

Computação Científica com Python

A cobra também é inteligente!



Marcel P. Caraciolo
@marcelcaraciolo

Quem é Marcel ?

Marcel Pinheiro Caraciolo - @marcelcaraciolo

Sergipano, porém Recifense.

Mestrando em Ciência da Computação no CIN/UFPE na área de mineração de dados

Diretor de Pesquisa e Desenvolvimento na Orygens

Membro e Moderador da Celúla de Usuários Python de Pernambuco (PUG-PE)

Membro do Muriçoca Labs - Labs de Projetos com Machine Learning

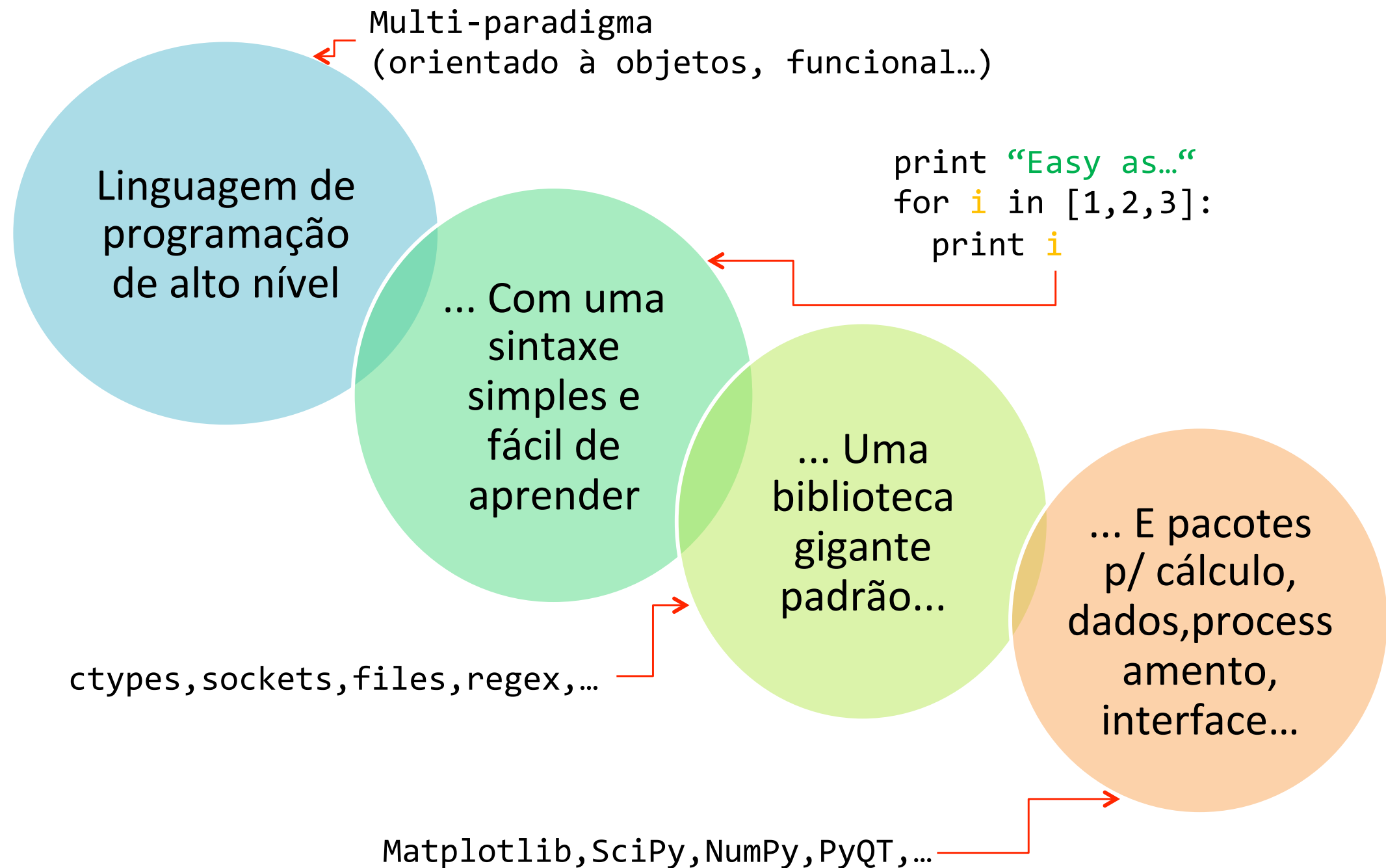
Minhas áreas de interesse: Computação móvel e Computação inteligente

Meus blogs: <http://www.mobideia.com> (sobre Mobilidade desde 2006)

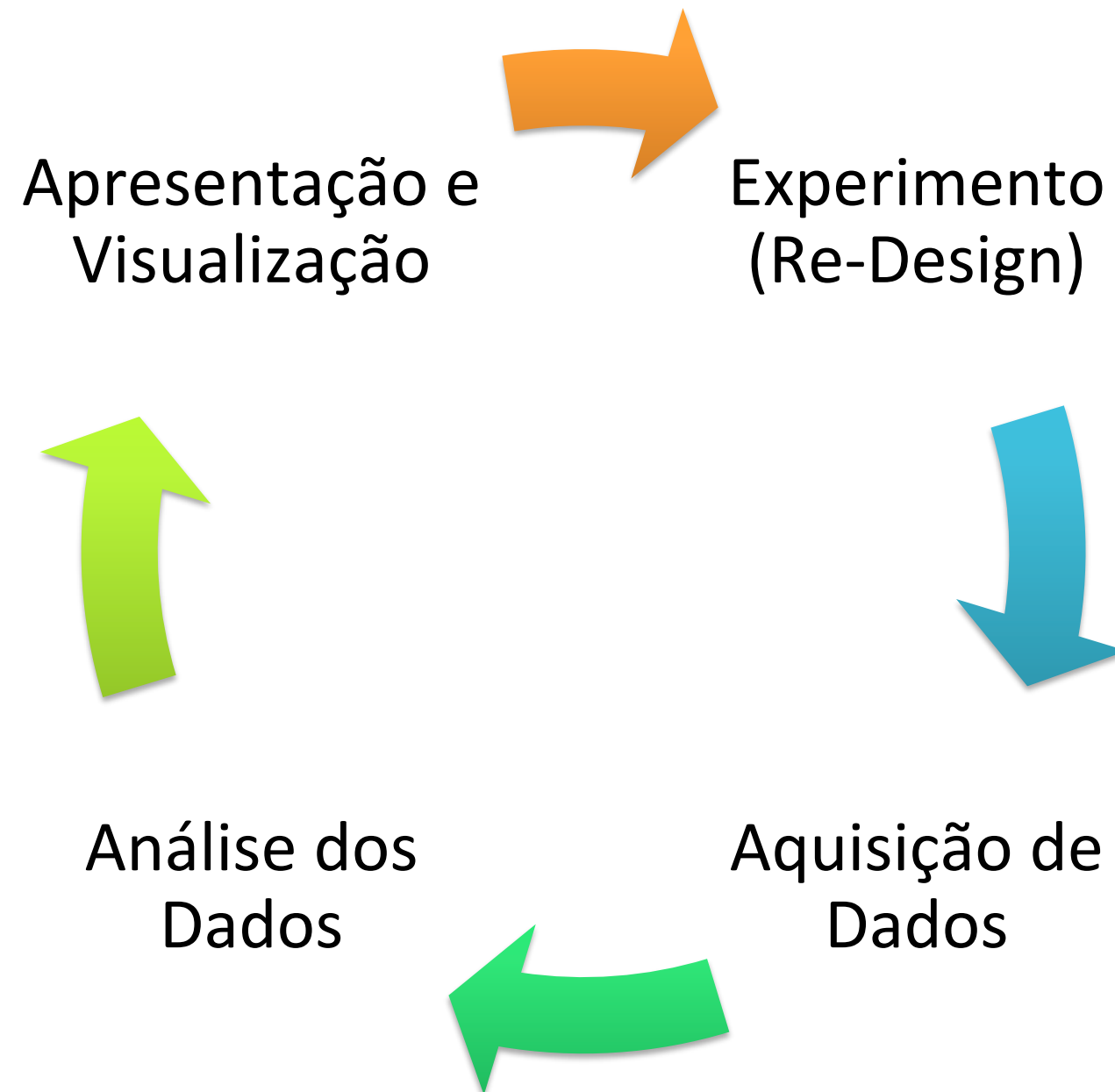
<http://aimotion.blogspot.com> (sobre I.A. desde 2009)

Jovem Aprendiz ainda nas artes pythonicas.... (desde 2007)

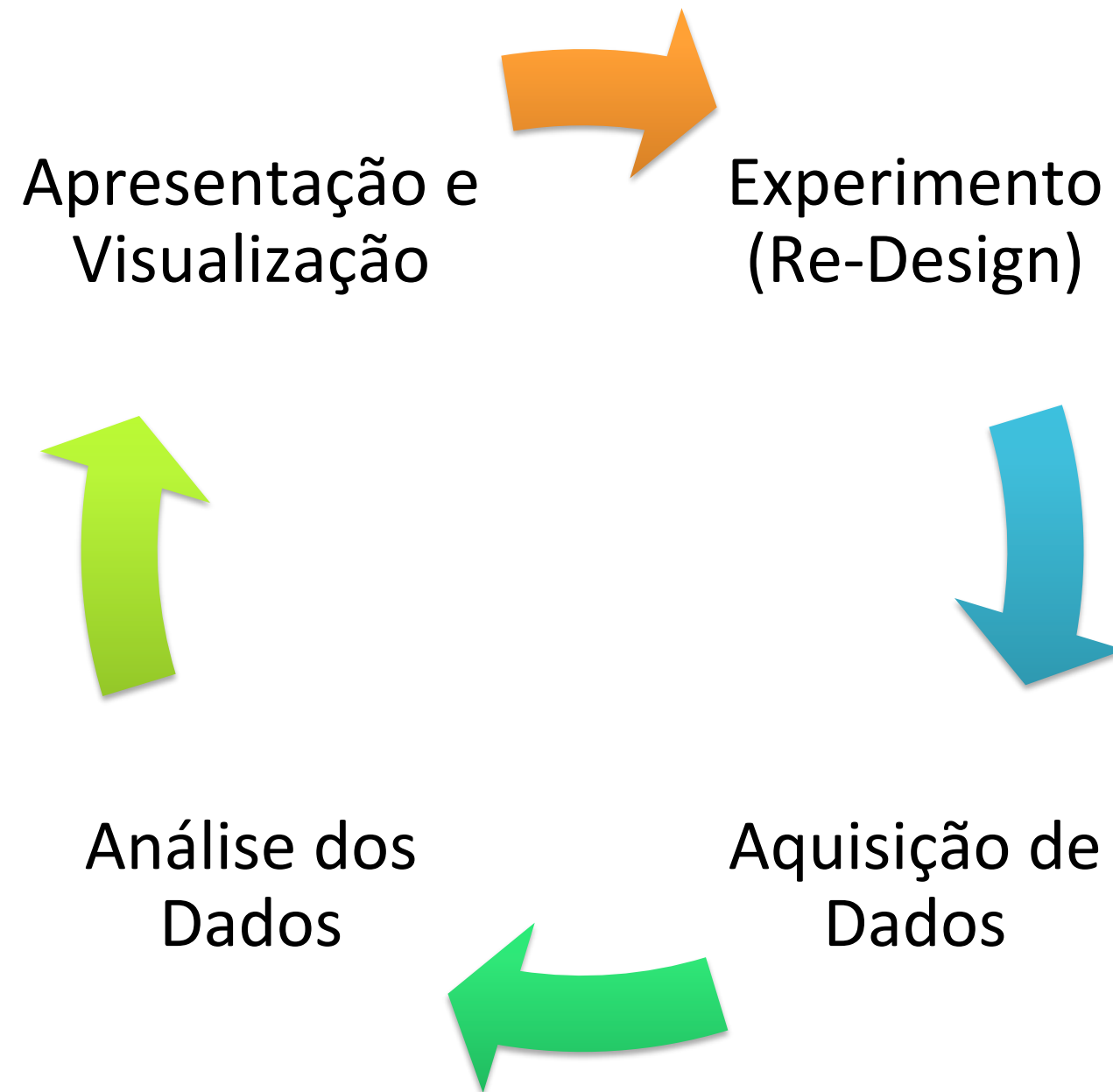
O que é Python ?



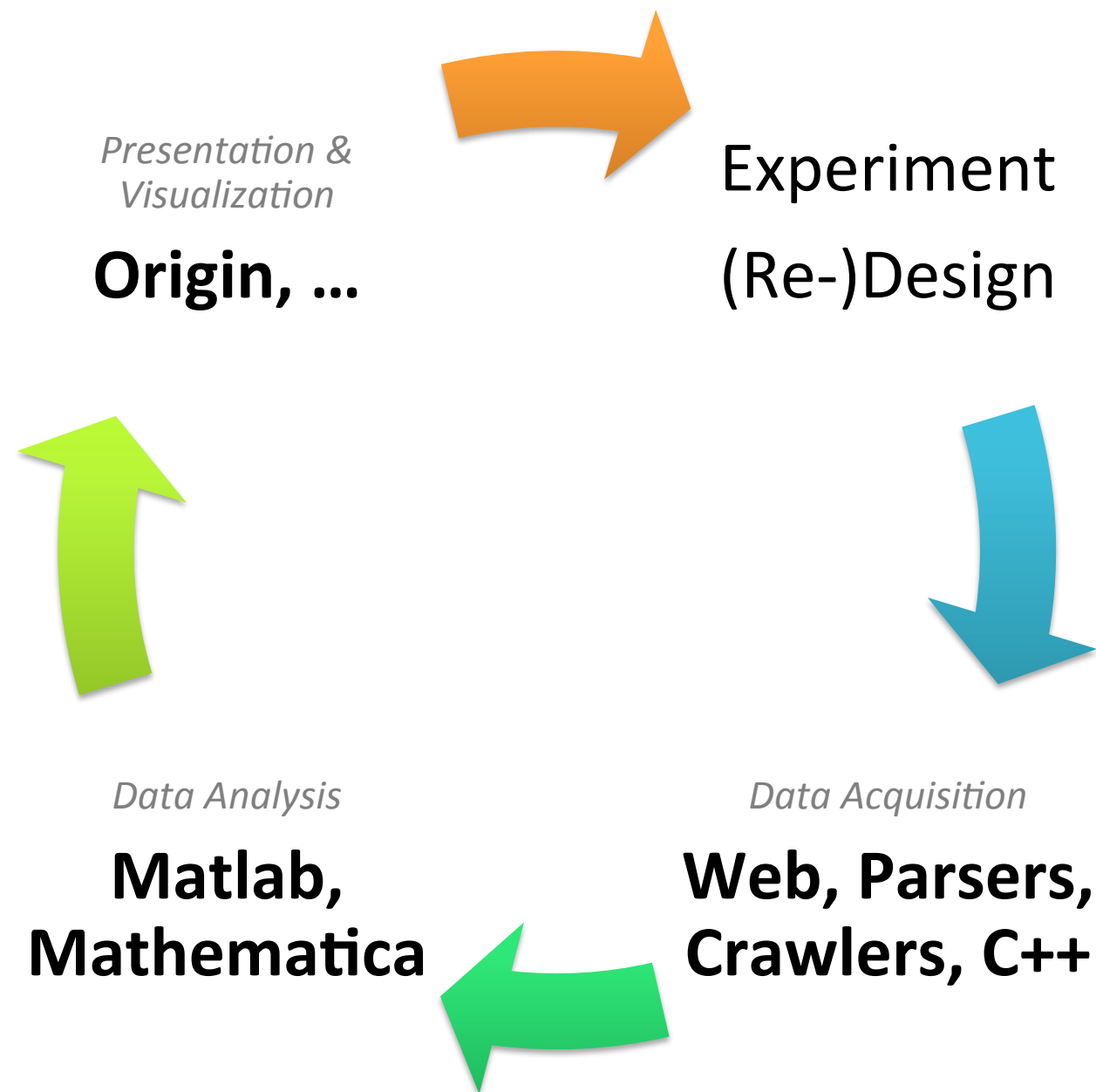
Ciclo de Desenvolvimento em P&D



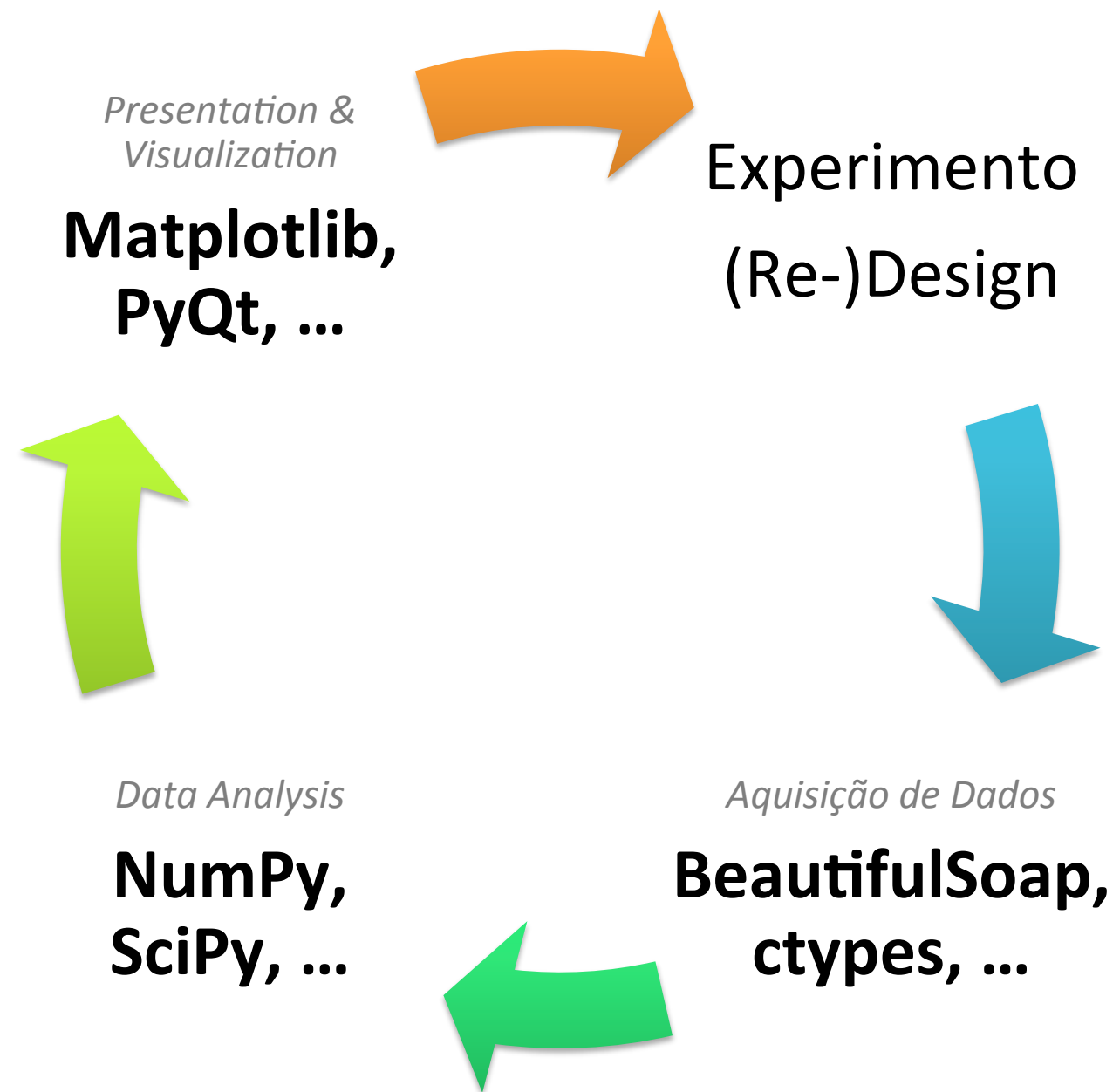
Ciclo de Desenvolvimento em P&D



Uso de software facilita muito!



Uso de **Python** facilita mais ainda!!





De onde vem os dados ?

Aquisição de Dados

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

About Citation Policy Donate a Data Set Contact

Search

Repository Web Google

[View ALL Data Sets](#)

Welcome to the UC Irvine Machine Learning Repository!


We currently maintain 200 data sets as a service to the machine learning community. You may [view all data sets](#) through our searchable interface. Our [old web site](#) is still available, for those who prefer the old format. For a general overview of the Repository, please visit our [About page](#). For information about citing data sets in publications, please read our [citation policy](#). If you wish to donate a data set, please consult our [donation policy](#). For any other questions, feel free to [contact the Repository librarians](#). We have also set up a [mirror site](#) for the Repository.

Supported By: In Collaboration With:

Latest News:

- 03-01-2010: [Note](#) from donor regarding Netflix data
- 10-16-2009: Two new data sets have been added.
- 09-14-2009: Several data sets have been added.
- 07-23-2008: [Repository mirror](#) has been set up.
- 03-24-2008: New data sets have been added!
- 06-25-2007: Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope
- 04-13-2007: Research papers that cite the repository have been associated to specific data sets.

Featured Data Set: Statlog (Landsat Satellite)

 Task: Classification
Data Type: Multivariate
Attributes: 36
Instances: 6435

Multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood

Newest Data Sets:

- 05-22-2011: [PEMS-SF](#)
- 02-07-2011: [YearPredictionMSD](#)
- 12-13-2010: [MiniBooNE particle identification](#)
- 11-03-2010: [AutoUniv](#)
- 11-03-2010: [Localization Data for Person Activity](#)
- 10-26-2010: [Steel Plates Faults](#)
- 09-13-2010: [Spoken Arabic Digit](#)
- 09-07-2010: [Cardiotocography](#)
- 08-04-2010: [Wall-Following Robot Navigation Data](#)

Most Popular Data Sets (hits since 2007):

- 209445: [Iris](#)
- 155045: [Adult](#)
- 135014: [Wine](#)
- 108915: [Breast Cancer Wisconsin \(Diagnostic\)](#)
- 88613: [Car Evaluation](#)
- 87239: [Abalone](#)
- 81394: [Poker Hand](#)
- 66641: [Forest Fires](#)
- 58813: [Yeast](#)

<http://archive.ics.uci.edu/ml/>

Repositórios de Dados

UCI, MovieLens, AWS, KDD, etc.

Marcel Caraciolo - @marcelcaraciolo

Usando o loadtxt()

```
from numpy import loadtxt

def class2int(s):
    if s == 'Iris-setosa':
        return 1
    elif s == 'Iris-versicolor':
        return 0
    else:
        return 2

ary1 = loadtxt('iris.data', delimiter=',',
               converters={4: lambda s: class2int(s)},
               skiprows=1)

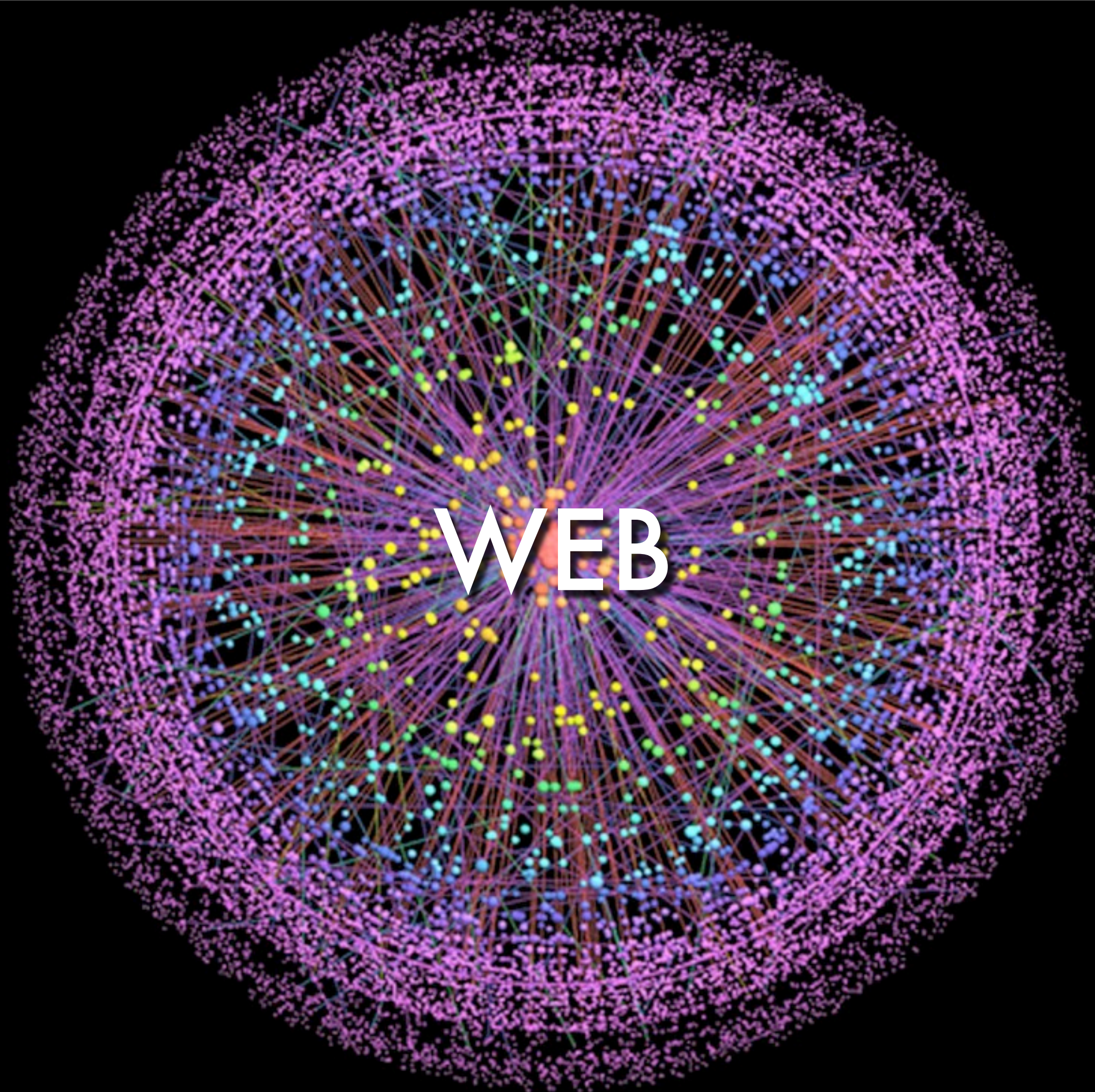
print ary1
```



<http://archive.ics.uci.edu/ml/datasets/Iris>

```
%n1, n2, n3, n4, class
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
...
```

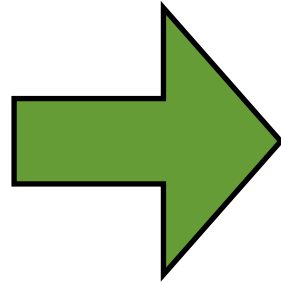
<http://docs.scipy.org/doc/numpy/reference/generated/numpy.loadtxt.html#numpy.loadtxt>



Bibliotecas como BeautifulSoup

```
<html>
<body>
  <table>
    <tr>
      <td>
        3.0
      </td>
      <td>
        4.0
      </td>
    </tr>
    <tr>
      <td>
        5.0
      </td>
      <td>
        6.2
      </td>
    </tr>
  </table>
</body>
</html>
```

3.0	4.0
5.0	6.2



```
from BeautifulSoup import BeautifulSoup

soup = BeautifulSoup(''.join(html))

table = soup.find('table')

rows = table.findAll('tr')
for tr in rows:
    cols = tr.findAll('td')
    for td in cols:
        text = ''.join(td.find(text=True))
        print text+"|",
    print
```

```
$ sudo aptitude install python-beautifulsoup
```



SciPy

- Conjunto de ferramentas para computação científica
- Álgebra Linear, Processamento

<http://www.scipy.org>

NumPy

- Módulo de alto nível em Python para trabalhar com vetores e matrizes
- Baseado em C; bem otimizado

<http://numpy.scipy.org>

História do Numpy e Scipy

Criado por Eric Jones e Travis Oliphant em
2001

Atualmente mantida por uma comunidade de
usuários



<http://enthought.com>



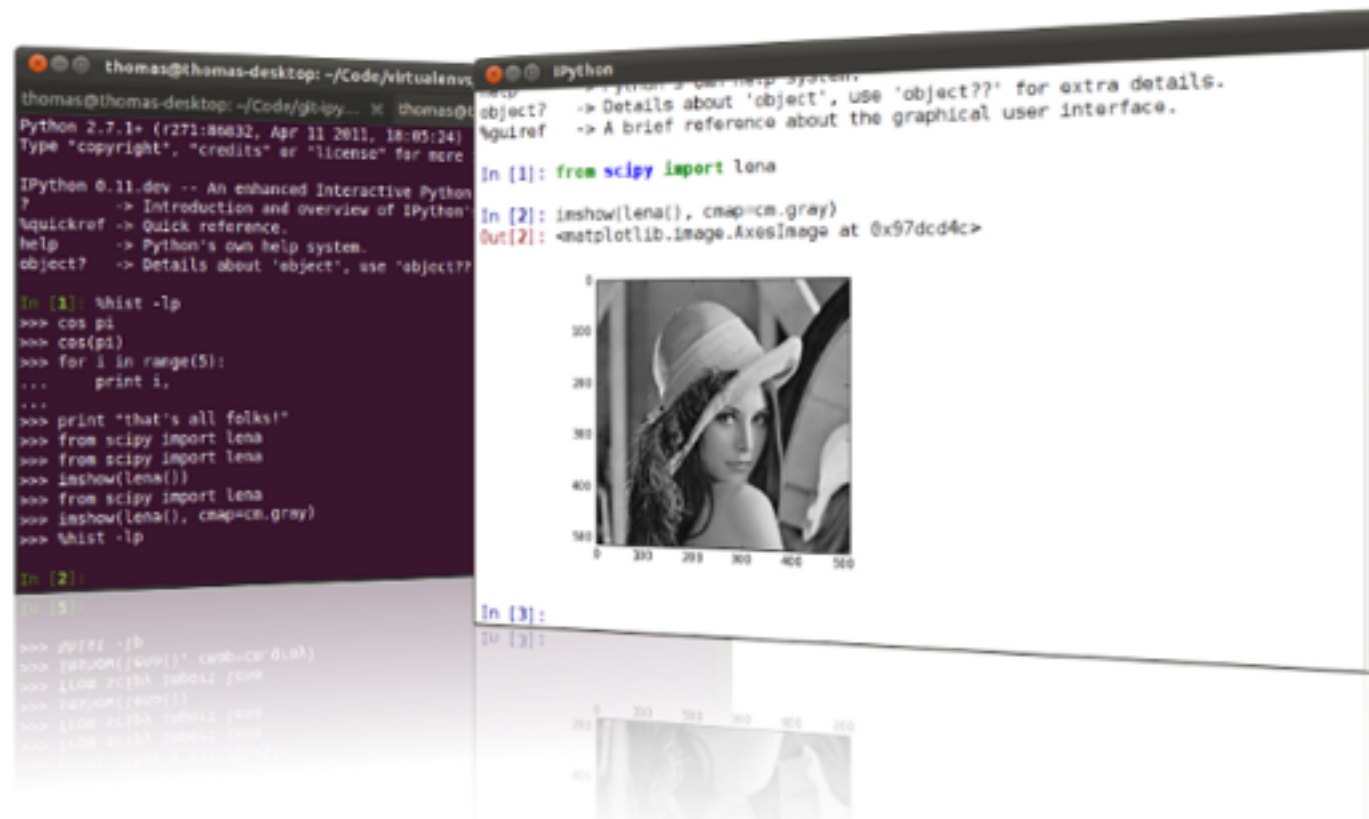
<http://conference.scipy.org/scipy2011/>

Numpy e Scipy como ambiente científico

Capaz de competir com outros softwares matemáticos como Matlab, Octave, R, Scilab, etc.

Ipython - <http://ipython.org/>

Ferramenta essencial para todo desenvolvedor que vai trabalhar com Python + Scipy + Numpy + Matplotlib



The screenshot displays the IPython environment. The left pane shows a terminal window with the following code and output:

```
thomas@thomas-desktop: ~/Code/virtualenvs
thomas@thomas-desktop: ~/Code/virtualenvs
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
Type "copyright", "credits" or "license()" for more

IPython 0.11.dev -- An enhanced Interactive Python
?      -> Introduction and overview of IPython
?quickref -> Quick reference.
help    -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.

In [1]: %hist -lp
Out[1]:
>>> cos pi
>>> cos(pi)
>>> for i in range(5):
...     print i,
...
0
1
2
3
4
>>> print "that's all folks!"
that's all folks!
>>> from scipy import lena
>>> from scipy import lena
>>> imshow(lena())
>>> from scipy import lena
>>> imshow(lena(), cmap=cm.gray)
>>> %hist -lp
Out[2]:
```

The right pane shows the IPython GUI with the following code and output:

```
In [1]: from scipy import lena
In [2]: imshow(lena(), cmap=cm.gray)
Out[2]: <matplotlib.image.AxesImage at 0x97dcd4c>
```

The GUI also displays a grayscale plot of the Lena image, with axes ranging from 0 to 500 on both the x and y dimensions.

Numpy e Scipy como ambiente científico

Comparando com o Matlab

http://www.scipy.org/NumPy_for_Matlab_Users

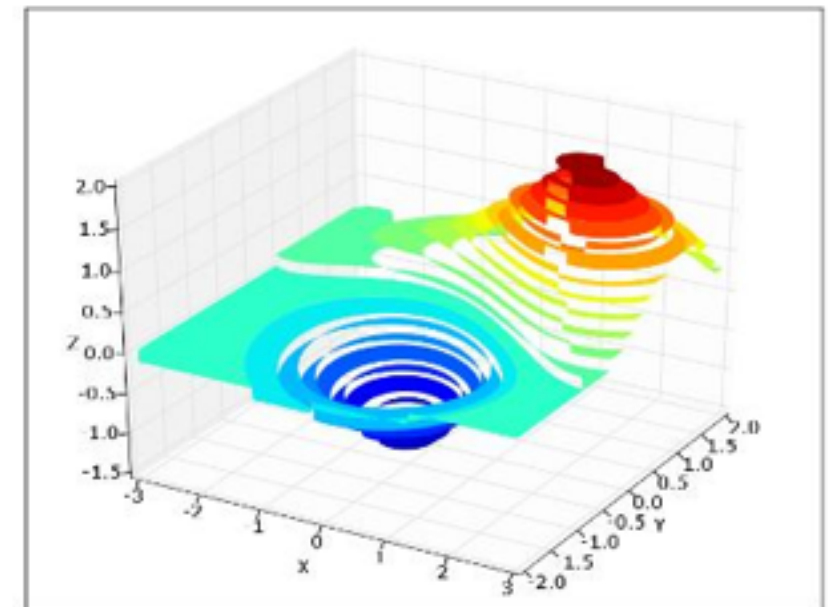
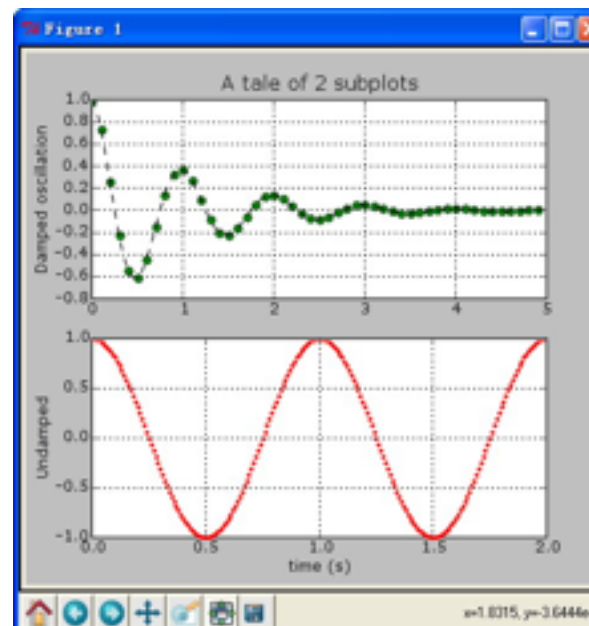
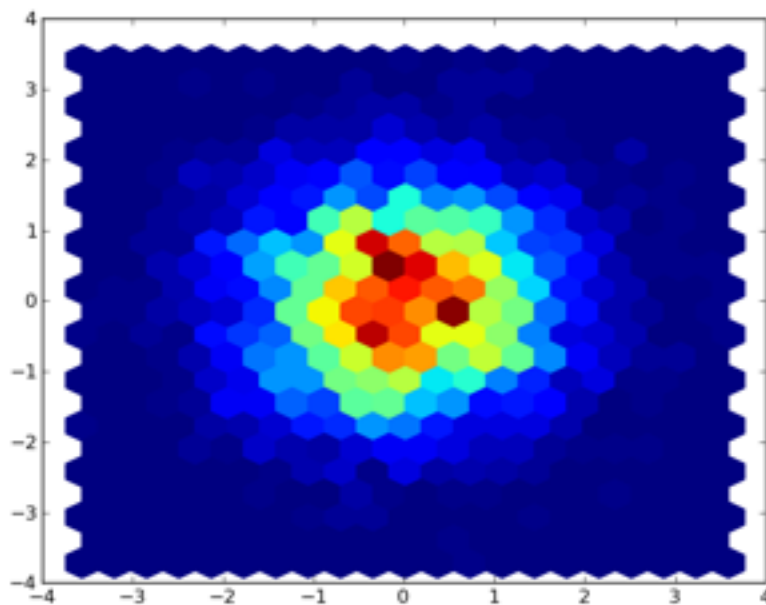
Scipy/Numpy	Matlab
Python com múltiplos argumentos, valores default, etc.	Definição de função de Matlab bem restrita
Programação Orientada a Objetos	Programação Procedural
Gratuito	Pago

Mas eu gosto de plotar gráficos!

Matplotlib - <http://matplotlib.sourceforge.net/>

Python 2D Plotting library

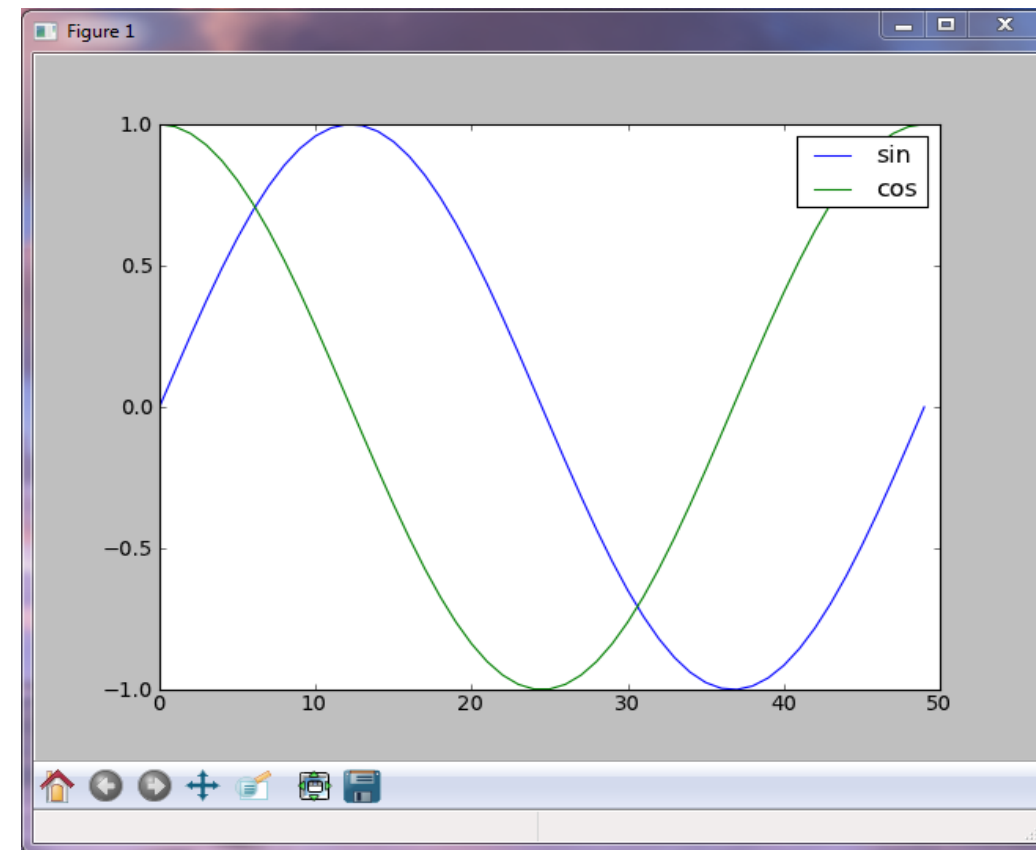
MATLAB plotting framework - *matplotlib.pyplot*



<http://www.scipy.org/Cookbook/Matplotlib>

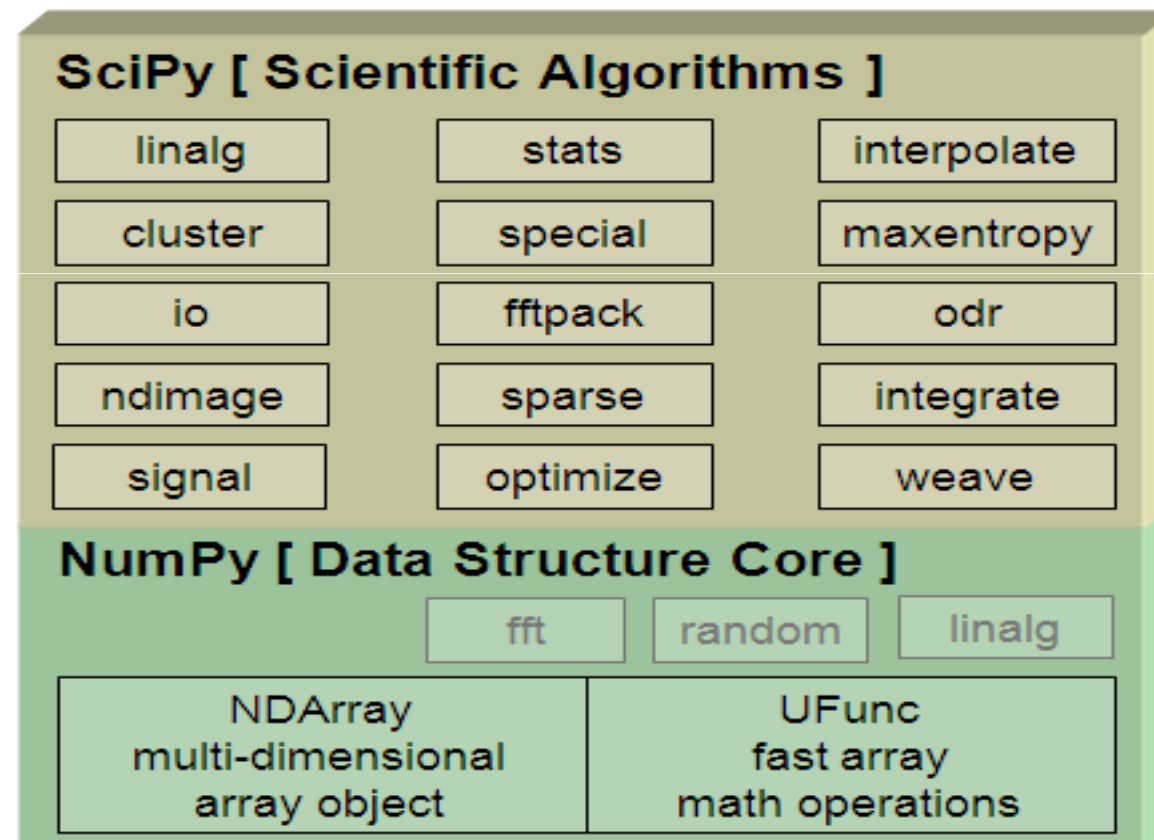
Plotando com o PyLab

```
$ ipython --pylab  
  
>>> plot(x,sin(x))  
  
>>> plot(x,cos(x))  
  
>>> legend(['sin', 'cos'])
```



Funcionalidades do Scipy

Organizado em subpacotes, abrangendo vários domínios da computação científica



Mas por onde começo ?

Scipy e Numpy disponível para diversas plataformas

<http://new.scipy.org/download.html>

ou

```
$ sudo apt-get install python-numpy
```

```
$ sudo apt-get install python-scipy
```



Explorando o Numpy

Arrays e Matrizes

```
>>> import numpy as np
```

```
>>> a = np.array([0,1,2,3,4,5],  
                 [10,11,12,13,14,15],  
                 [20,21,22,23,24,25],  
                 [30,31,32,33,34,35],  
                 [40,41,42,43,44,45],  
                 [50,51,52,53,54,55])
```

```
>>> a[0,3:5]  
array([3, 4])
```

```
>>> a[4:,4:]  
array([[44, 45],  
       [54, 55]])
```

```
>>> a[:,2]  
array([2, 12, 22, 32, 42, 52])
```

```
>>> a[2::2,::2]  
array([[20, 22, 24],  
       [40, 42, 44]])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Explorando o Numpy

Arrays e Matrizes

```
>>> import numpy as np

>>> a = np.matrix([0,1,2,3,4,5],
                  [10,11,12,13,14,15],
                  [20,21,22,23,24,25],
                  [30,31,32,33,34,35],
                  [40,41,42,43,44,45],
                  [50,51,52,53,54,55])

>>> m_t = a.transpose()

>>> mask = a < 30

>>> a[mask] #Matriz com valores < 30
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Operação de Arrays e Matrizes

Criação de Vetores

<code>numpy.zeros((M,N))</code>	<i>Vetor MxN de zeros</i>
<code>numpy.ones((M,N))</code>	<i>Vetor MxN de uns</i>
<code>numpy.empty((M,N))</code>	<i>Vetor MxN vazio (qualquer valor)</i>
<code>numpy.zeros_like(m)</code>	<i>Vetor de zeros com formato de m</i>
<code>numpy.ones_like(m)</code>	<i>Vetor de uns com formato de m</i>
<code>numpy.empty_like(m)</code>	<i>Vetor de vazio com formato de m</i>
<code>numpy.random.random((M,N))</code>	<i>Vetor com valores aleatórios</i>
<code>numpy.identity(N)</code>	<i>Matriz Identidade, N x N</i>
<code>numpy.array([(1,2,3),(4,5,6)])</code>	<i>Especifica os valores da matriz</i>
<code>numpy.matrix([(1,2,3),(4,5,6)])</code>	<i>Especifica os valores da matriz</i>
<code>numpy.arange(0.,1.,.3)</code>	<i>Vetor com Inicio I, fim F, passo P</i>
<code>numpy.linspace(0.1, 1, 10)</code>	<i>Vetor com N valores de I à F</i>

Arrays e Matrizes

```
>>> import numpy as np

>>> a = np.array([1,2,3],
                 [4,5,6],
                 [7,8,9])

>>> np.mean(a[0,:]) #media
>>> np.std(a[:,1]) #desvio-padrao
>>> np.min(a) #minimo
>>> np.max(a) #maximo
>>> b = a.T #transposta
>>> m = np.dot(a,b) #multipl.
>>> r = np.random.random((100,100))
>>> i = np.linalg.inv(r) #inversa
>>> eigval, eigvec = numpy.linalg.eig(r) #auto-vetores
```


Álgebra Linear (scipy.linalg)

```
>>> import numpy as np

>>> a = np.array([1,2,3],
                 [4,5,6],
                 [7,8,9])

>>> np.mean(a[0,:])
>>> np.std(a[:,1])
>>> np.min(a)
>>> np.max(a)
>>> b = a.T
>>> m = np.dot(a,b)
>>> r = np.random.random((100,00))
>>> i = np.linalg.inv(r)
>>> eigval, eigvec = numpy.linalg.eig(r)
```

Álgebra Linear (scipy.linalg)

Resolvendo sistemas de equações

```
>>> import numpy as np
>>> from scipy import linalg

>>> a = np.matrix('1 1 1; 1 -2 2; 0 1 2')
>>> b = np.matrix('0;4;2')
#Resolve a eq ax= b
>>> x = linalg.solve(a,b)
>>> print x
```

$$\begin{cases} x + y + z = 0 \\ x - 2y + 2z = 1 \\ y + 2z = 2 \end{cases}$$

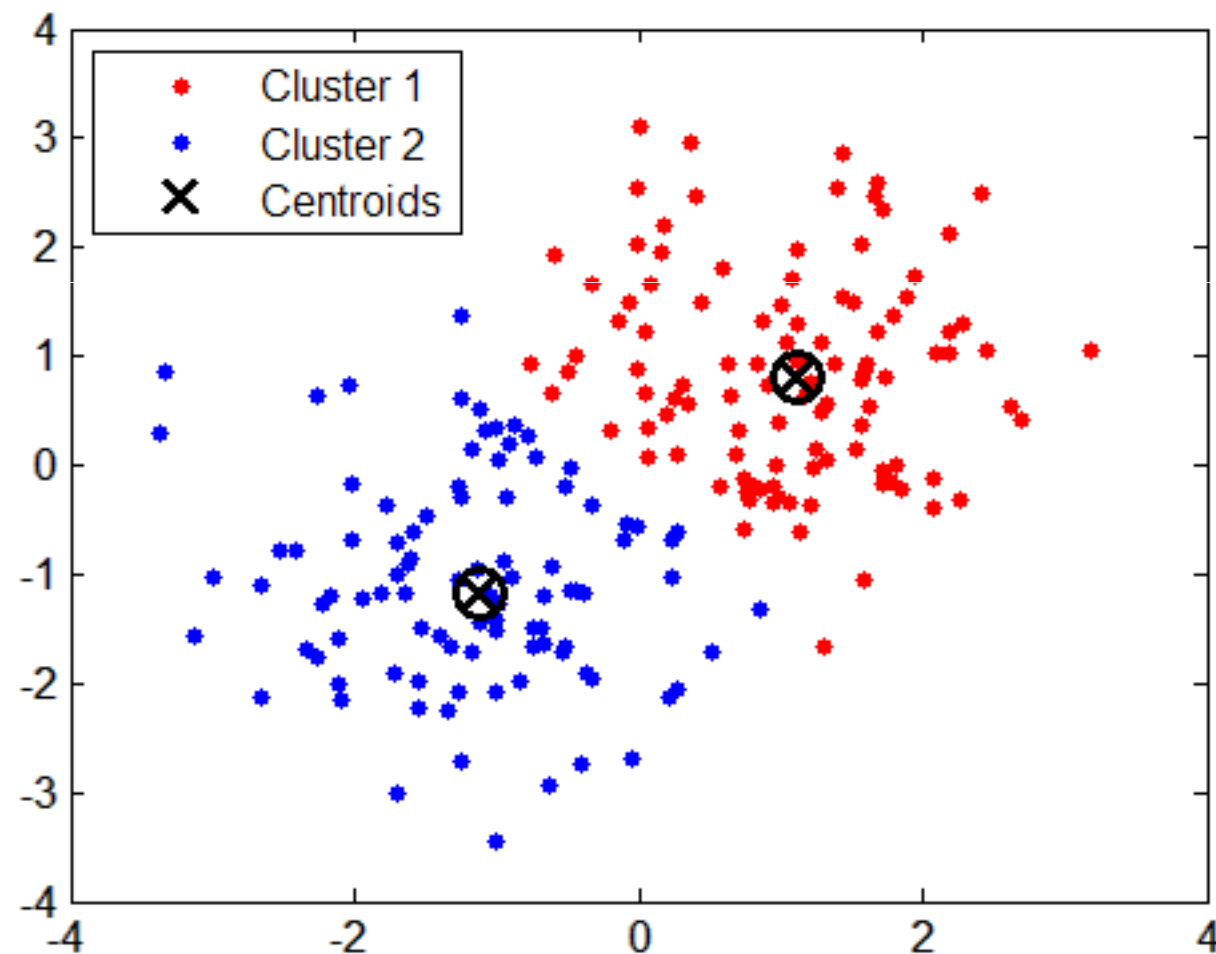
Estatísticas (scipy.stats)

```
>>> from scipy import stats
# 100 random values from a Poisson distribution with mu = 1.0
>>> s = stats.norm.rvs(loc=0.0,scale=1.0, size=100)
# basic statistics from the matrix.
>>> n, min_max, mean, var, skew, kurt = stats.describe(s)
>>> print("Number of elements: {0:d}".format(n))
>>> print("Minimum: {0:8.6f} Maximum: {1:8.6f}".format(min_max[0], min_max[1]))
>>> print("Mean: {0:8.6f}".format(mean))
>>> print("Variance: {0:8.6f}".format(var))
>>> print("Skew : {0:8.6f}".format(skew))
>>> print("Kurtosis: {0:8.6f}".format(kurt))
```

Clusterização (scipy.cluster)

Algoritmos de Agrupamento

Atualmente, apenas o **K-Means**



Explorando o Scipy

Clusterização (scipy.cluster)

- 1

```
import pylab
from numpy import array, random, vstack
from scipy.cluster.vq import vq, kmeans
```
- 2

```
# Gera duas classes normalmente distribuidas em duas dimensoes
class1 = array(random.standard_normal((100,2))) + array([5,5])
class2 = 1.5 * array(random.standard_normal((100,2)))
```
- 3

```
# concatena os dois arrays
data = vstack((class1,class2))
```
- 4

```
# Obtem os centroides e a variancia
centroids, variance = kmeans(data, 2)
```

Exemplos

Nº de clusters
- 5

```
# vq - vector quantization function
code, distance = vq(data, centroids)
```

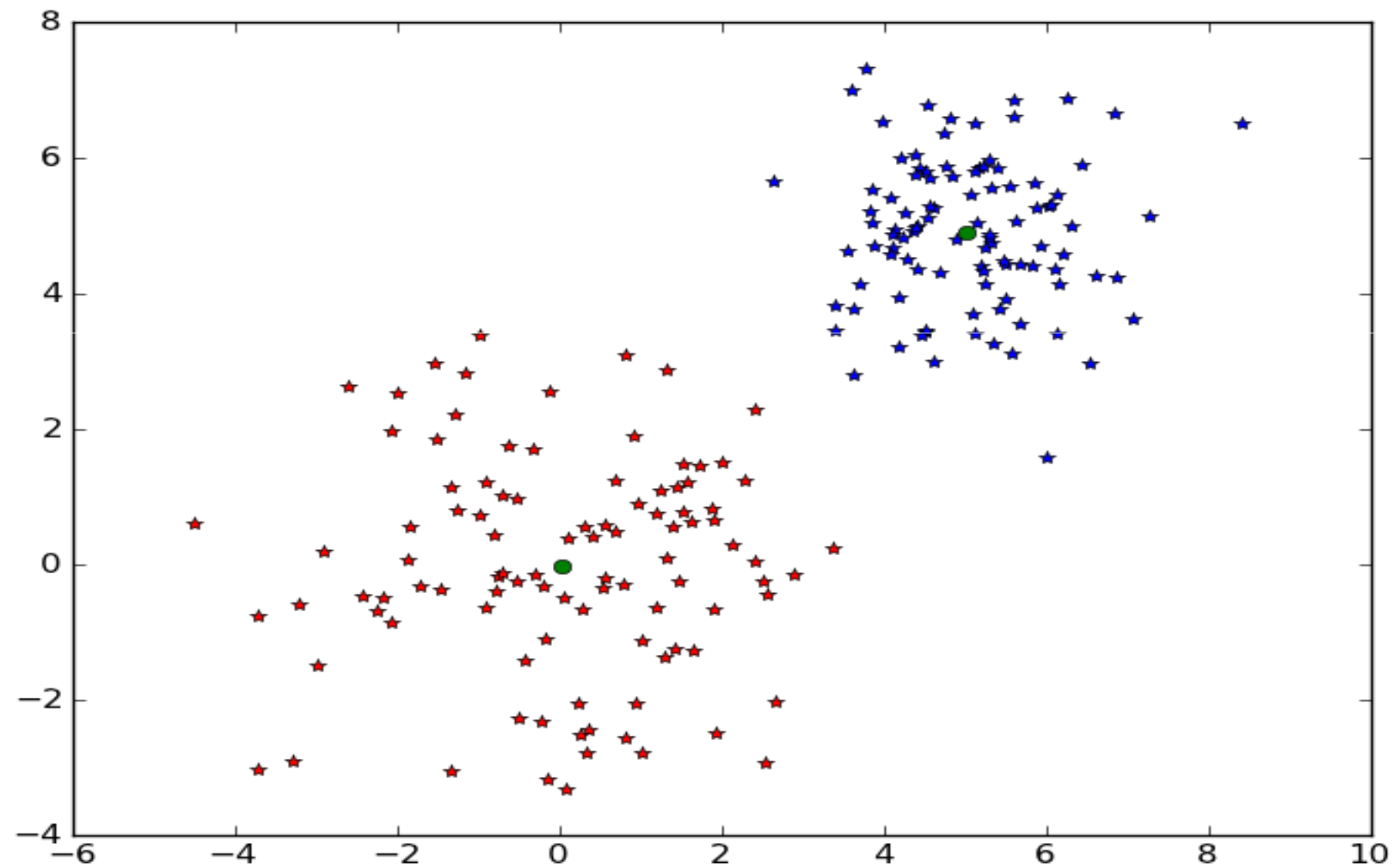
Obtêm matriz de classificação e de distâncias
- 6

```
# plota o grafico
pylab.plot([p[0] for p in class1], [p[1] for p in class1], '*')
pylab.plot([p[0] for p in class2], [p[1] for p in class2], 'r*')
pylab.plot([p[0] for p in centroids], [p[1] for p in centroids], 'go')
pylab.show()
```

Fonte: <http://www.slideshare.net/santiagosilas/computao-cientifica-com-numpy-e-scipy-7797060>



Clusterização (scipy.cluster)



Projetos interessantes usando o Scipy/Numpy

Scikit-learn <http://scikit-learn.sourceforge.net/stable/>

Toolkit de aprendizagem de máquina com algoritmos como PCA, SVM, Bayes, etc.

Divisi2 <http://csc.media.mit.edu/docs/divisi2/>

Toolkit para representação de matrizes esparsas e uso de SVD

Pandas <http://statsmodels.sourceforge.net/>

Toolkit para trabalho com dados estatísticos com Scipy e Numpy

Sympy - <http://code.google.com/p/sympy/>

Toolkit para manipulação para matemática simbólica

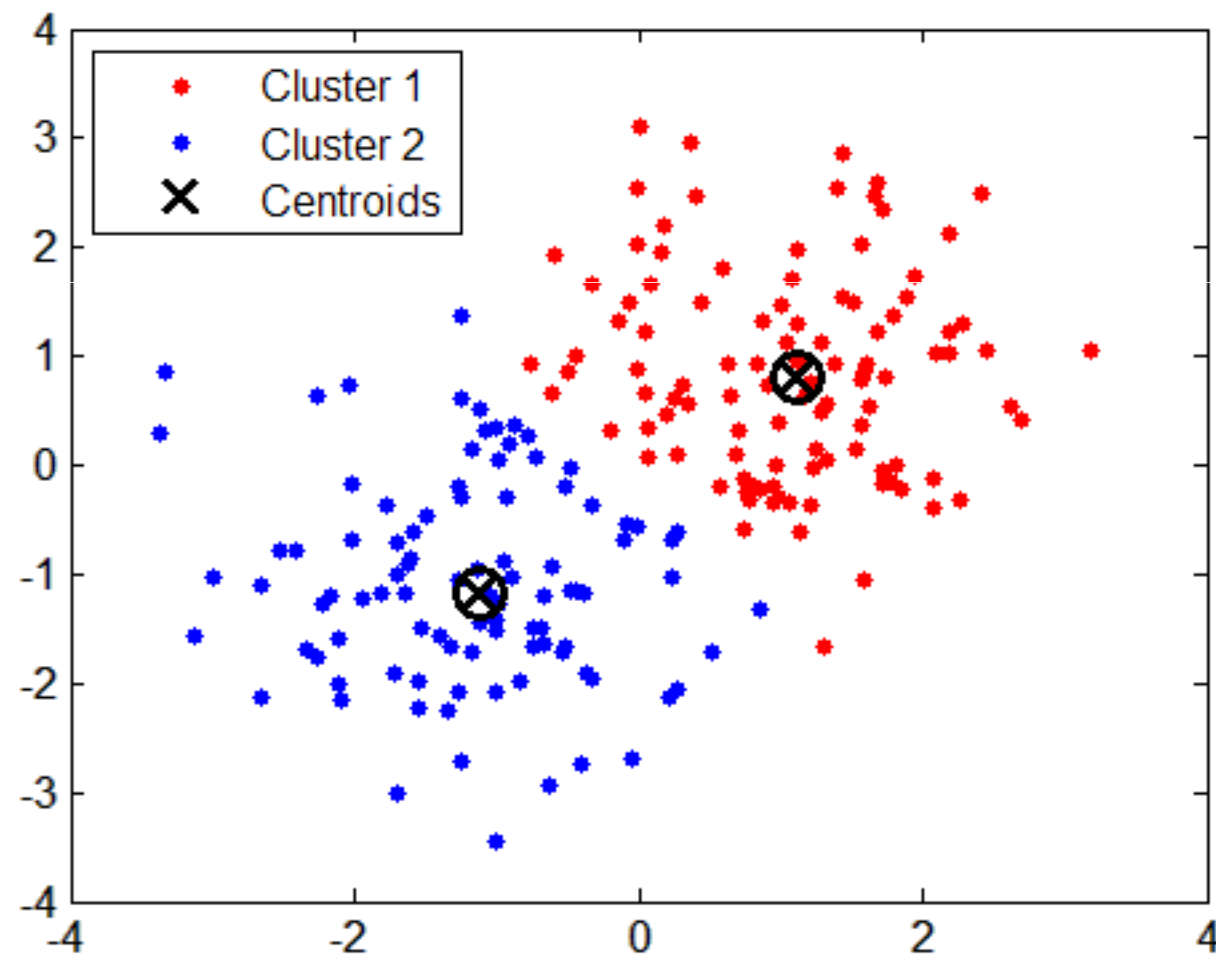
***Crab** <https://github.com/muricoca/crab>

Toolkit para construção de sistemas de recomendação com Scipy e Numpy

Clusterização (scipy.cluster)

Algoritmos de Agrupamento

Atualmente, apenas o **K-Means**



I. Documentação

<http://docs.scipy.org/doc/>

II. Tutoriais

<http://scipy-lectures.github.com/>

III. Listas de Discussão

http://www.scipy.org/Mailing_Lists

<http://pyscience-brasil.wikidot.com/>

IV. Livros

[Computação Científica com Python por Flávio Coelho, Editora Lulu.com](#)



Computação Científica com Python

A cobra também é inteligente!



Marcel P. Caraciolo
@marcelcaraciolo

marcel{@orygens, @muricoca} . com