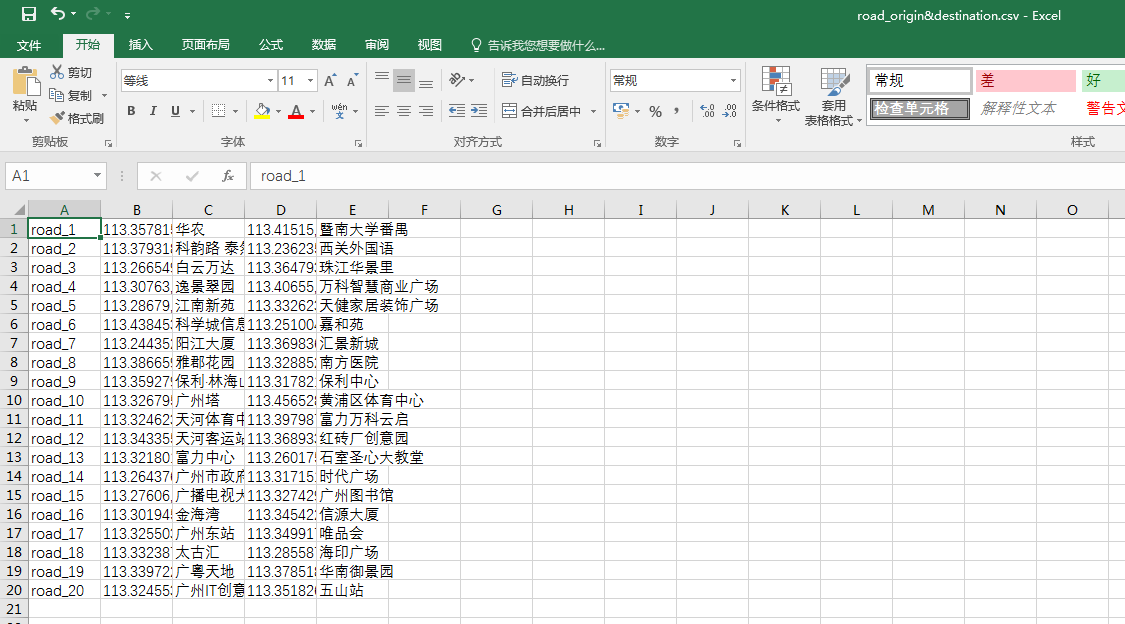
基于python的gps基础数据生成程序文档

程序设计思路大致流程：

1. 建立一份包含起点终点的csv文件
2. 调用高德地图API获取描述该路程的直线点以及该路程限速
3. 利用算法根据获取的gps点等距离生成gps坐标集来描述一段路程
4. 利用生成的路程，模拟汽车驾驶行为，形成gps轨迹
5. 利用算法计算每段路程的时间、速度、路程等基本要素
6. 上传到hdfs服务器

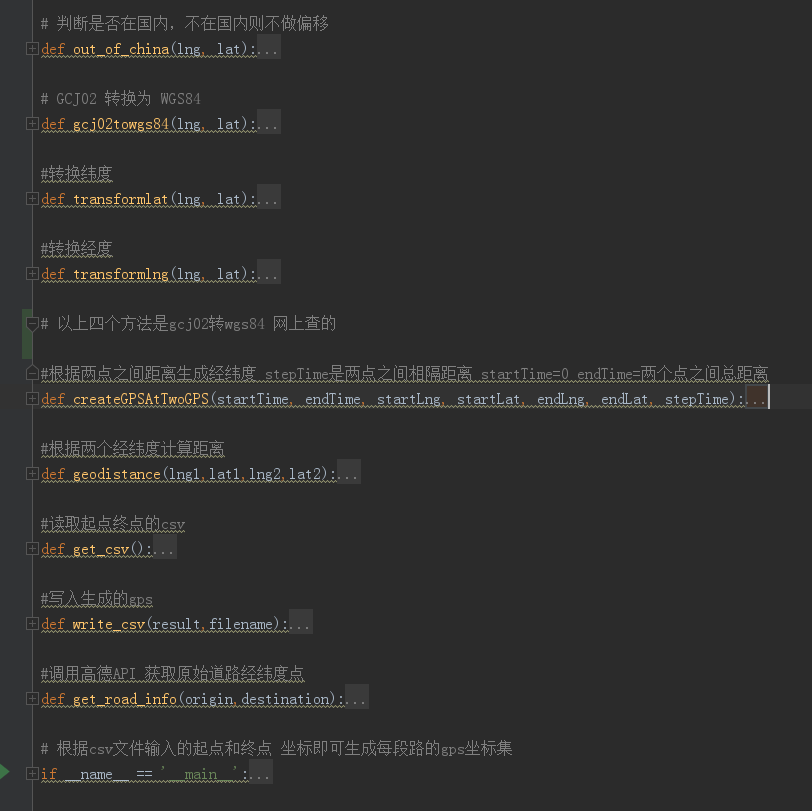
目前已实现功能：

1. GetRoadGPSGaoDeAPI.py该程序实现了根据包含起点终点的csv文件（目前是广州市内）road\_origin&destination.csv去调用高德API获取路段gps并等距离生成密集点，目前是0.2m一个gps点，csv中有多少段路程，程序跑完即可生成多少段路程的gps坐标集并由高德的gcj02坐标转换成wgs84的国际坐标。起点和终点的gps可以去高德地图的坐标拾取系统获取，road\_origin&destination.csv文件如下
2. GaoDeGetLimitspeedAPI.py 该程序实现了根据生成的road.csv文件去调用高德API获取路段的限速数据，目前是根据gps间隔达到30m调用一次，生成road\_limitspeed.csv后，需要人工手动纠正一些道路限速数据，由于gps没有海拔以及生成gps的小量误差会导致产生脏数据。目前运行该程序时，获取每段路的限速都要修改一下road.csv文件名。
3. CreateGPS.py该程序是利用上面生成的路程gps坐标集和限速值来随机生成模拟汽车行驶轨迹。原理就是控制汽车每一秒的速度最后生成一个行驶轨迹，并且上传到hdfs服务器
4. CreatTrafficBlackPoints.py该程序是根据获取下来的road.csv去随机生成一些点作为交通黑点（此数据暂时无法获取）,每段路取1-2个
5. Calculate.py该程序是验证计算平均速度 最大速度 交通黑点数 急刹车次数 急加速次数 开始时间 结束时间。
6. Upload.py：该程序把CreateGPS里面的上传功能独立开来，现在的上传方式是 读取upload文件夹下所有csv文件，上传的时候，在服务器上创建一个文件夹把这些文件都放上去。
7. GetLimitSpeedSection.py：20份限速文件已经是按照road.csv的点数生成的，该程序是把20份限速文件按不同的限速段，缩减限速点，每段取40m（200个点），生成在一份limitspeedAll的csv文件里面

程序描述：

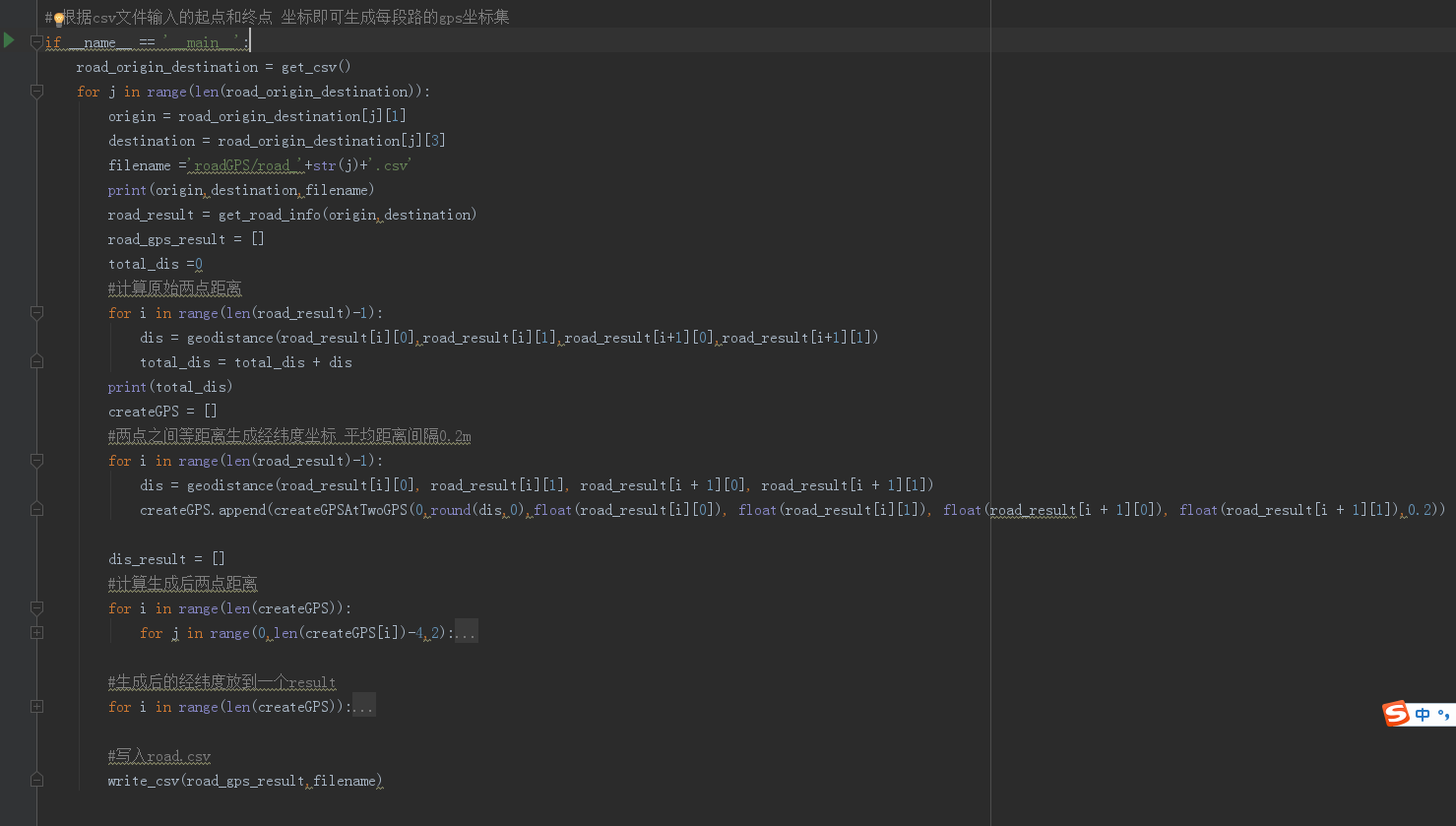
该程序主要就是三个GetRoadGPSGaoDeAPI、GaoDeGetLimitspeedAPI、CreateGPS三个文件，根据注释看程序也挺好理解的

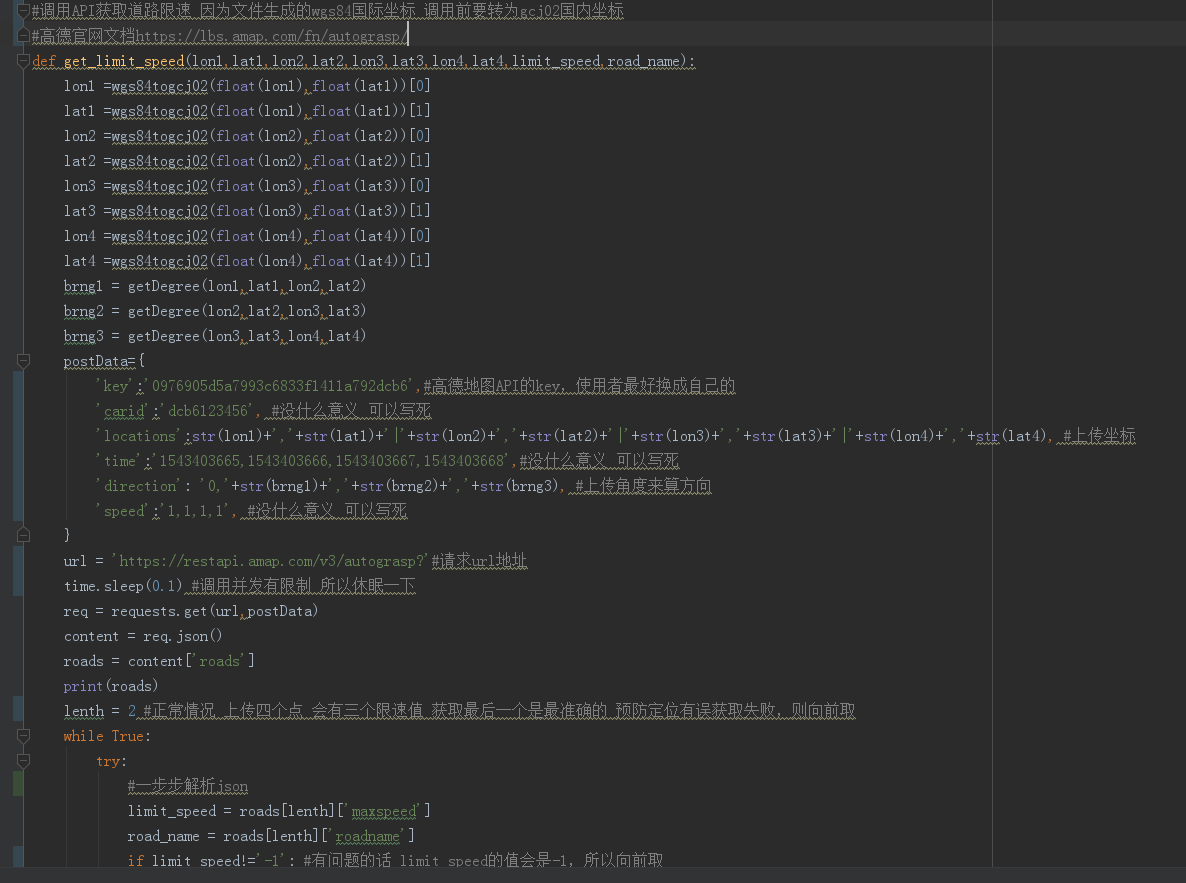
GetRoadGPSGaoDeAPI：获取路段gps坐标集



该程序写了注释，挺好理解的，说明一下的是调用高德API



下面是主程序也挺容易理解，计算的部分我只是想验证生成的gps坐标集之间的距离，经计算，平均距离0.2m，区间0.18m-0.22m吧

GaoDeGetLimitspeedAPI：获取路段限速，根据程序注释即可理解

调用高德抓路服务API间接获取限速，因为没有一个开放限速的API。只能间接获取，并且目前是每一段路依次手动获取。获取下来的数据也会有一些脏数据，原因可能是定位不够准确、高架与地下路重合。遍历路段的gps集合，当总距离超过30米时，调用一次API。注意每日调用量有限额。

CreateGPS：该程序是根据限速值随机不同的gps轨迹，汽车驾驶过程分为启动，匀速，加速，减速，急加速，急减速，停车，用随机概率生成。是利用每秒的速度秒速一整段路程，所以都是速度与距离的比较，距离是遍历原始gps坐标集（每点平均间隔0.2m），每个坐标点相加，总距离大于速度，才记录这个点。

生成的数据格式：posid，tripid，longitude，latitude，altitude（海拔无法获取，用速度值暂代），timestamp

还有一份trip-car的对应表，已经实现同一辆车，假设4段trip，分开两份文件，trip-car对应也会分开成1,2和3,4。