Data Mining using Python
— exercises for numeric Python

Finn Årup Nielsen

DTU Compute
Technical University of Denmark

September 15, 2014

File reading and simple computing

Consider a file with the following matrix $\mathbf{X}$:

```
1 2
3 4
```

Read and compute $\mathbf{Y} = 2 * \mathbf{X}$ now with NumPy!

Matrix rank

Compute the rank of the `array`:

```
>>> from numpy import *
>>> A = array([[1, 0], [0, 0]])
>>> rank(A)
2
```

Hmmmm ??? Not this one.

Find the matrix rank by computing the number of numerical non-zero singular values

Function header:

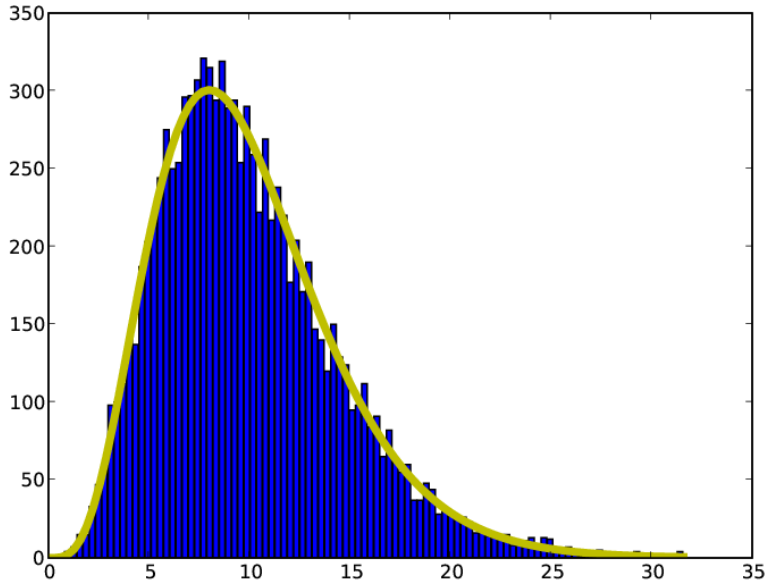
```
def matrixrank(A, tol=None):
    """Compute the matrix rank.

    >>> matrixrank(array([[1, 0], [0, 0]]))
    1
    """
```

Hint: use the `svd` function in `numpy.linalg`.

Statistical distributions

Generate 10'000 sets with 10 Gaussian distributed samples, square each element and sum over the 10 samples. Plot the histogram of the 10'000 sums together with the teoretically curve of the probability density function.

χ^2_{10} PDF from the `pdf()` function in the `scipy.stats.chi2` class



Coauthors

Read coauthors.csv — a tab-separated file with co-author matrix. Find the author with most coauthoring.

Plot the largest connected component part of the network with `NetworkX`.

Optimization

Minimize this function, e.g., with scipy:

```
f = lambda x, y: (1-x)**2 + 100 * (y-x**2)**2
```

You can also try to differentiate it with sympy.