

数据加载和存储

读写文本格式的数据

查看一个以逗号分隔的(CSV)文本文件:

```
In [11]: !cat input1.csv

Feature1, Feature2, Label
8,-11,1
7,7,-1
12,-20,1
```

我们可以使用`read_csv`将其读入一个`DataFrame`:

```
In[12]: df = pd.read_csv('input.csv')

In[13]: df
Out[13]:
   Feature1  Feature2  Label
0         8        -11      1
1         7         7      -1
```

一般地, 你可以用`read_table`来读取数据, 不过要记得指定分隔符:

```
In[17]: pd.read_table('input.csv', sep=',')
Out[17]:
   Feature1  Feature2  Label
0         8        -11      1
1         7         7      -1
```

有些文件是没有标题行的. 比如下面这个文件:

```
In[18]: !cat input1.csv

8,-11,1
7,7,-1
```

这时候, 我们就可以自己定义列名. 有两种方式: (1)让Pandas分配默认列名; (2)自己定义列名.

```
In[21]: pd.read_csv('input1.csv', header=None)
Out[21]:
   0  1  2
0  8 -11 1
1  7  7 -1

In[22]: pd.read_csv('input1.csv', names=['X1', 'X2', 'y'])
Out[22]:
   X1 X2 y
0  8 -11 1
1  7  7 -1
```

如果你想将y列作为DataFrame的索引，可以通过参数index_col指定 "y" . 当然也可以指定为其他的列.

```
In[24]: pd.read_csv('input1.csv', names=['X1','X2','y'], index_col='y')
Out[24]:
   X1 X2
y
1  8 -11
-1  7  7

pd.read_csv('input1.csv', names=['X1','X2','y'], index_col='X1')
Out[25]:
   X2 y
X1
8 -11 1
7  7 -1
```

你也可以将多列做成一个层次化索引. 具体做法是: 传入由列编号或列名组成的列表.

```
In[30]: pd.read_csv('input1.csv', names=['X1','X2','y'], index_col=['y','X1'])
Out[30]:
   X2
y X1
1 8 -11
-1 7  7
```