

Data Analysis

Velen Kong

摘要

基于Python的数据分析入门笔记，希望能自学掌握基本的数据分析能力，毕竟是统计学的基础之一。

内容安排主要参考

(美) Wes McKinney 著. Python for Data Analysis:2nd Edition [M]
USA: O'Reilly Media 2017

该书作者同时也是pandas库的作者，同时借用其在GitHub上的数据资料。

第三方库与Python入门

IPython & Jupyter

用到的库: ipython, jupyter, numpy, matplotlib, pandas, scipy, scikit-learn, statsmodels。

jupyter确实好用，扩展了`Tab`, `?`, `*`的功能。

还有一些快捷键仅限IPython使用。

另外就是一些常用的Magic Command, 比如

```
1 %run
2 %load
3 %debug
4 %xmode Plain # less detailed
5 %xmode Verbose # more detailed
6 %matplotlib inline # draw in Jupyter
7 import numpy as np
8 import matplotlib.pyplot as plt
9 plt.plot(np.random.rand(50).cumsum())
```

```
1 # ----- Day 0 -----
```

Python基础

#注释。

```
1 # this is comment
```

所有变量都是对象(object)。

基本的函数调用, 利用函数批量处理。

```
1 def append_element(some_list, element):
2     some_list.append(element)
3     return
4
5 data = [1, 2, 3]
6 append_element(data, 4)
7 data
8 # Out: [1, 2, 3, 4]
```

变量赋值类似引用(reference)。

动态类型，利用type()和isinstance()进行类型检查。后者可以传入tuple代替逻辑或操作。

```
1 a = 1.5
2 isinstance(a, (int, float))
3 # Out: True
```

print配合format输出。

```
1 a = 1; b = 1.5
2 print('a is {1}, b is {0}'.format(type(b),
3                                     type(a)))
3 # Out: a is <class 'int'>, b is <class 'float'>
```

Python中的object拥有各自的属性(attributes)和方法(methods), 可通过getattr, setattr, setattr操作。其中getattr可以直接使用返回对象，setattr不改变原class。

```
1 class A(object):
2     def set(self, a, b):
3         x = a
4         a = b
5         b = x
6         print a, b
7
8 a = A()
9 c = getattr(a, 'set')
10 c(a='1', b='2')
11 # Out: 2 1
```

可以利用try()检查容器是否是顺序，同时生成list。

```
1 def isiterable(obj):
2     try:
3         iter(obj)
4         return True
5     except TypeError:
```

```
6         return False
7     return
8
9 x = 'string'
10 iterable(x)
11 # Out: True
12 iterable(5)
13 # Out: False
14
15 if not isinstance(x, list) and iterable(x):
16     x = list(x)
17 x
18 # Out: ['s', 't', 'r', 'i', 'n', 'g']
```

import和as的使用。

is和is not的使用。

```
1 a = [1, 2, 3]
2 b = a
3 c = list(a)
4 d = None
5 print(a is b)
6 # Out: True
7 print(a is not c)
8 # Out: True
9 print(a == c)
10 # Out: True
11 print(d is None)
12 # Out: True
```

大部分Python的容器都是可变的(mutable)，而strings和tuples不可变(immutable)。

```
1 a = ['foo', 2, [4, 5]]
2 a[2] = (3, 4)
3 a
4 # Out: ['foo', 2, (3, 4)]
```

```
1 # ----- Day 1 -----
```

Numpy库
