根据摩拜单车出行轨迹画直方图

1752221 潘帅鑫

一,利用 python 处理数据

- 1. 处理思路:
 - (1). 读入 txt 文件, 建立数据 list;
 - (2). 利用循环, 计算出每个摩拜单车的出行时间, 单位为 s;
- (3). 自定义函数,根据 haversine 公式,定义计算两个经纬度之间的实际 距离的函数;
 - (4).调用上述函数,计算出出行距离和速度
 - (5). 将出行时间, 距离, 速度合成 list, 转置一下
 - (6). 写入 csv 文件
 - 2. 具体代码(带有注释,便于读懂)

```
#!/usr/bin/env python
import csv
```

```
file = 'mobike_reordered.txt'
bike_list = []
```

```
with open(file, newline='') as f:
```

```
reader = csv.reader(f, delimiter=',', quotechar='"')
next(reader)
```

for row in reader:

```
order_id = row[2]
bike_id = row[3]
user_id = row[4]
start_time = float(row[0])
```

```
start time = int(start time)
       end time = float(row[1])
       end_time = int(end_time)
       start_lon = float(row[5])
       end_lon = float(row[7])
       start_lat = float(row[6])
       end lat = float(row[8])
       track = row[9]
       trip info = [order id, bike id, user id, start time, end time,
(start_lon, start_lat), (end_lon, end_lat), track]
       bike_list.append(trip_info) #读入文件
use time = []
for i in range (len (bike list)):
   use_time.append(bike_list[i][4] - bike_list[i][3])#计算出行时间,
单位为s
from math import radians, cos, sin, asin, sqrt
def haversine (lon1, lat1, lon2, lat2):#定义计算两经纬度之间的距离的函数
   #将十进制数转化为弧度
   lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
   #haversine 公式
   dlon = lon2 - lon1
   dlat = 1at2 - 1at1
   a = \sin(d1at/2)**2 + \cos(1at1) * \cos(1at2) * \sin(d1on/2)**2
   c = 2 * asin(sqrt(a))
   r = 6371 #地球平均半径,单位为 km
   return c * r * 1000
```

```
path = []
for i in range(len(bike_list)):
   track list = bike list[i][7].split('#')#将 str 转为 list
   path t = 0
   for j in range(len(track list)):
           track_list1 = track_list[j].split(',')
           track list1 = list(map(float, track list1))#将字符型转化为
数值型
           if j < len(track list)-1:#防止出现列表调用超出
               track_list2 = track_list[j+1].split(',')
              track_list2 = list(map(float, track_list2))
              path f = haversine(track list1[0], track list1[1],
track list2[0], track list2[1]) #调用函数计算相邻两经纬度之间的距离
              path t += path f
                              #计算出出发到目的地的距离,单位为 m
   path.append(path_t)
speed = []
for i in range(len(bike_list)):
   speed_f = (path[i] / use_time[i]) * 3.6
   speed.append(speed_f) #计算速度,单位为km/h
mobike_list = [use_time, path, speed]
mobike list = zip(*mobike list) #转置为列
with open("mobike_bike_list.csv", 'w') as f:
   writer = csv.writer(f, lineterminator='\n')
   writer.writerows(mobike_list) #写入 csv 文件
```

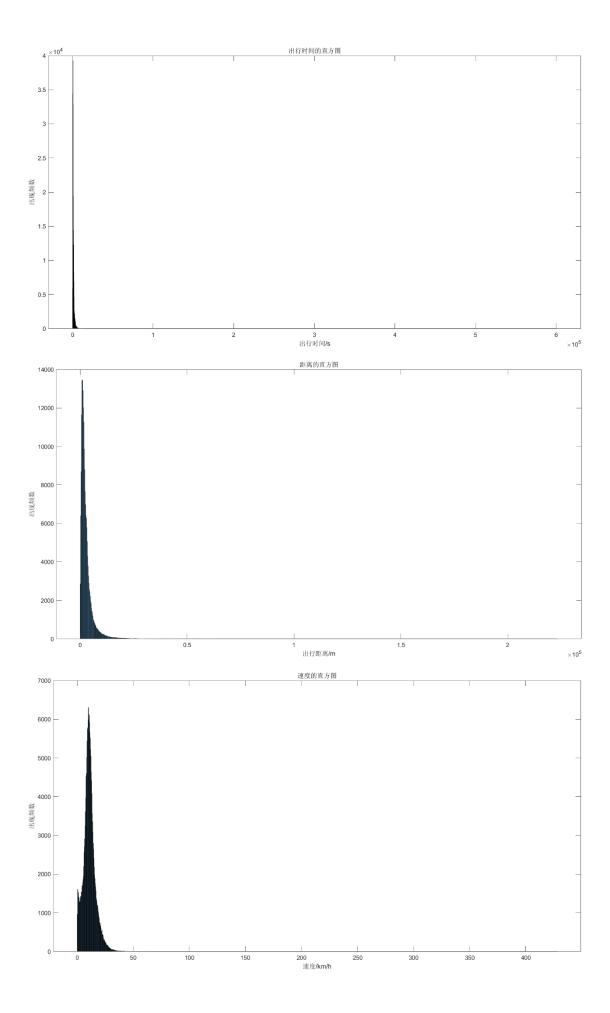
二,利用 matlab 画图

1. 处理思路:

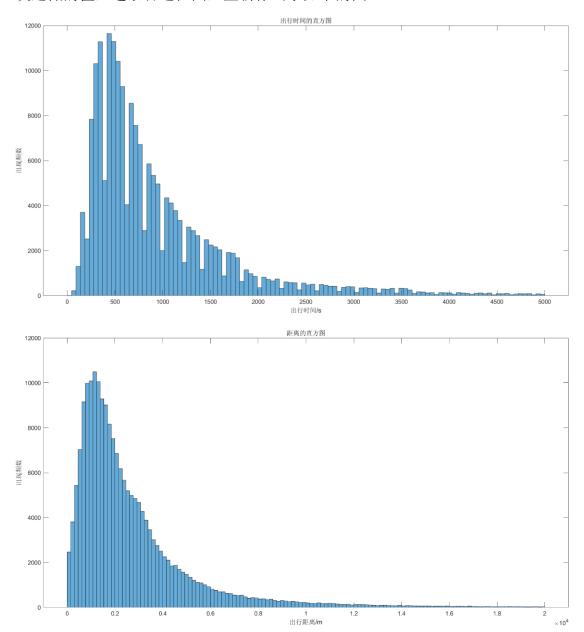
将上文输出的 csv 文件读入 matlab 中,利用 histogram()函数,画出出行时间,距离和速度的直方图,

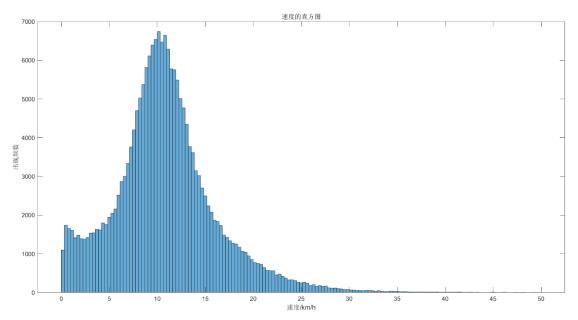
csv 文件如下图 (部分):

704 1653.537 8.455589 1705 3037.976 6.414495 439 410.5724 3.366881 750 2932.457 14.07579 1546 3561.533 8.293351 1196 3132.892 9.430111 1073 2248.746 7.544722 257 968.3935 13.56505 1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	Α	В	С	D	Е
439 410.5724 3.366881 750 2932.457 14.07579 1546 3561.533 8.293351 1196 3132.892 9.430111 1073 2248.746 7.544722 257 968.3935 13.56505 1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532	704	1653.537	8.455589		
750 2932.457 14.07579 1546 3561.533 8.293351 1196 3132.892 9.430111 1073 2248.746 7.544722 257 968.3935 13.56505 1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	1705	3037.976	6.414495		
1546 3561.533 8.293351 1196 3132.892 9.430111 1073 2248.746 7.544722 257 968.3935 13.56505 1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	439	410.5724	3.366881		
1196 3132.892 9.430111 1073 2248.746 7.544722 257 968.3935 13.56505 1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	750	2932.457	14.07579		
1073 2248.746 7.544722 257 968.3935 13.56505 1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	1546	3561.533	8.293351		
257 968.3935 13.56505 1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	1196	3132.892	9.430111		
1569 3874.402 8.889641 732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	1073	2248.746	7.544722		
732 4987.143 24.52693 893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	257	968.3935	13.56505		
893 4080.024 16.44802 426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	1569	3874.402	8.889641		
426 2707.755 22.88244 935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	732	4987.143	24.52693		
935 2970.443 11.437 632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	893	4080.024	16.44802		
632 333.5848 1.900166 3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	426	2707.755	22.88244		
3467 13353.76 13.86603 4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	935	2970.443	11.437		
4452 3715.491 3.00444 2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	632	333.5848	1.900166		
2924 2063.522 2.540589 551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	3467	13353.76	13.86603		
551 1408.523 9.202691 730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	4452	3715.491	3.00444		
730 2511.115 12.38358 467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	2924	2063.522	2.540589		
467 1788.122 13.78424 250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	551	1408.523	9.202691		
250 730.1313 10.51389 356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	730	2511.115	12.38358		
356 2425.815 24.53072 401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	467	1788.122	13.78424		
401 1655.936 14.86626 1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	250	730.1313	10.51389		
1542 3410.624 7.962546 540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	356	2425.815	24.53072		
540 380.0096 2.533398 314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	401	1655.936	14.86626		
314 1253.532 14.3717 301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	1542	3410.624	7.962546		
301 669.813 8.011052 1109 3992.835 12.96141 374 1146.215 11.03309	540	380.0096	2.533398		
1109 3992.835 12.96141 374 1146.215 11.03309	314	1253.532	14.3717		
374 1146.215 11.03309	301	669.813	8.011052		
EOC 100F 041 11 0000F	1109	3992.835	12.96141		
to the state of th	374	1146.215	11.03309		
	- FO	mobike b	ike list	(+)	



从上面的三个图可以看出,后面大部分的出现的频数很低,因此我们剔除频次过低的值,选取合适范围,重新作出了如下的图:





```
2. 具体代码如下:
clear;clc;
A = readmatrix('mobike_bike_list.csv');
figure();
histogram(A(:,1));
xlabel('出行时间/s')
ylabel('出现频数')
title('出行时间的直方图')
figure();
histogram(A(:,2));
xlabel('出行距离/m')
ylabel('出现频数')
title('距离的直方图')
figure();
histogram(A(:,3));
xlabel('速度/km/h')
ylabel('出现频数')
title('速度的直方图')
figure();
```

```
histogram(A(:,1),'BinLimits',[0,5000]);
xlabel('出行时间/s')
ylabel('出现频数')
title('出行时间的直方图')
figure();
histogram(A(:,2),'BinLimits',[0,20000]);
xlabel('出行距离/m')
ylabel('出现频数')
title('距离的直方图')
figure();
histogram(A(:,3),'BinLimits',[0,50]);
xlabel('速度/km/h')
ylabel('出现频数')
title('速度的直方图')
```