从城市交通看青岛

顾元成 41833001

钟鸿蔚 41733019

张传洋 41833030

刘伟键 41805063

数据库原理与应用

杨诚/张笃仲

2019.12

西南财经大学

（一）整体概要

1.1 研究背景

随着城市化建设的不断推进，交通拥堵情况愈发严峻。汽车数量的增加是导致城市交通堵塞的主要原因。每逢高峰时间，各类车辆从四面八方涌入市中心，造成不可避免地拥堵。

因此，如何更高效地到达目的地，途中尽可能地节约时间成为了城市居民心中的一个困扰，而这种困扰推动了时下许多智能交通信息管理平台逐渐兴起。而诸如此类的平台，大多通过对路口车流量等统计数据进行分析，结合大数据调度结果来指导车流行驶，避免交通路口拥堵，提高交通路口的智能调度和规划能力。

1.2 研究意义

本研究主要围绕车辆数据展开，通过对相关数据的分析，探究其中包含的关于“车流量”和“道路状况”两方面信息，旨在分析城市交通的具体动态变化和交通数据体现出来的城市人文特点。通过查阅相关资料，提出合理的假设，建立车流量与诸多因素的联系（如：天气，节日等等）。并且在对道路路口的车流信息统计特征分析基础上，根据对车流量的测量和特征提取结果，寻找前往一个目的地需要的最短路径数。

1.3 研究思路

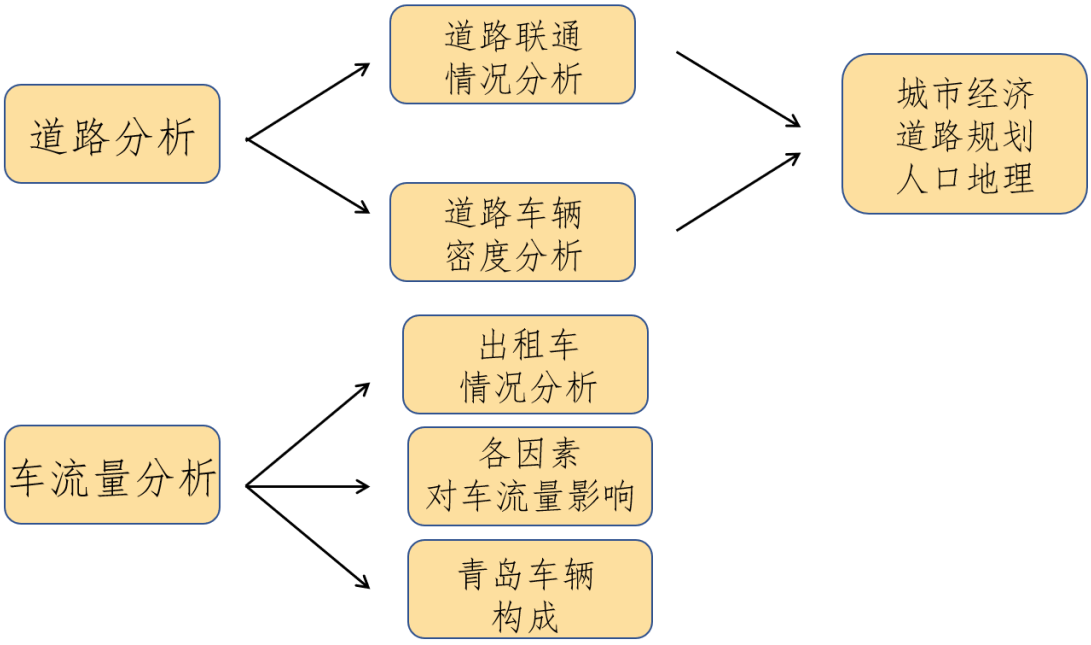

图 1 研究思路

本研究分成两部分——道路分析和车流量分析。道路分析包括道路连通分析与道路车辆密度分析，并且通过这两部分的研究提出关于城市经济，道路规划以及人口地理相关的分析与结论。车流量分析由三部分组成：出租车的情况分析，影响车流量的因素以及青岛的车辆构成（车牌分析），。并对每一部分提出了分析结果和相应结论。

（二）数据概况

2.1 数据来源

数据内容	数据来源
2019 年 8 月 1 日至 19 日青岛市交通卡口的过车数据	DataCastle 竞赛数据
2019 年 8 月 1 日至 19 日青岛市出租车 GPS 记录	DataCastle 竞赛数据
2019 年 8 月 1 日至 19 日天气数据	全国天气网

交通卡口的过车数据

a)交通卡口的编号及名字信息

字段内容描述：

字段名	字段描述
crossroadID	交通卡口的编号
crossroadName	对应名字

字段内容举例：

crossroadID	crossroadName
100205	人民路与重庆南路交叉口
100353	人民路与温州路交叉口

b)路网信息(交通卡口之间的连接信息)

字段内容描述：

字段名	字段描述
crossroadID1	交通卡口的编号
crossroadID2	交通卡口的编号

字段内容举例：

crossroadID1	crossroadID2
100205	100353
100031	100030

（注：一条记录中的两个交通卡口是相邻的，由一段道路连接，所以也可以看作是对一段道路的描述）

c)交通卡口过车数据

字段内容描述：

字段名	字段描述
direction	过车方向，取值 1-8，依次对应从北向开始顺时针旋转的八个方向
laneID	车道编号，从隔离护栏开始向右按序编号，取值正整数
timestamp	汽车通过卡口的时间
crossroadID	交通卡口的编号
vehicleID	汽车的编号（脱敏车牌号）

字段内容举例：

direction	laneID	timestamp	crossroadID	vehicleID
5	2	2019-08-01 13:28:36	100120	鲁 U-3d1d4c9c9bc6a990

出租车 GPS 记录

字段内容描述：

字段名	字段类型	字段描述
taxiID	ID	出租车的 ID
lng	浮点	某时刻某出租车 GPS 的经度
lat	浮点	某时刻某出租车 GPS 的纬度
GPSspeed	浮点	某时刻某出租车 GPS 的速度
timestamp	时间	该 GPS 记录的时间

字段内容举例：

taxiID	lng	lat	GPS speed	timestamp
鲁 Uf26f3202675eaa169874b5a56376f966	120.404	36.1897	0.0	2019-08-01 07:35:36

2.2 数据处理

1.根据时间处理不合理，删除了 8 月 2 日的数据（时间与日期不匹配）

2.进行部分计算时，排除了干扰项（如，计算出租车的平均速度时将）

3.根据天气情况，选取 8 日至 14 日为研究对象，期间天气较多变，易体现天气如何影响车流量。

（三）数据分析

3.1 道路分析

这一部分，我们对青岛市道路分别进行了联通结构和使用效率的分析。

（1）道路联通情况分析——网络展示

首先我们使用 Gephi 对节点数据进行了网络化，得到了一幅有向网如下。

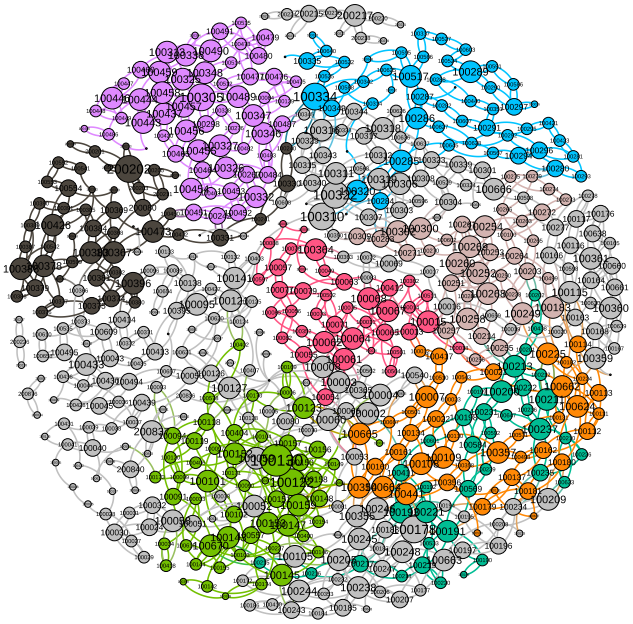

图 2 道路连通有向网

此网中，根据节点不同的出度入度比和连接紧密度等关系，进行了简单聚类分成了不同颜色，总体来看，颜色分类可以代表大致的行政区域关系。每个节点代表一个路口，每条连线代表路口间的路段。节点大小代表了该路口的权值大小，权值越大表明越多路通向此路口。

对比数据可知，颜色分类和行政区域的大致关系如下：

表 1 颜色分类与对应行政区域

颜色	对应区域
粉色	李沧区
黑色	崂山区, 市南区
灰色（左下）	市南区
红色	市南区
深浅绿色	市北区
橘色	市南市北
灰色（右上）	崂山区
蓝色	李沧区，城阳区

从颜色划分还可以发现，行政规划不能很好地区分道路的联通情况。比如，崂山区的道路在网络中被隔成两部分，由少数道路联通，黑色区域反而和市南区联系更紧密。从历史的角度来理解，地区的行政规划变化比道路规划更加灵活。查阅 1994 年青岛市政府发布的区政府重大调整概述发现，崂山区是因发展需要而划出的新区，部分土地属于市南区等其他区域。

青岛市市区行政区划调整后各区情况概况

一、市南区:保留原有行政区域范围、名称和机关驻地不变，只将东部与中韩镇交界区作小调整。辖区面积为30.11平方公里，辖14个街道办事处，人口为344250人，人口密度为每平方公里11433人。

保留市南区不变，是因为该区区域面积、人口数量和分布较为适中，且市区的多数旅游景点在该区，不予变更有利于保持其历史风貌、区域特色和社会知名度。

二、市北区：将原市北区、台东区和四方区鞍山路以南的吴家村、错埠岭2个街道办事处及李村镇的杨家群、河马石、夹岭沟、曲家庵子和中韩镇的浮山后、埠西等6个村以及高科技工业园7号规划线以西的区域合并。区机关设在原台东区机关驻地。

沿用市北区名称，与市南区对应，可以体现该区的地理位置。调整后的市北区面积为:28.63平方公里，辖23个街道办事处和5个自然村，人口为463067人，人口密度为每平方公里16174人。

三、四方区:将原沧口区李村河以南的水清沟、开平路、盐滩、郑州路、洛阳路等5个街道办事处和原崂山区李村河沿张村河以南、308国道以西的河崖、后台、小水清沟、河西、华光、大山、小河西、保儿、保儿西山，双山等10个自然村以及高科技工业园西北部,308国道以北小三角地的区域合并，区机关设在原四方区机关驻地。

调整后的四方区面积34.55平方公里，辖16个街道办事处和10个自然村，人口为331752人，人口密度为每平方公里9602人。

四、李沧区:将原沧口区李村河以北的四流中路、振华路、晓翁村、西流庄、永安路、营子、板桥坊、楼山后等8个街道办事处和楼山乡的区域与原崂山区李村镇张村河以北区域合并，设立新的李沧区。区机关所在地在原崂山区政府机关驻地。

新区取名两地原地名首字合并命名，既尊重并顺应群众习惯，又保持体现了该区的地域特点。调整后,该区面积为97.98平方公里，辖8个街道办事处,2个乡镇，52个村委会，人口为280836人，人口密度为每平方公里2866人。

五、崂山区:将中韩镇（即青岛高科技工业园划出浮山后、埠西2个村，7号规划线以西区域）与原崂山区的沙子口镇、北宅镇、王哥庄镇的区域合并，设立新的崂山区。区机关所在地在原中韩镇。该区沿用原崂山区名称是因为：崂山风景旅游区的大多数景点和崂山主峰都在该区，称崂山区有利于提高知名度。

设立新的崂山区，是因为该区是青岛市区向东部开发拓展的主要区域。除崂山风景区外，国家批准的高科技工业园、旅游度假区，还有新建的仰口旅游开发区均在该区之内。目前，高科技工业园已有各类企业176个,总投资人民币67.8亿元，美元3.5亿元，具有很大的发展潜力。沙子口、北宅、王哥庄镇的工业已有一定基础，可作为高科技工业园延伸发展的理想地域，同时，驻在该区的大专院校较多，因此，将其独立划为一个区，有利于充分发挥高科技和旅游资源的优势，尽快形成一个以高科技工业和以风景旅游为特色的新的经济区域。

该区面积为390.4平方公里，辖4个乡镇，135个村委会，人口为179490人，人口密度为每平方公里460人。

六、城阳区:在原崂山区所辖惜福镇、夏庄、流亭、城阳、棘洪滩、上马、红岛、河套8个镇的区域设立新城阳区，区机关设在城阳镇驻地。设立城阳区，有利于发挥城阳、流亭、棘洪滩等经济发达乡镇作为小城镇群的综合辐射功能，形成工农并举、城乡一体的格局，推动这一区域的整体发展。

该区面积为553.2平方公里，辖8个乡镇，227个村委会，人口为411940人，人口密度为每平方公里745人。

七、黄岛区:保留原黄岛区的行政区域范围、名称、机关驻地不变。面积为181.4平方公里，辖4个乡镇，107个村委会，人口为109686人，人口密度为每平方公里1611人。

图 3 青岛市区域划分

（2）道路车流量统计

接下来我们研究了道路车流量规律。（附录 1.1）

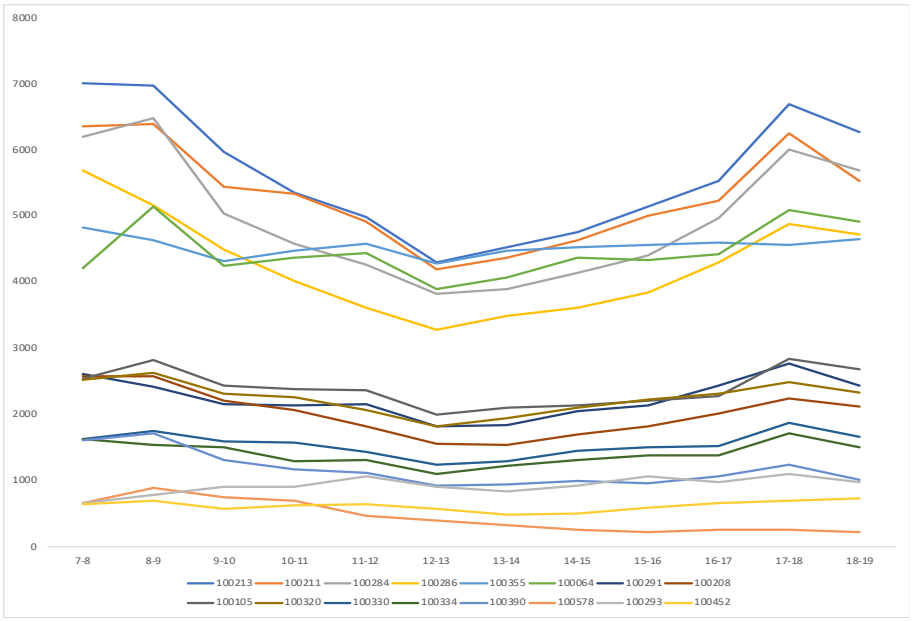

图 4　8 月 1 日部分道路分小时车流量

首先我们对每个路口的车流量进行了分小时统计，上图展示了我们随机抽取的四组路口随时间变化的车流量情况。可以发现，在车流量越大的路口，其车流量的早晚高峰车流量增幅越大，这个增幅随总车流量的减少而减少。一种合理的解释是，车多的地方也是工作者密集的地方。

但由于各条道路有不同的车道数，这不能去衡量一条道路的效率。因此我们为每条道路匹配了对应的车道数，求出了每条车道的车辆密度。

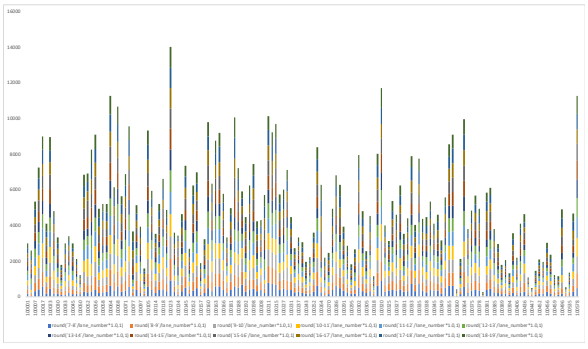

图 5　8 月 3 日路口分时段车辆密度堆积图

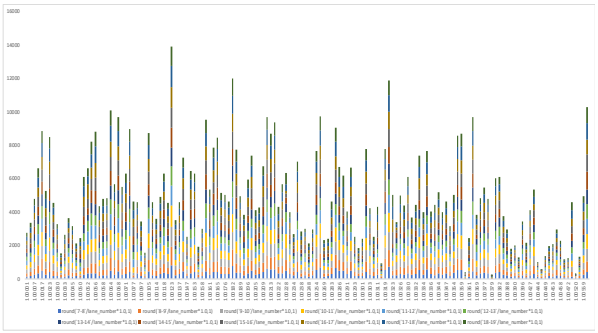

图 6　8 月 4 日路口分时段车辆密度堆积图

可以发现，虽然每天之间有一些区别，但总体密度趋势保持不变。

在知道不同道路的联通结构和车辆密度后，我们提出了一种交通规划假设：最四通八达的路段是否就是车流量密度最大的路段，从而保证了最优的交通便利程度和道路使用效率？

（3）道路规划合理性

用联通数和车辆密度分别列出排名前十的道路，进行对比，分析道路的应用有效性，评价青岛市的道路规划问题。

假设，我们将联通数前十的路段号和车辆密度前十的路段号进行了对比，发现居然全部不匹配。

表 2 联通排名和车辆密度排名关系

	联通排名	车辆密度排名
1	100130	100123
2	100178	100320
3	200202	100064
4	100334	100578
5	100305	100068
6	100122	100211
7	200837	100182
8	100121	100364
9	100000	100160
10	100205	100215

为解释该现象，我们将路段号和路段名在地图上进行对应，发现了不同的聚集关系。

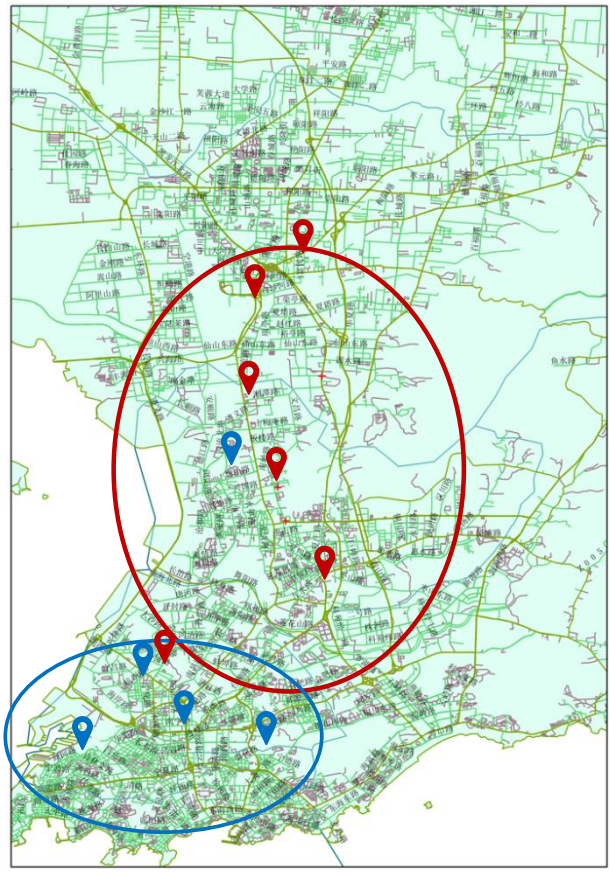

图 7 道路聚集关系

车辆密度排名前十的路段大部分位于蓝色区域，联通情况前十的路段大部分位于红色区域。查看地图中道路实际情况我们发现，蓝色区域为市南区，红色区域主要为李沧区。市南区为旧城区，人口密集，道路错综复杂；李沧区为进出青岛的交通枢纽，路长道宽。这样的道路设置或许是上述现象出现的原因。

那么为什么会有这样的规划呢？

我们又在网上发现了“青岛经济北迁” 的说法，我们发现，几乎每年都有关于青岛市政府北迁的传闻，每年市政府都要出来澄清谣言。

图 8 新闻截图

这样的舆论使得我们的分析更加具有价值。我们的发现表明目前青岛市的车辆仍然在市南区，市北区相对密集，但在道路规划开发的情况上，李沧区开始呈现优势，也侧面应证了青岛市经济北迁的可能性。但如果要从交通规划和车流量的角度去检验这一猜测，需要更多时间跨度的交通数据。

3.2 车辆分析

（1）车流量分析

首先，我们想要探究出租车供给情况随时间的变化，我们发现每天的出租车数量几乎一样，并且不随时间有明显变化。

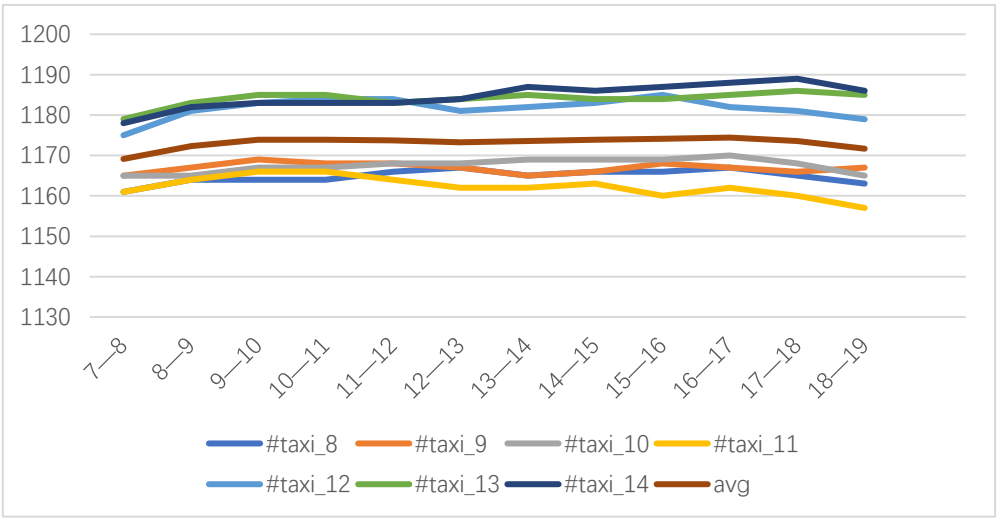

图 9 8 月 8 日至 14 日分小时出租车数量

代码示例（附录 1.2）

再根据每两天相同出租车（车牌）数量对比图：

表 3 出租车（车牌）数量对比表

table1	table2	sameid			
8	9	1170	10	11	776
8	10	1169	10	12	782
8	11	774	10	13	782
8	12	780	10	14	782
8	13	780	11	12	1172
8	14	780	11	13	1171
9	10	1171	11	14	1171
9	11	776	12	13	1188
9	12	782	12	14	1188
9	13	782	13	14	1188
9	14	782			

代码示例（附录 1.3）

从上图我们大体可以看出：8-10 日的出租车基本相同，11-14 日的出租车基本相同，虽然两个时间段之间的不同出租车接近 400 辆，但总体出租车数量并未有明显改变。

为了解释上述现象，我们查看了 18 天总体汽车数量变化：

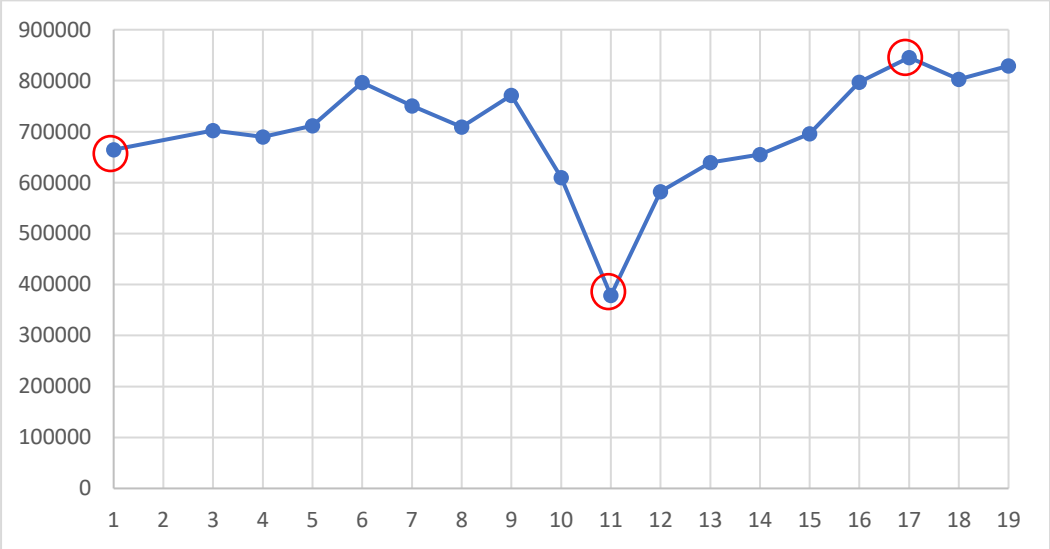

图 10 车流量折线图

代码示例（脚注）[1]

可以明显看出从 8 月 10 日开始，车流量明显下降。

[1] **select** count(distinct vehicleID)
from train_trafficFlow_13

表 4　青岛市 2019 年 8 月 1 日-8 月 19 日天气数据

日期	最高气温	最低气温	天气	风向
2019/8/1	31	26	阴	东南风3级
2019/8/3	28	25	阴	东南风2级
2019/8/4	30	26	阴	东南风3级
2019/8/5	29	26	阴	东南风3级
2019/8/6	29	25	阴	东南风3级
2019/8/7	29	25	多云	东南风3级
2019/8/8	30	26	阴	东南风2级
2019/8/9	30	26	阴	东南风3级
2019/8/10	30	24	阴	东南风5级
2019/8/11	26	25	大雨	东南风2级
2019/8/12	24	24	小雨	西南风4级
2019/8/13	25	23	阴	西北风5级
2019/8/14	26	23	小雨	西北风3级
2019/8/15	28	21	阴	西北风4级
2019/8/16	31	22	多云	西北风3级
2019/8/17	31	21	多云	西北风3级
2019/8/18	30	22	多云	西南风3级
2019/8/19	31	22	晴	西南风2级

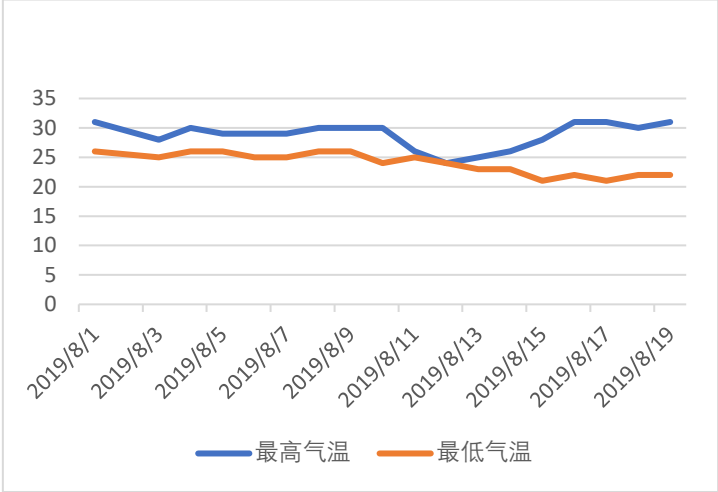

图 11　青岛市气温变化情况

通过对比天气数据的变化（见表 x 和图 12、13），可以明显看出天气在 8 月 11 日有明显的降温。其中青岛在 8 月 10 日和 11 日都发布了暴雨红色预警信号。

图 12　新闻截图

图 13 新闻截图

这就很好的解释了以上的两个现象。再回到我们之前讨论的相同出租车每两天数量对比上，我们做出了以下猜测：

1. 部分出租车司机得知恶劣天气来临，选择短时间内不出车工作。

2. 为了平衡青岛市中心城区的出租车运营情况，出租车公司可能进行了内部调度，促使出租车总量达到平衡状态。

因为车流量变化较大，且其变化规律可以被天气等因素较合理地解释，我们

可以得到相关出租车行业的结论：

1. 出租车供给弹性小

2. 出租车的需求有波动，但因供给弹性小，出租车司机较难自由选择是否上班（特殊情况除外）

3. 有潜在的乘车需求未被满足，因此滴滴打车这样的平台存在市场，从而填补这部分需求。

根据八月八日至十四日的分小时车辆数对比（附录 2.1）：

我们抽取九日（周五）和十日（周六）两天进行对比：

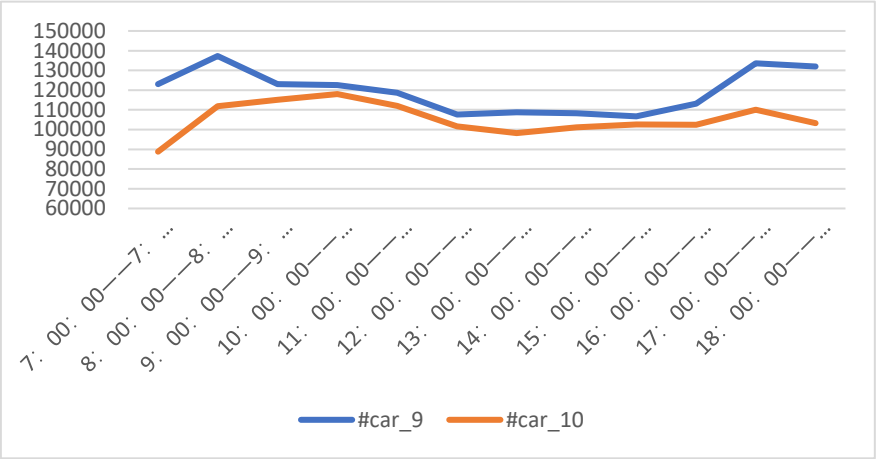

图 14 工作日和周末分小时车流量对比

我们可以得出：

1. 周末上午高峰期的时间段较工作日有所推迟且车辆数明显减少，可能是因为大部分工作人员周末休息，不像平时一样那么早出门上班。

2. 下午的高峰期较为一致，但车辆数有明显的减少

3. 除开高峰期，两条折现其余部分的差值较小，可以得出这些时间段的车辆并不受工作日/休息日的影响，可能是没有休息日的交通运输工作岗位，如：快递，公交车等等。

（2）车牌分析

建军节军队车牌统计

对 8 月 1 日各大战区、军区、军种的车辆建军节当天在青岛的分布进行统计。

例如，以下是对武警部队车辆的统计。

查询得知，v/z 代表中央军委车辆，k 代表空军，h 代表海军，b 代表北京军区，等等。最终结果如下：

表 5 武警部队车辆统计表

全体车辆 ： 664453			
非地方车辆	7077	北京军区	2495
地方车辆	657376	沈阳军区	191
武警	35	兰州军区	155
中央军委	352	济南军区	92

空军	732	南京军区	831
海军	614	广州军区	13
二炮	302	成都军区	14
东部战区	379	其他	872

（其中非地方车辆占 1.07%，地方车辆占 98.93%）

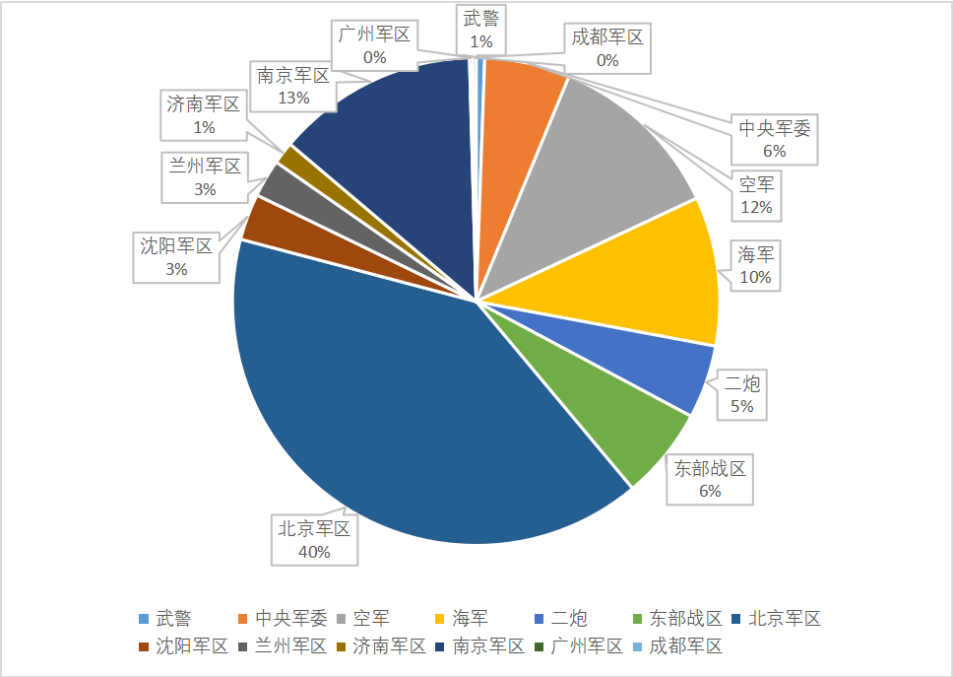

图 15 武警部队车辆占比

外省车辆流入青岛统计

统计各省级行政区（不含港澳台、山东省）的车辆（依据车牌识别）在青岛市的数量。

例如，以下是对某日四川省的统计：代码示例（脚注）[2]

类似地，对 7 天 30 个省级行政区的情况进行统计。最终结果见附录 2.2：

[2] **select distinct** vehicleID **from** train_trafficFlow_1
where (rtrim(vehicleID) like '[川][a-z]%')
order by vehicleID

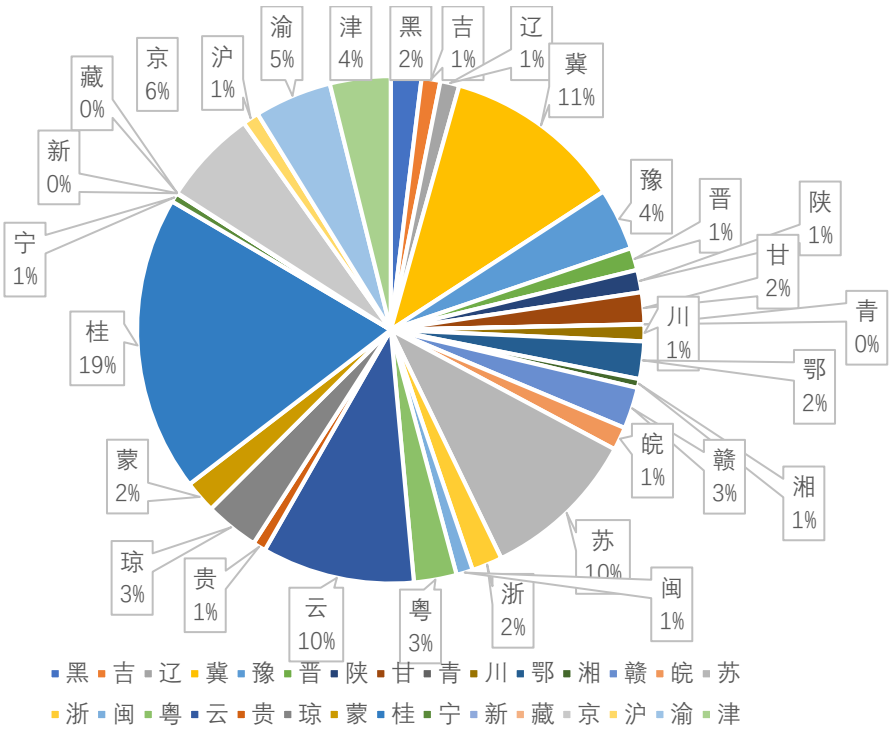

图 16 各省级行政区车辆数

表 6 平均车辆数排行

省份	平均车辆	排名						
桂	19622	1	赣	2747	11	吉	1247	21
冀	11933	2	鄂	2482	12	川	1105	22
苏	10429	3	蒙	2135	13	闽	1093	23
云	10119	4	甘	2099	14	沪	1075	24
京	6344	5	浙	2023	15	贵	860	25
渝	5128	6	黑	2020	16	宁	602	26
豫	4123	7	皖	1539	17	湘	590	27
津	4022	8	陕	1481	18	新	44	28
琼	3579	9	晋	1474	19	青	22	29
粤	2830	10	辽	1261	20	藏	3	30

上图是外省流入青岛车辆绝对数量的情况。考虑到地理上越靠近青岛越有动机自驾前往青岛，那么随机选取两辆来自不同外省的车，距离青岛较远的那辆车，具有更“爱青岛”的特征。于是，利用全国各省平均距离（实际按省会计算）表，将各省到青岛的距离纳入分析因素中（表 7 即为各省到山东平均距离，我们将其近似看作到青岛的距离）

表 7 全国各省平均距离

津	291.85	吉	1057.18	沪	707.23	青	1389.28
冀	299.68	闽	1170.59	鄂	722.94	贵	1502.16
京	356.94	粤	1170.59	浙	752.03	桂	1767.27
豫	391.45	甘	1211.18	辽	782.28	云	1878.85
晋	434.24	渝	1234.96	陕	804.90	琼	1940.11
皖	543.47	黑	1275.85	赣	900.89	蒙	1940.11
苏	543.47	川	1375.44	宁	981.15	藏	2536.20
沪	707.23	青	1389.28	湘	1010.56	新	2633.24

基于此表，我们定义“爱青岛指数”如下：

$$\text{爱青岛指数}=\frac{\text{某外省7天内在青岛的平均车辆数}}{1000}\times\left(1+\frac{\text{某外省到山东的距离}-\min\{\text{各省到山东的距离}\}}{\max\{\text{各省到山东的距离}\}-\min\{\text{各省到山东的距离}\}}\right)$$

进而我们得到 30 个省区市的“爱青岛指数排名”：

表 8 爱青岛指数排名

省份	排名	津	8	省份	排名	闽	23
桂	1	琼	9	浙	16	贵	24
冀	2	粤	10	陕	17	沪	25
苏	3	赣	11	皖	18	宁	26
云	4	鄂	12	吉	19	湘	27
渝	5	蒙	13	川	20	新	28
京	6	甘	14	晋	21	青	29
豫	7	黑	15	辽	22	藏	30

对外省省会 7 天内在青岛车辆数进行分析

统计各省级行政区行政中心（不含港澳台、山东省）的车辆（依据车牌识别）在青岛市的数量。

例如，以下是对某日成都市的统计：代码示例（脚注）[3]

类似地，对 7 天 30 个省级行政区行政中心的情况进行统计。最终结果如下：
进一步，计算各省级行政区行政中心的车辆在该省级行政区车辆总数中的比重，（附录 2.3）：

[3] **select distinct** vehicleID **from** train_trafficFlow_1
where (rtrim(vehicleID) like '[川][a]%')
order by vehicleID

表 9 各省级行政区行政中心的车辆在该省级行政区车辆总数中的比重

省	比重	闽	0.25	琼	0.14
青	0.55	蒙	0.25	苏	0.13
新	0.51	皖	0.24	粤	0.11
陕	0.45	吉	0.24	冀	0.10
藏	0.32	宁	0.23	津	0.10
辽	0.29	鄂	0.22	贵	0.09
豫	0.28	晋	0.20	渝	0.07
黑	0.28	赣	0.17	云	0.07
湘	0.26	沪	0.17	甘	0.06
川	0.26	浙	0.15	桂	0.06
		京	0.15		

使用 2017 年经济统计数据，得出各省会城市 GDP 占其省的比重（不考虑直辖市）：

表 10 各省会城市 GDP 占其省的比重

省	GDP占比	海南	9	广东	18
宁夏	1	湖南	10	福建	19
青海	2	云南	11	广西	20
吉林	3	安徽	12	山西	21
黑龙江	4	贵州	13	河南	22
湖北	5	新疆	14	河北	23
四川	6	辽宁	15	内蒙古	24
陕西	7	江西	16	江苏	25
甘肃	8	浙江	17		

对比发现，青海、黑龙江、四川、湖南四省，在经济上 GDP 较为集中于省会城市，而流出青岛的车辆也较为集中于省会城市。

（四）总结&应用

4.1 总结

首先，通过对青岛市道路进行联通结和使用效率的分析，我们可以发现道路的联通情况和行政划分的历史变化存在一致性。之后，通过车流量统计，可以发现在车流量越大的路口，早晚高峰车流量增幅最大，这个增幅随总车流量的减少而减少。为了严谨起见，通过车道数计算得出每条车道的车辆密度，发现每天之间有一些区别，但总体密度趋势保持不变。对于联通排名和车辆密度排名关系的不匹配关系，我们通过地图比对，发现了其中的聚集关系，即“经济北迁”的可能性。

其次，我们通过车辆分析进一步得出青岛交通的各方面信息。通过出租车数量的统计，发现总数基本保持一致，需求有所波动但供给弹性小。参考天气数据，

我们能明显地发现车流量会因恶劣天气而急剧减少。并且车流量的高峰时间段会因休息日而略微后移。

最后，针对数据中包含车牌信息，我们对其进行分析发现，其他省份车辆在青岛的数量随地理距离的增加而增加。通过自定义的“爱青岛指数”，我们能够清晰地得到排名，前三依次为桂，冀，苏。因为同省份各市车牌有明显区别，我们通过“计算各省级行政区行政中心的车辆在该省级行政区车辆总数中的比重”来侧面描述各个省份的经济集中度。

4.2 找路系统（最短可行路线）：

通过上述操作，我们拥有青岛市主要城区路段的联通数据，因此我们尝试使用 SQL 实现地图导航功能（代码见附录 1.4）。我们的方法基于一个假设：走的路段最少的路线是较优路线。

我们将路段联通数据视作一个有向图，使用递归查询实现了图的可重复广度优先搜索，由此我们可以通过输入出发点和目的地编码查看能到达目的地的最短路段数。

4.2.1 图的可重复广度优先搜索

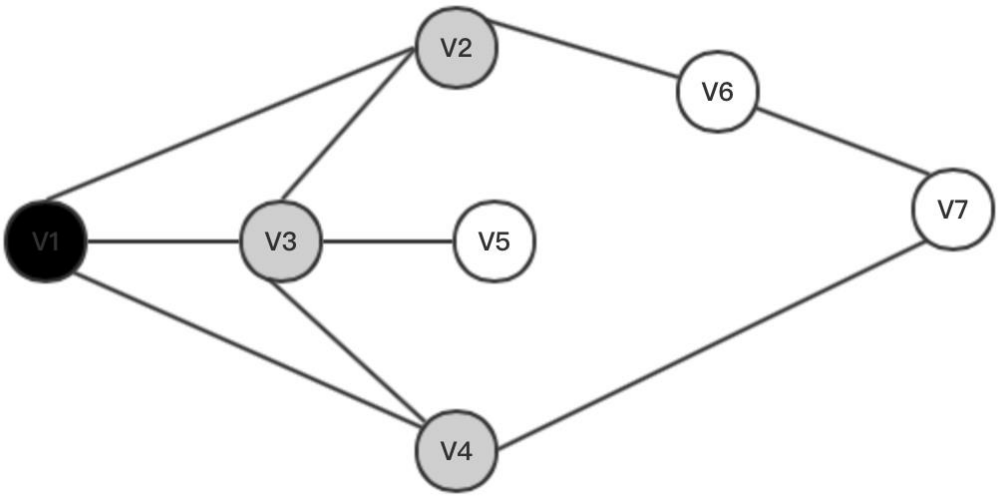

图 17 广度优先搜索示例

广度优先搜索（也称宽度优先搜索，缩写 BFS，以下采用广度来描述）是连通图的一种遍历算法这一算法也是很多重要的图的算法的原型。Dijkstra 单源最短路径算法和 Prim 最小生成树算法都采用了和宽度优先搜索类似的思想。其别名又叫 BFS，属于一种盲目搜寻法，目的是系统地展开并检查图中的所有节点，以找寻结果。换句话说，它并不考虑结果的可能位置，彻底地搜索整张图，直到找到结果为止。基本过程，BFS 是从根节点开始，沿着树(图)的宽度遍历树(图)的节点。如果所有节点均被访问，则算法中止。

4.2.2 算法思想：

（1）从图中某个顶点 v 出发，访问 v，然后将 v 进队。

（2）只要队列不空，则重复下述处理。

① 队头顶点 u 出队。

② 依次检查 u 的所有邻接点 w，访问 w，然后将 w 进队。

4.2.3 实现方法：

我们使用 mysql 中 group_contact 和 find_in_set 函数实现迭代查询，并将每一次的结果放到表中新的一行，设置一个迭代上限来终止函数，从而找到允许可迭代次数内目的地每一次出现的位置。

比如：我们把 100000 作为出发点，100059 作为目的地。

```
SELECT queryChildren(100000);
select * from testtest where `id` like "%100059%";
```

返回值为：

index	id
3	100059,100060,100562,100066,100064,100060,100411,100005,100439,100004,100004,100003,100008,100657
5	100059,100060,100060,100061,100562,100066,100066,100058,100365,100198,100188,100080,100034,100002,
6	100052,100059,100053,100060,100193,100060,100061,100053,100062,100562,100015,100494,100155,100066,1

返回表的第一列为自动生成的序号，说明从 100000 到 100059 需要的最短路段数为 3。

（五）研究报告评析

5.1 亮点

1.数据真实有效。数据均来自 DataCastle 竞赛数据网，由山东省大数据局、青岛市大数据发展管理局提供真实数据。

2.数据量大（约 1.1 亿条数据）。其中包含了 2019 年 8 月 1 日至 19 日出租车 GPS 数据和路口机动车监控数据两部分。其中出租车数据每天包含约 200 万条，路口机动车监控数据每天约 500 万条，经过数据处理后，实际使用数据约为 1.1 亿条。

5.2 不足

1.数据完整性不足。我们所探究的部分问题，如城市经济，道路规划以及人口地理相关问题，并不能在较小时间跨度下，给出极其详细准确的分析与结论。如果时间跨度能够更加大，包含近两年来的相关数据，更能准确地验证我们的猜想，得出结论。

2.区域较小。此次研究使用数据局限于青岛市三环以内，更全面的数据能帮助我们获取更多信息并验证更多猜想。

附录：

代码附录

1.1

```
*不断建表保存 一天 12 次*/
Create table am_7_8_9(
select crossroadID, COUNT(DISTINCT vehicleID) nums from train_trafficflow_9

where STR_TO_DATE(TIMESTAMP,'%Y-%m-%d %H:%i:%s')>=STR_TO_DATE('2019-08-09 07:00:00','%Y-%m-%d %H:%i:%s')
and STR_TO_DATE(TIMESTAMP,'%Y-%m-%d %H:%i:%s')<STR_TO_DATE('2019-08-09 08:00:00','%Y-%m-%d %H:%i:%s')
group by crossroadID
order by crossroadID)-- 连接
create table Aug_9_flux(
select a.crossroadID, a.nums '7-8', b.nums '8-9', c.nums '9-10', d.nums '10-11', e.nums '11-12',
f.nums '12-13', g.nums '13-14', h.nums '14-15', i.nums '15-16', j.nums '16-17', k.nums '17-18', l.nums '18-19'
from am_7_8_9 a, am_8_8_9 b, am_9_8_9 c, am_10_8_9 d, am_11_8_9 e,
am_12_8_9 f, pm_1_8_9 g, pm_2_8_9 h, pm_3_8_9 i, pm_4_8_9 j, pm_5_8_9 k, pm_6_8_9 l
where a.crossroadID=b.crossroadID and b.crossroadID=c.crossroadID and c.crossroadID=d.crossroadID
and d.crossroadID=e.crossroadID and e.crossroadID=f.crossroadID and f.crossroadID=g.crossroadID
and g.crossroadID=h.crossroadID and h.crossroadID=i.crossroadID and i.crossroadID=j.crossroadID
and j.crossroadID=k.crossroadID and k.crossroadID=l.crossroadID)-- 匹配车道数
create table Aug_9_flux_with_lane_no(select a.*, b.lane_number from Aug_9_flux a, lane_number b
where a.crossroadID=b.crossroadID)

除以车道数
Create table cars_no_per_lane_8_9
(select crossroadID, round(`7-8`/lane_number*1.0,1) , round(`8-9`/lane_number*1.0,1),
round(`9-10`/lane_number*1.0,1),round(`10-11`/lane_number*1.0,1), round(`11-12`/lane_number*1.0,1),
```

```
round(`12-13`/lane_number*1.0,1), round(`13-14`/lane_number*1.0,1), round(`14-15`/lane_number*1.0,1),
round(`15-16`/lane_number*1.0,1),round(`16-17`/lane_number*1.0,1), round(`17-18`/lane_number*1.0,1),
round(`18-19`/lane_number*1.0,1)
from Aug_9_flux_with_lane_no)
```

1.2

```
 declare @i int
create table day14(time_period int, car_number float)
set @i=0
while @i<=12
Begin
  insert into day14
     select @i+7, count(distinct taxiid) from trainTaxiGPS_14
     where convert(float,convert(datetime,timestamp))
     between (convert(float,convert(datetime,'2019-08-14 7:00:00'))+0.125*@i/3)
     and (convert(float,convert(datetime,'2019-08-14 07:59:59'))+0.125*@i/3)
  set @i=@i+1
end
select * from day14
```

1.3

```
create table Flow_number (taxiid nvarchar(50) , day int)
insert into Flow_number
select distinct vehicleID,8 day
from train_trafficFlow_8
union
select distinct vehicleID,9 day
from train_trafficFlow_9
union
select distinct vehicleID,10 day
from train_trafficFlow_10
union
select distinct vehicleID,11 day
from train_trafficFlow_11
union
select distinct vehicleID,12 day
from train_trafficFlow_12
```

```
union
select distinct vehicleID,13 day
from train_trafficFlow_13
union
select distinct vehicleID,14 day
from train_trafficFlow_14

declare @a int,@b int
create table Flow_contrast(table1 int, table2 int, sameid int)
set @a=8
set @b=9

while @a<=13
Begin
    while @b<=14
    begin
        insert into Flow_contrast
        select @a,@b,count(distinct a.taxiid)
        from Flow_number a,Flow_number b
        where a.day = @a
        and   b.day = @b
        and a.taxiid = b.taxiid
        set @b=@b+1
    end
set @a=@a+1
set @b=@a+1
end
select * from Flow_contrast
```

1. 4

```
drop table if exists `test1`;
create table `test1` as (
 select * from `路网上下游 ID`);
 delete from test1 where `downroadID`=100000;
drop table if exists `testtest`;
create table testtest(
`index` int(11) not null AUTO_INCREMENT,
`id` varchar(4000) default '0'
,
primary key(`index`)
);

drop table if exists `tmp`;
```

```
create table tmp(`id` varchar(16383));

DROP FUNCTION IF EXISTS queryChildren;
SET GLOBAL log_bin_trust_function_creators = 1;
DELIMITER ;;
CREATE FUNCTION queryChildren(star INT)
RETURNS VARCHAR(16383)
BEGIN
DECLARE sTemp VARCHAR(16383);
DECLARE sTempChd VARCHAR(16383);
DECLARE t int DEFAULT 0;
-- DECLARE x int DEFAULT 0;
SET sTemp='$';
SET sTempChd = CAST(star AS CHAR);
while t<6 do
SET sTemp= CONCAT(sTemp,',',sTempChd);
insert into testtest(id) values(sTempChd);
SELECT GROUP_CONCAT(downroadID) INTO sTempChd FROM test1 WHERE FIND_IN_SET(u
proadID,sTempChd)>0;
set t=t+1;
END WHILE;
-- INSERT INTO tmp SELECT * from testtest where `id` like dest;
RETURN sTemp;
END
;;
DELIMITER;
SELECT queryChildren(100000);
select * from testtest where `id` like "%100059%";
```

图表附录

2.1

time_period	#car_8	#car_9	#car_10	#car_11	#car_12	#car_13	#car_14	avg
7:00:00——7:59:59	122084	123064	88788	48483	100369	120271	120649	103387
8:00:00——8:59:59	133650	137285	111851	61248	105535	133458	132545	116510
9:00:00——9:59:59	110613	123015	115133	62267	85167	111433	114494	103160
10:00:00——10:59:59	103405	122618	117988	66297	87592	104300	107356	101365
11:00:00——11:59:59	105113	118611	111968	64519	86840	101466	104720	99034
12:00:00——12:59:59	103874	107695	101641	56781	76975	90328	95671	90424
13:00:00——13:59:59	110117	108824	98230	55354	83032	95847	99547	92993
14:00:00——14:59:59	110858	108269	101164	57318	87181	97251	102099	94877
15:00:00——15:59:59	103631	106701	102543	56082	88441	99103	102871	94196
16:00:00——16:59:59	110710	113218	102410	53749	98386	109352	110236	99723
17:00:00——17:59:59	134242	133588	110009	52796	127035	137779	135332	118683
18:00:00——18:59:59	127248	132009	103202	41464	112690	123544	127791	109707

八月八日至十四日的分小时车辆数图

2.2

各省级行政区车辆数表

晋	1474	958	1409	1307	1440	2085	1550	1572
陕	1481	1132	1199	1310	1574	1777	1734	1643
甘	2099	1028	2090	2120	1862	3228	2420	1947
青	22	14	25	19	13	25	26	32
川	1105	978	1054	1013	1084	1340	1121	1148
鄂	2482	2176	2478	2398	2427	2897	2541	2458
湘	590	475	570	616	588	726	607	545
赣	2747	2156	2574	2676	2642	3699	2860	2622
皖	1539	1378	1650	1592	1606	1534	1490	1525
苏	10429	7794	10739	12471	9262	11350	11797	9590
浙	2023	1960	1802	1851	2079	2189	2145	2132
闽	1093	718	1040	1160	997	1472	1168	1094
粤	2830	1949	2852	2918	2432	3727	3220	2714
云	10119	4840	9466	11501	8349	15899	12933	7848
贵	860	606	869	753	880	1152	910	851
琼	3579	1995	3586	3866	3134	4926	4091	3452
蒙	2135	1540	2034	2007	2315	2724	2108	2220
桂	19622	10655	18999	23611	16672	26010	25116	16292
宁	602	485	601	575	587	804	577	586
新	44	34	48	54	41	43	41	50
藏	3	5	1	3	5	2	2	1
京	6344	4943	6271	6079	6304	7648	6694	6466
沪	1075	960	1026	1063	1097	1211	1084	1087
渝	5128	2687	4786	5851	4031	8001	6313	4224
津	4022	3348	4219	4028	3788	4808	4108	3854

2.3

广度优先搜索示例表

省	省会平均	Day1	Day2	Day3	Day4	Day5	Day6	Day7
黑	562	471	560	513	539	697	586	567
吉	300	291	302	302	318	316	306	267
辽	367	327	324	376	380	411	397	355
冀	1230	896	1155	1363	1202	1383	1343	1270
豫	1154	787	1148	1172	1129	1288	1296	1260
晋	302	230	243	275	279	380	353	352
陕	666	493	471	548	762	776	817	792
甘	127	96	120	124	121	166	140	121
青	12	9	13	11	6	13	16	17
川	284	253	240	269	288	367	279	293
鄂	556	470	521	525	573	646	583	574
湘	153	123	143	161	160	186	162	138
赣	479	357	396	445	479	683	525	469
皖	373	373	352	369	415	381	341	378
苏	1400	1126	1532	1520	1331	1447	1417	1429
浙	296	240	279	305	326	335	291	293
闽	275	206	245	272	237	373	296	294
粤	299	244	275	288	276	397	352	261
云	673	450	613	651	626	1023	758	588
贵	76	42	77	57	79	112	95	70
琼	516	340	473	494	495	710	567	536
蒙	530	379	516	500	552	681	524	559
桂	1122	792	1065	1143	1067	1507	1259	1019
宁	137	112	130	117	144	186	128	145
新	23	20	23	21	24	23	24	25
藏	1	0	0	1	3	0	2	0
京	926	695	887	824	972	1186	953	966
沪	180	158	159	174	184	212	184	187
渝	372	215	323	365	350	610	404	337
津	410	322	404	415	352	569	411	394