Kvantitatív módszerek – Python gyakorlat

Kiegészítő anyag Csicsman József *Statisztika gyakorlati alkalmazásai* című kurzusához

Varga Viktor

2018. január 9.

1. **Bevezetés, Python rövid bemutatása**

2. **Python környezetek, programozási felületek**
   - Anaconda
   - Jupyter
   - Spyder

3. **Statisztikai szempontból fontos csomagok**
   - Python standard modulok
   - Matematikai alapok, statisztikai függvények: numpy, scipy.stats
   - Adatkezelés és lekérdezések egyszerűsítése, DataFrame-ek: pandas
   - Vizualizáció: matplotlib
   - További csomagok

Python

- általános célú, interpretált programozási nyelv
- rengeteg függvény- és osztálykönyvtár (ún. csomag, *package*) létezik hozzá
- tudományos célra való használata széles körben elterjedt
  - könnyen átlátható és tanulható
  - sok a matematikára, statisztikára, adatelemzésre kihegyezett csomag (adatmanipuláció, leíró statisztikák, összetett elemzések, pl. regresszió, főkomponens . . .)
  - támogatja a felfedező jellegű, értelmezőközpontú programozási paradigmát
  - sokban hasonlít az R nyelv használatához
- segítőkész közösség, „mindenre van megoldás”, csak keresni kell tudni
- szoftverfejlesztésben is népszerű (főleg webfejlesztésben)
- itt a nyelv és az értelmező 3.6.2-es verzióját használjuk

Python példakód: Fibonacci

- >>> jelzi a bevitelt, ... a blokk folytatását, ezeket nem kell beírni
- a sor eleji szóközök (4-esével) határozzák meg a hatáskört, pl. függvénydefiniálásnál (`def`), elágazó feltételeknél (`if`), ciklusoknál (`for`)
- komment a # szimbólummal a sor végéig

```
>>> def fib(n):                    # függvénydefiníció
...     a, b = 0, 1                 # értékadások
...     while a < n:
...         print(a, end=' ')       # képernyőre írás, végén szóközzel
...         a, b = b, a+b
...     print()                     # új sor
...
>>> fib(1000)                      # függvény meghívása
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Forrás: Python hivatalos weboldal

Python példakód: egyszerű aritmetika, lambda

```
>>> 17 / 3                                # osztás
5.666666666666667

>>> 17 // 3                               # kerekítő osztás, egészosztás
5

>>> 2 ** 3                                # hatványozás
8

>>> mult_by_two = lambda x: x*2           # lambda függvénydefiníció
>>> mult_by_two(10)
20
```

Forrás: Python hivatalos weboldal

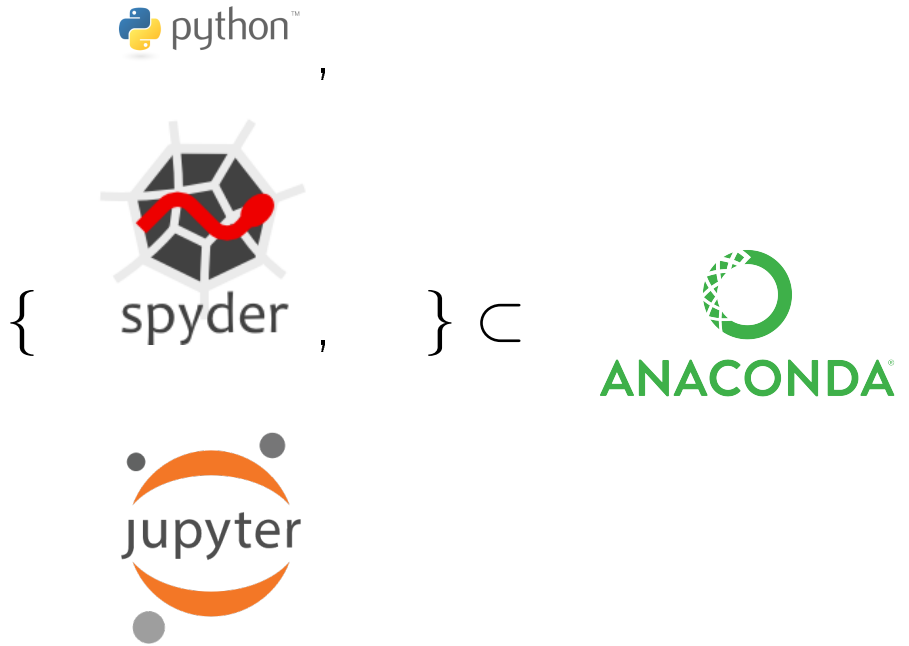
Python példakód: listabejárás

```
>>> numbers = [2, 4, 6, 8]          # listadefiníció
>>> numbers[2]                      # 3. (!) elem elérése
6
>>> product = 1
>>> for number in numbers:          # ciklus a lista elemein
...     product = product * number
...
>>> print('A szorzat:', product)
A szorzat: 384
```

Forrás: Python hivatalos weboldal

Python környezetek, programozási felületek

- **Spyder**: hasonló az RStudióhoz
- **Jupyter**: jegyzetfüzet alapú fejlesztőkörnyezet, hasonló az RStudio által nemrég bevezetett R Notebookhoz
- **Anaconda**: a teljes tudományos számításokhoz szükséges eszköztárat, köztük a fentieket is tartalmazza
- IDLE: a Pythonnal együtt adott minimális fejlesztői környezet
- JetBrains PyCharm: sokoldalú fejlesztői környezet, szoftverfejlesztéshez is jól használható

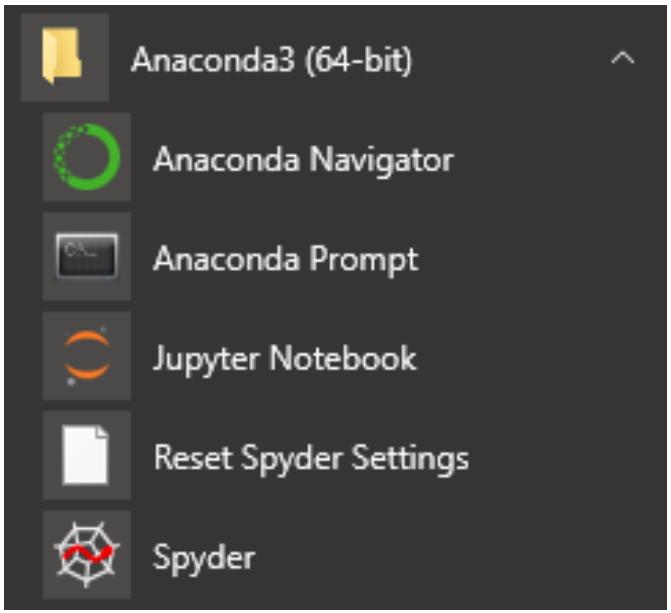


Anaconda

- Spyder, Jupyter, RStudio ...
- a legtöbb tudományos számításokhoz szükséges Python-csomagot tartalmazza, de telepíthetők újak is
- Anaconda Cloud: a megírt szkriptek, jegyzetfüzetek stb. online tárolására és megosztására használható felhő
  - a publikusra állított tartalom böngészhető
- Letöltés: `https://www.anaconda.com/download/`

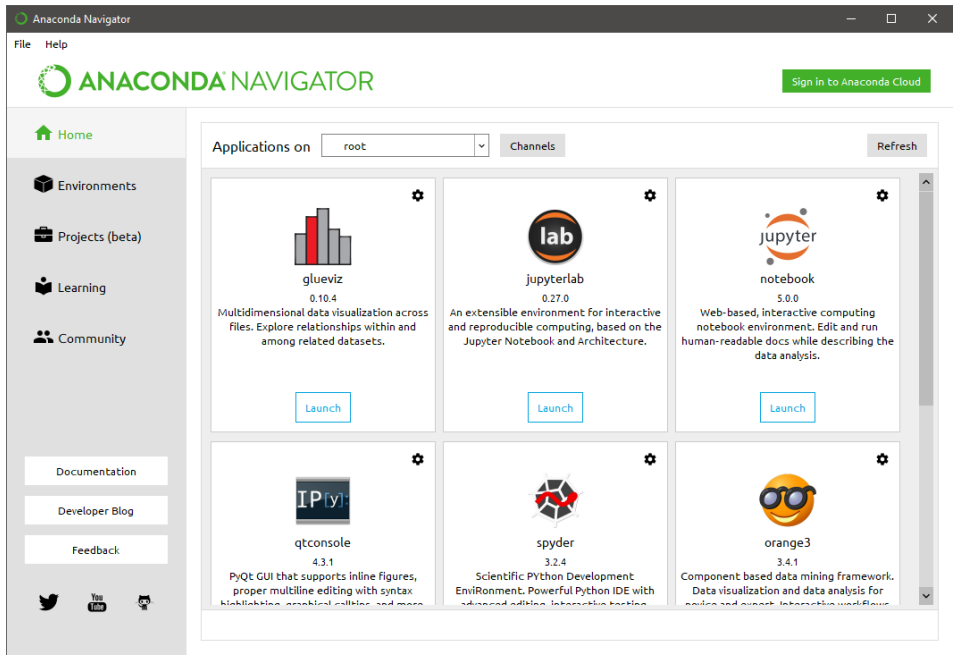
Anaconda - Használat

- **Anaconda Navigator**: az Anaconda által nyújtott eszközök egyszerű grafikus felületen
- *Anaconda Prompt*: konzol, többek között csomagok telepítésére (`conda` parancs), de a `python` paranccsal elérjük a Python értelmezőt (interpretert) is, az Anaconda által nyújtott csomagokkal együtt



Anaconda Navigator

- környezetek: külön definiálható csomaghalmazok
- oktatóanyagok
- alkalmazások
  - **Jupyter**, **Spyder**
  - JupyterLab: Jupyteren alapuló, böngészőalapú jegyzetfüzetkezelő és fejlesztőkörnyezet
  - Glueviz: Pythonra épülő adatvizualizációs eszköz
- egyéb segédeszközök



Jupyter

- böngészőből elérhető Python értelmező
- a jegyzetfüzet IPython-alapú weblap, cellákból áll, amelyek egyenként futtathatóak a háttérben futó kernelen
  - támogatott kernelek: Julia, Python, R nyelvek
- dokumentációszöveget külön Markdown típusú cellákban, ennek megfelelő jelöléssel érdemes megadni
- felépítése és működése miatt rendkívül használható demonstrációs célokra
- a közösség sok esetben meg is osztja ezeket a jegyzetfüzeteket (statikus formában), a hasznos linkek között találhatók ilyen jegyzetfüzet-gyűjtemények

Jupyter - Képernyőkép

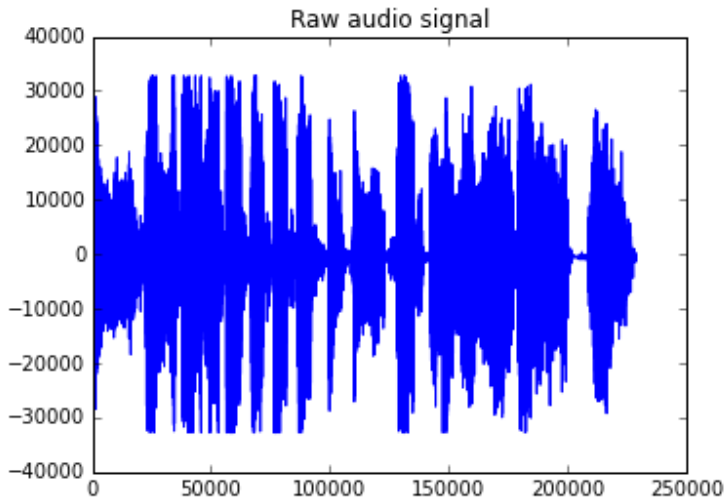
jupyter spectrogram Last Checkpoint: an hour ago (autosaved)

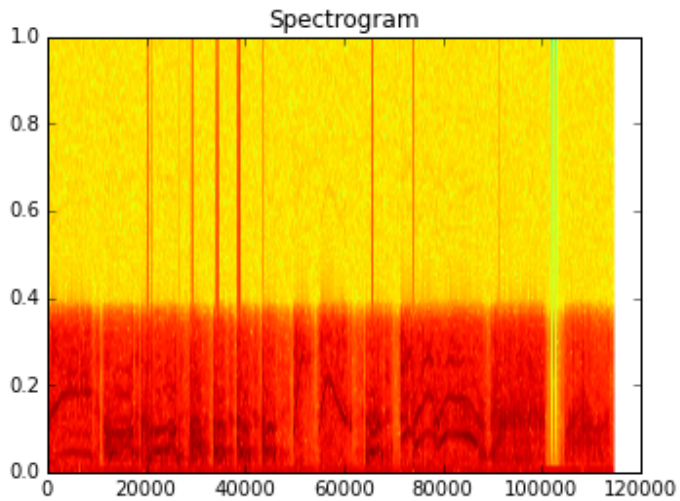
File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

In [1]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

In [2]:
```
import matplotlib.pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram')
plt.show()
```



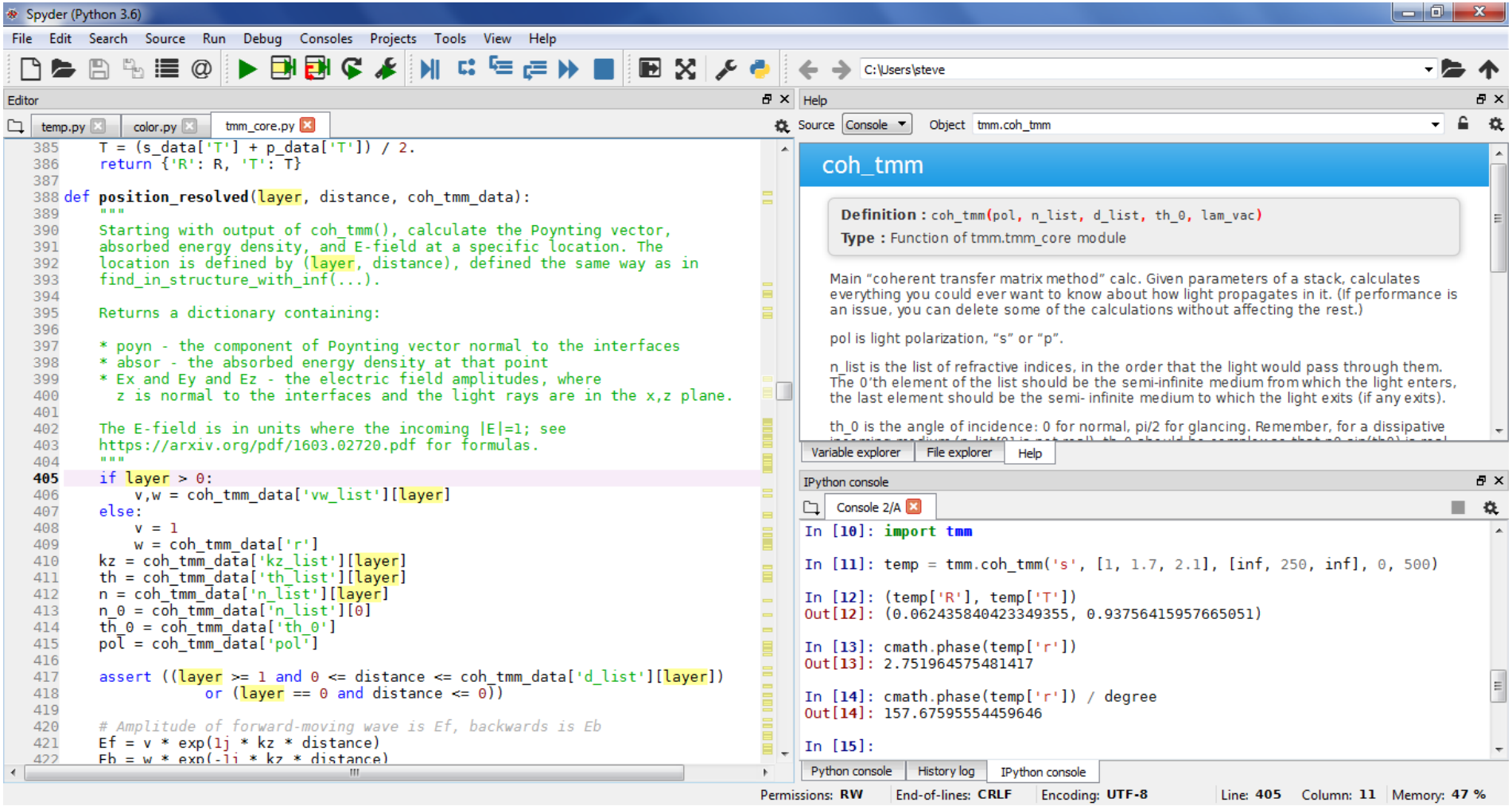


Forrás: https://rock-it.pl/content/images/2017/03/ipy-notebook-spectral.png

Spyder

- Scientific PYthon Development EnviRonment
- összetettebb fejlesztői környezet, több munkaablakkal, debug-funkciókkal, projektkezeléssel
- támogatja a (Jupyterhez hasonló) cellaalapú programozást
- munkaablakok
  - Editor: a fő kódolási felület
  - IPython Console: konzol, amely a futtatás során előállt kimenetet is mutatja (ha ezt választjuk)
  - Variable Explorer: a program futtatása során létrejött változók és aktuális értékük, változtathatók is
  - File explorer, History log, Help

Spyder - Képernyőkép



Forrás: http://sjbyrnes.com/wp-content/uploads/2012/03/spyder_screenshot.png

Python standard modulok, beépített függvények

- `time`, `datetime`, `calendar`: dátummal és idővel kapcsolatos modulok
  - pl. `datetime.datetime`, `.timedelta`, `.strftime()`, `.strptime()` stb.
- `random`: pszeudorandom számgenerálást, elemkiválasztást, rendezést stb. támogató modul
  - pl. `random.randint`, `.choice`, `.shuffle` stb.
- `statistics`: alapvető statisztikai függvények
  - pl. `statistics.mean`, `.median`, `.mode`, `.stdev`, `.variance`

Python standard modulok – Példák

```
>>> import random                              # csomag importálása
>>> import datetime, statistics
>>> d = datetime.datetime(2017, 11, 7) # dátum objektum
>>> d.strftime('%d/%m/%y')                     # dátumformázás
'07/11/17'
>>> random.randint(1, 3)                       # pszeudorandom szám [1; 3]
2
>>> a = [1, 2, 3, 4]
>>> random.choice(a)                           # ps-random választás
1
>>> random.shuffle(a)                          # ps-random keverés
>>> a
[2, 1, 3, 4]
>>> len(a), sum(a)                             # beépített aggregációk
(4, 10)
>>> statistics.stdev(a)                        # standard szórás
1.2909944487358056
```

numpy

- matematikai alapokat biztosító modul
- ellenőrzött adattípusok, vektorok és mátrixok bevezetése (`ndarray`) a számítások hatékonyságához
- bővebb és hatékonyabb matematikai függvénykönyvtár, mint a standard könyvtárban lévő
  - lineáris algebra, random generálás, alapvető statisztikai függvények
- almodulok: `numpy.linalg`, `.random`, ...
- önmagában való kezelése kissé nehézkes, de:
  - sok jó oktatóanyag és könyv van
  - sokszor meg lehet fogalmazni a problémát magasabb szinten (`pandas` csomag)

numpy – Példák

```
>>> import numpy as np
>>> x = np.array([[1., 2., 3.],          # 3x2-es mátrix
...               [4., 5., np.NaN]])
>>> x
array([[1., 2., 3.],
       [4., 5., nan]])
>>> x[0, 1]                              # 1. sor 2. eleme
2.0
>>> x.T                                  # transzponált
array([[ 1.,  4.],
       [ 2.,  5.],
       [ 3., nan]])
>>> np.add(x, x**2)                      # mátrixösszeadás
array([[ 2.,  6., 12.],
       [20., 30., nan]])
>>> (np.mean(x), np.nanmean(x))          # számtani közép
(nan, 3.0)
```

scipy

- `numpy`-ra épülő, komplexebb számításokat biztosító könyvtár
- sok esetben valójában interfészként funkcionál más nyelven (Fortran, C) hatékonyan megírt függvényekhez
- integrálás, optimalizáció, lineáris algebra, interpoláció
- almodulok: `scipy.optimize`, `.linalg`, `.stats`, `.io`, `...`

scipy – Példák

```
>>> from scipy.integrate import quad      # x^2 fuggveny hatarozott
>>> quad(lambda x: x**2, 1, 3)            # integralja [1; 3]-on
(30.375, 3.372302437298913e-13)           # (num. kvadratúra)

>>> import numpy as np
>>> from scipy.linalg import solve        # Ax=b egyenletrendszer
>>> A = np.array([[1, 2], [3, 4]])        # megoldasa a matrix
>>> b = np.array([[5], [6]])              # tulajdonságainak
>>> solve(A, b)                           # figyelembevételével
array([[-4. ],
       [ 4.5]])
```

scipy.stats

- a `scipy` statisztikai függvényeket, valószínűségi eloszlásokat tartalmazó almodulja
- folytonos, diszkrét és többváltozós eloszlások random generálása, sűrűség- és eloszlásfüggvény, statisztikák, momentumok stb.
  - eloszlások pl. `norm`, `poisson`, ...
  - `<eloszlás>.rvs`, `.pdf`, `.stats`, `.moment`, ...
- statisztikák, tesztek
  - describe, moment, skew, linregress,
  - skewtest, pearsonr, f_oneway, ttest_1samp

scipy.stats – Példák

```
>>> from scipy.stats import norm      # normalis eloszlas
>>> norm.stats(moments='mvsk')        # momentumai: mean, var, skew,
(array(0.0), array(1.0), array(0.0), array(0.0))              # kurt
>>> x = norm.rvs(size=4)              # valtozoertekek ps-random
>>> x                                                  # generalasa
array([-0.09877321, -1.50741484, -1.17870892,  0.20281403])

>>> np.linspace(                      # egyenletesen elhelyezett
...         norm.ppf(0.01),           # ertekek a normalis eloszlas
...         norm.ppf(0.99), 5)        # kvantilisfuggvenye alapjan
array([-2.32634787, -1.16317394,  0.        ,  1.16317394, \
       2.32634787])

>>> from scipy.stats import describe
>>> describe(x)                       # altalanos statisztikak
DescribeResult(nobs=10, minmax=(-2.3263478740408408, \
    2.3263478740408408), mean=0.0, variance=2.449828754798260 9, \
    skewness=5.425788447356446e-17, kurtosis=-1.224242424242246)
```

pandas

- gyakorlati adatelemzést jelentősen megkönnyítő, `numpy`-alapú csomag
- könnyen kezelhető adatstruktúrák, DataFrame-ek (2-dimenziós) és Series-ek (1-dimenziós) bevezetése
  - Jupyter környezetben HTML-alapú rendezhető táblázatként jelenik meg
- adatok fájlból való beolvasását és fájlba történő exportálását, bővítését, szűrését, törlését, tisztítását, konvertálását és elemzését támogató függvényeket nyújt
- idősor-elemzésekhez speciális módszereket biztosít (pl. ablakozás)
- egyszerű vizualizációra is van lehetőség
- érdemes használni a dokumentációt és az oktatóanyagot a funkciók gazdagságának feltérképezésére

pandas – Példák

```
>>> import pandas as pd
>>> import numpy as np
>>> dates = pd.date_range('2013-01-01', periods=6)
>>> df = pd.DataFrame(np.random.randn(6,4), index=dates,
...                   columns=list('ABCD'))
...
>>> df.head(2)
                   A         B         C         D
2013-01-01  0.469112 -0.282863 -1.509059 -1.135632
2013-01-02  1.212112 -0.173215  0.119209 -1.044236
```

Forrás: pandas oktatóanyag

pandas – Példák folytatása

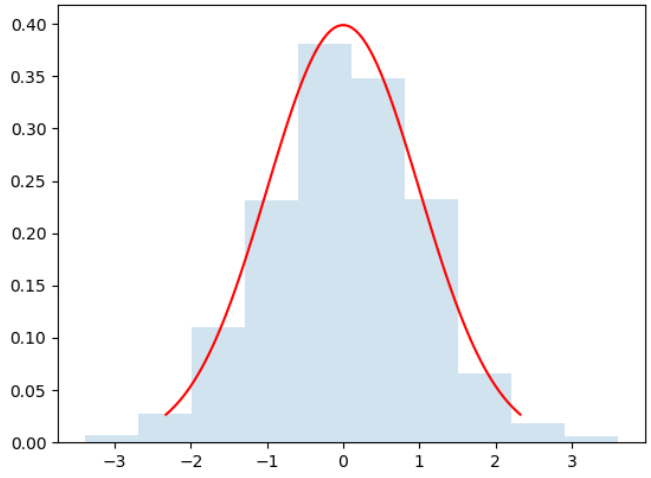
```
>>> df.describe()     # alapvető statisztikák
                A         B         C         D
count   6.000000  6.000000  6.000000  6.000000
mean    0.073711 -0.431125 -0.687758 -0.233103
std     0.843157  0.922818  0.779887  0.973118
min    -0.861849 -2.104569 -1.509059 -1.135632
25%    -0.611510 -0.600794 -1.368714 -1.076610
50%     0.022070 -0.228039 -0.767252 -0.386188
75%     0.658444  0.041933 -0.034326  0.461706
max     1.212112  0.567020  0.276232  1.071804
>>> df.A.head(2)      # adott oszlopra (változóra) hivatkozás
2017-01-01    0.469112
2017-01-02    1.212112
```

Forrás: pandas oktatóanyag

matplotlib

- az egyik legelterjedtebb vizualizációs könyvtár
- egyszerűbb grafikonokat is össze lehet állítani, de haladóbb felhasználók teljesen testre is szabhatják
- a `pyplot` modul biztosítja az ábrázoláshoz szükséges függvényeket
- kódban futtatva is megjeleníti külön ablakban, de Jupyter jegyzetfüzetbe is beágyazható a `%matplotlib inline` parancs lefuttatása után (hasonlóan a legtöbb más vizualizációs eszközhöz)

matplotlib – Példák




```
>>> from scipy.stats import norm
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> fig, ax = plt.subplots(1, 1)
>>> x = np.linspace(norm.ppf(0.01),
...                 norm.ppf(0.99), 100)
>>> ax.plot(x, norm.pdf(x), 'r-')
>>> r = norm.rvs(size=1000)
>>> ax.hist(r, normed=True, histtype='stepfilled', alpha=0.2)
>>> plt.show()
```

Forrás: SciPy dokumentáció oktatóanyag

További releváns csomagok

- `scikit-learn`: kifejezetten adatbányászatra és adatelemzésre szolgáló csomag
  - osztályozás, regresszió, klaszterelemzés, dimenziócsökkentés, ...
- `statsmodels`: `scipy`-ra épül, összetettebb statisztikai modelleket tartalmaz, a `pandas` DateFrame-ekkel is működik
- `sympy`: a `numpy` párja, szimbolikus matematikai eszközkészlet
- vizualizáció
  - `holoviews`, `seaborn`: magasabb szintű könyvtárak, „egyszerűbben szebbet”
  - `bokeh`: interaktív diagramok készítése (pl. webes böngészőbe)
  - `mpl_toolkits.mplot3d`: 3D diagramok készítése
- egyéb érdekes
  - `rpy2`: R szkriptek futtatása Pythonban

Hasznos weboldalak: Python általános

- Python 3 hivatalos dokumentációja oktatóanyaggal és útmutatókkal: `https://docs.python.org/3/`
- Könyvek, pl. innen: `https://wiki.python.org/moin/PythonBooks`
- Online kurzusok (Coursera, Udemy, edX stb.)
- Stack Overflow, „mindenre van megoldás”: `https://stackoverflow.com/`
- Online elérhető statikus Jupyter jegyzetfüzetek
  - `http://nb.bianp.net/sort/views/`
  - `https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks`

Hasznos weboldalak: Környezetek

- Anaconda: `https://www.anaconda.com/`
- Jupyter: `https://jupyter.org/`
- JupyterLab: `https://jupyterlab-tutorial.readthedocs.io/en/latest/`
- Spyder: `https://pythonhosted.org/spyder/`
- Glueviz: `http://www.glueviz.org/en/stable/`
- JetBrains PyCharm: `https://www.jetbrains.com/pycharm/`

Hasznos weboldalak: Csomagok referenciái

- numpy és scipy: https://docs.scipy.org/
- pandas: https://pandas.pydata.org/
- matplotlib: https://matplotlib.org/
- scikit-learn: http://scikit-learn.org/stable/
- statsmodels: http://www.statsmodels.org/stable/index.html
- bokeh: https://bokeh.pydata.org/en/latest/