一共有两个文件：（1）Gowalla\_user\_edges文件；（2）Gowalla\_totalCheckins文件

由于这两个文件都没有列名，我先给他们写一个列名：

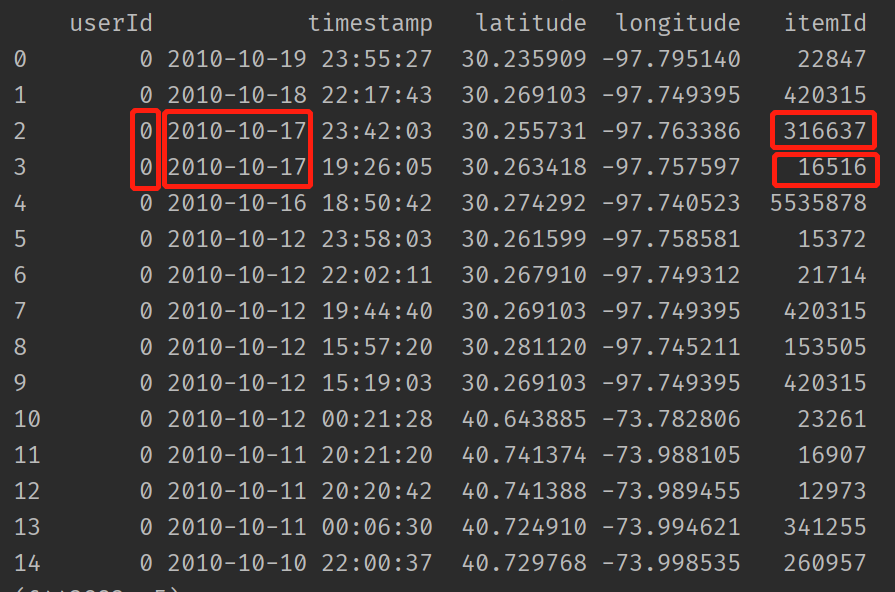
（1）Gowalla\_user\_edges文件的列名：names = [‘user1’, ‘user2’]

（2） Gowalla\_totalCheckins文件的列名：names = ['userId', 'timestamp', 'latitude', 'longitude', 'itemId']，其中，'latitude'和'longitude'在后面用不到，可以删除掉。

基于上述两个文件，主要进行以下三项工作：

**① 生成Gowalla\_item\_edges文件**，**格式和Gowalla\_user\_edges文件一致，列名为：names = [‘item1’, ‘item2’]。**

具体方法为：遍历itemId列（下图的最后一列）中的每个元素，将与这个元素具有相同用户编号（下图的userId列）和相同日期（下图的timestamp列）的元素，视作这个元素的相似元素。例如：316637这个项目和16516这个项目就是相似项目。这样，在Gowalla\_item\_edges文件中，就存在一行 316637 16516。



**② 生成user\_list文件和item\_list文件，以及train.txt和test.txt文件。**

具体过程为：仅保留Gowalla\_totalCheckins文件中的'userId'列和'itemId'列，将userId列和itemId中少于20的元素所在的行删除掉，生成rating文件。对rating文件中userId和itemId进行重新映射，生成user\_list文件（第一列表示原始的userId，第二类表示新的userId）和item\_list文件（第一列表示原始的itemId，第二类表示新的itemId）。基于新的userId和itemId，将rating生成追加格式（即：每一行的第一列表示新的userId，后面跟着多个新的itemId）。进而，从每一行的第二个元素开始，前百分之80用于生成train.txt，后百分之20用于生成test.txt。

**③生成user.txt文件和item.txt文件。**

具体流程为：过滤掉Gowalla\_user\_edges文件中没有出现在user\_list文件中原始userId的元素所在列，并将剩余的元素依据user\_list用新的userId表示，并生成追加的格式存为user.txt文件。对Gowalla\_item\_edges采取同样的操作，生成item.txt文件。