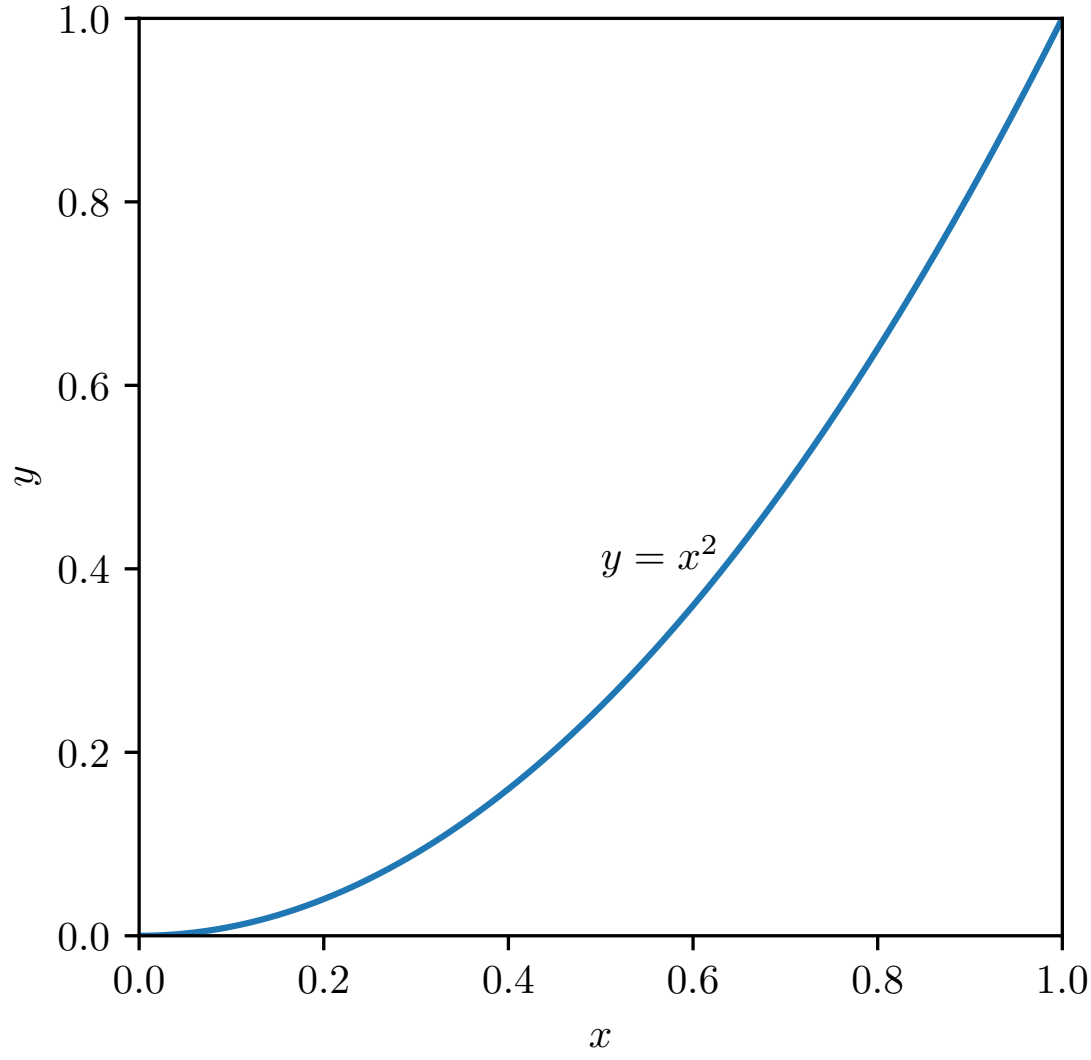
datslik.pdf


prigušena sinusoida
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
f(wt), fp, fm
0
180
360
540
720
900
1080
1260
1440
wt [deg]

Pylab, LaTeX, estetika

```
help(rc)

close('all')
x = linspace(0, 1, 101)
y = x**2
rc('text', usetex = True)
rc('font', family = 'serif')
figure(0, figsize = (6, 4))
plot(x, y)
xlabel(r'$x$')
ylabel(r'$y$')
text(0.5, 0.4, r'$y = x^2$')
savefig('kvadrat.pdf', bbox_inches = 'tight')
```

kvadrat.pdf



Pylab, LaTeX, „standardna“ „preambula“

```
rc('text', usetex = True)
rc('font', family = 'serif')
rc('font', size = 12)
rcParams['text.latex.preamble']=[r'\usepackage{amsmath}']
```

Pylab, linear algebra, 0th part

hoću da rešim sistem jednačina:

$x + y = 3$
$x - y = 1$

```
a = array([[1, 1], [1, -1]])
print(a)
b = array([[3], [1]])
print(b)
x = solve(a, b)
print(x)
b = array([3, 1])
print(b)
x = solve(a, b)
print(x)
```

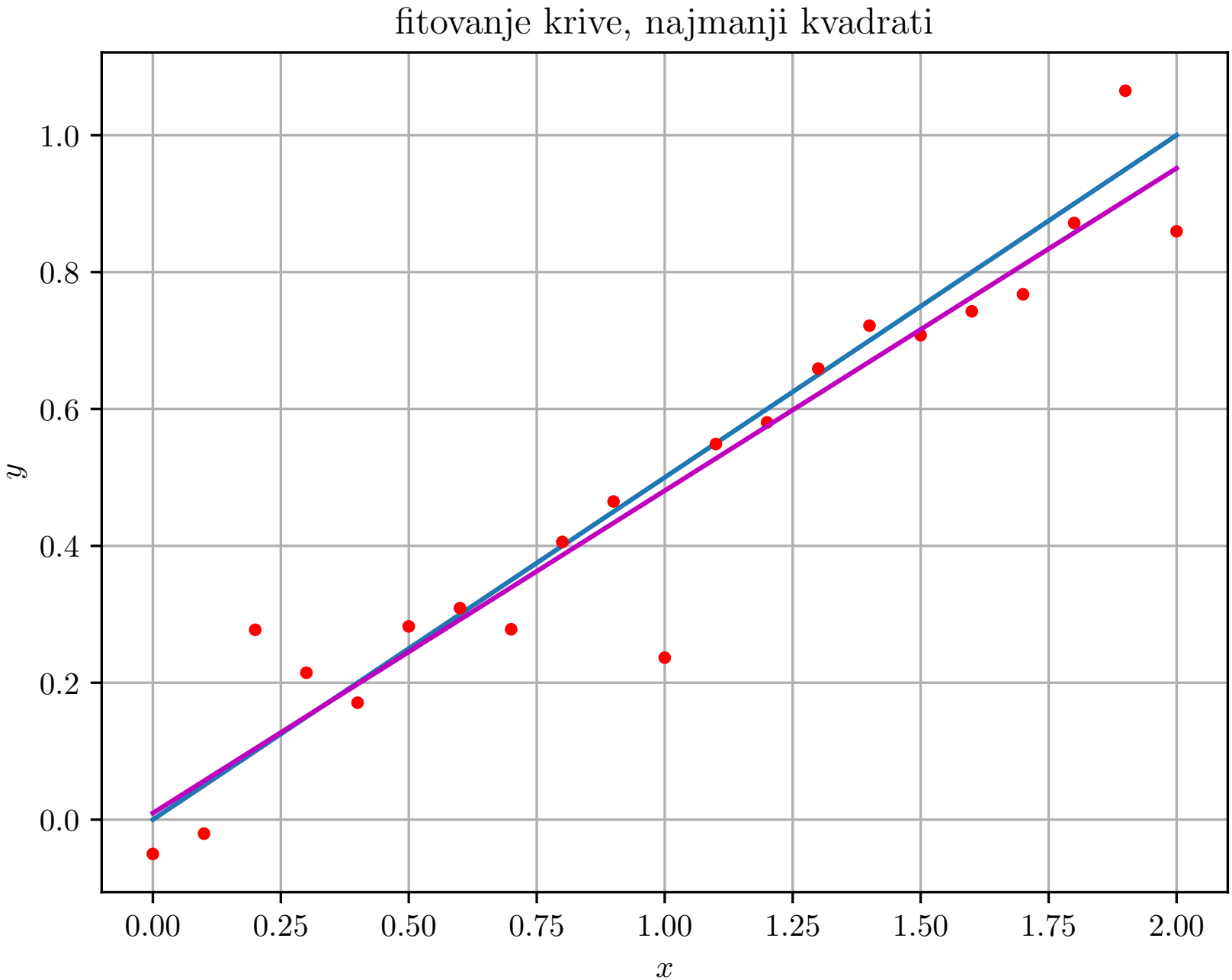
Pylab, linear algebra, 1st part

```
randn?
linalg.lstsq?

x = linspace(0, 2, 201)
y = x / 2

xe = linspace(0, 2, 21)
ye = xe / 2 + randn(21) * 0.1

A = array([xe, ones(len(xe))]).transpose()
t = linalg.lstsq(A, ye, rcond = None)
print(t)
type(t)
len(t)
a = t[0][0]
b = t[0][1]
```

Pylab, linear algebra, 2nd part

```
close('all')

rc('text', usetex=True)
rc('font', family='serif', size='16')

plot(x, y, 'b')
plot(xe, ye, 'r.')

yfit = a * x + b
plot(x, yfit, 'm')
xlabel(r'$x$')
ylabel(r'$y$')
savefig('fitovanje.pdf')
```

fitovanje.pdf



Jupyter

- „notebook“ okruženje za računanje
- lako dokumentovanje rezultata
- nastao iz ipython, language agnostic
- radi i sa python i sa julia i sa sage i sa ...
- pokrene web server, sve se radi kroz browser
- popularno i za cloud applications
- https://jupyter.org/
- https://en.wikipedia.org/wiki/Project_Jupyter
- sretaćete se sa ovim još ...
- za sada samo primer ...