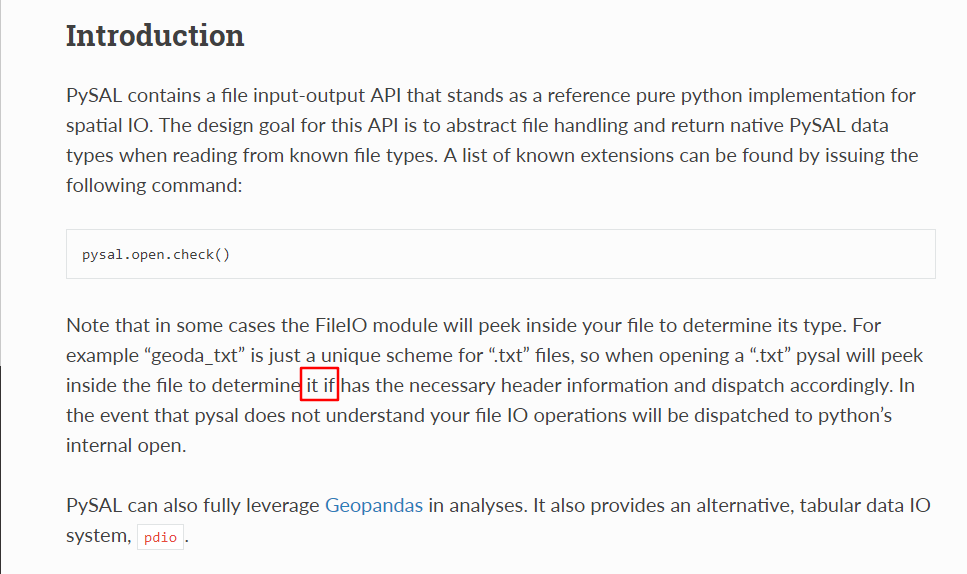
勘误1：官网教程Introduction部分。

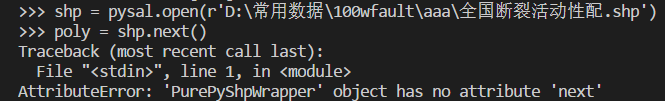


it if应该为if it。

勘误2：官网教程shp文件读写中，poly = shp.next()会报错。（下同）



用自己的数据进行测试，发现PurePyShpWapper没有next()方法，查看源代码，应为私有方法\_\_next\_\_（）

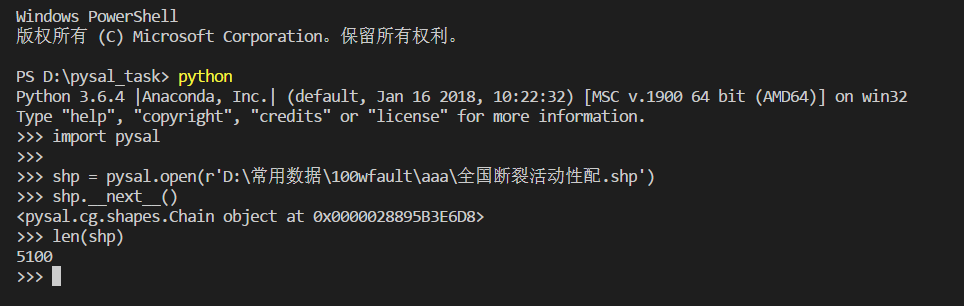


修改后可以运行：

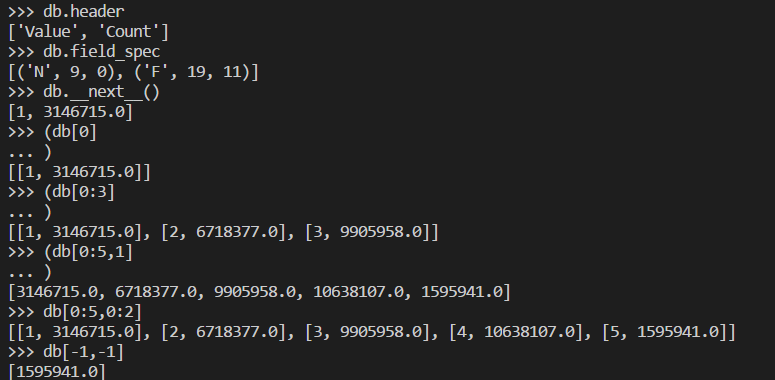


# File Input and Output

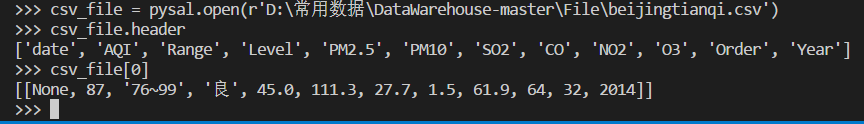
## Shp文件读取截屏：

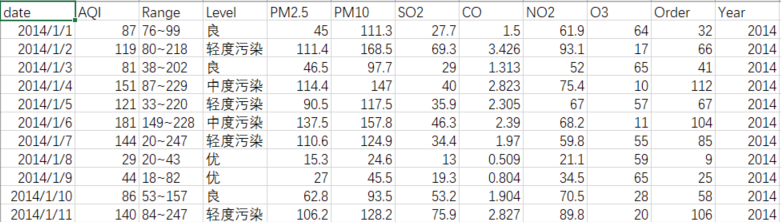


## dbf数据读取截屏：

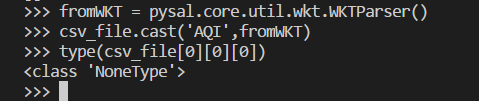


## csv文件读取截屏



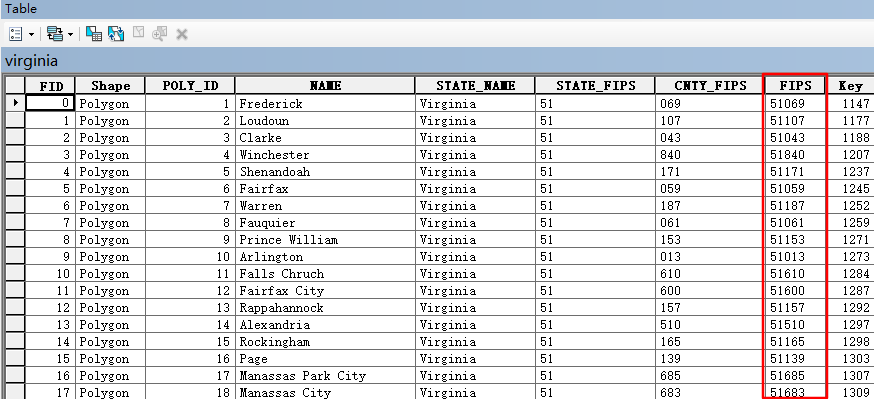
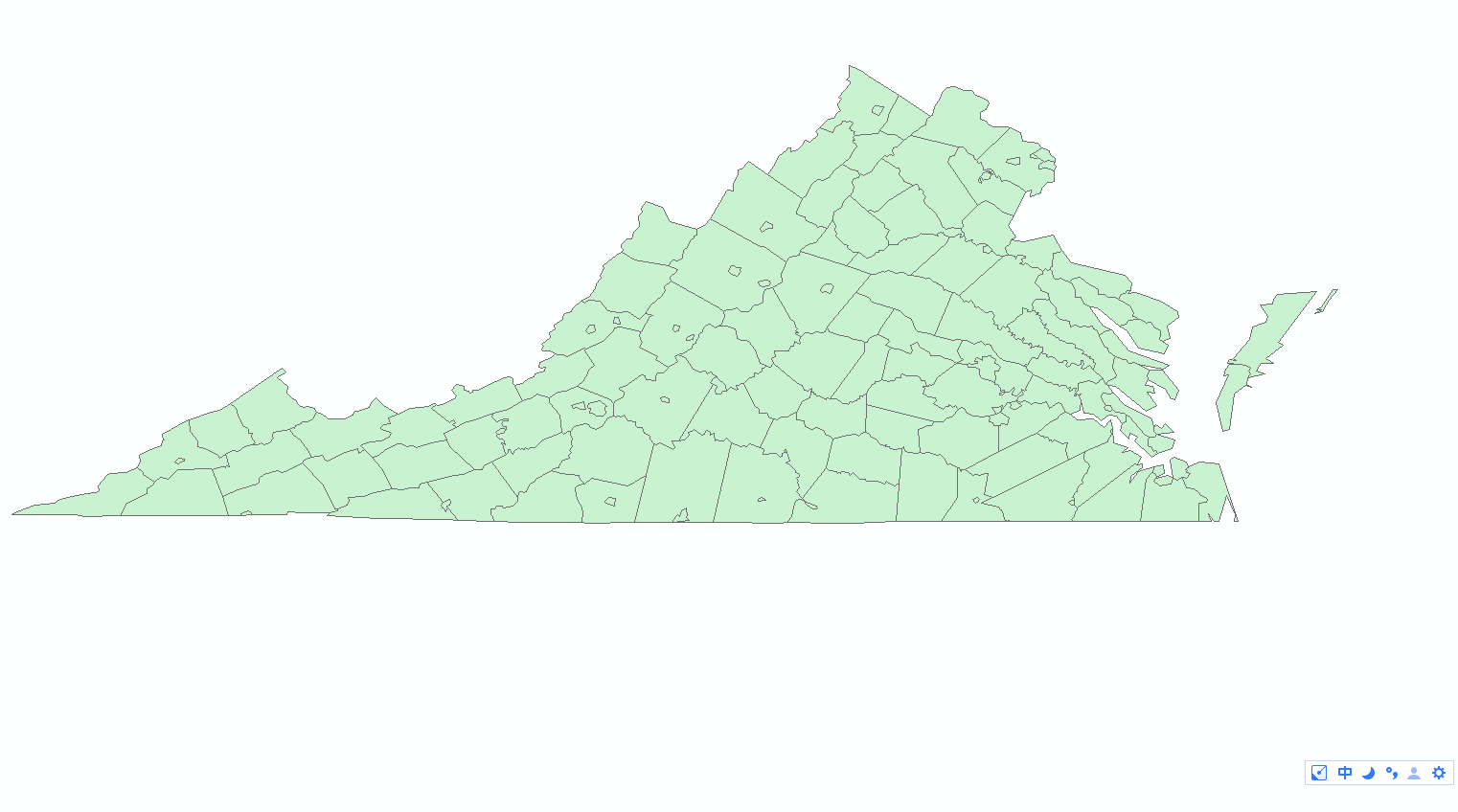


注:程序对date的类型解析出错。输出值为None。



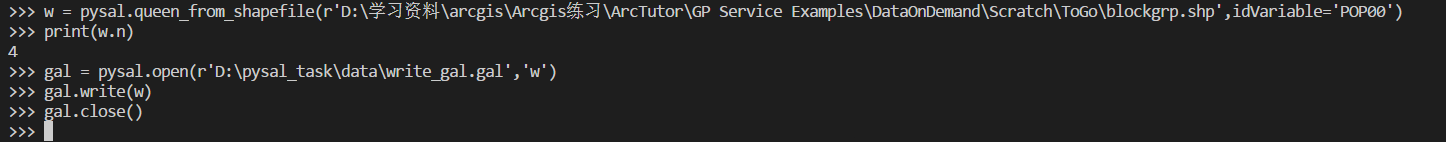
# Writing files

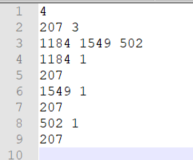
## [GAL Binary Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id32)



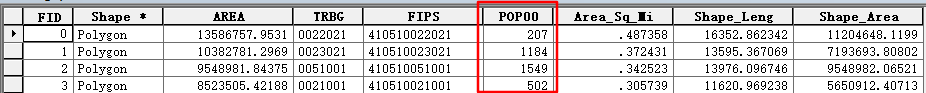
示例数据示意图（Virginia维吉尼亚[və(:)ˈdʒinjə]）

## Gal数据写入操作截图



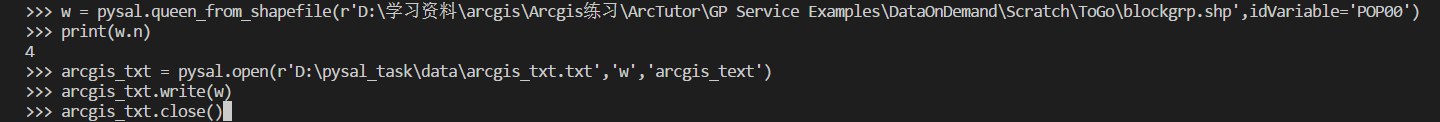


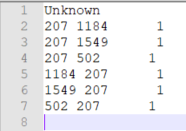
生成的gal数据。第一行的4代表数据共有4行。



使用数据

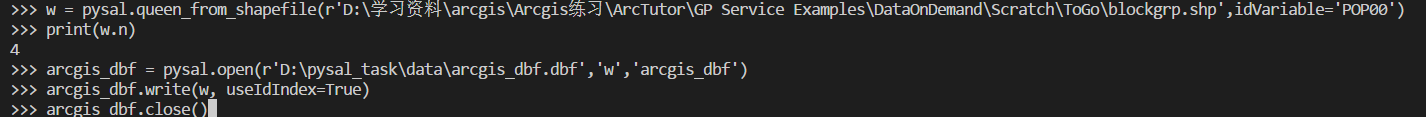
## [ArcGIS Text Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id34)



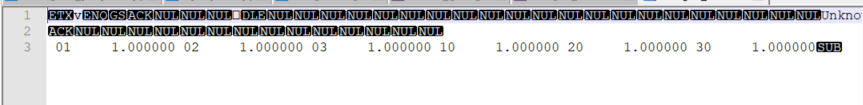


结果文件

## [ArcGIS DBF Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id35)

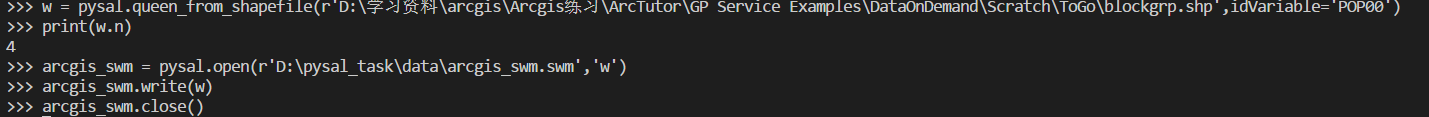


截屏文件

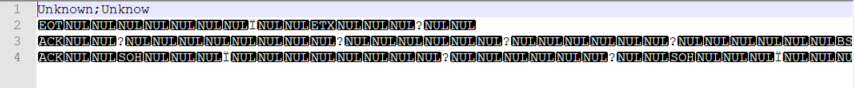


结果文件

## [**ArcGIS SWM Weights File**s](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id36)

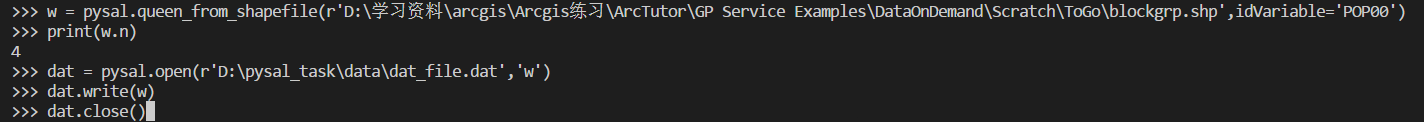


截屏文件

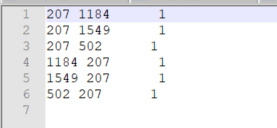


结果文件

## [DAT Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id37)

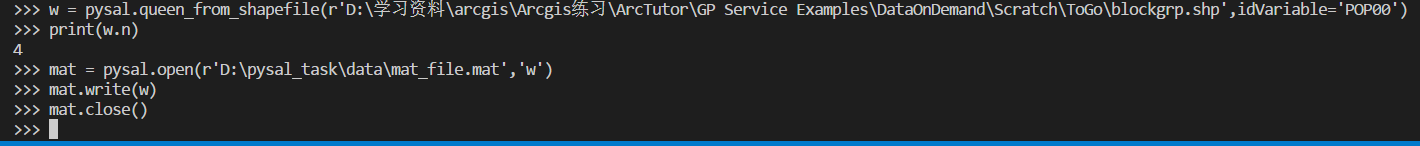


截屏文件



结果文件

## [MATLAB MAT Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id38)



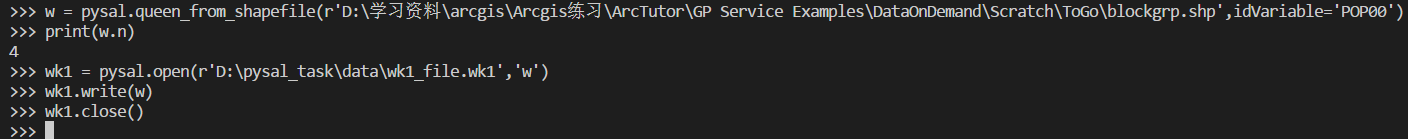
截屏文件



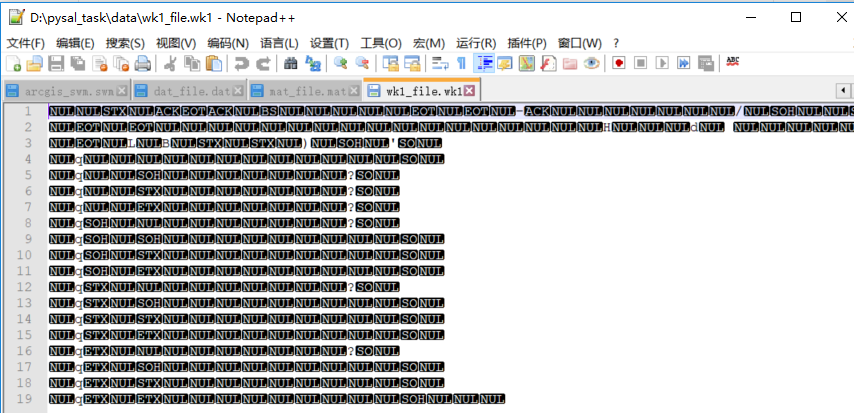
结果文件

## [LOTUS WK1 Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id39)

注意：wk1不是wkl.是数字“1”而不是字母“l”！！！！！！！！

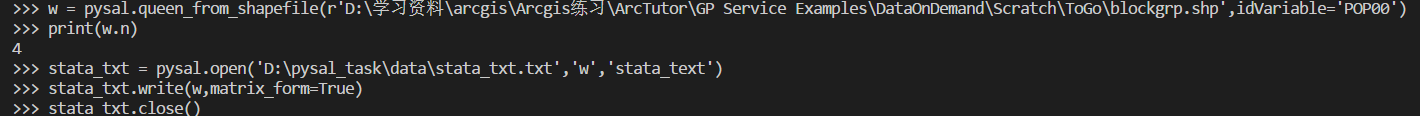


截屏文件

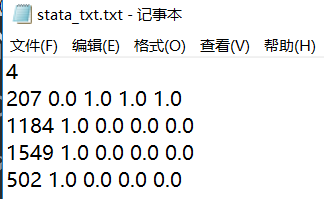


结果文件

## [GeoBUGS Text Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id40)

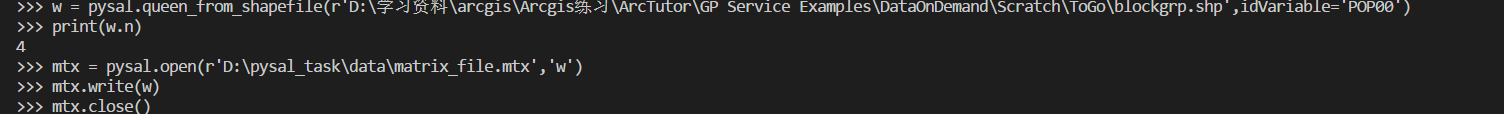


截屏文件

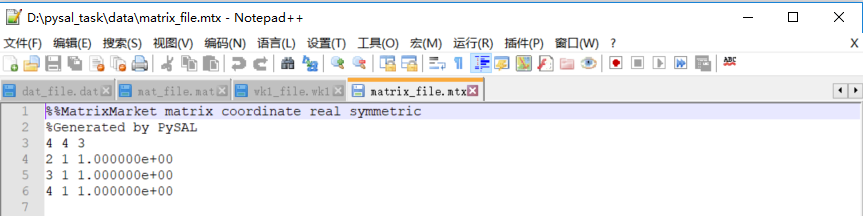


结果文件

## [MatrixMarket MTX Weights Files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id42)

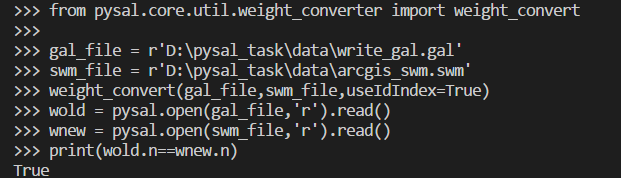


截屏文件



结果文件

## [Examples: Converting the format of spatial weights files](https://pysal.readthedocs.io/en/latest/users/tutorials/fileio.html#id43)



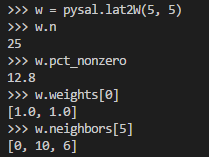
# Spatial Weights

## [Contiguity Based Weights](http://pysal.readthedocs.io/en/latest/users/tutorials/weights.html#id16)

基于接近度的权重。

Contiguity：[ˌkɒntɪ'ɡju:ətɪ] n. 接近; 一连串的事物; 一系列; 一大片;

为5x5的点阵构造一个简单的邻接矩阵。

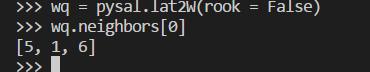




n是空间单位的数量，即权重存储在25x25的矩阵中。pct\_nozero显示矩阵的稀疏性。在上面的例子中，我们看到id 0（Python是零偏移量）的观察结果有两个ids [5,1]的邻居，每个邻居的权重都是1.0。

直方图属性是指示邻居关系的基数的一组​​元组。在这种情况下，我们有一个规则的格子，所以有4个单元有2个邻居（角单元），12个单元有3个邻居（边单元），9个单元有4个邻居（内部单元）。

在上面的例子中，晶格上的邻接的默认标准是作为邻居共享一个边的任意一对单元的rook。或者，我们可以使用皇后准则来包含格子的顶点来定义连续性：



# [Spatial Dynamics](https://pysal.readthedocs.io/en/latest/users/tutorials/dynamics.html#id26)(空间动力学)

## [Introduction](https://pysal.readthedocs.io/en/latest/users/tutorials/dynamics.html#id27)

PySAL实施了许多探索性方法来分析纵向空间数据的动态，或在多个时间段内对固定面积单位的观测。 例子包括美国总统选举中的投票模式的时间序列，遥感影像的时间序列，劳动力市场动态，区域商业周期等等。 实施两套广泛的空间动力学方法来分析这些数据类型。 第一种是基于马尔科夫的方法，第二种是基于Rank动态的。

此外，该模块中还包括方法来分析具有与其相关的空间和时间坐标的单个事件的模式。 例子包括个别病例或犯罪的地点和时间。 这里包括了确定这些事件模式是否表现出时空相互作用的方法。

## Markov Based Methods

The Markov based methods include classic Markov chains and extensions of these approaches to deal with spatially referenced data. In what follows we illustrate the functionality of these Markov methods. Readers interested in the methodological foundations of these approaches are directed to [1].

基于马尔可夫的方法包括经典马尔可夫链和这些方法的扩展以处理空间参考数据。 下面我们将说明这些马尔科夫方法的功能。 对这些方法的方法论基础感兴趣的读者可以参考[1]。

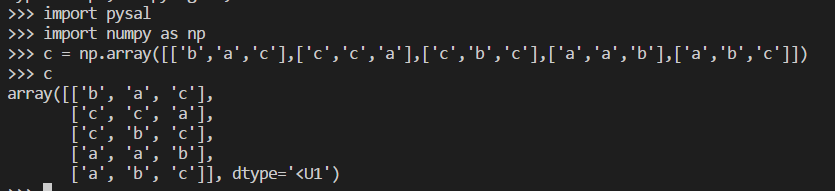
## [Classic Markov](https://pysal.readthedocs.io/en/latest/users/tutorials/dynamics.html#id29)

We start with a look at a simple example of classic Markov methods implemented in PySAL. A Markov chain may be in one of k different states at any point in time. These states are exhaustive and mutually exclusive. For example, if one had a time series of remote sensing images used to develop land use classifications, then the states could be defined as the specific land use classes and interest would center on the transitions in and out of different classes for each pixel.

我们首先看一下在PySAL中实现的经典马尔可夫方法的一个简单例子。 马尔可夫链可以在任何时间点处于k个不同状态之一。 这些状态是详尽而且相互排斥的。 例如，如果有一个用于开发土地利用分类的遥感影像的时间序列，那么可以将这些状态定义为特定的土地利用类别，并且兴趣将集中在每个像素的不同类别的转换中。

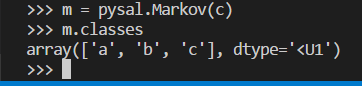
For example, let’s construct a small artificial chain consisting of 3 states (a,b,c) and 5 different pixels at three different points in time:

例如，让我们在三个不同的时间点构造一个由3个状态（a，b，c）和5个不同像素组成的小型人造链：



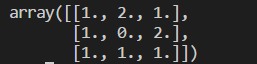
So the first pixel was in class ‘b’ in period 1, class ‘a’ in period 2, and class ‘c’ in period 3. We can summarize the overall transition dynamics for the set of pixels by treating it as a Markov chain:

因此，第一个像素在时期1中属于'b'类，时期2中属于类'a'，时期3中属于类'c'。我们可以通过将该像素集视为马尔可夫链来总结整体过渡动态：



The Markov instance m has an attribute class extracted from the chain - the assumption is that the observations are on the rows of the input and the different points in time on the columns. In addition to extracting the classes as an attribute, our Markov instance will also have a transitions matrix:

马尔可夫实例m具有从链中提取的属性类 - 假设是观察值在输入行上以及列上是不同时间点上。 除了将类作为属性提取外，我们的Markov实例还将具有转换矩阵：

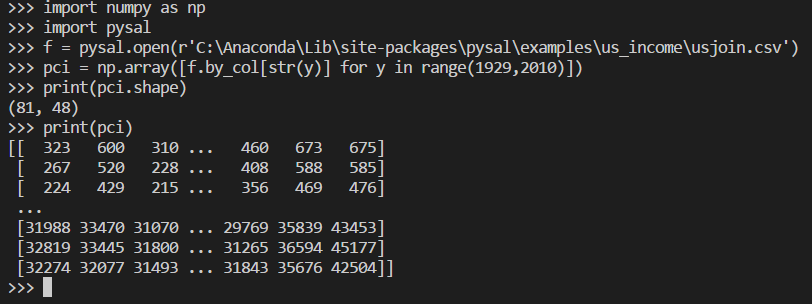


indicating that of the four pixels that began a transition interval in class ‘a’, 1 remained in that class, 2 transitioned to class ‘b’ and 1 transitioned to class ‘c’.

表明在类“a”中开始转换间隔的四个像素中，1个保留在该类中，2个转换为类“b”，并且1个转换为类“c”。

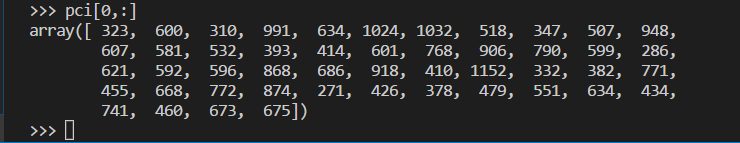
This simple example illustrates the basic creation of a Markov instance, but the small sample size makes it unrealistic for the more advanced features of this approach. For a larger example, we will look at an application of Markov methods to understanding regional income dynamics in the US. Here we will load in data on per capita income observed annually from 1929 to 2010 for the lower 48 US states:

这个简单的例子说明了马尔科夫实例的基本创建，但是小样本的大小使得这种方法的更高级特性不切实际。 举一个更大的例子，我们将看看马尔可夫方法在美国理解地区收入动态的应用。 在这里，我们将载入关于美国较低的48个州每年从1929年到2010年的人均收入数据：



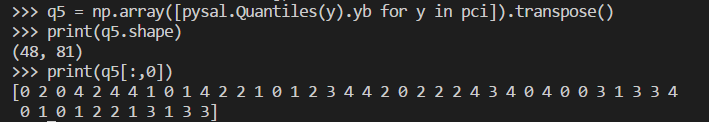
The first row of the array is the per capita income for the first year:

排列的第一行是第一年的人均收入：



In order to apply the classic Markov approach to this series, we first have to discretize the distribution by defining our classes. There are many ways to do this, but here we will use the quintiles for each annual income distribution to define the classes:

为了将经典的马尔可夫方法应用到本系列中，我们首先必须通过定义类来离散分布。 有很多方法可以做到这一点，但我们在这里将使用五分位数来计算每个年收入分配的类别：

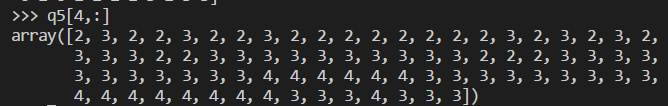


A number of things need to be noted here. First, we are relying on the classification methods in PySAL for defining our quintiles. The class Quantiles uses quintiles as the default and will create an instance of this class that has multiple attributes, the one we are extracting in the first line is yb - the class id for each observation. The second thing to note is the transpose operator which gets our resulting array q5 in the proper structure required for use of Markov. Thus we see that the first spatial unit (Alabama with an income of 323) fell in the first quintile in 1929, while the last unit (Wyoming with an income of 675) fell in the fourth quintile [2].

这里需要注意一些事情。 首先，我们依靠PySAL中的分类方法来定义我们的五分位数。 Quantiles类使用五分位数作为默认值，并将创建一个具有多个属性的类的实例，我们在第一行中提取的是yb - 每个观察值的类id。 第二件要注意的是转置运算符，它使我们的结果数组q5得到使用马尔科夫所需的适当结构。 因此，我们看到，第一个空间单位（阿拉巴马州323收入）在1929年下降到第一个五分位数，而最后一个单位（怀俄明州收入675）下降到第四个五分位数[2]。

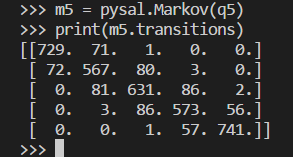
So now we have a time series for each state of its quintile membership. For example, Colorado’s quintile time series is:

所以现在我们为它的五分之一成员国的每个州制定一个时间序列。 例如，科罗拉多五分位时间序列是：



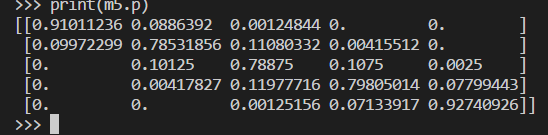
indicating that it has occupied the 3rd, 4th and 5th quintiles in the distribution at different points in time. To summarize the transition dynamics for all units, we instantiate a Markov object:

表明它已经在不同的时间点分布了三，四，五五分位数。 为了总结所有单元的过渡动态，我们实例化一个Markov对象：



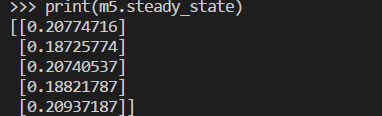
Assuming we can treat these transitions as a first order Markov chain, we can estimate the transition probabilities:

假设我们可以将这些转换看作一阶马尔可夫链，我们可以估计转移概率：



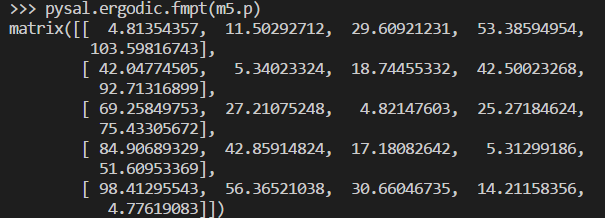
as well as the long run steady state distribution:

以及长期的稳态分布：



With the transition probability matrix in hand, we can estimate the first mean passage time:

使用转换概率矩阵，我们可以估计第一个平均通过时间：



Thus, for a state with income in the first quintile, it takes on average 11.5 years for it to first enter the second quintile, 29.6 to get to the third quintile, 53.4 years to enter the fourth, and 103.6 years to reach the richest quintile.

因此，对于一个收入在第一个五分位数的州来说，首先进入第二个五分位数平均需要11.5年，达到第三个五分位数平均需要29.6年，进入第四个分位平均需要53.4年，到达最富有的五分位数平均需要103.6年。

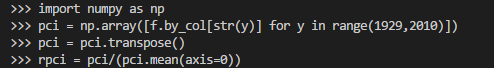
## [Spatial Markov](https://pysal.readthedocs.io/en/latest/users/tutorials/dynamics.html#id30)

Thus far we have treated all the spatial units as independent to estimate the transition probabilities. This hides a number of implicit assumptions. First, the transition dynamics are assumed to hold for all units and for all time periods. Second, interactions between the transitions of individual units are ignored. In other words regional context may be important to understand regional income dynamics, but the classic Markov approach is silent on this issue.

到目前为止，我们已经将所有的空间单位视为独立的来估计转移概率。 这隐藏了一些隐含的假设。 首先，假定所有单位和所有时间段的过渡动态都是成立的。 其次，个别单位转型之间的相互作用被忽略。 换句话说，区域背景对于了解地区收入动态可能很重要，但经典的马尔可夫方法在这个问题上没有提及。

PySAL includes a number of spatially explicit extensions to the Markov framework. The first is the spatial Markov class that we illustrate here. We first are going to transform the income series to relative incomes (by standardizing by each period by the mean):

PySAL包含许多对Markov框架的空间明确的扩展。 首先是我们在这里说明的空间马尔可夫类。 我们首先要把收入序列转化为相对收入（通过每个时期的均值标准化）：



Next, we require a spatial weights object, and here we will create one from an external GAL file:

接下来，我们需要一个空间权重对象，这里我们将从外部GAL文件创建一个：



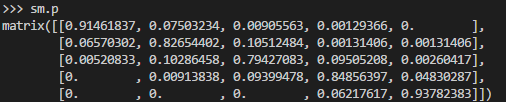
Finally, we create an instance of the Spatial Markov class using 5 states for the chain:

最后，我们使用链的5个状态创建一个空间马尔可夫类的实例：



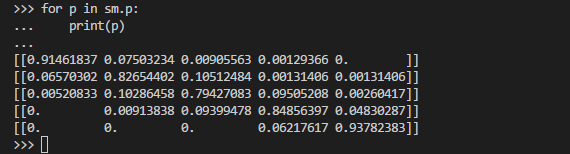
Here we are keeping the quintiles fixed, meaning the data are pooled over space and time and the quintiles calculated for the pooled data. This is why we first transformed the data to relative incomes. We can next examine the global transition probability matrix for relative incomes:

在这里，我们保持五分位数的固定，这意味着数据汇集在空间和时间上，而五分位数则用于汇总数据。 这就是为什么我们首先将数据转化为相对收入。 接下来我们可以检验相对收入的全局转移概率矩阵：



The Spatial Markov allows us to compare the global transition dynamics to those conditioned on regional context. More specifically, the transition dynamics are split across economies who have spatial lags in different quintiles at the beginning of the year. In our example we have 5 classes, so 5 different conditioned transition probability matrices are estimated:

空间马尔可夫使我们能够将全局转化动态与区域背景下的条件进行比较。 更具体地说，转型动态分布于年初有不同五分位数空间滞后的收入。 在我们的例子中，我们有5个类，所以估计5个不同的条件转移概率矩阵：

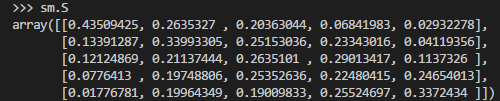


The probability of a poor state remaining poor is 0.963 if their neighbors are in the 1st quintile and 0.798 if their neighbors are in the 2nd quintile. The probability of a rich economy remaining rich is 0.977 if their neighbors are in the 5th quintile, but if their neighbors are in the 4th quintile this drops to 0.903.

如果他们的邻居处于第一分位阶层，贫困国家仍然很差的概率为0.963，如果他们的邻居处于第二分位阶层，则为0.798。 如果他们的邻居处于第五五分位数，富裕经济体富裕的概率为0.977，但如果他们的邻居处于第四分位数，则其下降到0.903。

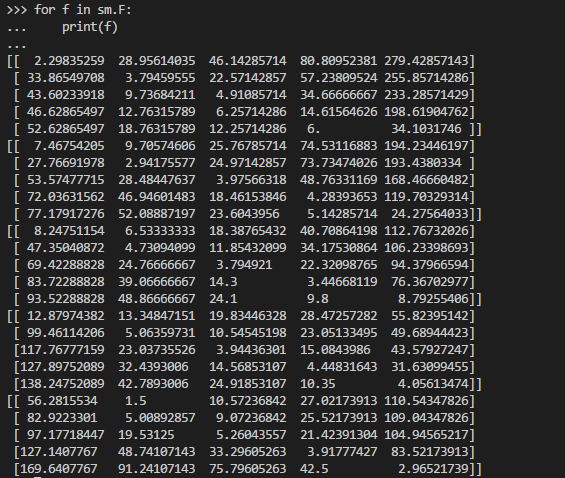
We can also explore the different steady state distributions implied by these different transition probabilities:

我们还可以探索这些不同转换概率所暗示的不同稳态分布：



The long run distribution for states with poor (rich) neighbors has 0.435 (0.018) of the values in the first quintile, 0.263 (0.200) in the second quintile, 0.204 (0.190) in the third, 0.0684 (0.255) in the fourth and 0.029 (0.337) in the fifth quintile. And, finally the first mean passage times:

贫穷（富）邻州的长期分布在第一分位数中为0.435（0.018），第二分位中为0.263（0.200），第三分位中为0.204（0.190），第四位中为0.0684（0.255）， 第五分位0.029（0.337）。 最后，第一个平均通过时间：



States with incomes in the first quintile with neighbors in the first quintile return to the first quintile after 2.298 years, after leaving the first quintile. They enter the fourth quintile 80.810 years after leaving the first quintile, on average. Poor states within neighbors in the fourth quintile return to the first quintile, on average, after 12.88 years, and would enter the fourth quintile after 28.473 years.

在离开第一个五分位数后，收入位于第一个五分位数的第一个五分位数的国家在2.298年后回到第一位五分位数。 他们在平均离开第一个五分位数后80.810年进入第四个分位数。 第四分位数的邻国中的贫穷国家在12.88年后平均返回第一分位数，并在28.473年后进入第四分位数。

again reflecting the dominance of quadrants 1 and 3 (positive autocorrelation). [[3]](https://pysal.readthedocs.io/en/latest/users/tutorials/dynamics.html#id16) Finally the first mean passage time for the LISAs is:

再次反映了象限1和3（主动自相关）的优势。 [3]最后，LISAs的平均通过时间是