Python para oceanógrafos

Minicurso pré-STO (2021)

Oc. Danilo A. Silva (danilo2.silva@usp.br)
Oc. Luiza P. Stein (luiza.stein@usp.br)

O que é python e por que devo usá-lo?

- Python é uma linguagem de programação utilizada em praticamente em qualquer tipo de serviço encontrado hoje em dia:

O que é python e por que devo usá-lo?

- *open source* (~~$$$$~~)
- relativamente fácil de aprender
- flexibilidade de aplicações
- comunidade científica ativa (com oceanógrafos inclusive)
- manutenção constante pela comunidade via github/gitlab
- revisão de pacotes científicos feito por cientistas!

O que é python e por que devo usá-lo?

Além disso:

- demanda de empresas da área
- serviços de download de produtos globais (mercator, ECMWF, NCEP/NCAR)
- NOAA migrando serviços de NCL para Python
- pacotes poderosos para lidar com modelos numéricos complexos

Onde rodar?

IDE (*Integrated Development Environment*):

- Pycharm
- Spyder
- VBStudio

C:/Users/TestUser/Documents/Spyder - Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\TestUser\Downloads\16740156420fb678bb5ba67c3ee3aae4-551407feca17b0f67bb9f85687f4db8d1b953678\16740156420fb678bb5ba67c3ee3aae4-551407feca17b0f67bb9f85687f4db8d1b953678

Project explorer

Spyder
Data
spyder
.github
conda.recipe
continuous_integration
doc
img_src
requirements
rope_profiling
scripts
spyder
app
tests
__init__.py
cli_options.py
mac_stylesheet.qss
mainwindow.py
restart.py
start.py
tour.py
config
defaults
fonts
images
locale
plugins
tests
utils
widgets
windows
workers
__init__.py
dependencies.py
interpreter.py
otherplugins.py
pil_patch.py
py3compat.py
pyplot.py
requirements.py
spyder_breakpoints
spyder_io_dcm
spyder_io_hdf5
spyder_profiler
spyder_pylint
.checkignore
.ciocheck
.ciocopyright
.codecov.yml
.coveragerc
.gitignore
.pep8speaks.yml
.project
.travis.yml
Announcements.md
appveyor.yml

Editor - C:\Users\TestUser\Downloads\16740156420fb678bb5ba67c3ee3aae4-551407feca17b0f67bb9f85687f4db8d1b953678\16740156420fb6...

temp.py | interpolation.py | __init__.py | umd_helper.py | umd_main.py | README.md

```
6
7  import pylab
8  from numpy import cos, linspace, pi, sin, random
9  from scipy.interpolate import splprep, splev
10
11 # %% Generate data for analysis
12
13 # Make ascending spiral in 3-space
14 t = linspace(0, 1.75 * 2 * pi, 100)
15
16 x = sin(t)
17 y = cos(t)
18 z = t
19
20 # Add noise
21 x += random.normal(scale=0.1, size=x.shape)
22 y += random.normal(scale=0.1, size=y.shape)
23 z += random.normal(scale=0.1, size=z.shape)
24
25
26 # %% Perform calculations
27
28 # Spline parameters
29 smoothness = 3.0  # Smoothness parameter
30 k_param = 2  # Spline order
31 nests = -1  # Estimate of number of knots needed (-1 = maximal)
32
33 # Find the knot points
34 knot_points, u = splprep([x, y, z], s=smoothness, k=k_param, nests=-1)
35
36 # Evaluate spline, including interpolated points
37 xnew, ynew, znew = splev(linspace(0, 1, 400), knot_points)
38
39
40 # %% Plot results
41
42 # TODO: Rewrite to avoid code smell
43 pylab.subplot(2, 2, 1)
44 data, = pylab.plot(x, y, 'bo-', label='Data with X-Y Cross Section')
45 fit, = pylab.plot(xnew, ynew, 'r-', label='Fit with X-Y Cross Section')
46 pylab.legend()
47 pylab.xlabel('x')
48 pylab.ylabel('y')
49
50 pylab.subplot(2, 2, 2)
51 data, = pylab.plot(x, z, 'bo-', label='Data with X-Z Cross Section')
52 fit, = pylab.plot(xnew, znew, 'r-', label='Fit with X-Z Cross Section')
53 pylab.legend()
54 pylab.xlabel('x')
```

Outline

interpolation.py
Generate data for analysis
Perform calculations
Plot results
imputerNan
Queue
__init__
appendleft
pop
with open(data_path + output_file_n...
with open(data_path + output_file_n...
with open(data_path + output_file_n...
print_file
Example Bitzer class
DataSet
__init__
prepare_dataset
Serie
something
Dataframe
something
foo
__init__
spam
with open(file) as f:
Base
Derived
for r, bar in zip(radii, bars):
with np.load(filename) as dem:

Variable explorer

Name	Type	Size	Value
array_int8	int8	(2, 3)	Min: -7 Max: 6
array_uint32	uint32	(2, 2, 3)	Min: 1 Max: 7
bars	container.BarContainer	20	BarContainer object of matplotlib.conta…
df	DataFrame	(3, 2)	Column names: bools, ints
filename	str	1	C:\ProgramData\Anaconda3\lib\site-packa…
list_test	list	2	[Dataframe, Numpy array]
nrows	int	1	344
r	float64	1	7.611082589334796
radii	float64	(20,)	Min: 0.4983036638535687 Max: 9.856848974942551
region	tuple	2	(slice, slice)
rgb	float64	(45, 45, 4)	Min: 0.0 Max: 1.0
series	Series	(1,)	Series object of pandas.core.series mod…
test_none	NoneType	1	NoneType object of builtins module

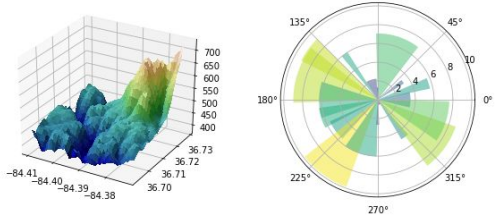
Variable explorer | Help | File explorer | Find in files | Breakpoints | Static code analysis | Profiler | Online help

IPython console

Console 1/A | Console 2/A | Custom Name/A

00:34:13

```
...:
...: ls = LightSource(270, 45)
...: # To use a custom hillshading mode, override the built-in shading
...: # in the rgb colors of the shaded surface calculated from "shade".
...: rgb = ls.shade(z, cmap=cm.gist_earth, vert_exag=0.1, blend_mode='soft')
...: surf = ax.plot_surface(x, y, z, rstride=1, cstride=1, facecolors=rgb,
...:                        linewidth=0, antialiased=False, shade=False)
...:
...: plt.show()
```



In [12]:

IPython console | History log | Internal console

Permissions: RW | End-of-lines: LF | Encoding: UTF-8 | Line: 26 | Column: 4 | Memory: 49 % | CPU: 15 %

Onde rodar?

IDE (*Integrated Development Environment*):
- Pycharm
- Spyder
- VBStudio

Outros:
- Google Colab
- Jupyter

Computação em nuvem (*Cloud computing*):
- escalabilidade usando conda + jupyter + github
- AWS ou Azure, por exemplo

Onde rodar?

IDE (*Integrated Development Environment*):
- Pycharm
- Spyder
- VBStudio

Outros:
- Google Colab
- Jupyter *

Computação em nuvem (*Cloud computing*):
- escalabilidade usando conda + jupyter + github
- AWS ou Azure, por exemplo

Ambientes virtuais

- O que são?
  - Empacotamento de bibliotecas e softwares isolados do seu sistema
- Por que usá-los?
  - Proteção do seu sistema
  - Portabilidade
  - Escalabilidade
- Cada projeto = 1 ambiente [!!!!]
  - Preservar as versões de cada biblioteca para reproduzir
  - Organização
  -

Ambientes virtuais: alguns nomes importantes

CONDA

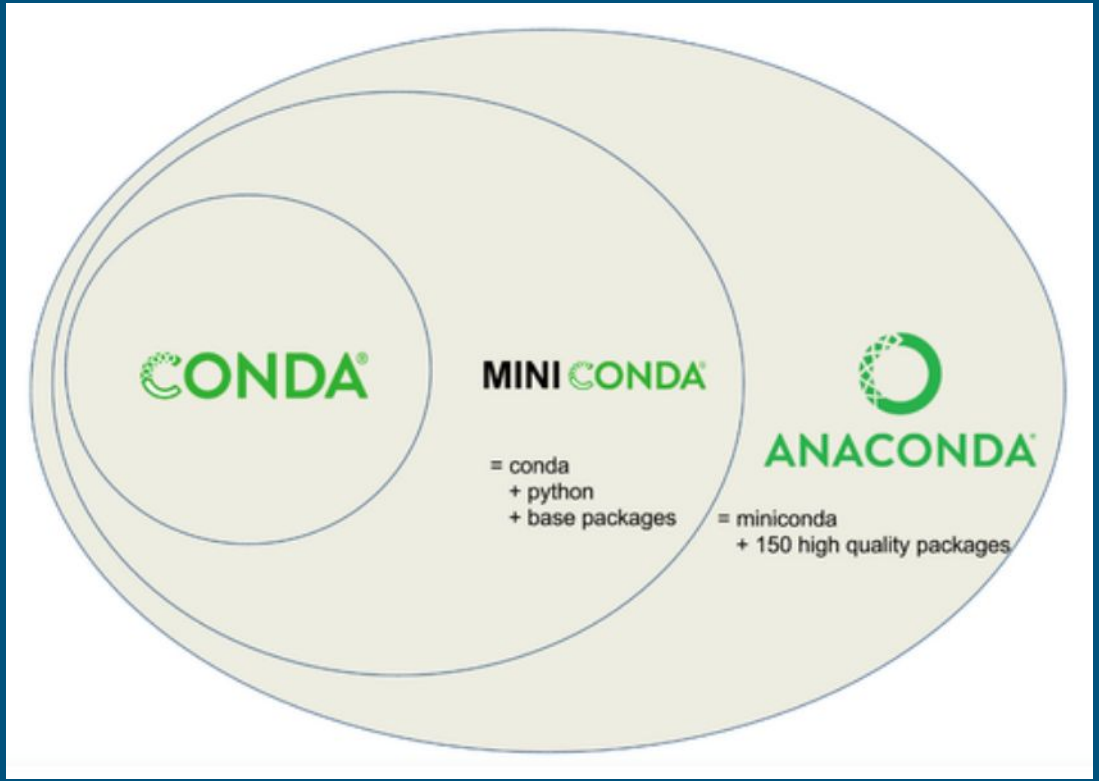
- gerenciador de ambientes virtuais

MINICONDA

- instalação limpa do conda com alguns pacotes úteis

ANACONDA

- instalação completa do conda com muitos pacotes



Ambientes virtuais: alguns nomes importantes

CONDA

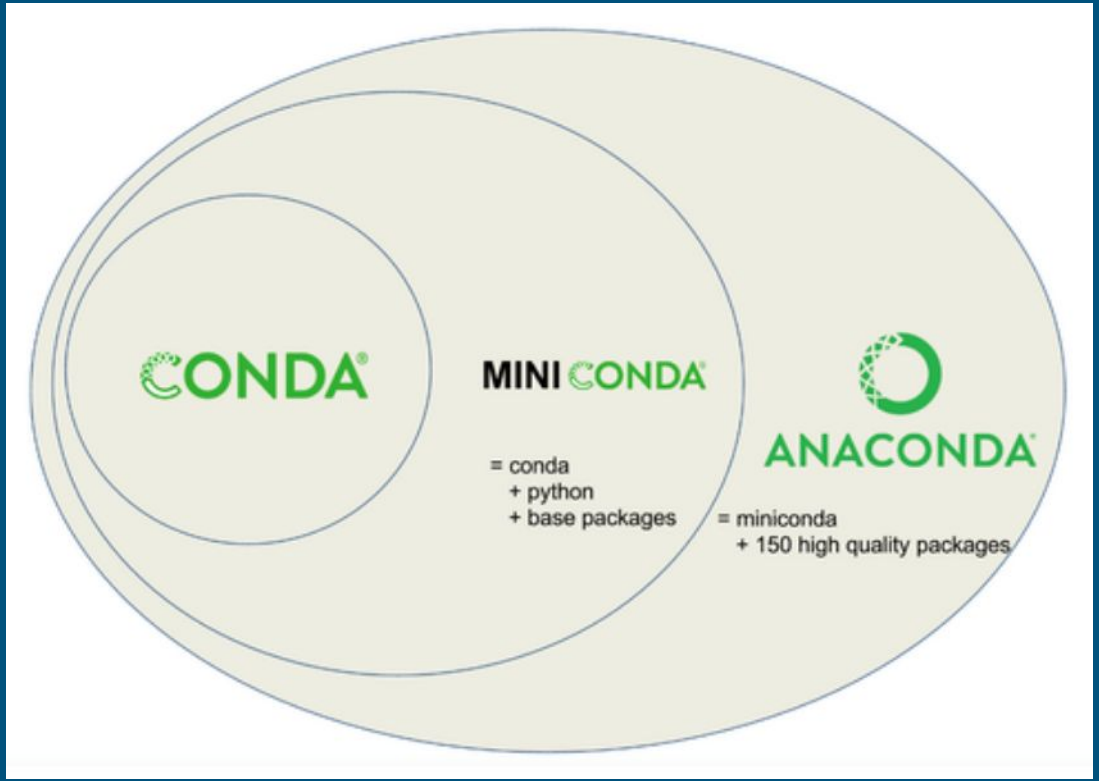
- gerenciador de ambientes virtuais

MINICONDA

- instalação limpa do conda com alguns pacotes úteis

ANACONDA

- instalação completa do conda com muitos pacotes



Ambientes virtuais: instalação

Linux e MacOS:

- https://whiteboxml.com/blog/the-definitive-guide-to-python-virtual-environments-with-conda

Windows:

- https://www.inatel.br/pesquisador/images/tutorial-de-instalacao-ambiente-python.pdf

Ambientes virtuais

Vejamos um pouco de terminal ...

Github

- controle de versionamento
- códigos, textos, mas não dados!
- material do curso:
  - https://github.com/nilodna/python-basico/tree/feature_presto_shortcourse