Python实践逻辑回归的准备工作

执行逻辑回归需要先安装几个Python中用于科学计算的包（package），本文介绍在Windows系统和在Linux系统安装这些Python包的方法。

1. Windows系统

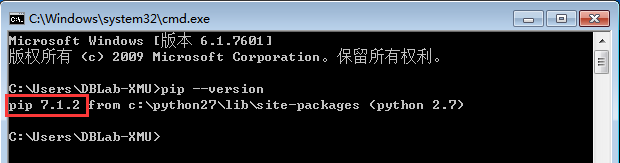
在Windows系统上安装package相对麻烦些，但跟着本教程的步骤，应该都可以顺利完成安装。

1. 安装pip

Pip(<https://pypi.python.org/pypi>)是一个安装和管理Python包的工具，如果使用的是Python 2.7.11版本（即按照教程” Python安装与基本使用.docx”进行安装），则默认已经安装了pip，在开始菜单中找到“运行”（或者使用键盘快捷键Win + r），输入“cmd”，打开Windows 命令提示符，执行如下命令进行检验：

|  |
| --- |
| pip --version |

若已经装好了pip，运行结果如下图如下，显示已安装的版本：



若已安装好pip，则可以跳过该部分剩余内容，开始安装所需的包。如果还未安装pip，则按照如下步骤进行安装：

1. 下载安装文件

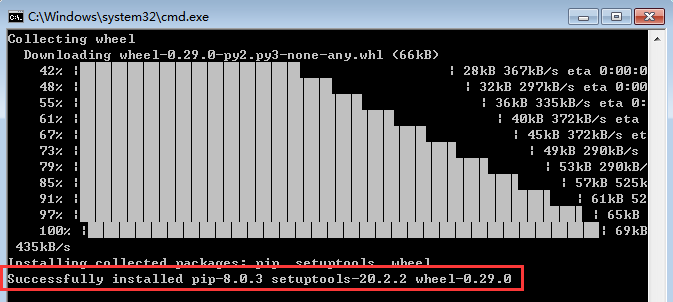
访问<https://bootstrap.pypa.io/get-pip.py>，下载该文件（若不是直接下载，则在页面中右键点击，选择另存为），或者使用备用下载地址: <http://pan.baidu.com/s/1c0NL4eW>。

1. 进行安装

为方便，我们将上述文件保存在D盘根目录下（即d:\get-pip.py），在cmd中，依次执行如下代码：

|  |
| --- |
| d:  cd d:\  python get-pip.py |

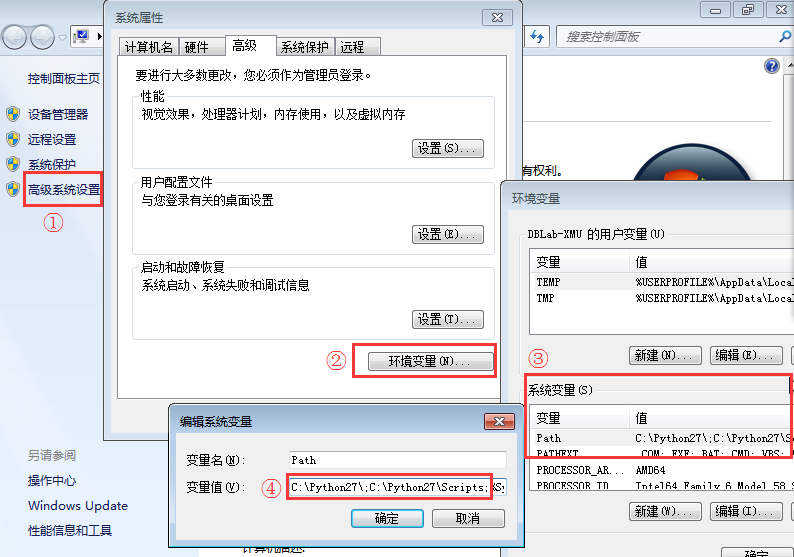
安装成功后如下图所示，提示“Successfully installed pip-xxx”：



安装pip可能会遇到的问题（若已安装成功请跳过这部分内容）：

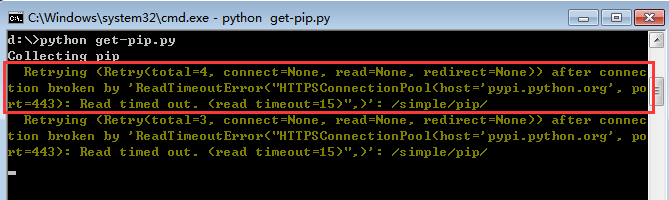
1. 提示“Python不是内部或外部命令，也不是可运行的程序或批处理文件”

需要将Python安装目录加入到环境变量中：右键点击“我的电脑”，选择“属性”，找到“系统属性”，点击“高级”面板的“环境变量”，在“系统变量”中找到“Path”变量，双击进行编辑，在最前面加上“c:\Python27\;C:\Python27\Scripts;”，其中“c:\Python27”为Python的安装目录，如下图所示：



添加环境变量后，重新开启cmd，再次执行命令安装pip即可。

1. 提示如下图“Retrying”



可以多尝试几次，或者下载离线安装包进行安装：[pip7.1.2下载地址](https://pypi.python.org/packages/source/p/pip/pip-7.1.2.tar.gz#md5=3823d2343d9f3aaab21cf9c917710196) | [备用下载](http://pan.baidu.com/s/1dEolOeH)

下载解压后，假设放在D盘根目录下（d:\ pip-7.1.2），执行如下命令进行安装：

|  |
| --- |
| d:  cd d:\get-pip  python setup.py |

1. 安装所需的包

安装好pip后，就可以在cmd中使用“pip install package-name”的方式来方便地在线安装python包了。例如，我们要安装一个名为 six 的包，只需要在cmd中执行如下命令即可：

|  |
| --- |
| pip install six |

在本实验中，我们需要安装如下几个包：

* Numpy
* Pandas
* Scipy
* Patsy
* Statsmodels

实践中，在线安装Numpy、Scipy时会遇到一些问题，可以解决但比较繁琐，因此推荐采用离线安装方式的方式来安装本作业所需的这几个包。

Windows的Python离线安装包可在<http://www.lfd.uci.edu/~gohlke/pythonlibs/> 中下载，下载后使用“pip install 离线安装包文件名”的命令进行安装。读者可直接下载我们已整理好的安装包（下面的32位、64位指的是你所装的Python版本，仅适用于Python 2.7系列版本）：

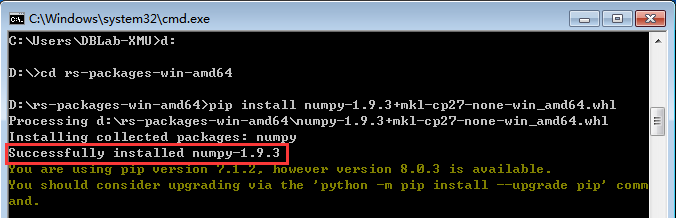
32位下载：<http://pan.baidu.com/s/1dEqomXj>

64位下载：<http://pan.baidu.com/s/1c0YHbvM>

以64位为例，下载后解压，假设放置于D盘根目录下（d:\rs-packages-win-amd64），首先安装numpy，在cmd中执行如下命令进行安装（注意32位的安装命令有所不同，主要是文件名不同）：

|  |
| --- |
| d:  cd d:\rs-packages-win-amd64  pip install numpy-1.9.3+mkl-cp27-none-win\_amd64.whl |

安装成功的话如下图所示，提示“Successfully installed numpy-1.9.3”：



对于提示”You are using pip version 7.1.2, however version 8.0.3 is available”，可以按照提示升级pip，也可以选择忽略该信息，并不会影响Python包的安装与使用。

接着，再依次执行如下命令安装其他Python包（需要额外安装一些依赖包，因此请注意安装顺序。同时注意32位的安装命令有所不同，主要是文件名不同。小技巧：可以利用tab键的补全功能，例如安装six-1.10.0-py2.py3-none-any.whl，只需要先输入pip install six，再按一下tab键，就能自动补全文件名了）：

|  |
| --- |
| pip install **six**-1.10.0-py2.py3-none-any.whl  pip install **python\_dateutil**-2.5.0-py2.py3-none-any.whl  pip install **pytz**-2015.7-py2.py3-none-any.whl  pip install **pandas**-0.16.2-cp27-none-win\_amd64.whl  pip install **patsy**-0.4.0-py2.py3-none-any.whl  pip install **scipy**-0.16.1-cp27-none-win\_amd64.whl  pip install **statsmodels**-0.6.1-cp27-none-win\_amd64.whl |

至此，就完成了所需Python包的安装。

1. 附录

Python一个比较常见的问题是编码问题，若遇到类似“UnicodeEncodeError: 'ascii' codec can't encode character u'\u6211' in position 0: ordinal not in range(128)”的问题，需在Python 的安装目录（默认为C:\Python27）下的Lib\site-packages文件夹中新建一个文件sitecustomize.py，添加如下代码并保存：

|  |
| --- |
| import sys  sys.setdefaultencoding('gb2312') |

该文件会更改Python的默认编码，以后出现类似的问题，可尝试将代码中的 gb2312 改为 utf-8 或 ascii ，然后再次尝试。

1. Linux系统

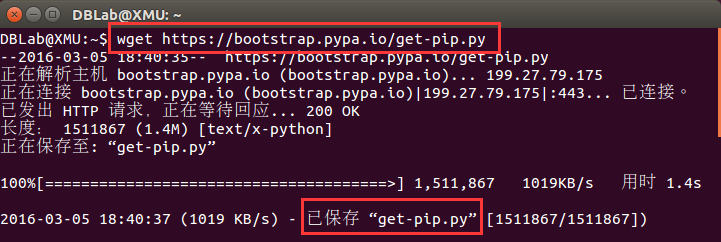
以Ubuntu 14.04为例，介绍如何在Linux系统中安装pip和Python包。

1. 安装pip

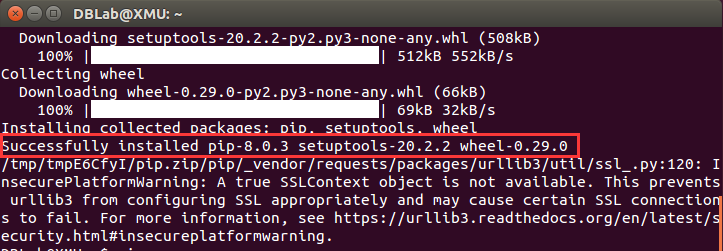
Pip(<https://pypi.python.org/pypi>)是一个安装和管理Python包的工具，一般Linux系统没有自带，需要自行安装。打开终端，执行如下命令进行安装：

|  |
| --- |
| wget https://bootstrap.pypa.io/get-pip.py  sudo python get-pip.py |

运行结果如下图所示：



截图1: 下载get-pip.py



截图2: 安装pip，成功则提示Successfully installed pip-xxx

如果无法安装成功，可尝试下载离线安装包进行安装：[pip7.1.2下载地址](https://pypi.python.org/packages/source/p/pip/pip-7.1.2.tar.gz#md5=3823d2343d9f3aaab21cf9c917710196) | [备用下载](http://pan.baidu.com/s/1dEolOeH)

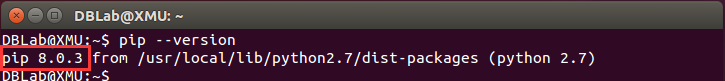
下载后解压，假设位于 ~/下载/pip.7.1.2 中，则执行如下命令进行安装：

|  |
| --- |
| cd ~/下载/pip7.1.2  sudo python setup.py |

安装完成后，执行如下命令检验是否安装成功：

|  |
| --- |
| pip --version |

若成功安装，则会显示pip的版本信息：



1. 安装所需的包

安装好pip后，可在终端中方便地使用“pip install package-name”的方式来在线安装所需的Python包了。

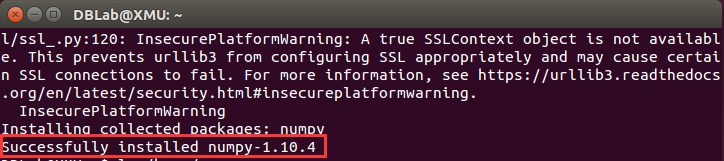
在本实验中，我们需要安装如下几个包：

* Numpy
* Pandas
* Scipy
* Patsy
* Statsmodels

以安装Numpy为例，在终端中执行如下命令：

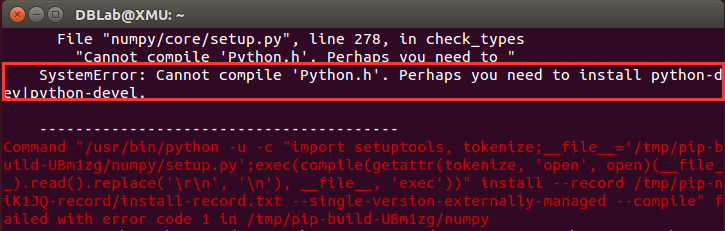
|  |
| --- |
| pip install numpy |

安装成功如下图所示，提示“Successfully installed numpy-xxx”：



安装过程中可能会遇到的问题：

1. 提示“SystemError: Cannot compile ‘Python.h’”的错误，如下图所示：

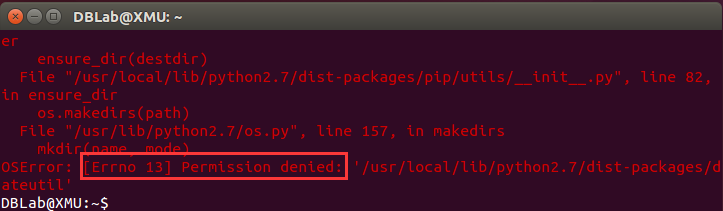


则需要执行如下命令安装Python-dev（CentOS系统对应的命令为：sudo yum install python-devel）：

|  |
| --- |
| sudo apt-get install python-dev |

安装Python-dev后再次执行pip install numpy进行安装即可。

1. 出现“Permission denied”的错误，如下图所示：



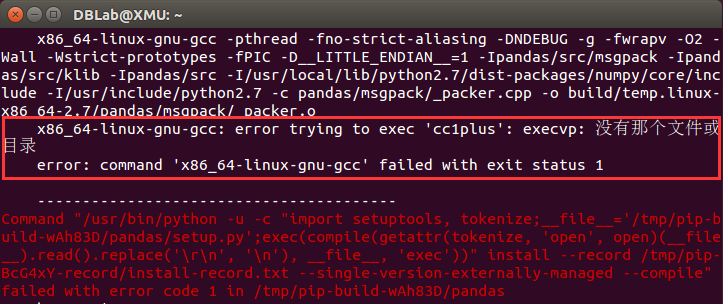
可以在命令前加上sudo，即“sudo pip install numpy”。

其他Python包的安装过程与numpy类似，在终端中依次执行如下命令（注意安装顺序，其中scipy的安装时间会比较长）：

|  |
| --- |
| pip install pandas  pip install patsy  pip install scipy  pip install statsmodels |

安装过程中可能会遇到的问题：

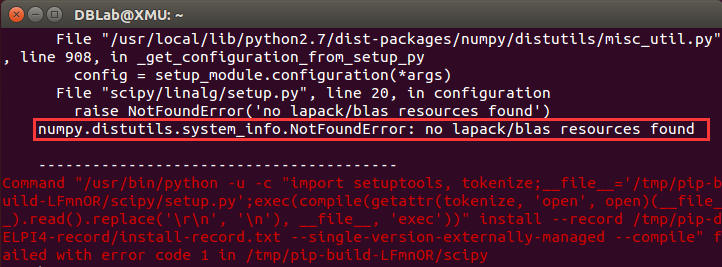
1. 提示“error: command ‘x86\_64-linux-gnu-gcc’ failed”的错误，如下图所示：



则需要执行如下命令安装g++（CentOS系统对应的命令为：sudo yum install gcc-c++）：

|  |
| --- |
| sudo apt-get install g++ |

1. 提示“no lapack/blas resources found”的错误，如下图所示：



则需要执行如下命令安装lapack/blas（CentOS系统对应的命令为：sudo yum install lapack lapack-devel blas blas-devel）：

|  |
| --- |
| sudo apt-get install gfortran libopenblas-dev liblapack-dev |

至此，就完成了所需Python包的安装。