Python 数据分析与数据挖掘（Python for Data Analysis&Data Mining）

Chap 1 概述和环境配置（Preliminaries）

1. Python环境配置和安装

2. 重要的Python库

3. Hello, Python

内容：

- Python环境配置

实践：

- Anaconda或Pycharm安装
- 基础环境ipython / ipython notebook
- Install package / Import package
- 几个Python小例子

1. Python环境配置和安装

由于人们使用Python所做的事情不同，所以没有普适的Python及其插件包的安装方案。要想使Python的科学计算环境满足我们这个课程的需要，建议使用下面的Python安装包之一。对于初学者，**建议使用第一种Anaconda安装包**。

了解如何安装IPython到http://ipython.org/install.html#i-am-getting-started-with-python (http://ipython.org/install.html#i-am-getting-started-with-python)

1). Anaconda

> 到Anaconda下载页面 (https://www.continuum.io/downloads)下载和安装适合系统的安装包，建议选择python2版本。Python2.7版本的知识与维护，将延续到2020年，以后将迁移到Python3。
>
> 下载Anaconda2，并安装到 你的驱动器:/Anaconda/目录下；
>
> 打开Anaconda Command Prompt (C:/Anaconda/)，在命令提示行中，执行如下的命令：
>
> - `conda update conda`
> - `conda update ipython ipython-notebook ipython-qtconsole`
>
> 要启动ipython notebook, 执行如下命令:
>
> - `cd <directory where you want to save notebook>`，建议大家新建C:/YourName/目录存放本课程的学习资料
> - `ipython notebook`

新版本执行jupyter notebook命令启动

2). PyCharm

> PyCharm 是一种Python IDE，从http://www.jetbrains.com/pycharm/download/ (http://www.jetbrains.com/pycharm/download/) 下载安装。 PyCharm带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完 成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。PyCharm的安装需要注册码，适合Python高级使用者。

3). Enthought Canopy

> 来自Enthought的面向科学计算的Python安装包https://www.enthought.com/ (https://www.enthought.com/) 。从 http://www.enthought.com (http://www.enthought.com) 下载 EPDFree的安装包，它可能是一个名字类似于epd_free-7.3-1-win-x86.msi的MSI安装包。EPDFull安装包面向高校老师是免费的，需要使用官方邮箱注册。
>
> - Enthought Python Distribution (EPD) 现在改名为 **Enthought Canopy!**

2. 重要的Python库

1). NumPy

即Numerical Python 的简称，是Python科学计算的基础包。数据分析的大部分内容都基于NumPy以及构建于其上的库。对于数值型数据，NumPy数组在存储和处理数据时要比内置的Python中的列表数据结构高效得多，因为NumPy数组是针对某些对象进行了大量的优化工作。可以提供的功能包括（不限于此）：

- 快速高效的多维数组对象ndarray
- 用于对数组执行元素级计算以及直接对数组执行数学运算的函数
- 用于读写硬盘上基于数组的数据集的工具
- 线性代数运算、傅立叶变换，以及随机数生成
- 用于将C、C++、Fortran代码集成到Python的工具（由C和Fortran编写的库可以直接操作NumPy数组中的数据，无需进行数据复制工作）
- 作为在算法之间传递数据的容器

2). pandas

名字源于panel data（面板数据，是计量经济学中关于多维结构化数据集的一个术语）以及Python data analysis（Python数据分析），很适合金融数据分析应用的工具，也是本书中使用的主要工具

- 兼具NumPy高性能的数组计算功能以及电子表格和关系型数据库（如SQL）灵活的数据处理能力
- 提供了复杂精细的索引功能，以便更为便捷地完成重塑、切片和切块、聚合以及选取数据子集等操作
- 对于金融行业的用户，pandas提供了大量适用于金融数据的高性能时间序列功能和工具
- 用的最多的pandas对象是DataFrame，是一个面向列（column-oriented）的二维表结构，含有行标和列标

3). matplotlib

matplotlib是最流行的用于绘制数据图表的Python库。它非常适合创建出版物上用的图表，跟IPython结合得很好，因而提供了一种非常好用的交互式数据绘图环境。绘制的图表也是交互式的，可以利用绘图窗口中的工具栏放大图表中的某个区域或对整个图表进行平移浏览。

4). IPython

IPython是Python科学计算标准工具集的组成部分，它将其他所有的东西联系到了一起。为交互式和探索式计算提供了一个强健而高效的环境。

- 它是一个增强的Python shell，目的是提高编写、运行、测试、调试Python代码的速度。
- 它主要用于交互式数据处理和利用matplotlib对数据进行可视化处理。

除了标准的基于终端的IPython shell外，该项目还提供了：

- 一个类似于Mathematica的HTML笔记本（通过Web浏览器连接IPython）
- 一个基于Qt框架的GUI控制台，其中含有绘图、多行编辑以及语法高亮显示等功能
- 用于交互式并行和分布式计算的基础架构

5). SciPY

SciPy是一组专门解决科学计算中各种标准问题域的包的集合。 NumPy 和SciPy的有机结合完全可以替代MATLAB的计算功能（包括其插件工具箱）。SciPy的主要包括下面这些包：

- scipy.integrate: 数值积分例程和微分方程求解器
- scipy.linalg: 扩展了由 numpy.linalg 提供的线性代数例程和矩阵分解功能
- scipy.optimize: 函数优化器（最小化器）以及根查找算法
- scipy.signal: 信号处理工具
- scipy.sparse: 稀疏矩阵和稀疏线性系统求解器
- scipy.special: SPECFUN（这是一个实现了许多常用数学函数（如伽玛函数）的Fortran库）的包装器
- scipy.stats: 标准连续和离散概率分布（如密度函数、采样器、连续分布函数等）、各种统计检验方法，以及更好的描述统计法
- scipy.weave: 利用内联C++代码加速数组计算的工具

有关anaconda的有用信息：

- conda info #显示工具的有用信息
- conda search pytables # 搜索库和软件包，可以下载和安装以及已经安装（由*标出）的pytables版本
- conda list ^pyt # 显示pyt开始的所有软件包

常用模块的命名惯例

- `import numpy as np`
- `import pandas as pd`
- `import matplotlib.pyplot as plt`

当看到np.arange时，就应该想到它引用的是NumPy中的arange函数。这样做的原因是：在Python软件开发过程中，不建议直接引入类似NumPy这种大型库的全部内容（不建议 `from numpy import *`）

安装库的命令

- pip install package-name

或者在anaconda模式下，使用

- conda install package-name

本书需要下面的包：

- Python中交互式Python解析器IPython，科学计算基础库：NumPy，SciPy，pandas，和绘图库matplotlib等大多数常用库默认在anaconda中安装
- 其他库如requests，beautifulsoup，statsmodels, PyTables, xlrd，xlwt等，它们被用在不同示例中，可以在后面的课程中需要的时候再安装

知乎：哪些 Python 库让你相见恨晚？ (https://www.zhihu.com/question/24590883)

本课程中一些重要的Python库包括以下：

- requests：人性化的HTTP请求库。
- BeautifulSoup：以 Python 风格的方式来对 HTML 或 XML 进行解析，迭代，搜索和修改。
- openpyxl：一个用来读写 Excel 2010 xlsx/xlsm/xltx/xltm 文件的库。
- xlwt / xlrd：读写 Excel 文件的数据和格式信息。
- python-docx：读取，查询以及修改 Microsoft Word 2007/2008 docx 文件。
- XlsxWriter：一个用于创建 Excel .xlsx 文件的 Python 模块。
- NLTK：一个先进的平台，用以构建处理人类语言数据的 Python 程序。
- jieba：中文分词工具。
- pickleDB：一个简单，轻量级键值储存数据库。
- scikit-learn：基于 SciPy 构建的机器学习 Python 模块。

绘图示例

Ipython notebook的一个非常好的特性是，能够生成和插入高质量的可定制数据图，作为交互式工作流的一部分。特别是matplotlib包和其他科学计算工作对于

IPython Notebook是可用的，功能非常强大，一旦理解了基本的流程就可以很轻松生成负责的图表。

> 有两种启动绘图的方式：（1）pylab （2）matplotlib，都可以使用pip安装。
>
> - $ pip install pylab
> - $ pip install matplotlib
>
> 在启动IPython Notebook时启用绘图的方式是：
>
> - ipython notebook --pylab=inline
> - ipython notebook --matplotlib=inline
>
> 在Python的源代码中启动绘图方式有两种：
>
> 方法1：
>
> - %matplotlib inline
> - import matplotlib.pyplot as plt
>
> 方法2：
>
> - %pylab inline

可以如下启动ipython notebook

```
ipython notebook --pylab=inline
```

3. Hello, Python

第一个Python代码

In [ ]:

```
# 运行方式1： 在ipython notebook模式下即现
print "Hello, Python!"
```

运行方式2：也可以将上面的代码存入到hello.py文件中，然后使用python hello.py方式运行；

运行方式3：启动ipython，然后在提示符后输入上述print语句

Python学习手册

- !pip install可以检查是否安装了某个库
- 任何前置了！号的命令行都将发送给系统的shell来处理
- 可以使用变量来存储命令的输出结果

In [ ]:

```
#!pip install numpy
```

- 可以使用如下的方法来存储命令的输出结果：

注意：这个方式可能等待时间很久，因为有的package正在后台下载安装中，慎用！

In [ ]:

```
a = !pip install numpy
a
```

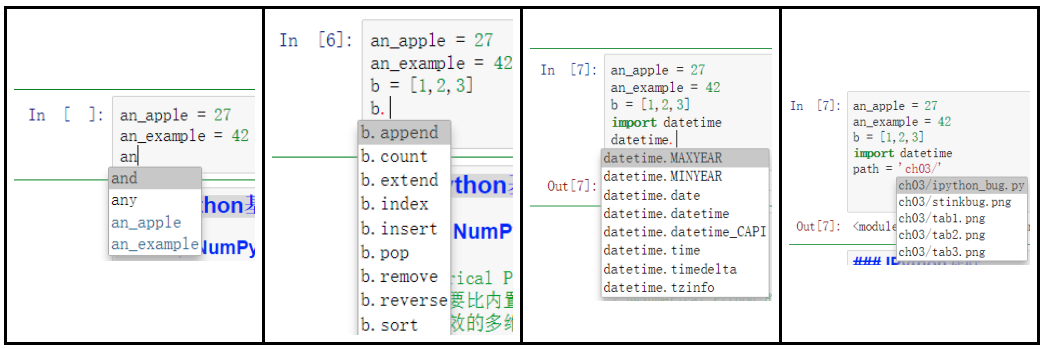
Python帮助手册

- Tab键的补全功能，输入函数名的前面字符，再按Tab键
- 在对象后面输入一个句点以便自动完成方法和属性的输入
- 调用help命令, help()
- 通过在函数名后面加上问号？进行查询。前提是要知道函数名，好处是不必输入help命令

Tab键自动完成

这个功能是对是对标准Python shell的主要改进之一，大部分交互式数据分析环境都有这个功能。在shell中输入表达式时，只要按下Tab键，当前命名空间中任何与已输入的字符串相匹配的变量（对象、函数等）就会被找出来。

下面左图IPython将定义的两个变量都显示出来了，此外还显示了Python关键字and和内置函数any。中图显示在任何对象后面输入一个句点以便自动完成方法和属性的输入。右图显示这个功能还可以应用在模块上。最右边的图显示，Tab键自动完成功能不只可以用于搜索命名空间和自动完成对象或模块属性。当你输入任何看上去像是文件路径的东西时（即使是在一个Python字符串中），按下Tab键即可找出电脑文件系统中与之匹配的东西：



如果再结合%run命令（参见后面内容），该功能将显著减少你敲键盘的次数。Tab键自动完成功能还可用于函数关键字参数（包括等号（=）！）

注意：

> IPython默认会隐藏那些以下划线开头的方法和属性，比如魔术方法（magic method）以及内部的“私有”方法和属性，其目的是避免在屏幕上显示一堆乱七八糟的东西（也为了避免把Python新人搞晕！）。其实这些也是可以通过Tab键自动完成的，只是你得先输入一个下划线才行。如果你就是喜欢能总是看到这些方法，直接修改IPython配置文件中的相关设置就可以了。

In [ ]:

```
#range?
```

In [ ]:

```
#help()
```

几个Python的小示例

- 导入绘图库，画图
- 导入两个常用库，numpy和pandas
- 导入外部数据，显示前面几行

In [ ]:

```
# 在python代码中有如下两种启动绘图的方式
# 方式 1
%matplotlib inline
import matplotlib.pyplot as plt

# 或者 等同于下面的方式 2
#%pylab inline
```

In [ ]:

```
# 导入两个常用的库，在导入库之前先安装库
# numpy和pandas在anaconda中已经默认安装
import numpy as np
import pandas as pd

# 启动绘图
%matplotlib inline
import matplotlib.pyplot as plt

plt.plot(np.arange(10)) # 画出从0到9的曲线，使用numpy
#plt.plot(range(10)) #同上，使用python
```

In [ ]:

```
# 导入两个常用的库，在导入库之前先安装库
# numpy和pandas在anaconda中已经默认安装
import numpy as np
import pandas as pd

# 启动绘图
%matplotlib inline
import matplotlib.pyplot as plt

#绘制正弦曲线
x = np.linspace(-10, 10, 1000)
y = np.sin(x)
plt.plot(x,y)
```

In [ ]:

```
# 在python代码中有如下两种启动绘图的方式
# 方式 1
#%matplotlib inline
#import matplotlib.pyplot as plt
# 方式 2
%pylab inline

img = plt.imread('image/stinkbug.png')
imshow(img)
```

In [ ]:

```
namelist = pd.read_csv('data/xiaoshan.csv')
namelist.head()# 最前五行，不含头行title
```

In [ ]:

```
namelist.tail() # 最后五行
```

In [ ]:

```
f = 'data/Pujiang2008.csv'
namelist = pd.read_csv(f)
namelist.head()# 最前五行，不含头行title
```

In [ ]:

```
# python 读取中文文件名/中文路径的方法
#encoding: utf-8
#import sys
#reload(sys)
#sys.setdefaultencoding('utf-8')
#f = 'L01/萧山街道.csv'
#uf = unicode(f, "utf8")
#namelist = pd.read_csv(uf)
#namelist.head()
```

In [ ]:

```

```