

城市公共自行车服务系统运行状况和效率分析

——基于温州市鹿城区公共自行车系统运营实践的研究

摘 要

本文基于温州市鹿城区公共自行车系统的 20 天内借车和还车原始数据，构建模型对公共自行车的使用频次分布、供求状况和自行车租赁的站点位置特征进行了分析，进而探讨了公共自行车系统的有效性，挖掘其背后的系统运行规律和机理，并分析了这一系统的其他特征，如潮汐现象等，最终建立在长期可持续发展的基础上提出解决方案。

针对累计借还车频次、各日借车卡数量和累计借车次数分布的计算，我们主要采用 **MATLAB** 遍历过所有数据并进行统计运算，得到累积分布结果的二维直方图，采用拟合的方法得到其近似分布。我们还构建了自行车用户群体的效用函数模型，探讨高于某一保留效用阈值的城市居民采用公共自行车出行偏好程度。发现用户选择公共自行车出行的效用函数关于出行时间的变化近似为一个矩形脉冲函数，其拐点约为 28 分钟。当出行时间少于 28 分钟时，用户更倾向于选择公共自行车出行；在 28 分钟之后，用户倾向于选择其他的交通工具，仅在随机因素的作用下选择公共自行车。

在公共自行车服务系统站点设置和锁桩数量的配置的分析中，我们在鹿城区公共自行车管理中心网站上找到每个自行车站点的位置，在电子地图服务提供商网站上查询出该站点位置的坐标。对于距离的界定，我们采用欧氏距离、直角距离和地图实际距离三种方法度量站点间距。第三种方法较为精确，我们采用颜色替换，形态学处理，道路生长细化的方法克服了第三种方法的度量困难，最终获得城市道路信息，用于两地之间的实际道路路径求取。

我们将一天中的 24 小时划分成以 0.2 小时为间隔的若干时段，分别统计落入每个时段的借还车频次，即可用来反映借还车频次时刻分布，我们发现许多站点都存在较明显的早晚高峰现象，反推出的效用函数具有稳定性和可靠性，表明我们的研究是可信的。在峰值的搜寻中，我们借助 **MATLAB** 使用均值滤波的方法减弱随机因素造成的影响，对站点的借还车高峰时刻进行统计。通过计算等效时间峰值统一不同站点的时间峰值维数，然后采用 **K-means** 算法进行聚类分析，从而根据借还车高峰时刻对站点进行了归类。

基于上述初步结果，我们提取了数据中蕴含的信息，进而对鹿城区的公共自行车系统配置效率和运营状况进行分析和评价。我们发现，公共自行车系统存在明显的“潮汐现象”，在高峰时段可能有供不应求的问题，亟需建立良好的调度系统已平衡供求关系。我们拓展了前面的研究，从站点选址的社会福利优化问题以及合理的配套调度机制设计几个角度进一步分析了公共自行车服务系统的运行规律，并提出了相应的改进建议。

关键词：公共自行车 效用函数 分布估计 统计分析 聚类分析

一 问题提出

1.1 题目背景与研究意义

在倡导健康、节能、环保的生活理念的今天，公共自行车得到了越来越多城市居民的青睐，在国内掀起了一场“绿色革命”^[1]。作为同城短区间非机动车载人设备，公共自行车是具有可持续发展性的重要代步手段，日益成为人们生活中不可或缺和替代的主要交通工具之一。作为公共交通末端交通工具的公共自行车不仅弥补了公共交通线网密度不足的缺陷，而且向公共交通提供客源，如图 1 所示。我国现已有众多城市主动将自行车纳入公共交通领域，意图让公共自行车交通与公共交通实现无缝对接，破解交通末端“最后一公里”难题，实现低碳出行，美化城市。

随着公共自行车的普及使用，如何构建合理、高效的公共自行车服务系统是一个非常关键的问题。自行车租赁的站点位置及各站点自行车锁桩和自行车数量的配置，对系统的运行效率与用户的满意度有重要的影响。浙江省温州市鹿城区公共自行车系统的数据能够为我们的机制设计提供参考。根据温州市鹿城区公共自行车管理中心网站 (<http://www.wzbicycle.com>) 提供的数据，鹿城公共自行车的投放量约为 5000 辆，设立 180 个服务网点。启动当天，首先在五马街区和新城街区推出 80 个服务网点，其余 100 个服务网点计划在 10 月设立。服务网点主要位于机关、企事业单位、商业楼宇、公共场所、住宅小区等人员密集地区，间距一般在 300 米至 500 米^[2]。

本文试图基于温州市的地形环境和公共自行车站点建设的现有情况，构建数学模型探讨温州市市民公共自行车的使用水平和分布规律，并对城市自行车服务系统进行评价。目前，我国公共自行车的系统建设尚处于初步阶段，我们的研究不仅针对特定城市的公共自行车系统进行了系统性分析，而且能够为全国其他城市的公共自行车系统优化建设提供参考。

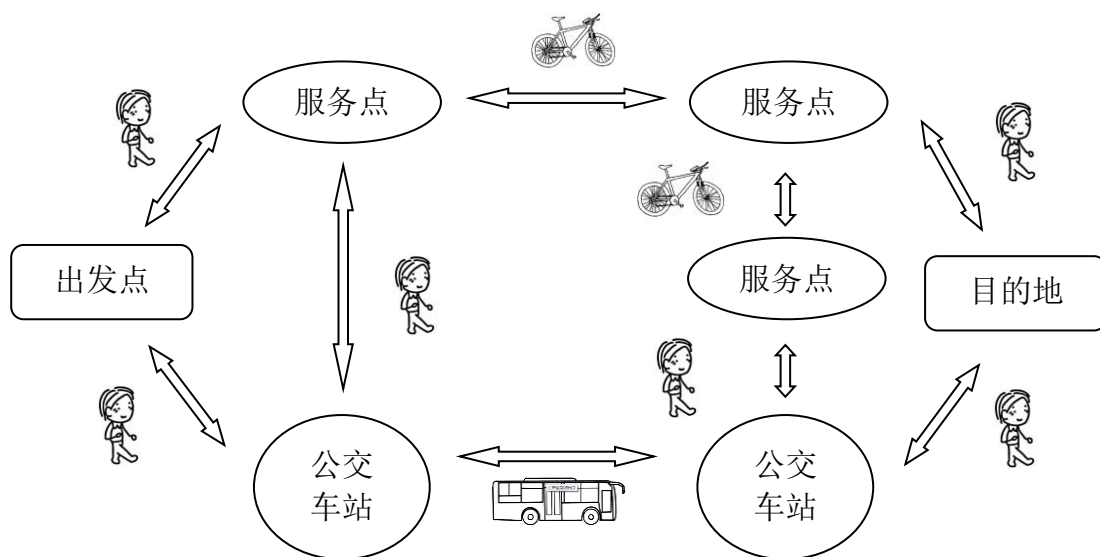


图 1：城市公共自行车系统

1.2 问题重述

借助鹿城区公共自行车客流数据和站点分布图，本文试图解决以下几个问题：

首先，分析公共自行车的使用频度和每次使用的时长：我们统计各站点 20 天中每天及累计的借车频次和还车频次,并对所有站点按累计的借车频次和还车频次分别给出它们的排序，并统计分析每次用车时长的分布情况。

其次，基于借车卡数据对使用主体区分，统计 20 天中各天使用公共自行车的不同借车卡（即借车人）数量，并统计数据中出现过的每张借车卡累计借车次数的分布情况。

然后，我们试图寻找所有已给站点合计使用公共自行车次数最多的一天，并按照以下几个角度探讨站点客流量特点和规律：

- 1) 我们合理界定两站点之间的距离，在此基础上找出自行车用车的借还车站点之间（非零）最短距离与最长距离，对借还车是同一站点且使用时间在 1 分钟以上的借还车情况进行统计。
- 2) 我们选择借车频次最高和还车频次最高的站点，分别统计分析其借、还车时刻的分布及用车时长的分布。
- 3) 寻找各站点的借车高峰时段和还车高峰时段，在给定的温州市鹿城区地图上标注或列表给出高峰时段各站点的借车频次和还车频次，并对具有共同借车高峰时段和还车高峰时段的站点分别进行归类。

进而建立在前面分析的基础上，我们探讨上述统计结果携带了哪些有用的信息，并由此对目前公共自行车服务系统站点设置和锁桩数量的配置做出评价。

最后，我们试图找出公共自行车服务系统的其他运行规律，提出改进建议。

二 问题分析

城市公共自行车系统是一个复杂的系统，“牵一发而动全身”，其机制的设计和模式的优化是城市公共基础设施建设服务的核心。在这一部分中，我们对要解决的问题进行梳理整合，为后文的研究构建逻辑架构。

2.1 基本思路

针对本文的几个关键问题点，我们首先从数据出发，通过对公共自行车使用频率和时长的数据观察统计，分析其中规律，进而采用效用函数合理刻画用户需求和选择模式，合理预测出行时间，建立模型刻画用户偏好。然后我们采用题目中给定的地图，借助电子地图寻找借还车频次高峰，并基于数据和初步结论探索最后的开放式问题——公共自行车系统的运行规律和政策建议。

2.2 具体分析

我们首先进行数据的预处理，利用 Matlab 导入数据，对数据进行整合，剔除异常值和缺漏值，得到用于分析的样本数据。我们分以下几个步骤解决：

- 1) 针对问题一，即计算公共自行车的使用频度和每次使用的时长，我们遍历过所

- 有数据之后，即可以得到各站点每天的借车还车频次的统计数据。对每个站点的全部 20 天的借车与还车数据分别求和，既可以得到每个站点的累计借车与还车频次。用还车时间减去借车时间可以得到用户的用车时间，精确到秒。
- 2) 利用用户效用函数来刻画用户的选择行为，建立自行车出行时间分布的效用函数模型。求解效用函数模型的具体形式，对用户群体的偏好特征进行分析。
 - 3) 针对问题二，我们按照借车卡区分不同用户个体，对用户群体的需求模式进行探索，方法类同问题一。通过累计各个借车人在 20 日中累计借车次数，我们做出频率分布图，拟合得到借车频次的近似分布特征。
 - 4) 针对问题三，共有三种方法界定站点距离——欧式距离，直角距离和实际地图距离，我们可以分别用三种方法求解。第三种方法最为精确，但实现难度较大，我们尝试采用 MATLAB 画图的方法，利用题目给出的地图计算实际距离。在此基础上，寻找借还车频次高峰，并对站点位置、锁桩数量的优劣进行评价。
 - 5) 第四、第五问要求我们发散思维，寻找数据背后的运行规律，我们建立在数据特征的基础上，进一步挖掘公共自行车系统的运行特点，并回顾和借鉴了一些现有研究的方法，最终提出政策建议。

三 基本假设

H3.1: 假设该城区所有的用户都是同质的，即有相同的效用函数。由于生活地理位置和风俗人情接近，影响出行状况的外界不可抗因素相近，这一假设比较合理，并可以简化我们的分析。当用户同质时，我们可以采用代表性用户的效用函数完成推导过程。

H3.2: 不同用户出行需求不同，我们假设用户需要出行的时长（ T ）近似服从指数分布。

H3.3: 假设公共自行车站点设置在研究时间区间内没有明显变更，借车站点序号没有较大的变化，我们可以在一个相对静态的公共自行车硬件系统环境下进行分析。

四 符号说明

符号	意义
$\mathbf{A}_{m \times n} = [a_{ij}]$	第 i 个站点第 j 天的借车次数
$\mathbf{B}_{m \times n} = [b_{ij}]$	第 i 个站点第 j 天的还车次数
m	站点数目的最大值
n	天数的最大值
a_{i0}	第 i 个站点的累计借车频次
b_{i0}	第 i 个站点的累计还车频次

T	用户需要出行的时长
$U(t)$	用户自行车出行的效用
$Q(x)$	标准正态分布的累计分布函数

五 数据预处理

分析发现公共自行车系统的数据记录中存在错误的记录。因此，在对系统使用情况进行统计分析之前，首先要对数据进行预处理，以去除这些错误数据。分析发现，数据记录错误的形式包括以下几种：

- 1) 借车站点序号错误，例如 2012 年 11 月 8 日的序号为 No.16548 的记录，借车站点为“调试站 1”，站点序号为 1000；
- 2) 还车站点序号错误，例如 2012 年 11 月 8 日的序号为 No.30983 的记录，还车站点名称为空白，站点序号为 29999；
- 3) 借还车时间错误，例如 2012 年 11 月 8 日的序号为 No.30983 的记录，借车时间为“2012/11/5 17:50:07”，还车时间为“2012/11/5”，还车时间只有日期没有时刻；
- 4) 用车时间间隔错误，例如 2012 年 11 月 5 日的序号为 No.19436 的记录，借车时间为“2012/11/5 14:24:43”，还车时间“2012/11/5 14:31:23”，实际用车时间为 6 分 40 秒，记录用车时间为 0 分钟。

综上所述，制定筛选条件对错误数据进行滤除。正确的数据需要满足的条件为：

- 1) 借车站点序号小于 1000；
- 2) 还车站点序号小于 1000；
- 3) 借还车时间格式正确，且还车时间晚于借车时间；
- 4) 还车时间减去借车时间所得到的时间间隔（精确到秒）与记录中的时间间隔的误差小于 2 分钟。

我们使用 MATLAB 导入数据，并编写程序根据上述条件对全部 20 天的数据记录进行筛选，最终得到一共 637336 条记录，其中借车时间范围为 2012/11/1 5:59:38 至 2012/11/20 21:33:57，还车时间范围为 2012/11/1 6:02:12 至 2012/11/21 7:37:20。数据一共有 180 个借车站点，站点序号分别为 1~107,109~181；180 个还车站点，序号范围和借车站点相同。

六 公共自行车总体使用情况统计分析

6.1 借车还车频次统计

针对导入的数据，使用 MATLAB 编写程序对各站点每天的借车还车频次进行统计，并分别使用矩阵 $\mathbf{A}_{m \times n} = [a_{ij}]$, $\mathbf{B}_{m \times n} = [b_{ij}]$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$ 来表示借车频次和还车频次的统计结果。其中 a_{ij} 表示第 i 个站点第 j 天的借车次数， b_{ij} 表示第 i 个站点第 j 天的

还车次数。 m 和 n 分别表示站点数目的最大值和天数的最大值，根据数据记录的范围，分别取 $m=181, n=20$ 。

程序初始化后建立 $\mathbf{A}_{m \times n}$ 和 $\mathbf{B}_{m \times n}$ 两个矩阵并把它们置为零矩阵。程序遍历所有数据记录，读取每一条数据记录之后，都对该条数据记录中借车日期、借车站点、还车日期、还车站点所对应的矩阵中的元素加 1。遍历过所有数据，即得到各站点每天的借车还车频次的统计数据。对每个站点的全部 20 天的借车与还车数据分别求和，可得每个站点的累计与还车频次，第 i 个站点的累计借车与还车频次分别用 a_{i0} 和 b_{i0} 来表示，显然有：

$$\begin{aligned} a_{i0} &= \sum_{j=1}^n a_{ij} \\ b_{i0} &= \sum_{j=1}^n b_{ij} \end{aligned} \quad (6.1)$$

各个站点的每天与累计借车与还车频次如表 1 和表 2 所示，限于篇幅，表 1 和表 2 中仅列出了前 10 个站点前 10 天的借还车频次以及累计频次，完整的数据请参见附录。

表 1：各个站点的每天与累计借车频次（部分）

站点	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}	a_{i8}	a_{i9}	a_{i10}	累计
1	91	148	35	152	124	107	88	15	9	6	1760
2	114	112	25	82	135	135	120	41	21	15	1744
3	179	188	47	60	194	189	189	73	65	13	2672
4	235	286	86	255	243	322	259	112	65	66	5232
5	139	165	67	167	161	164	168	53	33	16	2412
6	105	100	50	102	117	118	134	47	19	13	1760
7	106	109	14	55	143	137	157	34	35	5	1675
8	95	80	15	47	106	116	114	36	41	10	1533
9	353	365	177	303	400	460	393	167	124	86	6483
10	354	359	198	311	366	334	376	165	82	53	5821

表 2：各个站点的每天与累计还车频次（部分）

站点	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}	a_{i8}	a_{i9}	a_{i10}	累计
1	93	145	48	167	116	103	98	23	11	4	1821
2	117	110	27	88	125	136	127	53	22	16	1811
3	176	193	65	74	199	187	219	115	54	9	2915
4	224	297	108	283	242	310	297	115	87	61	5582
5	154	153	80	166	182	176	190	61	49	16	2613
6	112	97	41	116	119	119	132	56	33	15	1829
7	106	106	23	59	140	134	157	67	19	3	1798
8	89	89	15	41	101	110	110	54	23	4	1478
9	334	351	171	315	408	447	400	178	121	72	6416
10	341	332	216	314	361	326	347	150	84	52	5741

对累计借车频次和还车频次进行排序，得到的结果如表 3 所示，表 3 中仅给出借车与还车频次最高的前 10 名站点，完整的排序结果见附录。

表 3：站点的借还车频次排序

借车站点累计频次排序			还车站点累计频次排序		
排名	站点序号	累计借车频次	排名	站点序号	累计还车频次
1	42	12286	1	56	12308
2	56	11950	2	42	12148
3	19	9864	3	19	9985
4	63	9670	4	63	9945
5	49	9270	5	49	9398
6	33	8068	6	33	8392
7	47	7747	7	69	7848
8	64	7607	8	47	7731
9	69	7567	9	101	7683
10	101	7525	10	64	7558

6.2 用车时间的区间频次统计

用还车时间减去借车时间可以得到用户的用车时间，精确到秒。

我们首先分析用车时间的范围，通过统计全部 637336 条记录，发现最短用车时间为 2 秒，最长用车时间为 6601.5 分钟，即 4 天 14 小时 1 分 30 秒。我们采用 t 来表示用户的用车时间，以 100 分钟为间隔首先初步统计各用车时间区间内的用车频次，如表 4 所示。

表 4:用车时间的区间频次统计

t 区间	[0,100)	[100,200)	[200,300)	[300,400)	...	[6500,6600)	[6600,+∞)
频率	0.9967	0.9992	0.9996	0.9998	...	1.0000	1

从表 4 中可以看出，虽然用车时间的范围较大，但是 $t < 200$ 所占的比例为 0.9992，说明绝大多数用车时间小于 200 分钟。因此我们只对用车时间小于 200 分钟的情况进行统计。此外，用车时间小于 1 分钟的情况通常为用户在同一站点迅速借车并还车，和用户真正使用公共自行车的情况不同，因此 $t < 1$ 的情况也不予统计。

统计用户用车的时间得到用户用车时间的均值为 15.5907 分钟，标准差为 11.6429 分钟。以 1 分钟为单位统计用户用车时间的频次，得到图 2。观察图 2 可以发现，当用车时间较短的时候，用户用车的频次随时间的增长而增长，在约 8 分钟左右达到最大值，20 天内约有 3.5 万次借车时间为 8 分钟，之后用户用车的频次随时间的增加而减少。为了定量刻画用户用车时间的分布以及分布产生的机理，我们构建了一个效用函数模型，如 6.3 节所示。

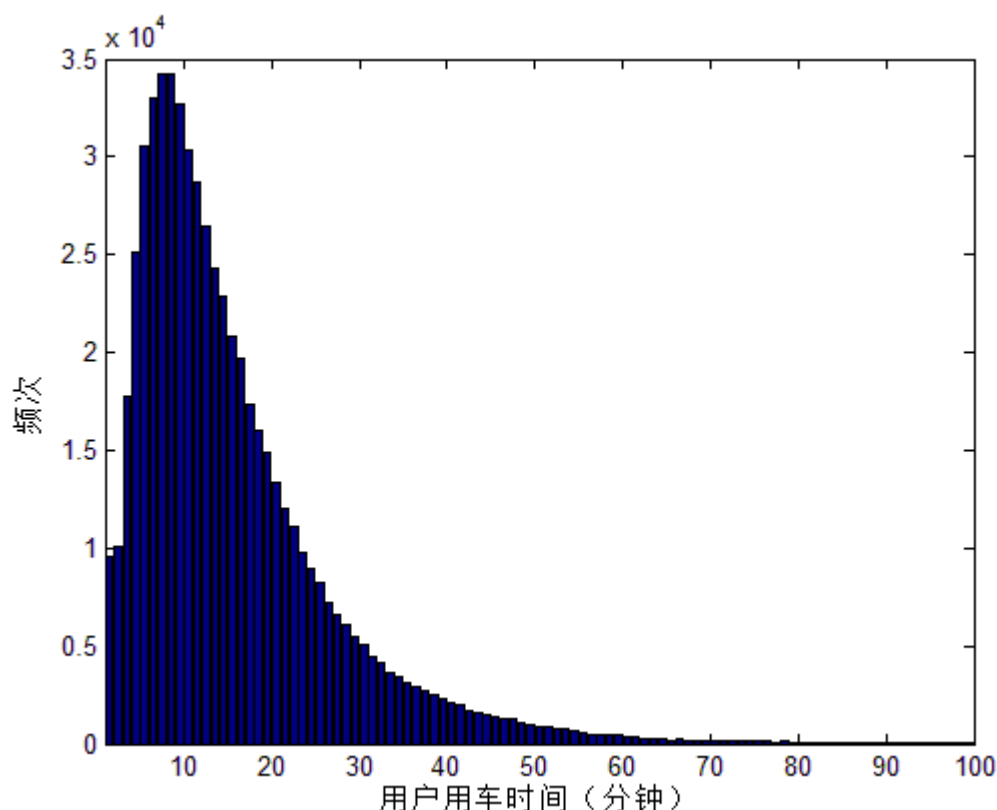


图 2：用户用车时间的频次分布

6.3 效用函数模型

公共自行车的推广主要针对的人群有三类：短距离出行者、长距离换乘出行者和非常规目的出行者等。短距离出行者具体是指在自行车的出行适宜的范围（5km）内，用公共自行车的单一交通方式完成的出行的，包括通勤、购物以及娱乐为目的的交通出行者。长距离换乘出行者指使用公共自行车作为公共交通的一个换乘工具的日常通勤者或者节假日出门娱乐、探访亲友或者购物的人群。非常规目的出行者是指以健身或者旅游为目的的出行者。

用户使用公共自行车出行与否显然和出行的时间相关，出行时间间接反映了出行距离的远近，自行车是一种人力驱动的出行交通工具,在短时间条件下,出行者体力充沛,其速度优势得到体现,但是在时间超过一定的程度,其速度优势就完全暴露。因此出行时间是自行车方式选择的重要因素。表 8 反映了统计资料中披露的部分城市自行车出行时耗分布。从表 8 来看自行车交通的出行时间大部分都控制在在 30 分钟以内。绝大部分城市自行车出行时间控制在 20 分钟以内的比例达到了 60%以上，时间控制在 30 分钟以内的比例都达到了 85%左右（郑州除外），出行时间超 30 分钟的出行比例较小，我们可以认为,自行车的最佳出行时耗在 30 分钟以内。^[3]

表 8：部分城市自行车出行时耗分布（%）¹

出行时耗	<10min	<20min	<30min	<40min	<50min	统计年份
郑州	4.2	42.5	62.3	89.2	93.8	2000
苏州	18.4	62.7	81	87	94	2000
无锡	28.1	65.9	87.1	93.3	96.9	1996
常熟	40.2	80.8	95.6	96.8	97.7	2001
昆山	7.4	62.4	84.8	96.5	97.4	2001

我们构建了一个模型来描述用户用车时长的分布规律。我们假设用户需要出行的时长是一个随机变量，用 T 来表示，并用 $p(T=t)$ 表示 T 的概率密度函数。用户根据自己的出行需求选择要采用的交通工具。显然，用户选择出行交通工具需要考虑一些因素，例如出发起点、出发终点、出行距离、出行用时、出行时段以及交通路况、天气等一系列因素。用户会综合考虑这些因素以确定出行所采用的交通工具。

我们引入效用函数来描述用户选择自行车出行的偏好。效用函数是对用户使用公共自行车获得的综合收益的量化，显然，当收益较高的时候，用户更倾向于使用公共自行车。设用户采用公共自行车出行的效用函数为 $U(t)$ ，则可以设用户使用公共自行车的条件 S_1 为效用函数大于某一阈值 δ ，这一阈值可以视为用户选择出行方式的保留效用。如式(6.2)所示：

$$S_1 : U(t) > \delta \quad (6.2)$$

由于除了 t 之外其他因素也可能影响用户使用公共自行车的效用，所以 $U(t)$ 应该是以 t 为参数的随机变量。为了简化起见，设效用函数是和距离相关的确定部分 $u(t)$ 与一随机变量的和，即

$$U(t) = u(t) + n \quad (6.3)$$

式(6.3)中， $u(t)$ 为 t 的函数， n 是一个随机变量，用以刻画影响用户选择的随机因素。由于随机因素可以看成许多正、负的随机分布的综合效应，根据中心极限定理，我们可以假定 n 服从正态分布，即 $n \sim N(\mu, \sigma^2)$ 。

将式(6.3)代入式(6.2)中可得

$$\begin{aligned}
S_1 : U(t) &> \delta \\
\Leftrightarrow S_1 : u(t) + n &> \delta \\
\Leftrightarrow S_1 : \frac{u(t) - \delta + \mu}{\sigma} + \frac{n - \mu}{\sigma} &> 0 \\
\Leftrightarrow S_1 : u_0(t) + n_0 &> 0
\end{aligned} \quad (6.4)$$

其中， $u_0(t)$ 和 n_0 分别为归一化之后的确定效用函数分量和随机效用函数分量，并且有 $n_0 \sim N(0,1)$ 。用随机变量 Y 来表示用户使用公共自行车出行的用时。用户选用公共自

¹ 数据参见单晓峰（2007）：城市自行车交通合理方式分担率及其路段资源配置研究，东南大学博士论文，2007

行车出行需要满足两个条件，第一，用户需要出行的时间达到一定水平；第二，在此基础上，并综合各种其他因素，用户最终选用公共自行车。因此， Y 的密度函数可以表示为：

$$p(Y=t) = p(S_1)p(T=t) \quad (6.5)$$

由于 $n_0 \sim N(0,1)$ ，故 n_0 的概率密度函数为：

$$p(n_0 = x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (6.6)$$

把式(6.6)代入式(6.5)可得

$$p(Y=t) = p(T=t) \cdot \int_{-u_0(d)}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \quad (6.7)$$

式(6.7)为使用公共自行车出行时间分布的效用函数模型。

6.4 出行时间的效用函数模型求解

下面我们求解用户出行时间的效用函数模型，通过求解模型，可以解出效用函数的表达形式和用户出行时间的分布。设 $Q(x)$ 为标准正态分布的累计分布函数，即

$$Q(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{s^2}{2}} ds$$

根据对称性我们可知 $\int_{-u_0(d)}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = Q(u_0(t))$ ，则有：

$$p(Y=t) = p(T=t) \cdot Q(u_0(t)) \quad (6.8)$$

解出 $Q(u_0(t))$ 可以表示为：

$$Q(u_0(t)) = \frac{p(Y=t)}{p(T=t)} \quad (6.9)$$

由于 $Q(u_0(t))$ 为单调非减连续函数，我们必可以求解其反函数，从而解出用户的效用函数表达式：

$$u_0(t) = Q^{-1} \left(\frac{p(Y=t)}{p(T=t)} \right) \quad (6.10)$$

我们可以利用样本数据的统计结果近似代替 Y 的分布，即用样本的出现的频率分布来代替随机变量的密度函数，如图 1 所示。因为数学期望在样本趋近于无穷大的时候收敛于数学期望，因此这种近似是有效的。

对于用户需要出行的时间 T ，其分布是独立于该问题的，并且只要任给一个 T 的分布就可以解出一个效用函数的表达式， T 分布的具体形式并不影响解决问题的方法的一般性。因此我们可以假设一种 T 的分布。常识性的知识表明用户需要出行某一时间的概

率随时间的增加而下降，即用户有更大的可能在更近的范围内活动。因此不妨设 T 服从指数分布，即 T 的密度函数为：

$$p(T=t) = \lambda e^{-\lambda t} \quad (6.11)$$

可以设 $\lambda=1/100$ 。由于仅统计 100 分钟以内的出行时间，因此需要对 T 的概率密度函数进行归一化，归一化之后概率密度函数如图 3 所示。

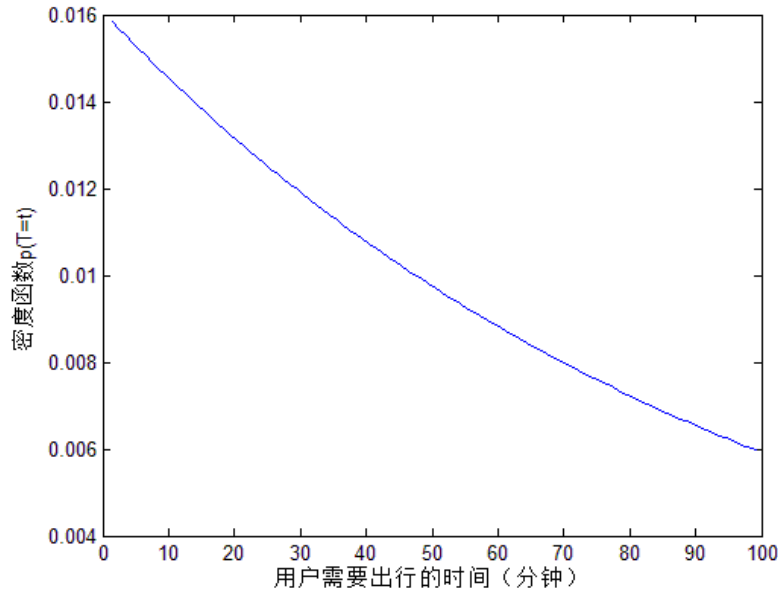


图 3：用户需要出行的时间 T 的密度函数

$Q^{-1}(x)$ 没有解析解，但是正态分布的累积分布函数有使用数值方法计算出的函数值表，可以使用反查该表的方法来得到 $Q^{-1}(x)$ ，其函数图象如图 4 所示。

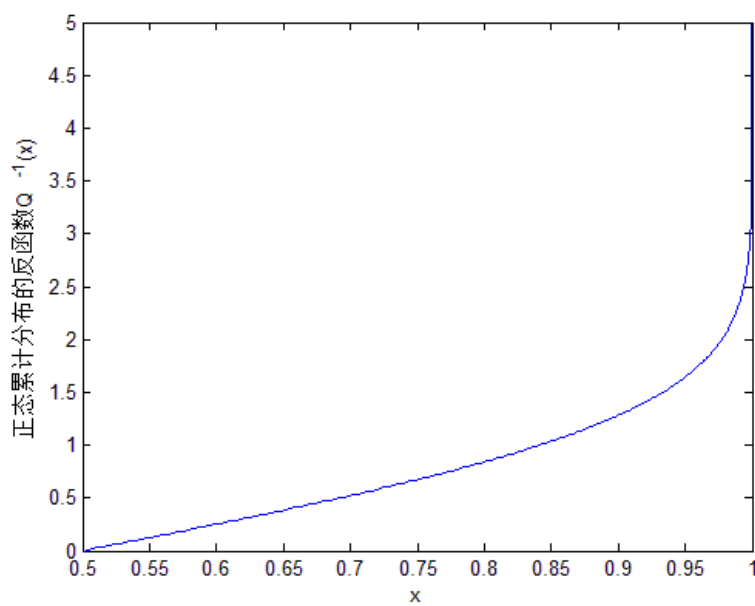


图 4：正态累计分布的反函数 $Q^{-1}(x)$ 的函数图象。

综上所述, 将 $p(Y=t)$ 、 $p(T=t)$ 和 $Q^{-1}(x)$ 的值代入式 6.10, 即可解得效用函数 $u_0(t)$, $u_0(t)$ 的函数图象如图 5 所示。

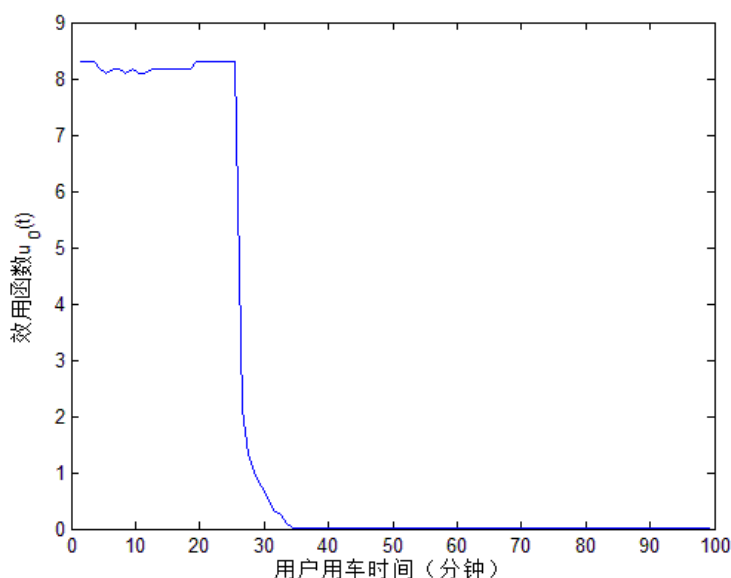


图 5: 效用函数 $u_0(t)$ 的函数图象

从图 5 中可以看出, 用户选择公共自行车出行的效用函数近似为一个矩形脉冲函数, 其拐点约为 28 分钟。当出行时间小于 28 分钟时, 用户更倾向于选择公共自行车出行; 在 28 分钟之后, 用户倾向于选择其他的交通工具, 仅在随机因素的作用下选择公共自行车。

综上, 即可用效用函数 $u_0(t)$ 以及式 6.7 来描述用户选择公共自行车出行时间的分布。

6.5 借车卡累计借车次数统计分析

在这一部分的分析中, 我们按照借车卡区分不同用户个体, 对用户群体的需求模式进行探索。根据温州市鹿城区借车卡办理方法规定, 办卡采用实名制, 一张借车卡, 只能借一辆公共自行车。^[2]因此我们假定用户和借车卡之间是一一对应的关系, 即各位用户各采用一张借车卡。通过计算 20 天中各天使用公共自行车的不同借车卡 (即借车人) 数量, 计算统计数据中出现过的每张借车卡累计借车次数的分布情况, 我们可以看出借车人的偏好规律。

类似 6.1 部分中的分析, 我们采用 MATLAB 统计每日使用公共自行车的借车人数量, 如表 9 所示:

表 9: 各天使用公共自行车的借车人数量

天数 j	1	2	3	4	5	6	7	8	9	10
借车人数量	16840	17462	9671	14677	17985	18708	18887	10600	7038	4152
天数 j	11	12	13	14	15	16	17	18	19	20
借车人数量	15097	18194	19531	19462	18673	11335	15412	15311	19188	20020

通过累计各个借车人在 20 日中累计借车次数，我们做出如下频率分布图：

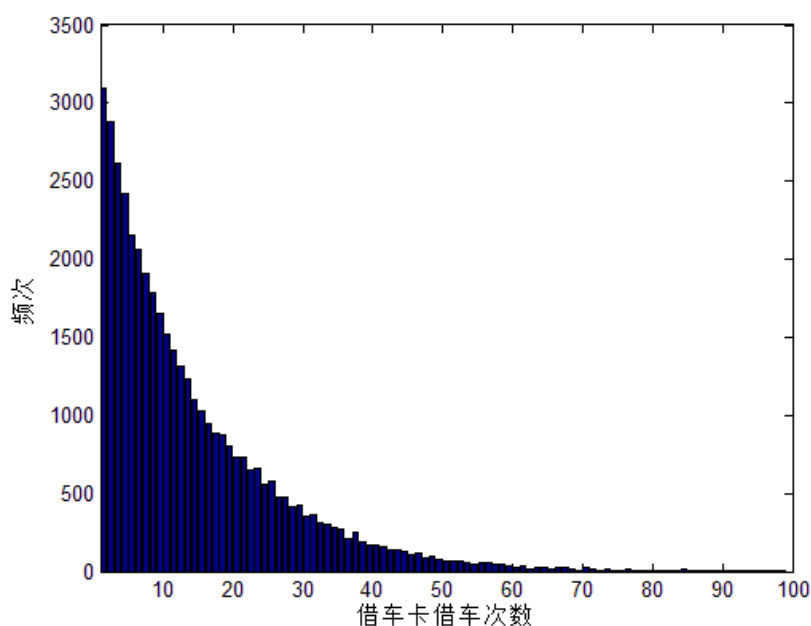


图 6：累计借车次数

横坐标表示各个借车卡累计借车次数，纵坐标表示各个次数出现的频数。从图 6 中我们可以看到，大多数借车人的借车次数在 30 次以下，随着借车次数的逐渐增加，相应借车次数的借车人数逐渐下降。我们依然采用简单统计方法分析各用车频次：

表 10：累计借车次数的区间频次统计

次数区间	[0,50)	[50,100)	[100,150)	[150,200)	[200,250)	[250,300)	[300,350)
累计频率	0.978202	0.999295	0.99989	0.999956	0.999956	0.999978	0.999978
次数区间	[350,400)	[400,450)	[450,500)	[500,550)	[550,600)	[600,+∞)	
累计频率	0.999978	0.999978	0.999978	0.999978	0.999978	1	

我们发现存在离群值，样本点 99.2% 的用户借车次数都在 100 次以下，只有极少数人超过 100 次，根据常识判断这些值可能是统计错误，我们作为异常值去除。观察频数直方图，我们发现其分布近似于指数分布。基于这一假设，我们利用 MATLAB 拟合，得到拟合图如图 7 所示：

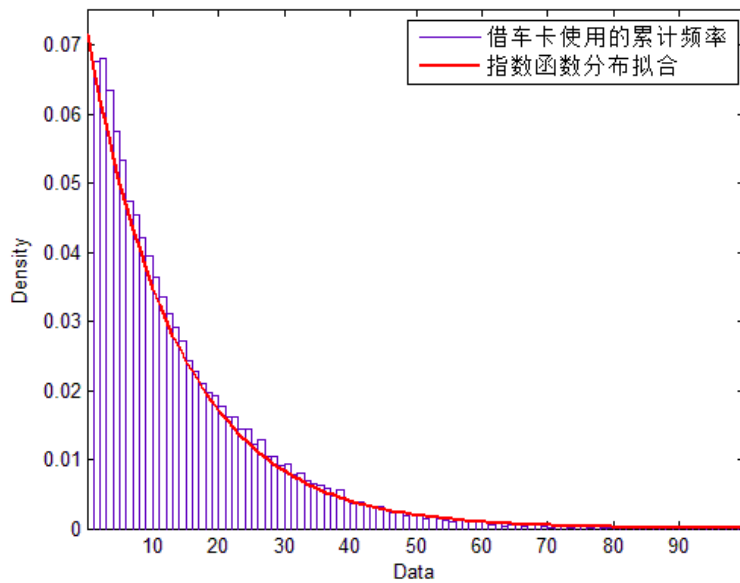


图 7：指数分布拟合图

从表 11 中可以看到，拟合结果良好，标准差 0.065，表明指数分布能够较好地拟合该借车频数的累计概率分布。

表 11：拟合结果

分布 (Distribution)： 指数分布 (Exponential)			
对数似然率 (Log likelihood)： -164938			
区间 (Domain)： $0 \leq y < \text{Inf}$			
均值 (Mean)： 13.9364			
方差 (Variance)： 194.225			
参数估计			
	系数	估计值	标准差
	Parameter	Estimate	Std. Err.
	mu	13.9364	0.0654207
参数估计协方差			
Estimated covariance of parameter estimates:			
	mu	0.00427987	

七 地理信息数据的获取

7.1 公共自行车站点的坐标信息

对城市公共自行车的空间使用情况进行统计，需要已知城市地理信息以及公共自行车站点的地理位置信息。题目附件 2 中给出的鹿城区公共自行车站点分布图，仅描述了自行车站点的分布，并没有每个自行车站点的具体坐标。我们在温州市鹿城区公共自行车管理中心网站上找到每个自行车站点的位置，再在电子地图服务提供商（例如谷歌地图）网站上查询出该站点位置的坐标^[4]。为了表示方便，我们定义北纬 $28^{\circ}2'17.68''$ ，东

经 120°38'19.13”，以正南为 y 轴正方形，以正东为 x 轴正方形，以 1m 为单位建立直角坐标系。由于鹿城区面积较小，故可以不考虑地球曲率。将各站点的地理坐标映射到直角坐标系下，得到站点的坐标如表 12 所示，完整的表格参加附录中。

表 12：公共自行车站点的坐标（部分）

站点序号	站点名称	y 刻度	x 刻度
1	科技馆	8114.064	4378.759
2	温州大剧院	8255.842	4209.716
3	吴桥路加油站	3838.912	3882.536
4	银泰百货	4515.084	2475.662
5	星河广场	2453.85	916.104
6	绣山卫生院	8539.398	3735.305
7	市政府西	7972.286	3713.493
8	市政府东	8217.671	3697.134
9	小南门立交桥	3718.946	2333.884
10	市九中	4340.588	1134.224
11	鹿城区审批中心	3528.091	3909.801
12	桥儿头公交站	5414.829	3511.732
13	公共自行车中心	3757.117	3740.758
14	南浦医院	5453	4209.716
15	温州建国医院对面	4596.879	5005.854
16	金色家园	4580.52	4187.904
17	区政府西	3549.903	1706.789
18	区政府东	3626.245	1706.789
19	开太百货	4324.229	2044.875
20	南浦桥	5529.342	3762.57

将这些公共自行车站点的坐标标记在地图上，如图 8 所示。



图 8：公共自行车站点的在地图上的位置

7.2 城市道路地理信息

计算两个公共自行车站点时将会使用到城市道路信息。城市道路信息可以使用由现有的 GIS 信息，也可以通过图片格式的地图获得，使用图片格式的地图获得道路信息的方法主要包括颜色替换，形态学处理，道路生长细化等等，最终获得城市道路信息，用于两地之间的实际道路路径求取。颜色替换之后的城市地图如图 9 所示。

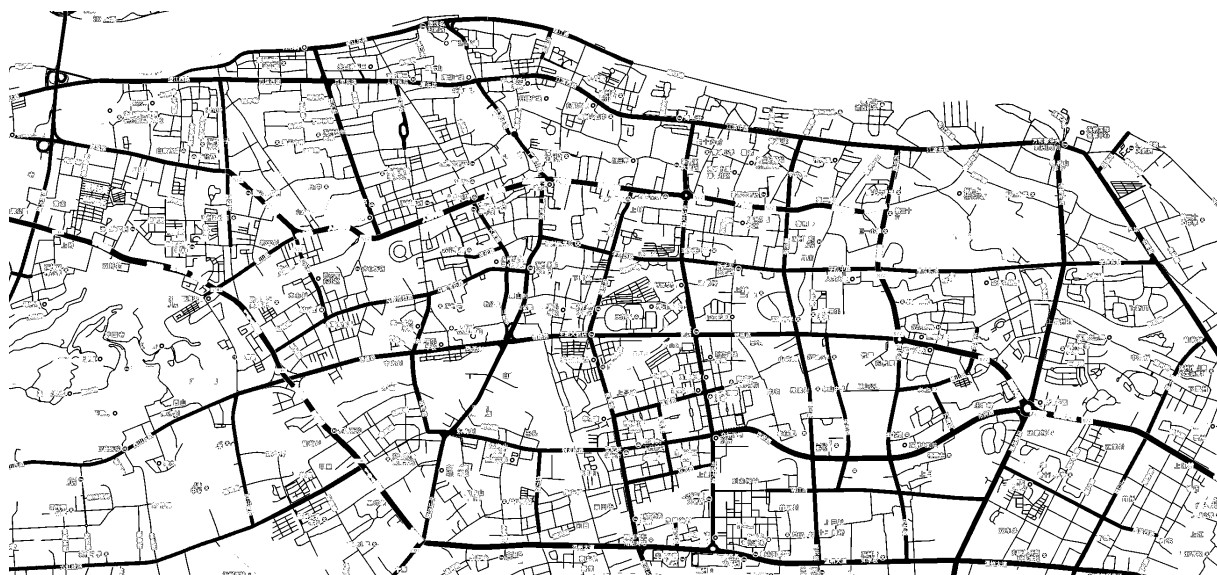


图 9：经过颜色替换得到的包含道路信息的城市地图

八 公共自行车时间、空间使用情况统计分析

8.1 站点距离的界定

对公共自行车站点的空间规划研究是建立在合理界定站点距离的基础上的。在本研究中，我们主要采用三种方法来对公用自行车站点间的距离进行刻画。

1) 欧氏距离

度量地理空间中的距离最简单的方法是欧氏距离（Euclid Distance），也称欧几里得度量、欧几里得距离，它是在 m 维空间中两个点之间的真实距离。本研究基于鹿城区的城市交通地图，属于二维空间中的欧氏距离，即为两点之间的直线段距离。

2) 直角边距离

我们注意到，欧氏距离虽然计算方法简单，但是不符合公共交通的一般特点。在城市规划和道路建设中，两个站点往往不能通过直线连接相互连通，因此，采用近似直角边距离和的方式近似求距离对现实的拟合度更高。但是，这种方法假设道路全部是“横平竖直”的，这也与实际有一定的差异，会存在计算偏差。

3) 城市道路实际距离

这一方法建立在实际的城区道路网络基础上，衡量的是站点之间城市道路的真实距离，因此是最为精确的。其优点是对空间距离的衡量拟合度最高，但缺点是操作难度较大。

8.2 借/还车的最短距离和最长距离

分别统计每天所有站点的借车和还车频次，并将两者的加和最大的一天作为公共自行车使用最大的一天。经统计发现，2012 年 11 月 20 日公共自行车的使用次数最大，借还总频次为 84441 次。本章以下问题均针对此日的数据进行讨论。

根据 8.1 节定义的距离计算方法，分别采用欧式距离法和直角边距离法计算各站点之间的距离。根据数据记录，有的站点之间没有连接，将这些没有连接站点之间的距离视为 0。分别统计不同站点之间距离的最大距离，并对其进行排序，即可得到用车借还站点之间的最短距离和最长距离（非零距离，并且只统计时间在 1 分钟以上的借还车情况），如表 13 所示，完整的表格参见附录。

表 13：借车站点的最近与最远还车站点及距离（部分）

借车 站点	欧式距离				直角边距离			
	最近还 车站点	最近还车 距离（米）	最远还 车站点	最远还车 距离（米）	最近还 车站点	最近还车 距离（米）	最远还 车站点	最远还车 距离（米）
1	2	220.6	123	5223.4	2	310.8	18	7159.8
2	28	472.3	142	6482.8	8	550.8	142	8081.3
3	11	312.0	148	4499.8	11	338.1	111	5578.4
4	71	269.4	72	5026.0	71	376.3	67	6472.7
5	175	282.1	114	8008.0	175	398.1	114	9602.7
6	8	324.0	140	6364.0	8	359.9	140	7067.1
7	8	245.9	131	5520.0	8	261.7	38	6549.1
8	7	245.9	133	5656.3	7	261.7	133	6778.1
9	43	123.0	73	5979.1	129	141.8	72	7399.7
10	21	239.0	115	5272.1	21	321.7	1	7018.0
11	3	312.0	116	6214.2	3	338.1	116	7001.7
12	78	208.4	160	4352.7	78	229.0	160	5720.2
13	11	284.7	72	5490.8	11	398.1	72	5954.7
14	79	359.9	157	5830.9	79	365.4	157	8010.5
15	75	531.2	144	4517.9	75	730.7	144	6309.1
16	166	319.3	58	4653.4	166	359.9	138	5692.9
17	18	76.3	76	5564.6	18	76.3	76	7028.9
18	17	76.3	114	6707.1	17	76.3	114	7639.7
19	71	215.2	113	5940.1	71	245.4	113	6925.3
20	23	270.9	160	4560.9	12	365.4	160	6085.5

8.3 借/还车频次最高站点

通过统计可知，2012 年 11 月 20 日的借车频次最高的站点为 42 号站点，还车频次最高的站点为 56 号站点。对这两个站点统计借还车时刻分布以及用车时长分布。对于用车时刻，将一天中的 24 小时划分成以 0.2 小时为间隔的若干时段，分别统计落入每个时段的借还车频次，即可用来反映借还车的时刻分布，统计的结果如图 10 所示。

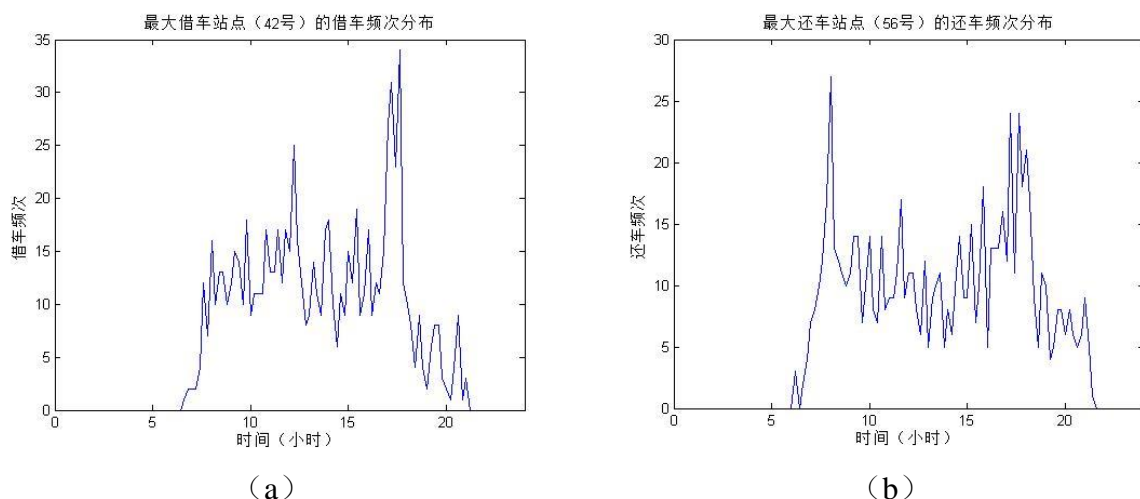


图 10: 最大借还车站点的借还车频次时刻分布

通过观察可以发现，借还车的时刻分布呈现一定的分布规律，例如 42 号站点的借车频次在 17:30 分左右出现峰值，在峰值左右的频次大于其他部分的频次，说明在该时刻前后借车行为比较集中；56 号站点的还车频次有两个峰值，分别出现在 8:00 和 17:30 左右，说明该站点在早晨和晚上都出现了还车比较集中的现象。

按照 6.2 节的方法来统计两个站点的用车时长，得到用车时长的分布如图 11 所示。根据 6.3 和 6.4 节的方法求解效用函数分布模型，得到的效用函数的图象如图 12 所示。

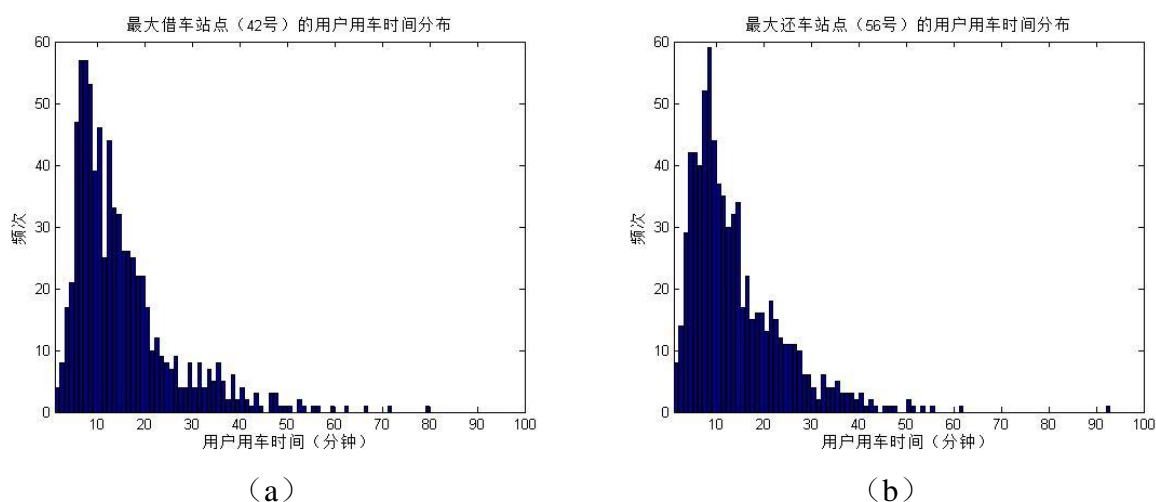


图 11: 最大借还车站点的用户用车时长分布

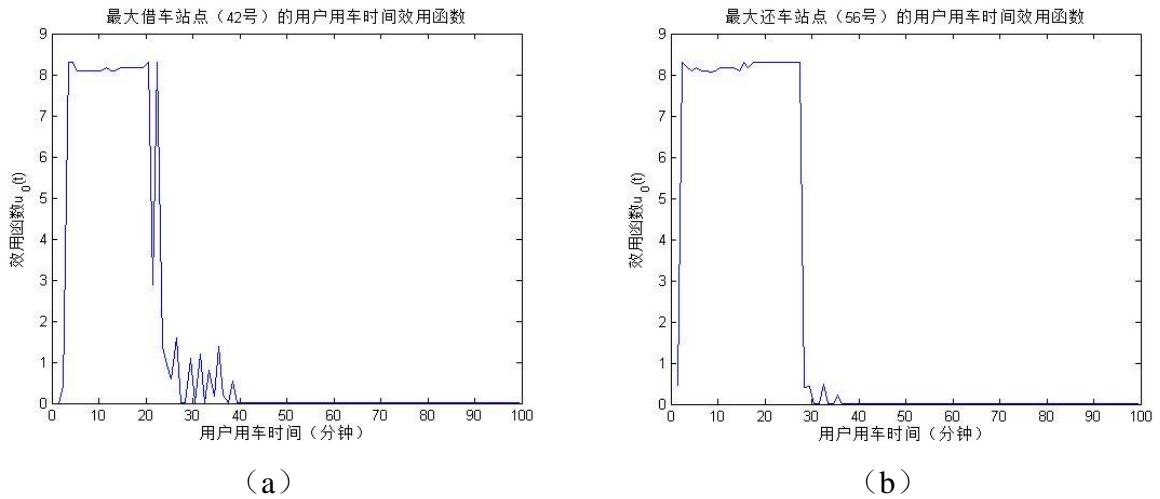


图 12：最大借还车站的用户用车效用函数

从图 11 和图 12 可以看出，42 号和 56 号站点的用车时长分布以及用户选择公共自行车的效用函数与 6.2 和 6.4 节中的总体情况大致相同，说明效用函数模型具有较好的稳定性。

8.4 峰值搜索算法

由 8.3 的分析可知，图 10 中的借车与还车时刻分布具有一定的规律，表现为存在一定的显著高于平均值的高峰时段。我们希望找到每个站点的这些峰值的位置，用来评价公共自行车系统的运行状况。

峰值一定是频率密度函数的极大值，但是极大值并不一定是峰值。随机因素造成的波动同样可能在频率密度函数中产生极大值。可以使用均值滤波的方法减弱随机因素造成的影响。均值滤波即使用某一点附近的一个小区间内的点的平均值来代替这个点的数值。图 13 为借还车时刻频率密度函数均值滤波的效果。

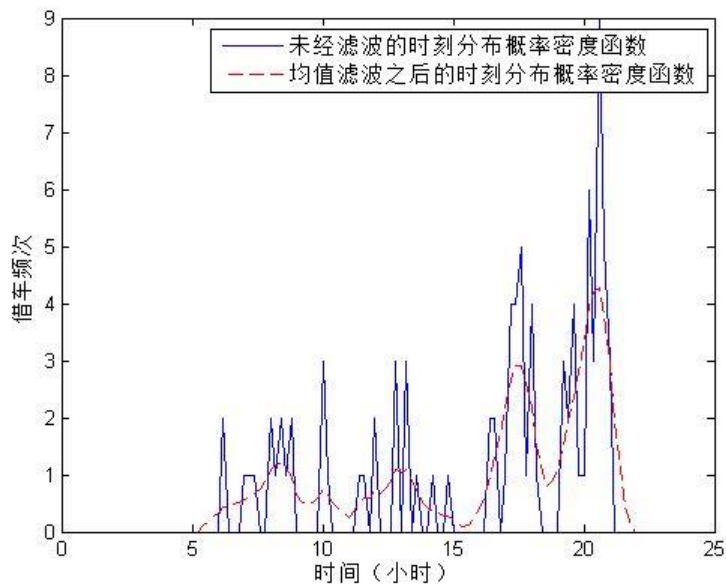


图 13:用车时刻分布频率密度函数的均值滤波效果

经过均值滤波之后，频率分布函数变得平滑了。对滤波之后的结果求极大值，定义当极大值大于整个取值范围的 60% 才为峰值时刻。当多个峰值之间的距离小于 2 个小时的时候，将它们视为同一个峰值，并用平均的时间代替峰值时间。使用 MATLAB 实现峰值搜索算法，其结果如图 14 所示。

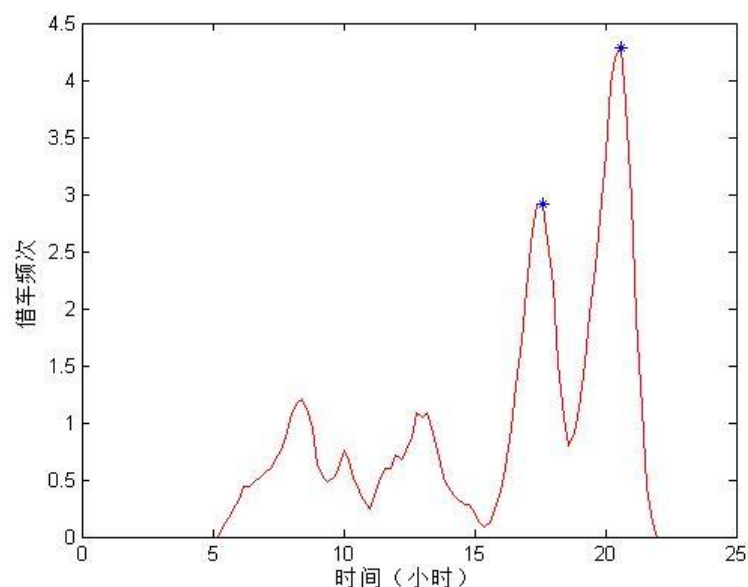


图 14：峰值搜索算法的运行结果

8.5 用车高峰时段统计与归类

根据 8.4 节所述的峰值搜索算法对站点的借还车高峰时刻进行统计。得到各个站点的借车和还车高峰数据如表 14 所示，完整的表格见附表。

表 14：各站点的借车和还车高峰时间表

站点序号	借车高峰	还车高峰
1	17:36, 20:36	19:23
2	8:12, 16:23	8:12, 15:35
3	17:11	8:12
4	7:48, 13:35, 14:23, 17:11, 19:11	9:12, 11:24, 12:47, 16:00, 17:36
5	17:11	10:12, 10:48, 18:00, 19:11, 19:48
6	8:12, 17:36, 18:36	17:23, 20:23
7	17:48	8:12
8	17:36	8:24
9	8:00, 13:47, 16:11, 17:48	8:24, 11:00, 15:47, 17:48
10	8:00, 15:24, 17:11	8:24, 17:23
11	17:11	8:12
12	7:48, 17:23	17:36
13	10:00	8:24, 10:00

14	7:48	7:48, 17:23
15	8:00, 10:36, 17:11	8:12, 17:23, 18:00
16	7:24, 12:35, 17:23	17:48
17	9:36, 17:23	8:12, 17:23
18	17:23	8:24, 17:23
19	8:00, 13:00, 15:35, 17:23, 19:23	8:24, 11:36, 12:23, 13:00, 15:47, 17:23, 19:11
20	8:00, 17:36	17:48

(续表)

8.6 高峰时段的聚类分析

可以使用聚类分析的方法对不同站点的高峰时段特征进行分类。聚类分析的思路是将参数空间中的点分成若干类,使得每类中的点的差异最小。经典的聚类方法如 K-means 算法,通过构造一个类中心,通过不断迭代的方法改变类中心的距离以及分类结构,使得同一类中的点距离类中心的距离之和最小。

由于不同的站点具有的峰值数目不同,即所在的空间的维数不同,因而无法直接使用 K-means 算法进行聚类,此时可以将不同峰值数目的点折算成具有相同数目的等效峰值的点。例如,设等效峰值数目为 20 个,对于不足 20 个峰值的站点,需要将一些峰值时间复制成若干个。峰值时间复制的数目与峰值频次的大小成正比。

经过折算之后,即可使用 k-means 算法对 20 维等效峰值时刻空间进行聚类,使用欧式距离作为空间中点的距离,设置类数目为 5 个。得到结果后将同类的站点使用相同的颜色标记在地图上。借车站点高峰时刻聚类分析的结果如图 15 所示,还车站点高峰时刻聚类分析的结果如图 16 所示。

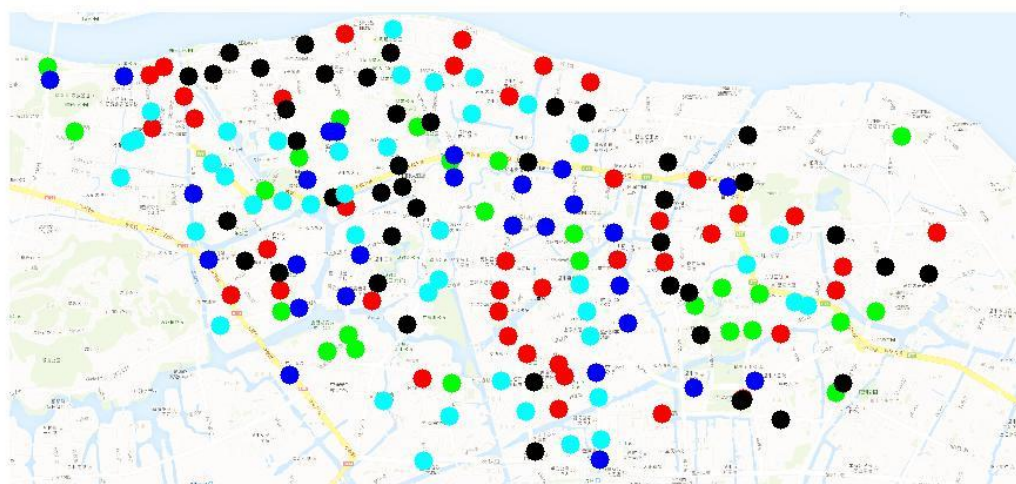


图 15: 借车站点高峰时刻聚类分析的结果

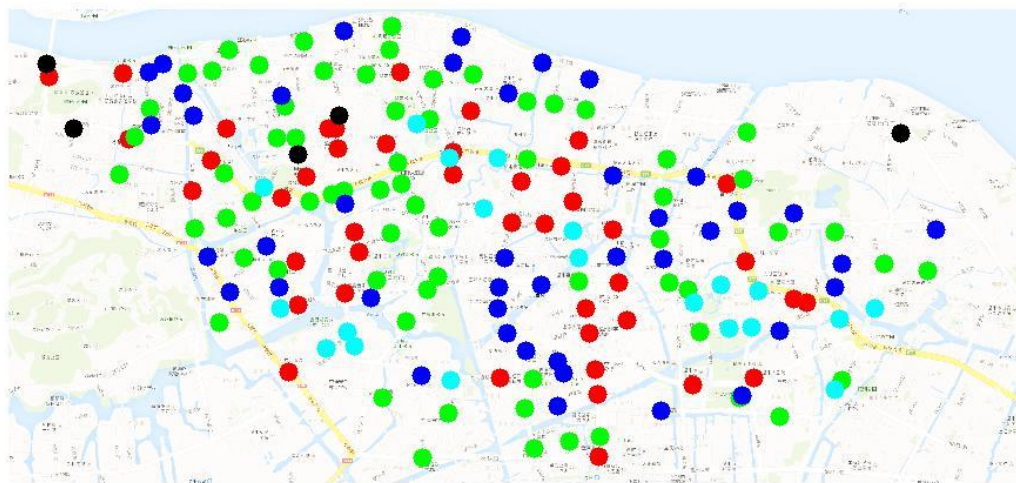


图 16: 站点高峰时刻聚类分析的结果

九 其他信息和公共自行车系统设置评价

9.1 潮汐现象

我们观察上面的计算和统计结果，注意到最大借还车站点的借还车频次时刻分布（即图 10）显示，42 号站点的借车频次在 17:30 分左右出现峰值，高于最大自行车提供量即 30 辆，出现供给不足的“晚高峰”，而 56 号站点公共自行车也存在早高峰和晚高峰两个较为明显的峰值，供给非常紧张，这一现象也符合自行车交通的基本规律。而其他时段则不同程度出现低谷（例如，用户需求量低于 5 辆）等现象。不同的时间区间用户的用车需求波动较大，体现出“潮汐现象”的特征。^[1]

潮汐现象主要是指租车或还车不能持衡，出现“供不应求”或者“供过于求”的不均衡现象，在一定时间段内租车量持续高于或者低于还车量，并且以一定时间为循环周期重复出现，严重影响了系统的运行效率。一系列新闻报道已经从侧面证实了这一困难，潮汐现象是自行车系统的设计和运行中普遍存在的规律。¹从我们的统计分析来看，部分地段服务点的公共自行车在使用过程中有明显的峰值和低谷，基于用户的频率分布也有显著的差异，有较为明显的自行车需求和潮汐现象。表明鹿城市的公共自行车系统的效率依然有提高空间。

9.2 潮汐现象的表现

潮汐现象主要表现为：①租赁点还车数大于借车数，造成停车桩已满而无法还车的现象；②租赁点的借车数大于还车数，造成自行车全空，没有自行车可以租借的现象^{[5][6]}。这两种情况均影响了公共自行车的可持续使用。潮汐现象有一个特点就是当天内出现借还逆转，并且以日为周期重复出现。

¹ 参见新闻报道：“湖东公共自行车频现“潮汐”现象”<http://jsnews.jschina.com.cn/system/2012/04/09/013101671.shtml>，“市民指河西公共自行车早晚高峰借车难还车难”<http://jiangsu.sina.com.cn/news/m/2013-05-23/074956286.html> 等

城市边远地带的大型居住区、大型的工业园区、公交枢纽站或转运站、商务集中区等类型的服务点更容易出现潮汐现象，这些地区的用地功能复合度较低，城市居民出行需求趋同，出行时间集中，短时间内对车辆的需求量大而波动频繁是潮汐现象形成的主要原因。在本文的研究框架下，温州市鹿城区位于浙江省东南部、温州市中部，是温州市的政治、经济、文化中心，潮汐现象相对较弱；而城市公共自行车系统在其他城市的推行中，不可避免地会遇到不同程度的潮汐问题。

9.3 潮汐现象的解决

解决潮汐现象的最好方法是实现自行车的高效率、低成本调度，提高公共自行车流转量，减少因空架或满架造成的阻碍使用的情况，及时满足供给、需求的相对均衡，保证市民的正常使用。目前我国公共自行车系统的应用尚处于初级阶段，调度模式落后，主要利用人工巡查的方式对服务点进行监控。目前的调度主要是静态调度，一般选就近的原则进行服务点间的调配，未形成系统科学、结构清晰的调度模式。这也是亟待解决的问题。在 10.2 部分中我们会有进一步的阐释。

十 公共自行车服务系统的其他运行规律和政策建议

在前面的研究中，我们从公共自行车的需求规律、用户的效用水平和站点的地理空间配置等几个角度对公共自行车服务系统的规律进行了分析和评价。除这些方面之外，公共自行车服务系统作为一个复杂、庞大的体系，还涉及许多其他方面，体现出一些独特的运行规律。我们选取有代表性的两个方面简要分析。

10.1 站点选址的社会福利最大化问题

城市公共自行车系统建设不仅是解决城市交通问题的重要手段，而且作为基础设施建设的重要组成部分，这一系统的平稳、可持续运行还有赖于良好的运营商业模式和策划。鹿城新闻网显示，鹿城要推行公共自行车系统，鹿城捷信小额贷款公司 20 名股东经过商议，决定承担前期建设的所需全部资金 1200 万元，并出资 500 万元设立“公共自行车发展专项基金”，用于下阶段的扩建、维护和管理等后续费用。¹政府项目的市场化运作已经成为当今社会公共品提供的主流趋势。作为公共物品，公共自行车的站点设置应当建立在政府财政预算的基础上实现收益和社会福利最大化。我们构建简单的社会福利模型对这一问题进行分析。

社会福利最大化模型受制于政府的财政预算约束，最大化社会收益和投资公共自行车企业收益。相关的文献研究参见何博（2009）^[6]的站点选址模型：

¹参见鹿城新闻报道：<http://www.wzbicycle.com/article.asp?Action=View&ArticleID=260&Catalog=2>

$$\begin{aligned}
\max W &= \sum_{i=1}^s \left(\frac{365P_i}{\gamma} N \right) + \sum_{j=1}^l (g_j Y_j N + q_j Y_j N \sigma) - \sum_{j=1}^l (t_1 X_j + t_2 Y_j + t_3 Y_j N) \\
s.t. & \sum_{j=1}^l w_{ij} Y_j \geq 1 \\
& \sum_{j=1}^l w_{ij} Y_j \leq T \\
& Y_j \in \{0,1\}
\end{aligned} \tag{10.1}$$

s, l 分别为需求点的数目、备选租赁点的数目；

g_j, q_j 分别为备选租赁点 j 每年的商业收入、每年的政府补贴；

t_1, t_2, t_3 分别为基础设施单价、每个租赁点建设费用以及运营费用；

σ 为运营模式分类系数哑变量, 完全将市场化模式取 0, 其他模式取 1；

γ 为公共自行车高峰小时系数；

T 为租赁点个数的最大规模；

X_j 为备选租赁点 j 的基础设施数量；

Y_j 如果在 j 处建设租赁, 取值为 1, 否则取值为 0；

P 为需求点 i 的出行者所需要支付的租赁费用；

w_{ij} 如果需求点与备选租赁点的距离小于最大值则取 1, 否则取 0；

对于基于预算约束的规划模型求解可以得出社会福利最优解。更广泛经济意义上来说, 该机制的设置存在两个维度的问题：①权衡问题 (trade off)：建立更为广泛的自行车站点的福利增进、经济利益与建造租赁地点成本和财政补贴、预算约束的权衡。这一机制设计可以用行为博弈论的方法求解。②委托-代理问题：以政府为主导的城市公共自行车系统建设市场化, 面临作为委托方 (principle) 的政府与作为代理方 (agent) 的承建企业之间的利益分歧, 政府的目标是社会福利最大化, 而企业是以股东利益最大化为目标的组织, 其主要目标是利润最大化, 这可能与社会最优的目标之间存在一定冲突。

10.2 公共自行车系统运行的调度机制

公共自行车“想借时难还亦难”问题可以通过有效的自行车调度来解决。在推广城市公共自行车服务系统的同时, 必须建立起配套的调度系统。公共自行车行车系统主要通过后台控制中心和区域调度中心来完成车辆调配。通过各服务点自行车停放情况的跟踪统计, 当车辆数低于某一个阈值时, 启动调度, 并对站点实行实时监控, 保证调度最优化。调度系统主要包括两个方面的作用：①日内调度：每个服务点的公共自行车租赁需求存在季节差异和日内时间段差异；②需求低谷期的车辆入库：当出现极端恶劣天气时, 将多余的车辆进行入库管理能够提高车辆的使用寿命。^[5]区域调度的层级实现方法如下图：

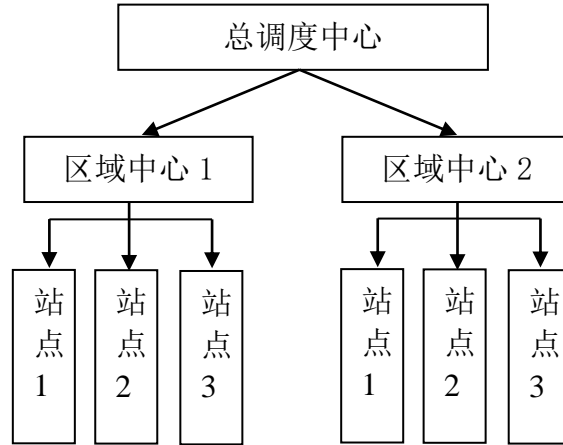


图 17：自行车区域调度系统

通过构建基于模糊时间窗的用户满意度函数、惩罚函数，叶丽霞（2013）^[5]系统地采用了基于 A-Priori 优化的方法和实时再优化方法计算最优调度方式和规模。另外，节约算法、插入算法、遗传算法、模拟退火算法等方法都可以用于解决最优调度问题。

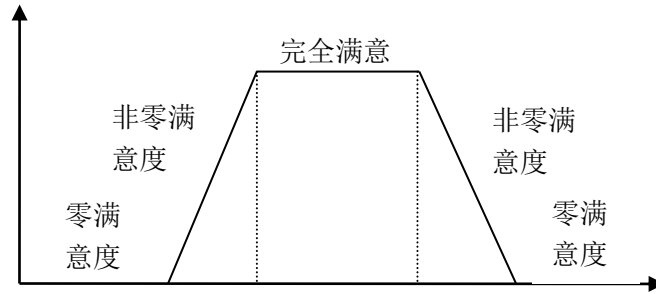


图 18：模糊时间窗的用户满意度区间图

十一 结论

城市公共自行车为代表的新型公共交通计划实施后，城市交通拥堵现象带来了有效的缓解，为“最后一公里现象”找到了行之有效的解答^[7]。良好的公共自行车服务系统设计有助于这一举措的顺利实施和可持续发展。本文建立在温州市鹿城区的公交系统基础上对于城市公共自行车系统的客流量、供求关系和分布规律进行了研究，在总结数据的基础上对公共自行车系统一般规律进行探索，并提出了一些问题和可行的解决方法。本文的结论主要揭示了以下几个方面的问题：

1) 通过原始数据的初步整合处理、剔除异常值，我们求出了借还车频率、借车人分布，并通过拟合方法发现，借车人的借车次数普遍在 30 次以下，并且群体的借车次数的近似服从指数分布。

2) 通过建立用户效用函数，我们可以求解城市居民采用公共自行车出行的偏好程度随出行时间的变化规律。我们发现，用户选择公共自行车出行的效用函数关于出行时间的变化近似为一个矩形脉冲函数，当出行时间少于 28 分钟时，用户更倾向于选择公共自行车出行；在 28 分钟之后，用户倾向于选择其他的交通工具，仅在随机因素的作用下选择公共自行车。

3) 我们采用了三种方法界定站点距离——欧式距离，直角距离和实际地图距离，我们借助于温州市实际电子地图通过 MATLAB 进行相应处理，获得市级道路信息，从

而成功解决了距离的度量问题。

4) 我们的统计分析来看, 部分地段服务点的公共自行车在使用过程中有明显的峰值和低谷, 基于用户的频率分布也有显著的差异, 有较为明显的自行车需求和潮汐现象。另外, 站点设置的社会福利经济学分析以及配套的调度机制设计也是公共自行车系统运营中需要解决的重要问题。

本文的研究对政府建设城市公共自行车租赁系统具有一定的参考价值。公共自行车系统是一个复杂的系统, 在城市的运用过程中会涉及到多个角色的相互作用, 包括政府、企业、出行者, 也会涉及到地理、政治、经济、交通等非常广泛的领域^[8]。而我们的这一研究尚粗浅, 还有许多没有涵盖的部分, 建立在社会福利、用户需求等多方权衡下的最优化的公共自行车系统的构建将是未来可以继续探索的方向。

参考文献

- [1] 鲍娜, 城市公共自行车租赁点选址决策及调度模型研究, 长安大学硕士学位论文, 2012。
- [2] 温州市鹿城区公共自行车管理中心网站, <http://www.wzbicycle.com>, 2013.09。
- [3] 单晓峰, 城市自行车交通合理方式分担率及其路段资源配置研究, 东南大学博士论文, 2007。
- [4] 谷歌(google)地图, <http://ditu.google.cn/maps?hl=zh-CN&tab=wl>, 2013.09。
- [5] 叶丽霞, 城市公共自行车调度系统研究, 南京理工大学硕士学位论文, 2013。
- [6] 何博, 城市公共自行车系统的应用研究, 西南交通大学硕士学位论文, 2012。
- [7] 李正浩, 城市公共自行车租赁站远期发展规模分析 [J][J], 交通节能与环保, 2010, 2: 015。
- [8] 王志高, 孔喆, 谢建华等, 欧洲第三代公共自行车系统案例及启示[J], 城市交通, 2009, 7(4)。

附录

附录一:

1) 6.1 章节中各个站点的每天与累计借车频次

站点	1	2	3	4	5	6	7	8	9	10	11
1	91	148	35	152	124	107	88	15	9	6	169
2	114	112	25	82	135	135	120	41	21	15	70
3	179	188	47	60	194	189	189	73	65	13	84
4	235	286	86	255	243	322	259	112	65	66	315
5	139	165	67	167	161	164	168	53	33	16	158
6	105	100	50	102	117	118	134	47	19	13	89
7	106	109	14	55	143	137	157	34	35	5	49
8	95	80	15	47	106	116	114	36	41	10	39
9	353	365	177	303	400	460	393	167	124	86	325
10	354	359	198	311	366	334	376	165	82	53	323
11	122	132	33	36	141	130	131	45	36	16	63
12	252	249	131	235	210	226	282	108	56	33	223

13	161	145	62	100	116	126	109	73	98	53	46
14	285	333	121	252	333	306	300	154	71	53	230
15	111	98	46	64	118	104	99	35	32	17	74
16	214	232	106	203	203	223	189	65	64	30	190
17	272	273	114	197	286	343	300	124	83	43	208
18	247	238	74	181	237	309	287	106	78	43	168
19	563	522	254	552	528	618	573	211	172	133	571
20	221	229	96	177	213	230	219	96	56	47	175
21	356	359	207	339	402	423	397	154	107	63	341
22	127	132	47	100	123	148	130	62	37	27	88
23	269	276	151	264	272	290	290	111	86	55	315
24	275	315	144	296	285	277	305	110	71	46	289
25	165	152	72	122	159	163	153	65	39	15	121
26	254	250	129	247	261	254	255	111	59	39	244
27	154	166	91	160	161	160	169	92	47	31	173
28	46	41	25	37	40	44	46	21	15	4	36
29	121	118	44	80	90	103	149	38	31	18	64
30	119	129	60	108	123	133	105	58	32	22	103
31	175	192	102	170	205	233	203	84	63	46	169
32	411	374	129	301	346	364	406	152	113	72	248
33	500	474	253	353	478	510	494	235	140	120	392
34	271	324	102	225	248	234	243	88	74	47	186
35	271	278	152	251	272	279	296	125	111	37	240
36	304	294	163	275	291	337	314	135	74	46	265
37	230	233	113	230	246	237	269	84	47	33	198
38	392	371	235	312	380	400	426	170	121	69	282
39	387	450	212	373	461	432	455	227	139	90	409
40	226	206	98	208	212	226	206	111	68	40	199
41	248	237	104	215	212	229	237	84	59	48	174
42	798	807	267	597	772	850	831	306	271	143	526
43	277	263	124	215	283	271	274	132	87	60	223
44	394	382	200	388	395	427	483	185	133	105	387
45	190	214	99	235	217	206	197	75	37	23	253
46	206	248	107	191	316	252	239	105	76	39	191
47	497	457	206	382	499	488	494	201	165	95	338
48	399	419	131	372	423	424	456	189	139	80	347
49	546	552	268	398	609	606	598	248	172	109	372
50	342	321	133	260	364	366	370	144	104	50	261
51	148	163	66	129	172	190	199	60	51	28	114
52	325	394	111	242	333	332	356	110	100	37	213
53	398	379	199	352	398	480	398	190	106	87	385
54	359	321	130	280	353	366	394	141	92	96	293
55	354	366	241	349	394	464	395	229	92	72	366
56	645	711	286	589	734	818	760	267	253	168	653

57	413	414	147	296	436	415	431	146	116	91	277
58	91	100	21	53	112	94	90	25	24	10	42
59	342	318	105	231	323	348	378	145	106	64	240
60	292	312	175	278	319	302	326	146	88	68	253
61	263	230	98	204	248	255	288	120	69	45	206
62	393	383	184	370	455	436	450	156	125	77	369
63	601	635	280	549	556	633	560	245	172	129	539
64	428	420	222	365	428	484	500	232	140	82	347
65	362	350	127	255	412	422	403	168	124	48	275
66	179	205	75	198	183	163	196	69	73	33	198
67	127	147	32	82	139	140	154	47	43	9	83
68	246	245	71	197	252	258	251	67	49	20	171
69	550	423	169	333	478	475	509	170	125	78	346
70	406	363	116	202	361	349	391	137	102	45	216
71	369	373	149	368	441	459	460	156	116	86	323
72	105	114	41	99	137	144	123	49	24	17	73
73	49	63	18	50	58	52	51	23	13	10	57
74	126	129	58	112	126	121	123	60	44	18	103
75	84	84	42	61	101	103	107	40	22	16	61
76	153	146	55	120	139	144	138	55	47	20	116
77	171	150	45	96	141	146	175	74	44	24	135
78	294	268	169	264	282	298	335	165	90	44	298
79	341	326	192	388	321	321	328	130	86	64	360
80	182	193	61	136	183	177	200	86	47	23	113
81	134	151	52	88	136	136	151	46	40	10	74
82	183	205	74	156	212	219	239	106	53	31	144
83	136	136	60	108	135	116	135	64	36	34	115
84	162	161	86	180	176	188	176	94	39	28	180
85	224	259	111	295	213	215	258	71	59	53	339
86	25	28	5	7	30	31	30	23	9	2	8
87	253	276	137	302	274	294	256	117	69	32	284
88	269	230	124	236	228	275	265	106	58	51	187
89	163	194	123	181	191	237	182	74	52	36	184
90	28	35	14	36	41	49	48	20	5	4	29
91	90	89	28	108	87	85	88	30	16	3	96
92	223	236	85	152	184	253	219	102	72	25	158
93	25	66	12	91	40	44	31	12	6	7	46
94	93	93	62	80	115	120	120	55	27	25	89
95	231	252	124	195	202	248	253	122	51	38	199
96	70	73	22	48	56	69	78	25	9	12	40
97	90	81	43	42	71	72	73	29	23	24	55
98	268	263	71	153	290	265	261	130	86	23	190
99	122	159	85	118	162	156	158	70	26	17	117
100	206	208	92	180	173	233	203	96	55	35	188

101	432	481	233	432	441	482	429	196	146	71	420
102	116	125	61	117	117	141	106	55	23	21	118
103	270	270	100	160	264	309	300	116	75	42	144
104	175	166	75	155	149	199	197	72	49	31	165
105	292	314	114	241	289	332	290	95	95	48	259
106	196	171	120	232	209	190	194	81	64	57	228
107	158	217	67	165	196	169	179	89	39	29	140
108	0	0	0	0	0	0	0	0	0	0	0
109	134	140	57	102	145	151	159	54	33	15	108
110	97	131	71	145	107	97	101	68	11	6	165
111	61	70	46	57	75	78	86	36	26	15	69
112	114	83	44	111	96	96	100	41	22	8	92
113	77	83	33	76	76	86	93	40	19	14	78
114	118	112	73	143	94	114	142	48	30	20	131
115	104	107	43	117	90	111	106	44	23	15	103
116	74	75	27	52	52	51	59	34	17	9	55
117	145	201	76	160	167	188	180	79	35	29	151
118	365	355	193	389	357	386	380	183	113	69	405
119	49	63	34	49	67	62	55	23	20	15	59
120	30	25	10	34	36	28	24	11	8	3	53
121	100	91	82	116	145	162	144	76	36	27	120
122	123	119	50	105	117	135	113	64	41	23	73
123	193	194	118	265	211	194	267	119	67	46	222
124	219	256	119	243	238	247	237	91	68	55	242
125	108	101	35	82	138	158	109	43	36	28	93
126	282	290	97	188	245	293	277	103	71	39	170
127	171	133	40	92	148	157	171	82	46	15	79
128	379	380	157	304	423	413	454	190	149	80	338
129	201	212	97	206	212	207	199	75	67	38	163
130	285	319	202	321	377	322	376	149	104	72	313
131	213	209	146	227	221	224	253	113	90	67	279
132	88	90	56	97	107	138	106	46	27	25	88
133	237	243	93	222	245	238	195	97	63	51	214
134	270	252	132	206	256	282	270	133	98	56	213
135	267	266	115	290	295	321	305	117	90	70	280
136	197	168	116	186	255	221	244	81	76	28	200
137	142	164	93	162	171	177	172	72	43	28	197
138	133	114	93	162	140	152	149	71	49	45	169
139	123	116	65	132	141	139	144	67	28	24	106
140	131	116	65	101	99	114	126	52	39	21	120
141	175	143	66	104	174	217	198	66	56	22	143
142	76	73	29	41	75	92	88	39	21	14	48
143	47	50	35	64	69	73	59	28	21	14	65
144	93	127	59	129	107	127	154	69	40	29	144

145	87	92	63	104	95	109	136	57	33	26	113
146	168	182	110	213	163	200	215	102	68	37	206
147	113	145	71	172	131	150	143	41	20	16	169
148	203	209	79	144	229	232	214	80	71	19	157
149	268	308	115	164	277	292	290	112	87	49	183
150	118	126	38	105	174	172	169	74	37	20	102
151	81	81	56	86	102	96	111	37	28	26	102
152	82	83	50	75	100	102	117	42	24	26	87
153	9	22	17	18	20	12	20	6	1	1	22
154	293	236	102	166	323	325	273	119	99	50	171
155	45	32	22	43	39	37	52	13	4	4	53
156	29	31	27	30	43	32	40	28	4	7	31
157	72	85	48	64	97	101	113	46	41	20	88
158	46	59	33	65	74	71	90	25	20	3	73
159	71	77	27	46	79	88	89	41	23	11	83
160	111	129	70	124	134	145	155	62	31	34	168
161	54	75	30	77	94	86	79	29	16	11	85
162	14	18	9	23	15	20	21	5	8	2	9
163	127	141	86	227	166	199	164	74	51	31	205
164	93	114	72	138	118	121	152	76	36	37	117
165	85	83	31	69	117	108	105	42	41	15	106
166	79	86	24	74	107	112	107	48	24	10	63
167	95	109	65	109	112	129	141	54	39	18	122
168	152	207	109	130	204	192	221	92	64	46	209
169	134	171	87	159	137	186	183	76	37	30	169
170	138	146	80	143	146	166	188	90	65	44	167
171	165	187	102	186	231	255	252	129	85	67	206
172	192	187	76	179	208	251	289	123	80	56	189
173	106	100	36	102	116	110	116	42	27	14	106
174	91	112	62	99	130	130	134	71	36	28	106
175	94	123	47	125	127	141	133	82	47	38	151
176	140	142	92	160	142	174	193	69	44	36	187
177	139	189	88	158	213	214	255	121	70	59	219
178	55	68	18	31	93	106	91	37	35	11	44
179	308	322	106	211	320	349	374	124	105	52	227
180	50	94	57	94	111	128	117	54	44	35	142
181	66	104	54	92	124	148	152	52	48	32	104
站点	12	13	14	15	16	17	18	19	20	总数	
1	75	86	96	59	37	107	159	106	91	1760	
2	108	102	87	118	64	72	91	118	114	1744	
3	189	184	210	173	78	108	78	189	182	2672	
4	261	281	319	437	172	417	447	359	295	5232	
5	130	157	154	119	45	113	121	136	146	2412	
6	97	98	134	91	41	91	86	104	124	1760	

7	116	135	121	108	41	37	37	120	116	1675
8	106	115	137	111	42	48	35	122	118	1533
9	392	407	416	408	204	309	296	409	489	6483
10	321	369	372	345	149	346	300	347	351	5821
11	138	148	143	157	69	38	53	134	136	1901
12	223	320	222	246	135	225	195	249	255	4075
13	115	153	130	115	51	59	69	97	127	2005
14	268	317	319	323	175	193	230	278	315	4856
15	93	130	105	102	53	72	70	119	102	1644
16	194	276	180	169	109	178	169	189	214	3397
17	310	298	315	269	109	208	180	278	298	4508
18	238	241	251	250	113	152	155	253	271	3892
19	632	641	640	601	252	593	551	644	613	9864
20	211	295	276	264	132	196	208	274	255	3870
21	409	388	489	392	174	309	357	413	440	6519
22	122	147	123	124	74	85	106	128	146	2076
23	257	283	322	266	134	252	320	263	333	4809
24	258	296	317	268	128	277	264	267	261	4749
25	194	190	177	153	71	113	118	173	168	2583
26	226	255	257	240	118	203	209	244	281	4136
27	138	147	178	163	90	147	142	176	203	2788
28	36	54	42	36	23	47	37	51	41	722
29	145	120	127	88	55	47	50	127	103	1718
30	114	110	121	118	72	111	108	121	124	1991
31	197	217	217	207	113	143	165	199	243	3343
32	350	423	377	337	153	285	237	383	386	5847
33	480	547	522	482	204	424	408	504	548	8068
34	239	223	211	177	97	183	174	210	224	3780
35	256	295	283	289	135	246	228	293	305	4642
36	323	325	306	296	113	236	237	322	324	4980
37	237	242	256	211	87	163	200	231	246	3793
38	420	428	406	439	184	347	302	402	447	6533
39	458	465	487	467	236	369	363	456	458	7394
40	188	210	223	173	94	168	174	165	174	3369
41	185	207	225	146	92	155	167	156	191	3371
42	770	815	793	735	323	523	518	820	821	12286
43	290	278	273	256	150	259	181	289	235	4420
44	432	439	456	406	191	330	369	451	487	7040
45	218	230	166	205	87	195	224	185	166	3422
46	227	252	245	239	116	188	207	244	227	3915
47	486	498	520	449	244	372	332	487	537	7747
48	395	413	429	387	175	361	308	403	407	6657
49	576	656	631	631	255	419	379	593	652	9270
50	325	394	386	356	167	274	225	385	336	5563

51	175	187	194	157	75	117	117	134	193	2669
52	343	360	342	317	128	237	233	343	379	5235
53	358	352	448	373	228	350	352	430	419	6682
54	363	374	349	326	196	288	312	350	337	5720
55	407	392	436	398	174	358	374	434	442	6737
56	685	846	798	742	343	592	550	744	766	11950
57	393	423	425	423	164	241	275	400	399	6325
58	108	100	100	86	37	56	34	101	104	1388
59	322	363	338	362	168	245	256	368	349	5371
60	346	360	317	270	148	236	252	308	302	5098
61	230	299	309	240	132	192	180	259	257	4124
62	431	405	402	398	216	333	350	430	395	6758
63	550	600	583	552	282	473	500	587	644	9670
64	458	497	514	497	283	354	378	458	520	7607
65	329	400	429	356	187	252	244	376	439	5958
66	208	184	167	185	60	166	181	168	173	3064
67	113	149	145	120	48	95	83	130	128	2014
68	245	250	281	252	121	154	160	252	245	3787
69	485	547	525	502	220	341	331	476	484	7567
70	322	391	446	357	154	209	238	387	331	5523
71	427	426	430	398	180	381	360	424	456	6782
72	115	134	126	108	49	88	69	117	110	1842
73	39	64	63	74	25	38	49	49	40	885
74	132	147	130	133	62	107	110	140	156	2137
75	106	79	107	81	50	66	64	95	95	1464
76	141	153	156	137	63	103	89	141	141	2257
77	175	126	148	134	60	117	98	139	149	2347
78	299	384	306	324	180	288	308	323	352	5271
79	302	313	367	296	165	340	292	275	334	5541
80	183	251	196	186	95	136	132	209	198	2987
81	128	148	136	141	65	85	98	121	133	2073
82	192	214	213	176	75	118	131	203	217	3161
83	128	123	155	119	89	119	109	150	156	2223
84	152	187	163	175	92	145	165	175	172	2896
85	196	228	204	207	98	256	290	220	275	4071
86	40	27	32	23	21	7	6	31	33	418
87	264	267	289	258	130	199	289	256	254	4500
88	264	271	224	223	109	202	198	283	252	4055
89	193	216	362	181	103	189	199	235	223	3518
90	27	57	40	27	7	29	30	34	45	605
91	82	106	88	100	25	66	69	83	94	1433
92	198	250	199	234	127	163	150	207	246	3483
93	42	47	44	23	9	58	52	27	44	726
94	107	107	121	90	67	89	108	121	119	1808

95	245	220	253	237	153	180	210	202	250	3865
96	69	74	71	62	17	35	51	59	61	1001
97	81	70	73	49	27	62	56	100	78	1199
98	289	306	266	307	187	200	199	338	359	4451
99	149	160	195	170	91	146	128	158	164	2551
100	212	233	225	203	118	150	186	205	238	3439
101	465	474	465	423	223	393	359	473	487	7525
102	123	125	136	116	58	106	95	125	117	2001
103	289	330	310	311	132	224	184	287	292	4409
104	207	213	230	227	110	160	194	219	197	3190
105	268	295	266	304	134	253	264	312	326	4791
106	204	227	208	224	126	200	222	260	224	3637
107	190	171	184	174	95	127	109	205	195	2898
108	0	0	0	0	0	0	0	0	0	0
109	130	162	170	125	78	115	97	145	157	2277
110	113	130	126	103	36	73	137	83	98	1898
111	58	69	106	79	50	65	92	81	117	1336
112	82	86	100	91	38	85	98	98	107	1592
113	70	81	84	85	35	61	61	73	63	1288
114	114	121	130	105	56	133	115	118	112	2029
115	90	113	94	109	57	104	98	94	118	1740
116	65	59	61	57	30	46	57	60	62	1002
117	181	176	157	159	64	129	139	169	181	2766
118	360	430	390	412	224	365	363	394	379	6512
119	65	74	69	60	27	55	66	70	61	1043
120	26	31	27	17	6	78	107	101	123	778
121	140	196	172	143	88	135	133	147	169	2422
122	123	146	129	121	62	127	95	109	123	1998
123	228	229	232	241	115	218	244	231	254	3888
124	209	241	257	239	146	230	256	246	248	4087
125	112	159	136	108	56	77	91	123	137	1930
126	274	308	322	279	106	206	160	290	289	4289
127	167	171	188	181	92	113	80	229	212	2567
128	401	468	493	423	222	432	401	484	524	7115
129	205	193	235	223	126	201	180	208	196	3444
130	340	378	340	315	189	271	297	324	368	5662
131	253	256	241	239	131	267	262	257	263	4211
132	114	119	125	103	62	133	96	113	123	1856
133	262	228	260	255	130	202	201	249	235	3920
134	254	258	268	263	143	241	238	218	287	4338
135	313	342	322	262	139	272	283	348	314	5011
136	263	253	234	229	101	206	193	263	270	3784
137	165	165	158	155	65	170	191	175	220	2885
138	112	167	170	213	99	171	173	170	186	2738

139	114	111	131	128	67	100	134	126	124	2120
140	119	150	118	130	68	139	139	139	170	2156
141	198	240	223	187	98	147	140	232	225	3054
142	87	111	93	66	34	65	57	87	85	1281
143	73	74	92	84	40	47	57	70	66	1128
144	127	147	151	137	105	149	167	152	181	2394
145	117	135	139	138	52	129	95	147	137	2004
146	192	254	246	240	133	199	192	204	234	3558
147	110	120	121	114	33	122	168	162	129	2250
148	232	275	280	258	121	175	157	285	302	3722
149	293	321	344	295	151	224	208	326	340	4647
150	119	224	161	160	78	129	92	208	188	2494
151	129	128	135	133	52	120	111	131	154	1899
152	103	147	134	115	48	99	100	138	129	1801
153	14	24	13	32	1	23	19	13	12	299
154	272	319	301	317	153	204	206	377	371	4677
155	33	41	46	79	11	36	45	37	58	730
156	40	40	34	62	14	59	34	45	36	666
157	103	102	120	87	63	96	109	103	125	1683
158	89	98	91	100	45	65	64	89	100	1300
159	89	97	103	90	51	84	68	112	121	1450
160	164	181	211	163	87	155	156	184	193	2657
161	87	97	105	90	40	87	79	107	101	1429
162	20	20	21	38	7	27	16	19	16	328
163	167	195	205	153	74	194	193	189	177	3018
164	151	161	159	141	69	134	123	165	163	2340
165	109	136	127	107	69	106	69	124	129	1778
166	124	149	121	119	67	54	63	100	129	1660
167	160	168	244	187	81	124	146	161	161	2425
168	232	270	275	269	133	204	192	303	301	3805
169	147	144	148	195	72	168	166	166	218	2793
170	197	178	182	173	119	176	152	155	166	2871
171	256	302	289	286	170	281	273	301	318	4341
172	265	300	272	321	144	249	228	305	333	4247
173	100	129	119	113	37	108	110	122	123	1836
174	164	147	153	143	70	114	148	191	187	2316
175	144	177	166	143	78	133	146	162	168	2425
176	166	211	210	194	123	190	214	209	205	3101
177	215	219	216	213	129	210	237	254	276	3694
178	85	100	110	113	59	59	52	96	107	1370
179	379	367	387	351	149	250	257	401	405	5444
180	121	119	127	132	74	135	163	136	160	2093
181	156	160	177	166	76	134	121	173	185	2324

2) 6.1 章节中各个站点的每天与累计还车频次

站点	1	2	3	4	5	6	7	8	9	10	11
1	93	145	48	167	116	103	98	23	11	4	167
2	117	110	27	88	125	136	127	53	22	16	87
3	176	193	65	74	199	187	219	115	54	9	103
4	224	297	108	283	242	310	297	115	87	61	304
5	154	153	80	166	182	176	190	61	49	16	161
6	112	97	41	116	119	119	132	56	33	15	80
7	106	106	23	59	140	134	157	67	19	3	50
8	89	89	15	41	101	110	110	54	23	4	31
9	334	351	171	315	408	447	400	178	121	72	348
10	341	332	216	314	361	326	347	150	84	52	331
11	120	134	47	34	149	147	147	90	18	17	63
12	263	243	118	211	218	236	266	78	83	32	204
13	122	87	56	60	70	104	98	70	71	43	35
14	292	327	121	253	312	311	308	144	69	56	223
15	99	98	45	71	101	110	103	45	24	13	67
16	206	223	88	194	190	199	178	56	59	34	178
17	295	274	119	195	296	342	311	148	70	51	208
18	247	242	89	184	258	323	309	129	81	51	171
19	583	544	252	567	523	605	588	218	187	137	628
20	206	205	89	172	223	215	204	84	59	46	160
21	361	364	204	349	390	413	383	143	103	69	331
22	140	127	58	89	128	156	126	78	42	32	86
23	255	294	141	261	272	310	307	92	102	60	307
24	277	287	133	293	301	279	298	84	84	49	264
25	168	167	63	118	157	163	162	78	39	25	126
26	255	259	108	230	294	228	250	98	65	36	228
27	177	188	84	151	169	166	175	80	55	28	168
28	55	40	22	41	46	56	45	18	9	3	35
29	128	114	42	66	96	113	147	54	29	15	56
30	125	127	57	114	105	127	107	43	41	27	110
31	168	195	106	157	185	220	204	84	53	30	166
32	390	368	124	269	349	368	410	180	95	53	243
33	518	502	263	390	502	512	552	257	134	106	423
34	251	304	98	209	217	227	228	101	67	37	190
35	289	297	138	273	302	285	297	116	100	45	242
36	287	300	140	284	289	326	329	130	92	50	272
37	235	242	133	246	258	232	276	93	59	31	228
38	385	372	236	293	380	394	437	189	128	70	277
39	402	424	193	384	485	454	462	190	138	85	397
40	246	210	86	193	229	234	225	98	64	49	213
41	233	215	99	216	213	216	210	81	64	45	174
42	800	788	310	543	774	852	813	372	213	129	507

43	271	250	110	210	271	258	268	131	77	58	203
44	414	405	222	399	424	422	498	197	131	95	414
45	189	234	82	250	213	194	188	75	31	24	246
46	221	236	106	188	315	226	248	95	75	33	188
47	492	453	213	390	477	487	459	218	180	89	364
48	389	433	145	368	412	432	445	173	143	71	352
49	532	578	271	399	609	613	596	255	186	112	386
50	357	323	127	271	348	365	355	133	101	54	252
51	163	177	74	120	182	186	199	92	46	33	118
52	328	411	129	268	331	374	365	131	91	53	240
53	388	386	202	368	385	475	403	164	122	77	383
54	339	294	147	278	340	356	376	139	78	84	313
55	356	354	230	335	392	468	400	231	90	78	355
56	682	714	334	616	722	833	792	275	271	157	660
57	394	415	156	300	434	440	459	192	95	84	272
58	109	108	30	46	119	104	112	43	19	7	54
59	335	316	112	241	316	347	354	146	97	71	231
60	295	338	213	319	331	333	353	164	91	77	303
61	298	246	119	204	260	282	280	119	76	42	206
62	400	402	194	361	456	465	482	164	131	66	380
63	629	641	279	556	613	620	577	249	189	122	549
64	433	407	223	371	418	479	488	234	124	77	343
65	361	339	118	256	403	412	399	176	98	57	261
66	174	188	79	206	176	169	188	57	79	37	208
67	148	152	49	93	147	155	163	74	25	7	86
68	244	259	85	198	250	254	262	79	57	26	169
69	540	466	166	359	483	491	524	190	134	73	359
70	382	360	121	194	364	355	366	150	85	54	210
71	368	382	170	363	429	457	440	165	123	82	346
72	131	138	54	99	139	152	154	58	28	20	78
73	52	64	14	51	56	46	54	22	12	9	55
74	109	124	57	104	130	126	105	50	37	17	93
75	82	89	40	68	89	111	109	53	29	10	60
76	142	156	58	131	147	150	142	56	39	16	118
77	147	137	47	96	137	146	161	74	36	23	125
78	290	285	151	270	300	301	335	116	103	54	294
79	324	311	177	375	307	325	334	109	101	58	342
80	185	207	90	134	170	180	194	93	50	18	109
81	124	145	59	75	139	134	139	52	29	15	77
82	208	240	73	160	190	220	225	111	51	27	165
83	142	154	70	119	153	130	142	73	35	37	114
84	175	158	78	178	177	184	187	53	45	27	183
85	230	257	130	289	208	209	250	90	53	48	330
86	26	27	13	6	25	34	29	23	9	3	9

87	261	285	150	330	270	290	284	133	85	47	300
88	272	253	134	248	239	291	274	98	72	55	202
89	155	193	121	167	180	228	163	76	50	38	176
90	36	38	19	34	43	48	43	20	5	5	33
91	84	96	24	119	76	95	108	25	20	6	93
92	218	217	86	150	164	248	203	76	60	29	167
93	31	66	13	96	46	36	31	8	6	7	53
94	85	98	55	87	109	112	120	44	17	23	71
95	212	225	102	183	200	236	239	87	77	34	186
96	76	68	22	46	60	78	85	21	10	8	39
97	84	84	34	49	68	75	66	30	21	21	58
98	252	237	72	145	264	244	247	114	81	23	185
99	142	163	89	115	177	166	144	48	31	22	127
100	199	194	79	178	175	214	204	80	68	30	189
101	432	499	208	396	431	485	456	186	167	81	451
102	125	124	66	122	124	131	129	41	33	25	107
103	259	283	106	163	261	296	302	118	71	53	176
104	171	170	78	171	156	198	194	82	60	32	164
105	274	295	115	219	291	299	277	97	100	64	250
106	204	197	101	235	208	205	202	88	72	58	225
107	172	202	77	169	183	159	178	93	38	34	113
108	0	0	0	0	0	0	0	0	0	0	0
109	125	149	56	86	137	148	167	59	35	20	110
110	105	119	75	149	109	117	101	46	16	7	176
111	60	69	33	59	80	78	83	32	27	11	66
112	100	65	36	104	91	93	95	36	23	10	79
113	77	74	22	70	77	94	102	35	38	18	72
114	131	120	54	144	109	121	114	41	34	21	132
115	98	109	48	111	95	95	95	37	22	14	101
116	64	68	18	54	51	49	50	22	20	6	59
117	153	208	63	150	186	187	183	73	47	31	149
118	372	323	184	359	369	412	362	137	118	69	387
119	45	55	28	52	70	59	53	25	13	19	52
120	40	31	22	33	36	30	25	12	7	12	64
121	94	90	81	110	121	140	134	79	29	27	110
122	125	111	55	95	116	124	117	72	34	26	81
123	178	203	101	241	211	215	244	93	68	45	204
124	221	236	106	230	234	242	225	85	79	44	242
125	121	94	40	88	124	145	107	49	29	24	79
126	298	333	107	178	284	296	302	139	64	38	188
127	169	128	41	89	124	153	158	55	34	10	77
128	366	367	170	305	399	396	438	157	157	87	321
129	194	202	88	190	191	204	199	54	52	46	154
130	295	315	178	309	394	325	361	130	124	80	308

131	191	226	125	218	232	214	241	86	68	57	248
132	86	92	55	102	107	118	118	42	25	23	99
133	238	229	93	218	215	222	193	93	68	42	181
134	261	246	125	211	253	266	278	102	97	50	214
135	253	258	123	293	285	297	281	105	98	65	288
136	205	172	105	187	248	218	246	84	70	40	205
137	134	165	89	178	161	176	179	70	53	24	203
138	129	121	88	157	145	148	162	59	51	38	150
139	126	125	65	131	139	135	147	54	28	30	97
140	136	114	55	97	102	111	115	51	43	22	113
141	171	145	50	104	173	205	195	89	48	25	141
142	73	54	36	47	69	85	78	40	22	12	53
143	45	37	27	59	59	64	51	28	22	14	67
144	100	118	60	124	121	135	142	72	36	27	132
145	94	111	47	105	103	101	131	42	42	24	107
146	163	170	108	199	197	177	210	75	65	35	230
147	106	150	66	171	130	130	133	35	19	9	160
148	215	213	80	152	240	254	246	117	57	22	146
149	263	303	120	171	314	294	303	148	75	45	192
150	128	107	43	99	174	169	153	52	40	17	88
151	79	77	49	105	86	95	107	40	30	22	107
152	83	95	41	83	93	105	114	41	40	18	74
153	15	23	15	21	12	24	18	7	1	0	17
154	293	267	100	205	315	336	268	139	94	65	213
155	35	36	26	51	38	44	44	13	3	5	50
156	20	37	25	28	45	33	40	25	6	14	31
157	74	86	41	63	94	106	118	38	32	18	92
158	54	68	36	83	86	94	107	37	21	9	86
159	75	67	31	36	85	98	85	49	28	13	76
160	115	123	66	129	133	136	158	55	29	38	156
161	61	74	46	76	91	90	83	46	15	10	91
162	15	18	5	24	20	18	19	8	3	3	16
163	132	150	89	220	175	175	170	85	57	36	213
164	97	113	70	135	130	140	150	83	38	39	132
165	84	87	41	70	106	110	94	61	37	22	104
166	72	75	35	62	101	116	93	49	23	13	63
167	89	112	55	105	117	141	129	41	41	31	118
168	150	214	87	145	194	188	218	86	53	57	191
169	135	174	71	158	154	188	172	57	40	32	171
170	144	149	66	138	135	165	174	84	55	43	156
171	166	176	97	198	228	256	245	105	83	60	185
172	183	188	83	181	235	274	285	108	80	66	191
173	101	98	32	96	125	117	123	43	27	13	98
174	94	96	59	107	126	136	137	65	46	30	100

175	95	122	53	114	124	156	157	63	42	44	134
176	124	148	80	148	144	174	183	61	58	33	187
177	125	180	80	150	201	210	245	114	69	53	216
178	54	61	22	33	98	111	86	59	22	10	57
179	305	322	111	212	325	359	390	167	95	56	227
180	36	81	53	90	103	123	113	50	43	32	118
181	66	83	59	92	122	143	138	57	43	23	109
站点	12	13	14	15	16	17	18	19	20	总数	
1	82	93	94	60	31	120	163	116	87	1821	
2	124	105	88	119	69	75	94	120	109	1811	
3	186	208	225	205	94	124	106	197	176	2915	
4	278	303	335	502	201	490	469	371	305	5582	
5	140	161	167	135	70	132	131	140	149	2613	
6	105	103	146	97	45	96	77	108	132	1829	
7	126	125	137	113	79	54	56	120	124	1798	
8	102	103	121	104	57	43	40	124	117	1478	
9	394	395	427	397	166	285	306	427	474	6416	
10	307	367	378	329	152	328	322	343	361	5741	
11	153	155	140	161	89	41	44	134	134	2017	
12	248	281	246	225	140	213	186	226	250	3967	
13	92	110	98	89	40	40	56	74	79	1494	
14	262	302	307	324	163	209	228	272	330	4813	
15	102	129	113	109	57	72	70	108	108	1644	
16	201	241	178	178	95	166	170	191	207	3232	
17	318	313	312	261	128	200	191	264	282	4578	
18	253	259	259	281	136	153	163	271	283	4142	
19	633	655	630	597	241	590	540	627	640	9985	
20	225	276	269	244	120	210	170	264	249	3690	
21	385	397	463	383	153	290	365	415	420	6381	
22	136	147	137	117	68	98	98	147	164	2174	
23	267	305	322	244	145	245	299	278	315	4821	
24	278	317	281	266	124	256	241	279	273	4664	
25	188	186	182	171	78	118	120	184	183	2676	
26	236	223	263	227	99	205	200	266	279	4049	
27	149	174	166	157	85	129	147	191	203	2842	
28	40	52	42	29	24	41	31	36	43	708	
29	137	125	130	94	56	49	48	121	118	1738	
30	101	114	106	109	56	100	110	115	115	1909	
31	180	191	209	195	93	140	170	196	236	3178	
32	336	412	368	337	149	277	223	378	377	5706	
33	512	540	514	506	251	429	431	495	555	8392	
34	206	211	189	174	114	181	167	205	208	3584	
35	278	319	291	312	140	232	229	307	311	4803	
36	314	322	328	314	139	268	221	319	325	5049	

37	229	239	274	218	94	208	237	248	250	4030
38	425	424	427	435	188	323	283	408	442	6516
39	422	480	473	466	228	372	352	470	443	7320
40	203	235	213	211	100	172	193	182	198	3554
41	200	205	214	159	84	147	156	172	158	3261
42	770	823	791	732	341	511	488	806	785	12148
43	277	259	261	261	120	241	161	278	236	4201
44	406	435	484	418	188	383	397	450	483	7265
45	217	226	178	190	75	171	239	168	157	3347
46	231	241	245	233	115	185	221	240	236	3878
47	460	500	532	435	244	384	330	481	543	7731
48	398	414	421	399	178	347	300	422	400	6642
49	591	642	629	645	272	424	392	594	672	9398
50	321	376	391	334	171	268	224	370	333	5474
51	191	214	205	177	78	128	121	130	214	2848
52	332	384	362	346	151	260	251	351	391	5549
53	359	370	445	374	204	340	356	428	388	6617
54	336	367	352	316	193	275	298	320	340	5541
55	406	415	437	388	184	339	357	427	451	6693
56	736	876	800	757	375	570	563	788	787	12308
57	389	419	427	415	189	259	277	403	395	6414
58	98	105	120	90	47	60	38	110	99	1518
59	309	349	334	334	164	239	257	339	314	5205
60	347	358	360	297	178	239	292	309	309	5506
61	241	293	305	254	129	210	193	269	260	4286
62	417	442	403	460	203	349	326	439	438	6978
63	562	625	619	557	285	512	537	582	642	9945
64	449	522	495	498	275	367	369	481	505	7558
65	364	391	433	355	186	255	270	381	463	5978
66	206	179	162	194	77	169	204	188	172	3112
67	115	156	164	127	50	94	94	137	143	2179
68	233	255	287	250	135	153	170	240	257	3863
69	489	564	546	550	233	377	328	473	503	7848
70	326	378	443	350	155	218	214	381	330	5436
71	410	440	425	415	186	378	370	408	476	6833
72	131	148	125	108	49	107	76	119	134	2048
73	36	63	57	64	23	34	44	41	42	839
74	136	149	132	123	53	99	102	146	146	2038
75	108	92	102	92	49	72	53	89	92	1489
76	150	144	142	128	79	101	102	146	130	2277
77	164	126	143	127	59	111	103	135	140	2237
78	319	366	332	335	178	281	285	348	340	5283
79	280	351	322	285	141	314	296	273	331	5356
80	184	263	218	179	96	150	118	198	198	3034

81	115	140	141	135	73	99	88	123	143	2045
82	212	199	222	171	90	136	136	184	218	3238
83	147	124	156	124	73	113	135	156	163	2360
84	165	180	172	150	77	134	162	171	166	2822
85	187	224	198	211	98	253	288	224	257	4034
86	38	27	34	28	21	13	4	19	33	421
87	270	261	269	250	118	212	312	280	259	4666
88	246	279	232	229	107	230	217	267	260	4205
89	190	204	344	189	100	190	202	223	221	3410
90	31	49	35	34	11	44	30	35	38	631
91	84	111	96	98	21	85	88	87	90	1506
92	199	227	200	221	105	160	156	199	232	3317
93	40	47	49	34	10	65	50	40	36	764
94	103	114	103	84	52	85	95	117	115	1689
95	253	234	234	232	95	172	199	223	258	3681
96	59	69	76	58	23	42	49	59	65	1013
97	77	74	64	48	30	65	55	100	88	1191
98	273	277	268	306	174	191	199	336	354	4242
99	139	172	200	166	95	126	125	172	182	2601
100	211	217	223	192	85	153	190	199	252	3332
101	483	504	475	445	242	399	355	503	485	7683
102	118	127	142	128	66	102	102	128	133	2073
103	300	329	318	324	162	244	192	285	293	4535
104	216	200	237	203	104	159	197	202	216	3210
105	258	282	261	278	137	256	235	303	293	4584
106	215	233	195	217	128	198	213	263	253	3710
107	185	158	192	169	91	113	124	184	197	2831
108	0	0	0	0	0	0	0	0	0	0
109	122	160	149	120	82	123	103	149	149	2249
110	106	133	115	91	30	71	159	92	108	1925
111	71	70	97	72	40	63	93	80	123	1307
112	74	82	89	96	36	77	89	99	108	1482
113	82	92	90	75	33	57	58	85	58	1309
114	127	140	154	109	67	130	114	130	129	2121
115	89	108	99	103	49	101	100	100	113	1687
116	56	64	50	61	19	46	58	59	57	931
117	192	170	163	155	62	136	142	169	171	2790
118	370	419	375	410	213	365	383	408	390	6425
119	62	74	74	64	33	52	54	62	64	1010
120	45	29	32	30	14	77	97	109	122	867
121	123	186	154	128	82	117	125	128	159	2217
122	119	143	119	122	68	120	121	102	131	2001
123	208	230	223	233	102	195	223	233	237	3687
124	208	241	246	251	124	216	248	241	241	3960

125	108	148	127	108	49	77	81	115	126	1839
126	297	329	369	300	143	196	164	292	294	4611
127	140	149	188	184	84	107	81	214	210	2395
128	409	462	458	407	246	428	370	473	509	6925
129	194	167	212	203	124	184	167	190	208	3223
130	355	362	337	299	170	263	293	349	345	5592
131	243	246	230	229	109	244	238	259	247	3951
132	106	121	129	108	58	144	88	120	129	1870
133	247	242	252	242	125	188	192	241	213	3734
134	247	264	271	239	126	240	226	219	283	4218
135	294	317	312	253	124	263	287	298	313	4807
136	237	239	243	235	115	196	202	264	243	3754
137	152	157	194	180	75	177	200	180	215	2962
138	121	176	176	186	85	153	166	175	176	2662
139	114	119	122	111	78	96	117	135	135	2104
140	107	135	117	125	82	130	139	132	174	2100
141	206	245	213	193	80	136	150	230	250	3049
142	85	106	86	75	28	51	45	77	88	1210
143	70	66	91	83	30	54	44	67	61	1039
144	125	135	154	144	93	148	150	157	178	2351
145	104	149	135	141	47	127	101	134	142	1987
146	213	255	253	205	135	210	171	208	229	3508
147	102	109	127	103	40	139	155	141	128	2153
148	245	285	280	255	142	163	159	290	323	3884
149	303	319	334	318	143	227	210	330	339	4751
150	122	188	150	145	66	109	111	182	181	2324
151	121	123	133	117	57	110	101	120	127	1806
152	116	145	116	119	42	93	94	112	131	1755
153	16	27	15	36	2	25	11	18	14	317
154	277	329	296	314	150	244	216	367	362	4850
155	30	41	35	70	14	39	41	36	51	702
156	37	39	37	59	21	51	35	52	34	669
157	97	106	125	99	57	88	102	104	131	1671
158	94	109	106	104	50	79	74	106	122	1525
159	90	95	95	89	48	77	61	124	124	1446
160	157	178	198	177	88	159	157	176	196	2624
161	89	91	111	82	43	89	84	99	95	1466
162	26	17	20	34	8	29	23	20	19	345
163	168	202	212	167	70	185	192	211	193	3102
164	149	181	159	148	77	125	138	165	180	2449
165	108	142	126	115	73	94	89	120	130	1813
166	111	151	109	123	67	63	59	103	117	1605
167	138	167	240	175	86	134	112	139	153	2323
168	230	253	280	246	156	190	204	295	303	3740

169	165	153	156	192	65	177	160	187	212	2819
170	184	169	168	174	91	159	137	160	167	2718
171	260	301	300	261	168	266	269	314	289	4227
172	269	298	300	326	135	257	244	313	363	4379
173	119	127	118	117	35	107	126	120	113	1855
174	160	141	149	139	85	109	140	183	191	2293
175	158	168	166	136	71	123	150	163	189	2428
176	170	205	205	197	109	203	206	222	197	3054
177	197	219	212	203	127	200	229	258	247	3535
178	91	103	100	93	63	68	65	95	116	1407
179	377	427	414	360	146	240	267	398	427	5625
180	115	111	134	126	57	127	144	135	158	1949
181	130	166	166	166	78	145	117	162	186	2251

3) 6.1 中的站点的借还车频次排序

借车站点累计频次排序				还车站点累计频次排序		
排名	站点序号	累计借车频次		排名	站点序号	累计还车频次
	1	42	12286	1	56	12308
	2	56	11950	2	42	12148
	3	19	9864	3	19	9985
	4	63	9670	4	63	9945
	5	49	9270	5	49	9398
	6	33	8068	6	33	8392
	7	47	7747	7	69	7848
	8	64	7607	8	47	7731
	9	69	7567	9	101	7683
	10	101	7525	10	64	7558
	11	39	7394	11	39	7320
	12	128	7115	12	44	7265
	13	44	7040	13	62	6978
	14	71	6782	14	128	6925
	15	62	6758	15	71	6833
	16	55	6737	16	55	6693
	17	53	6682	17	48	6642
	18	48	6657	18	53	6617
	19	38	6533	19	38	6516
	20	21	6519	20	118	6425
	21	118	6512	21	9	6416
	22	9	6483	22	57	6414
	23	57	6325	23	21	6381
	24	65	5958	24	65	5978
	25	32	5847	25	10	5741
	26	10	5821	26	32	5706
	27	54	5720	27	179	5625

28	130	5662	28	130	5592
29	50	5563	29	4	5582
30	79	5541	30	52	5549
31	70	5523	31	54	5541
32	179	5444	32	60	5506
33	59	5371	33	50	5474
34	78	5271	34	70	5436
35	52	5235	35	79	5356
36	4	5232	36	78	5283
37	60	5098	37	59	5205
38	135	5011	38	36	5049
39	36	4980	39	154	4850
40	14	4856	40	23	4821
41	23	4809	41	14	4813
42	105	4791	42	135	4807
43	24	4749	43	35	4803
44	154	4677	44	149	4751
45	149	4647	45	87	4666
46	35	4642	46	24	4664
47	17	4508	47	126	4611
48	87	4500	48	105	4584
49	98	4451	49	17	4578
50	43	4420	50	103	4535
51	103	4409	51	172	4379
52	171	4341	52	61	4286
53	134	4338	53	98	4242
54	126	4289	54	171	4227
55	172	4247	55	134	4218
56	131	4211	56	88	4205
57	26	4136	57	43	4201
58	61	4124	58	18	4142
59	124	4087	59	26	4049
60	12	4075	60	85	4034
61	85	4071	61	37	4030
62	88	4055	62	12	3967
63	133	3920	63	124	3960
64	46	3915	64	131	3951
65	18	3892	65	148	3884
66	123	3888	66	46	3878
67	20	3870	67	68	3863
68	95	3865	68	136	3754
69	168	3805	69	168	3740
70	37	3793	70	133	3734
71	68	3787	71	106	3710

72	136	3784	72	20	3690
73	34	3780	73	123	3687
74	148	3722	74	95	3681
75	177	3694	75	34	3584
76	106	3637	76	40	3554
77	146	3558	77	177	3535
78	89	3518	78	146	3508
79	92	3483	79	89	3410
80	129	3444	80	45	3347
81	100	3439	81	100	3332
82	45	3422	82	92	3317
83	16	3397	83	41	3261
84	41	3371	84	82	3238
85	40	3369	85	16	3232
86	31	3343	86	129	3223
87	104	3190	87	104	3210
88	82	3161	88	31	3178
89	176	3101	89	66	3112
90	66	3064	90	163	3102
91	141	3054	91	176	3054
92	163	3018	92	141	3049
93	80	2987	93	80	3034
94	107	2898	94	137	2962
95	84	2896	95	3	2915
96	137	2885	96	51	2848
97	170	2871	97	27	2842
98	169	2793	98	107	2831
99	27	2788	99	84	2822
100	117	2766	100	169	2819
101	138	2738	101	117	2790
102	3	2672	102	170	2718
103	51	2669	103	25	2676
104	160	2657	104	138	2662
105	25	2583	105	160	2624
106	127	2567	106	5	2613
107	99	2551	107	99	2601
108	150	2494	108	164	2449
109	167	2425	109	175	2428
110	175	2425	110	127	2395
111	121	2422	111	83	2360
112	5	2412	112	144	2351
113	144	2394	113	150	2324
114	77	2347	114	167	2323
115	164	2340	115	174	2293

116	181	2324	116	76	2277
117	174	2316	117	181	2251
118	109	2277	118	109	2249
119	76	2257	119	77	2237
120	147	2250	120	121	2217
121	83	2223	121	67	2179
122	140	2156	122	22	2174
123	74	2137	123	147	2153
124	139	2120	124	114	2121
125	180	2093	125	139	2104
126	22	2076	126	140	2100
127	81	2073	127	102	2073
128	114	2029	128	72	2048
129	67	2014	129	81	2045
130	13	2005	130	74	2038
131	145	2004	131	11	2017
132	102	2001	132	122	2001
133	122	1998	133	145	1987
134	30	1991	134	180	1949
135	125	1930	135	110	1925
136	11	1901	136	30	1909
137	151	1899	137	132	1870
138	110	1898	138	173	1855
139	132	1856	139	125	1839
140	72	1842	140	6	1829
141	173	1836	141	1	1821
142	94	1808	142	165	1813
143	152	1801	143	2	1811
144	165	1778	144	151	1806
145	1	1760	145	7	1798
146	6	1760	146	152	1755
147	2	1744	147	29	1738
148	115	1740	148	94	1689
149	29	1718	149	115	1687
150	157	1683	150	157	1671
151	7	1675	151	15	1644
152	166	1660	152	166	1605
153	15	1644	153	158	1525
154	112	1592	154	58	1518
155	8	1533	155	91	1506
156	75	1464	156	13	1494
157	159	1450	157	75	1489
158	91	1433	158	112	1482
159	161	1429	159	8	1478

160	58	1388	160	161	1466
161	178	1370	161	159	1446
162	111	1336	162	178	1407
163	158	1300	163	113	1309
164	113	1288	164	111	1307
165	142	1281	165	142	1210
166	97	1199	166	97	1191
167	143	1128	167	143	1039
168	119	1043	168	96	1013
169	116	1002	169	119	1010
170	96	1001	170	116	931
171	73	885	171	120	867
172	120	778	172	73	839
173	155	730	173	93	764
174	93	726	174	28	708
175	28	722	175	155	702
176	156	666	176	156	669
177	90	605	177	90	631
178	86	418	178	86	421
179	162	328	179	162	345
180	153	299	180	153	317
181	108	0	181	108	0

4) 自行车租车站点坐标

站点序号	站点名称	y 刻度	x 刻度
1	科技馆	8114.064	4378.759
2	温州大剧院	8255.842	4209.716
3	吴桥路加油站	3838.912	3882.536
4	银泰百货	4515.084	2475.662
5	星河广场	2453.85	916.104
6	绣山卫生院	8539.398	3735.305
7	市政府西	7972.286	3713.493
8	市政府东	8217.671	3697.134
9	小南门立交桥	3718.946	2333.884
10	市九中	4340.588	1134.224
11	鹿城区审批中心	3528.091	3909.801
12	桥儿头公交站	5414.829	3511.732
13	公共自行车中心	3757.117	3740.758
14	南浦医院	5453	4209.716
15	温州建国医院对面	4596.879	5005.854
16	金色家园	4580.52	4187.904
17	区政府西	3549.903	1706.789
18	区政府东	3626.245	1706.789
19	开太百货	4324.229	2044.875
20	南浦桥	5529.342	3762.57

21	东阿外楼	4231.528	921.557
22	鱼鳞浹	6205.514	4842.264
23	温迪路农贸市场	5736.556	3937.066
24	春晖路口	6080.095	4040.673
25	鹿城法院	6843.515	3626.245
26	南浦小区	6140.078	4155.186
27	绵江家园	7225.225	4536.896
28	上田菜场	8533.945	4591.426
29	市审批中心	7590.576	3446.296
30	秀山中学	7645.106	3751.664
31	云锦大厦	6516.335	4367.853
32	数码广场	5747.462	2006.704
33	国光大厦	3222.723	1968.533
34	广信大厦	3195.458	1794.037
35	安平大厦	3031.868	1374.156
36	工人文化宫	2791.936	1063.335
37	松台广场	3304.518	2186.653
38	市二医院	3975.237	1161.489
39	墨斗小区	3500.826	1128.771
40	海悦名邸酒店	3724.399	730.702
41	安澜轮渡码头	3277.253	828.856
42	街心公园	4885.888	1995.798
43	群艺大楼	3598.98	2361.149
44	东瓯智库	4193.357	1859.473
45	白鹿州公园	4645.956	3326.33
46	巽山公园	4760.469	3195.458
47	阳光花苑	4776.828	2699.235
48	沃尔玛（欧洲城）	4711.392	1205.113
49	医学院	5262.145	2502.927
50	中西医结合医院	7525.14	3320.877
51	新城车站	8692.082	3419.031
52	裕达大厦农业银行	6434.54	3762.57
53	马鞍池公园北	4242.434	2753.765
54	吴桥路观松楼	3735.305	3359.048
55	中山公园北	4531.443	1663.165
56	五马美食林	3658.963	1903.097
57	建设大厦	4924.059	2164.841
58	新城大道体检中心	9199.211	3620.792
59	文景花苑东	5676.573	2230.277
60	公园路去茶去	4667.768	1614.088
61	丽都大厦	5153.085	1161.489
62	大南门农贸	4127.921	2322.978
63	体育中心西	5502.077	3010.056
64	县前头	4296.964	1532.293

65	小南门电力局	3838.912	2742.859
66	体育中心南	5905.599	3271.8
67	大自然家园	9139.228	4324.229
68	站前东小区	6543.6	4793.187
69	时代广场	6396.369	3511.732
70	区地税局	6499.976	4122.468
71	江滨东杨府山路口	4356.947	2257.542
72	质量监督局	9226.476	4226.075
73	汤家桥路英豪花园	9580.921	3511.732
74	方正大厦	4406.024	3637.151
75	龙方家园	4874.982	4553.255
76	宏源路数码大厦	8833.86	3451.749
77	喜来登酒店	6314.574	3244.535
78	桥儿头交运国旅	5436.641	3304.518
79	世纪联华	5812.898	4215.169
80	牛山北路文杰酒业	4144.28	4406.024
81	火车站对面	6532.694	5000.401
82	海港大厦	4258.793	676.172
83	南塘一组团	5818.351	4913.153
84	南浦街道	6085.548	4493.272
85	市图书馆	7574.217	4275.152
86	测试点	3768.023	3740.758
87	雪山路-勤奋路口	2077.593	2704.688
88	鹿城路旅集散中心	2399.32	2153.935
89	妙果寺	3031.868	2393.867
90	拉菲度假酒店	5911.052	1041.523
91	中瑞曼哈顿	5016.76	790.685
92	十四中学	5747.462	1428.686
93	瓯江路鹿城广场	6429.087	1205.113
94	惠民路与航标路口	7285.208	2006.704
95	特警支队	7252.49	2388.414
96	鹿城实验中学	8130.423	2208.465
97	江滨路府东路口	8174.047	1744.96
98	市公安局	6761.72	3249.988
99	公交上徒门始发站	7197.96	2595.628
100	蒲鞋街道卫生中心	7247.037	3015.509
101	洪殿奥康	6696.284	2175.747
102	府东家园公交站	7950.474	2262.995
103	新南亚大酒店	6320.027	1821.302
104	东门商业步行街	4929.512	1046.976
105	温州十九中	7317.926	3249.988
106	金桥路桃源居	6729.002	2988.244
107	远东大酒店	6118.266	2083.046
108	秀山公园北门	7879.585	3277.253

109	浙江工贸学院	8157.688	3037.321
110	杨府山公园停车场	8528.492	2737.406
111	巨一花园	8064.987	2530.192
112	绿洲花园农行外	9226.476	3064.586
113	黎明街道卫生中心	10164.39	3130.022
114	新田园人本超市	10257.09	2715.594
115	汤家桥北云中花园	9150.134	3293.612
116	汤家桥北路新田路	9684.528	3064.586
117	江滨路车站大道	6390.916	1521.387
118	上陡门住宅公交站	7612.388	2192.106
119	学院东路丰源路口	9139.228	2742.859
120	会展中心	9875.383	1750.413
121	市电力局	3026.415	3511.732
122	温八医	3211.817	3479.014
123	杏花路十七中路口	3004.603	3293.612
124	马鞍池路杏花路口	2988.244	3119.116
125	东南剧场	6309.121	2999.15
126	学院大厦	6254.591	2731.953
127	人力资源社保局	6696.284	2715.594
128	马鞍池吴桥路口	3887.989	2944.62
129	马鞍池西路 89 号	3735.305	2459.303
130	水心邮电	3190.005	3042.774
131	水心汇昌路口	2464.756	3342.689
132	浙南农贸市场对面	2339.337	3653.51
133	清明桥站	2704.688	2432.038
134	杏花路水心菜场	2857.372	2890.09
135	温州大厦	3348.142	2432.038
136	瓯昌饭店	2426.585	2601.081
137	大世界超市	1597.729	1674.071
138	西城菜场	1575.917	1499.575
139	瓯海二高	2622.893	3004.603
140	水心过境路口	2219.371	2988.244
141	勤奋路花坦小区	2219.371	2988.244
142	勤奋路财政局	2050.328	2333.884
143	粗康桥公交站	2055.781	1575.917
144	粗康桥	1946.721	1346.891
145	百里路勤奋路口	1990.345	1150.583
146	人才大厦	7770.525	2731.953
147	杨府山南大门	8692.082	2557.457
148	物华天宝	8304.919	3331.783
149	均瑶宾馆对面	6249.138	2432.038
150	滨江街道办事处	7208.866	2808.295
151	金讯达大厦	6036.471	1461.404
152	迎潮大厦	8119.517	4395.118

153	妇女儿童中心	8092.252	4406.024
154	黎明路电信大厦	4929.512	1941.268
155	时代海景	1723.148	1057.882
156	三桥下	441.693	1057.882
157	广化街道	468.958	1183.301
158	过境路黄龙	1248.737	2159.388
159	公交集团	1352.344	1810.396
160	半腰桥特陶市场	1417.78	1788.584
161	黄龙商贸城北	736.155	1706.789
162	望江路广化桥路口	1276.002	1156.036
163	九山公园	2835.56	2290.26
164	温四中	2421.132	1701.336
165	过境路宽带路口	3108.21	4144.28
166	南郊派出所	4896.794	4231.528
167	双龙路王子花苑	5714.744	4515.084
168	上村小区	5572.966	2644.705
169	帆影广场对面	1565.011	1145.13
170	银都花苑	4018.861	3402.672
171	大士门石坦小学	2982.791	1799.49
172	东方灯具市场	5120.367	1526.84
173	渡江美景园	3075.492	1483.216
174	信河嘉会里路口	3075.492	1483.216
175	百里小学	2268.448	1128.771
176	马鞍池南	4084.297	3222.723
177	都市花苑	5938.317	2661.064
178	鹿城区公安局	3669.869	1575.917
179	国际贸易中心	5420.282	2001.251
180	繁华公寓	5534.795	1346.891
181	鹿城路嘉乐迪	2252.089	2028.516

4) 借车站点的最近与最远车站点及距离

借车站点	欧式距离				直角边距离			
	最近还车站点	最近还车距离(米)	最远还车站点	最远还车距离(米)	最近还车站点	最近还车距离(米)	最远还车站点	最远还车距离(米)
1	2	220.6	123	5223.4	2	310.8	18	7159.8
2	28	472.3	142	6482.8	8	550.8	142	8081.3
3	11	312.0	148	4499.8	11	338.1	111	5578.4
4	71	269.4	72	5026.0	71	376.3	67	6472.7
5	175	282.1	114	8008.0	175	398.1	114	9602.7
6	8	324.0	140	6364.0	8	359.9	140	7067.1
7	8	245.9	131	5520.0	8	261.7	38	6549.1
8	7	245.9	133	5656.3	7	261.7	133	6778.1
9	43	123.0	73	5979.1	129	141.8	72	7399.7
10	21	239.0	115	5272.1	21	321.7	1	7018.0

11	3	312.0	116	6214.2	3	338.1	116	7001.7
12	78	208.4	160	4352.7	78	229.0	160	5720.2
13	11	284.7	72	5490.8	11	398.1	72	5954.7
14	79	359.9	157	5830.9	79	365.4	157	8010.5
15	75	531.2	144	4517.9	75	730.7	144	6309.1
16	166	319.3	58	4653.4	166	359.9	138	5692.9
17	18	76.3	76	5564.6	18	76.3	76	7028.9
18	17	76.3	114	6707.1	17	76.3	114	7639.7
19	71	215.2	113	5940.1	71	245.4	113	6925.3
20	23	270.9	160	4560.9	12	365.4	160	6085.5
21	10	239.0	116	5859.0	82	272.7	58	7666.9
22	68	341.6	137	5591.9	68	387.2	137	7776.0
23	20	270.9	159	4872.8	79	354.4	159	6510.9
24	26	129.3	157	6296.8	26	174.5	157	8468.5
25	98	385.0	181	4861.5	98	458.1	36	6614.5
26	24	129.3	175	4914.1	24	174.5	175	6898.0
27	85	436.2	160	6424.9	85	610.7	160	8555.8
28	2	472.3	33	5923.6	2	659.8	33	7934.1
29	50	141.5	36	5357.7	50	190.9	36	7181.6
30	29	310.2	5	5915.2	29	359.9	5	8026.8
31	70	245.9	36	4979.1	70	261.7	36	7028.9
32	59	234.5	157	5342.3	59	294.5	157	6101.9
33	34	176.6	72	6414.2	34	201.8	72	8261.3
34	33	176.6	1	5556.4	33	201.8	1	7503.3
35	174	117.5	152	5917.0	174	152.7	152	8108.6
36	5	368.8	119	6565.7	5	485.3	152	8659.4
37	33	233.0	150	3953.5	135	289.0	22	5556.6
38	21	351.1	114	6471.2	10	392.6	114	7836.0
39	40	456.6	120	6404.8	38	507.1	152	7885.0
40	39	456.6	120	6234.9	41	545.3	112	7836.0
41	39	374.1	152	6013.8	39	523.5	152	8408.5
42	154	69.8	114	5419.2	154	98.2	114	6091.0
43	9	123.0	152	4957.0	9	147.2	152	6554.5
44	19	226.9	120	5683.1	19	316.3	72	7399.7
45	46	173.9	67	4602.7	46	245.4	67	5491.2
46	47	496.5	72	4583.4	47	512.6	72	5496.6
47	4	344.2	114	5480.3	4	485.3	114	5496.6
48	104	269.4	152	4668.1	104	376.3	152	6598.1
49	168	341.6	114	4999.5	168	452.6	113	5529.3
50	29	141.5	87	5482.3	29	190.9	87	6063.7
51	76	145.5	141	6487.0	76	174.5	36	8255.8
52	69	253.7	160	5391.2	69	289.0	160	6990.7
53	4	389.5	112	4993.7	65	414.4	152	5518.4
54	170	286.9	72	5559.2	170	327.2	67	6369.1

55	60	144.9	72	5349.0	60	185.4	72	7257.9
56	18	199.0	73	6136.6	18	229.0	28	7563.3
57	42	173.3	120	4968.6	42	207.2	120	5365.8
58	73	397.0	131	6740.2	73	490.8	131	7012.6
59	32	234.5	161	4968.1	32	294.5	67	5556.6
60	55	144.9	120	5209.4	55	185.4	112	6009.2
61	104	251.2	120	4758.9	104	338.1	152	6200.1
62	71	238.2	76	4839.4	71	294.5	76	5834.7
63	78	301.6	157	5354.4	78	359.9	157	6859.9
64	55	268.5	120	5582.7	55	365.4	148	5807.4
65	128	207.6	120	6117.5	128	250.8	120	7028.9
66	77	409.9	144	4402.0	77	436.2	144	5883.8
67	72	131.3	165	6033.7	72	185.4	154	6592.7
68	81	207.5	158	5913.8	81	218.1	158	7928.7
69	52	253.7	87	4393.5	52	289.0	40	5453.0
70	31	245.9	162	6007.5	31	261.7	162	8190.4
71	19	215.2	58	5030.5	19	245.4	58	6205.5
72	67	131.3	130	6151.4	67	185.4	9	7399.7
73	58	397.0	4	5170.7	58	490.8	4	6101.9
74	45	392.7	76	4431.7	45	550.8	96	5153.1
75	166	322.5	144	4342.3	166	343.5	144	6134.6
76	51	145.5	140	6630.7	51	174.5	35	7879.6
77	69	279.4	169	5192.9	69	349.0	169	6849.0
78	12	208.4	120	4702.9	12	229.0	120	5992.8
79	23	288.4	120	4751.7	23	354.4	175	6630.8
80	16	487.7	116	5700.3	16	654.4	157	6898.0
81	68	207.5	174	4931.8	68	218.1	174	6974.4
82	21	246.9	73	6030.4	21	272.7	73	8157.7
83	167	411.3	113	4697.6	167	501.7	35	6325.5
84	26	342.5	141	4148.8	26	392.6	34	5589.3
85	27	436.2	173	5294.7	30	594.4	40	7394.3
86	80	764.3	67	5402.8	80	1041.5	67	5954.7
87	140	317.0	148	6258.8	142	398.1	148	6854.4
88	181	193.4	51	6418.7	181	272.7	152	7961.4
89	163	222.0	70	3875.0	163	299.9	70	5196.7
90	92	420.3	164	3551.7	151	545.3	7	4733.2
91	104	270.7	112	4784.6	104	343.5	112	6483.6
92	180	227.9	156	5318.7	180	294.5	67	6287.3
93	117	318.6	137	4854.1	117	354.4	136	5398.5
94	118	376.1	138	5731.8	95	414.4	138	6216.4
95	99	214.3	87	5184.6	99	261.7	141	5632.9
96	102	188.0	87	6073.1	102	234.5	87	6549.1
97	96	465.6	35	5155.5	96	507.1	35	5513.0
98	106	263.8	169	5606.8	106	294.5	169	7301.6

99	95	214.3	88	4818.9	95	261.7	88	5240.3
100	150	210.7	159	6016.6	150	245.4	159	7099.8
101	149	515.4	156	6353.7	127	539.8	156	7372.5
102	96	188.0	160	6549.9	96	234.5	145	7072.5
103	117	308.2	157	5885.8	117	370.8	157	6489.1
104	61	251.2	72	5345.1	61	338.1	72	7476.1
105	50	219.0	173	4595.6	50	278.1	173	6009.2
106	98	263.8	159	5504.2	98	294.5	159	6554.5
107	103	330.5	158	4870.1	32	447.1	165	5071.3
108	0	0.0	1	0.0	0	0.0	1	0.0
109	148	329.2	87	6089.2	148	441.7	88	6641.8
110	147	243.2	82	4741.2	147	343.5	82	6330.9
111	102	290.7	124	5110.8	102	381.7	40	6140.1
112	115	241.4	33	6103.0	115	305.4	82	7356.1
113	114	424.7	56	6620.1	114	507.1	56	7732.4
114	116	670.5	130	7074.7	116	921.6	86	7514.2
115	112	241.4	158	7982.4	112	305.4	158	9035.6
116	112	458.1	143	7772.6	112	458.1	143	9117.4
117	103	308.2	138	4815.0	103	370.8	67	5551.2
118	102	345.4	158	6363.7	102	409.0	165	6456.4
119	112	333.3	65	5300.3	112	409.0	65	5300.3
120	114	1037.9	155	8181.6	114	1346.9	155	8844.8
121	122	188.3	6	5517.5	122	218.1	6	5736.6
122	121	188.3	72	6060.9	121	218.1	67	6772.6
123	124	175.3	51	5688.9	124	190.9	152	6216.4
124	123	175.3	51	5711.7	123	190.9	51	6003.8
125	126	272.7	89	3332.7	126	321.7	82	4373.3
126	125	272.7	142	4223.1	149	305.4	142	4602.3
127	106	274.6	159	5420.1	106	305.4	159	6249.1
128	65	207.6	112	5339.8	65	250.8	152	5682.0
129	9	126.5	81	3779.2	9	141.8	81	5338.5
130	124	215.7	114	7074.7	124	278.1	114	7394.3
131	132	335.2	110	6093.9	132	436.2	152	6707.2
132	131	335.2	114	7973.1	131	436.2	114	8855.7
133	163	192.9	152	5759.7	163	272.7	152	7377.9
134	139	260.9	120	7109.9	139	349.0	120	8157.7
135	37	249.2	152	5159.4	37	289.0	152	6734.5
136	133	325.4	148	5923.6	133	447.1	148	6609.0
137	138	175.9	152	7066.7	138	196.3	152	9242.8
138	137	175.9	152	7155.6	137	196.3	152	9439.1
139	134	260.9	58	6605.1	134	349.0	58	7192.5
140	87	317.0	102	5776.8	139	419.9	102	6456.4
141	87	317.0	51	6487.0	139	419.9	152	7307.0
142	181	366.0	115	7164.4	87	398.1	152	8130.4

143	144	253.7	152	6687.1	144	338.1	152	8882.9
144	145	201.1	152	6884.4	145	239.9	152	9221.0
145	144	201.1	120	7907.8	144	239.9	152	9373.7
146	111	357.0	140	5557.1	111	496.2	40	6047.4
147	110	243.2	164	6329.1	110	343.5	164	7127.1
148	109	329.2	34	5335.8	109	441.7	39	7007.1
149	126	300.0	157	5913.5	126	305.4	157	7028.9
150	100	210.7	134	4352.3	99	223.6	82	5082.2
151	92	290.9	169	4482.6	92	321.7	67	5965.6
152	100	1632.3	156	8371.7	100	2252.1	156	11015.1
153	117	3349.0	161	7835.7	117	4586.0	161	10055.3
154	42	69.8	72	4866.6	42	98.2	72	6581.8
155	169	180.6	152	7214.6	169	245.4	152	9733.6
156	162	840.1	153	8351.1	162	932.5	153	10998.7
157	161	587.7	152	8297.4	161	790.7	152	10862.4
158	159	364.0	115	7982.4	159	452.6	115	9035.6
159	160	69.0	152	7244.0	160	87.2	152	9351.9
160	159	69.0	152	7190.8	159	87.2	152	9308.3
161	157	587.7	152	7857.6	159	719.8	152	10071.7
162	138	456.0	52	5779.7	155	545.3	52	7765.1
163	133	192.9	152	5687.8	133	272.7	152	7388.8
164	181	368.3	152	6303.0	88	474.4	152	8392.2
165	11	480.9	67	6033.7	11	654.4	67	6211.0
166	16	319.3	143	3888.9	75	343.5	143	5496.6
167	79	315.6	137	5002.1	84	392.6	137	6958.0
168	49	341.6	120	4394.4	177	381.7	72	5234.9
169	155	180.6	120	8332.4	155	245.4	152	9804.5
170	54	286.9	115	5132.4	54	327.2	115	5240.3
171	34	212.7	152	5755.3	34	218.1	152	7732.4
172	61	366.8	161	4387.9	61	398.1	152	5867.4
173	39	553.7	120	6805.1	178	687.1	152	7955.9
174	35	117.5	51	5940.8	35	152.7	152	7955.9
175	145	279.0	51	6819.7	145	299.9	152	9117.4
176	128	340.4	51	4612.0	128	474.4	6	4967.7
177	126	324.1	160	4604.0	168	381.7	160	5393.0
178	17	177.5	8	5018.2	17	250.8	8	6669.0
179	32	327.2	114	4889.3	32	332.6	113	5872.9
180	92	227.9	157	5068.5	92	294.5	76	5403.9
181	88	193.4	51	6588.4	88	272.7	152	8234.0

5) 各站点的借车和还车高峰时间表

站点	借车高峰	还车高峰
1	17:36, 20:36	19:23
2	8:12, 16:23	8:12, 15:35

3	17:11	8:12
4	7:48, 13:35, 14:23, 17:11, 19:11	9:12, 11:24, 12:47, 16:00, 17:36
5	17:11	10:12, 10:48, 18:00, 19:11, 19:48
6	8:12, 17:36, 18:36	17:23, 20:23
7	17:48	8:12
8	17:36	8:24
9	8:00, 13:47, 16:11, 17:48	8:24, 11:00, 15:47, 17:48
10	8:00, 15:24, 17:11	8:24, 17:23
11	17:11	8:12
12	7:48, 17:23	17:36
13	10:00	8:24, 10:00
14	7:48	7:48, 17:23
15	8:00, 10:36, 17:11	8:12, 17:23, 18:00
16	7:24, 12:35, 17:23	17:48
17	9:36, 17:23	8:12, 17:23
18	17:23	8:24, 17:23
19	8:00, 13:00, 15:35, 17:23, 19:23	8:24, 11:36, 12:23, 13:00, 15:47, 17:23, 19:11
20	8:00, 17:36	17:48
21	8:00, 9:48, 17:11	8:35, 17:23
22	8:12, 17:36	8:24, 17:23
23	8:00	17:23
24	7:48	17:23
25	8:00, 17:48	8:24, 17:00
26	7:48, 17:48	17:48
27	7:48	17:36
28	8:00, 17:00, 19:23	7:36, 17:48
29	16:36	7:48
30	8:12, 17:36	8:00, 16:36, 17:48, 20:11
31	7:48, 17:11	8:24, 17:11
32	11:12, 13:47, 15:47, 17:11	9:00, 10:36, 13:35, 15:35, 17:36
33	8:24, 10:36, 16:48	8:00, 10:00
34	7:48, 17:23, 19:23	8:12, 17:23, 19:23
35	8:00	17:23
	8:00, 9:48, 11:36, 13:47, 14:35,	
36	16:48, 18:11	8:24, 12:35, 13:23, 14:35, 17:48
37	8:24, 10:00, 20:36	8:12, 9:00, 19:23
	8:00, 9:36, 10:12, 11:12, 15:24,	
38	17:00	8:00, 14:47, 15:24, 17:11
39	8:12, 17:23	8:00, 17:36
40	8:00, 9:24, 10:12, 16:11, 17:48	17:23, 19:00
		8:12, 11:00, 11:48, 12:23, 14:23, 15:24, 16:23, 17:00,
41	7:48, 10:24, 14:11, 19:11	17:36, 18:36, 19:11, 20:00, 20:36
42	12:00, 17:11	8:24
43	14:00, 17:00	10:24, 11:12, 14:11, 15:24, 17:48, 20:48

44	11:12, 17:11	8:24, 11:24, 13:35, 17:23
45	8:12	8:00, 8:35, 17:36, 19:23
46	8:00, 17:36	8:00, 11:48, 14:35, 17:36
47	8:12, 16:11, 17:23	8:24, 17:48
48	8:12, 12:00, 17:36	8:35, 17:48
49	8:00, 11:24, 17:00	7:48
50	8:12, 11:12, 17:00	8:00, 9:36, 14:00, 16:11, 17:48, 19:00
51	17:11	8:12, 12:47, 17:48
52	17:36	8:24, 17:36
53	8:00, 17:11, 20:23	8:35, 10:36, 17:36, 19:11
54	8:00, 15:47, 17:23	8:12, 11:12, 17:36
55	8:00, 9:48, 15:11, 17:00	8:12
56	8:00, 17:23	8:00, 9:12, 11:36, 15:11, 17:36
57	8:12, 17:23	8:24, 17:23
58	11:36, 17:11	8:00
59	8:00, 11:48, 17:23	8:48, 11:48, 17:23
60	8:12, 9:24, 10:48, 17:23	8:12, 14:47, 17:36
61	8:00, 17:23, 20:11	8:48, 17:48
62	8:35, 17:23	8:35, 17:23
63	8:00, 17:11, 18:48	17:36
64	7:36, 8:12, 17:00	8:12, 15:00, 17:23
65	8:12, 17:23	8:24, 17:36
66	8:00, 17:11, 19:48	17:48, 20:00
67	17:11	8:12
68	17:23	8:24, 17:36
69	8:12, 17:11	8:24, 17:36
70	8:00, 17:23	8:12, 17:23
71	17:23	8:24, 15:24, 17:48
72	17:11, 18:00	8:24, 9:12, 17:48, 20:23
73	8:12, 17:36	8:00
74	8:00, 17:11	8:00, 17:23
75	7:48, 17:00	8:00, 17:36
76	8:12, 17:36	8:24, 17:48
77	17:48	8:24, 17:36
78	8:00	17:48
79	8:00, 17:00	8:00, 17:11
80	17:00	7:48, 18:00
81	8:00, 17:11	8:12, 17:48
82	7:48, 17:11	7:48, 17:36
83	8:00, 17:11	7:48, 14:11, 17:48
84	7:48	17:36
85	11:12, 17:11, 19:48	8:12, 19:11
86	17:36	7:48
87	7:48, 17:36	8:00, 18:00

88	8:00, 17:11	8:24, 17:23
89	8:12, 10:36, 17:00	8:24, 13:47, 17:11
90	7:36, 13:35, 17:11, 19:36	13:35, 15:11, 18:00, 19:36
91	16:00, 17:00, 19:23, 20:36	19:48
92	8:00, 17:00	8:12, 17:36
93	17:23, 20:48	19:23
94	8:00, 17:48	8:35, 17:36
95	8:00	8:12, 16:36, 17:48
96	7:48, 17:00	7:12, 17:36
97	7:12, 8:48, 12:47, 17:00	8:12, 17:11
98	8:12, 17:23	8:12, 17:23
99	8:00	16:00, 17:36
100	8:00	17:48
101	8:12, 17:23	17:36
102	8:00	8:00, 11:12, 17:48
103	8:12, 17:11	8:24, 17:23
104	7:48, 17:48	17:48
105	8:00, 17:23, 20:48	8:12, 17:48
106	8:00	17:23
107	7:48, 17:36	8:12, 17:36
108		
109	8:12, 16:36, 17:48, 18:23, 19:48	8:00, 18:00
110	7:36	7:00, 7:36, 19:00, 20:36
111	8:00, 17:23, 19:00	18:00, 19:23
112	7:48, 10:48, 13:47, 16:11	13:47, 17:23
113	7:48, 19:23	8:00, 17:48
114	7:36	17:48
115	8:00, 17:36	17:48
116	8:00	8:00, 10:12, 17:00, 17:48, 20:48
117	8:35, 17:36	8:24, 17:48
118	7:48	17:48
119	7:48, 14:00, 15:11, 17:11	8:24, 12:35, 15:11, 16:48, 18:00, 19:11
120	11:24, 14:00	9:36
121	8:00	8:00
122	9:00, 11:12, 16:48	8:12, 16:36
123	8:00, 17:36	17:36
124	7:48	11:36, 15:47, 17:23
125	8:12, 13:47, 17:36	8:12
126	17:00	8:12
127	8:12, 17:23	8:00, 17:23
128	8:24, 11:24, 15:24, 17:23 7:36, 8:12, 10:36, 12:23, 13:00,	8:24, 10:00, 17:23
129	13:35, 15:35, 17:11, 20:00	17:23
130	8:12	8:12, 11:12, 17:36

131	7:48	17:36
132	7:36, 17:23	8:00, 12:11, 17:11
133	10:24, 14:47, 17:00	8:35, 9:48, 16:11, 17:36
134	7:48, 9:48, 16:11, 17:36, 19:00	17:36
135	8:00, 9:00, 17:23	8:35, 17:36
	8:00, 10:00, 11:12, 13:47, 17:23,	
136	19:23	8:12, 17:36
137	7:48, 17:23, 19:00	17:36
138	7:36	7:48, 10:00, 15:47, 17:48
139	7:48	11:00, 17:23, 20:23
140	8:24, 10:00, 18:00	15:47, 17:36
141	8:00, 17:11	8:12, 17:36
142	17:36	8:24, 17:23
143	7:36, 14:47, 17:11	17:36
144	7:48, 17:36	17:36
145	8:00, 19:11	8:12, 11:36, 17:36, 20:00
	7:36, 13:47, 16:00, 16:36, 17:11,	
146	19:11	17:23
147	20:00	19:00
148	17:36	8:12
149	15:35, 17:11	8:24, 17:36
150	8:24, 12:00, 17:11	8:12, 16:00, 17:36
151	8:00	8:12, 16:48, 17:36
152	7:48, 10:00, 14:35, 17:11, 19:00	8:35, 17:11
153	18:00, 19:23	8:24, 19:11
154	17:36	8:24, 17:48
155	7:00, 8:12, 15:35, 19:11	15:35, 19:11
156	7:48, 15:35, 16:23, 17:00	10:36
157	8:00, 10:12, 14:35, 17:11	8:00, 10:12, 18:11
158	17:11	8:24, 17:36
159	8:12, 9:00, 15:00, 17:23	8:12, 17:23
160	8:24, 17:23	8:24, 17:36
161	17:00	9:12
162	10:12, 14:47, 15:47, 16:23	10:00, 16:11
163	8:24, 10:00, 17:23	8:12
164	8:35, 16:48	7:12, 17:11
165	8:00, 10:00, 16:48, 17:36	8:12, 17:36
166	8:12	8:12
167	7:36, 17:23	8:24, 17:48
168	8:00, 17:36	8:00, 11:00, 17:36
169	8:00, 13:00, 15:00, 17:00	17:36
170	7:24, 17:23	17:48
171	8:00, 17:00	8:12, 11:12, 15:35, 17:23
172	7:48, 17:23	8:24, 17:23

173	7:48, 17:00 7:24, 8:00, 8:35, 9:12, 9:48, 13:35,	8:24, 9:48, 17:00, 18:36, 19:11, 19:48
174	16:48, 17:23	8:24, 13:47, 17:48
175	8:24, 11:36, 17:11	9:36, 17:23
176	7:48	8:48, 15:24, 17:23, 19:00, 19:48
177	8:00, 17:11	8:12, 17:23
178	17:23	8:12, 11:48
179	17:00	8:24
180	7:36, 9:00, 13:47	17:00, 17:36
181	8:00, 9:48, 17:11	8:24, 17:23

附录二¹:

本论文中采用的 **MATLAB** 代码如下:

```
% function csv_data_load
%csv_data_load
day_no_vec = 1:20;
bike_record = zeros(1000000,10);
bike_item_cnt = 0;
for i_day = day_no_vec
    filename = sprintf('day%02d.csv',i_day);
    fid = fopen(filename,'r');
    tline = fgetl(fid);
    while(~feof(fid))
        tline = fgetl(fid);
        comma_index = find(tline == ',');
        line_item{1} = tline(1:comma_index(1)-1);
        line_item{15} = tline(comma_index(14)+1:end);
        for i_line_item = 2:14
            line_item{i_line_item} =
tline(comma_index(i_line_item-1)+1:comma_index(i_line_item)-1);
        end
        item_no = str2num(line_item{1});
        item_no = i_day * 1e5 + item_no
        bike_sn = str2num(line_item{3});
        start_station = str2num(line_item{5});
        start_time = datenum(line_item{7});
        end_station = str2num(line_item{9});
        end_time = datenum(line_item{11});
        duration = str2num(line_item{12});
        user_sn = str2num(line_item{15});
        start_pos = str2num(line_item{6});
        end_pos = str2num(line_item{10});
        if ~isempty(start_time) && ...
            ~isempty(end_time) %&& ...
%            start_time - datenum([2012 11 i_day]) < 1 && ...
%            end_time - datenum([2012 11 i_day]) < 20
        bike_item_cnt = bike_item_cnt + 1;
        bike_record(bike_item_cnt,:) = [
            item_no,
            bike_sn,
            start_station,
            start_time,
```

¹ 附录二仅包含于电子版中, 打印版中不包含。

```

        end_station,
        end_time,
        duration,
        user_sn,
        start_pos,
        end_pos,
    ];
    end
end
fclose(fid);
end
bike_record = bike_record(1:bike_item_cnt,:);
save bike_record.mat bike_record

% load data & pre-process
% load data
clear all;
load('bike_record.mat');
n_item = size(bike_record,1);
item_no = bike_record(:,1);
bike_sn = bike_record(:,2);
start_station = bike_record(:,3);
start_time = bike_record(:,4);
end_station = bike_record(:,5);
end_time = bike_record(:,6);
duration = bike_record(:,7);
user_sn = bike_record(:,8);
start_pos = bike_record(:,9);
end_pos = bike_record(:,10);

% error analyze
% time
min_start_time = datevec(min(start_time));
max_start_time = datevec(max(start_time));
min_end_time = datevec(min(end_time));
max_end_time = datevec(max(end_time));

% duration
duration_float = (end_time - start_time) * 24 * 60;
duration_float_max = max(duration_float);
time_error_index = find(duration_float < 0);
time_error_item_no = item_no(time_error_index);
duration_error = abs(duration_float - duration);
duration_error_max = max(duration_error);

```

```

duration_error_index = find(duration_error > 2);
duration_error_item_no = item_no(duration_error_index);
duration_unique = unique(duration);
duration_too_long_index = find(duration > 1000);
duration_too_long_item_no = item_no(duration_too_long_index);

% station
start_station_unique = unique(start_station);
end_station_unique = unique(end_station);
start_station_error_index = find(start_station > 181);
start_station_error_item_no = item_no(start_station_error_index);
end_station_error_index = find(end_station > 181);
end_station_error_item_no = item_no(end_station_error_index);

% position
start_pos_unique = unique(start_pos);
end_pos_unique = unique(end_pos);

% error data delete
correct_index = find(...
    duration_float > 0 & ...
    duration_error < 2 & ...
    start_station <= 181 & ...
    end_station <= 181);

% save data
item_no = item_no(correct_index);
bike_sn = bike_sn(correct_index);
start_station = start_station(correct_index);
start_time = start_time(correct_index);
end_station = end_station(correct_index);
end_time = end_time(correct_index);
duration = duration(correct_index);
user_sn = user_sn(correct_index);
start_pos = start_pos(correct_index);
end_pos = end_pos(correct_index);
duration_float = duration_float(correct_index);

save bike_record_corrected.mat item_no bike_sn start_station start_time end_station
end_time duration user_sn start_pos end_pos duration_float

clear all;
load bike_record_corrected;

```

```

start_time_vec = datevec(start_time);
start_day = start_time_vec(:,3);
end_time_vec = datevec(end_time);
end_day = end_time_vec(:,3);
start_station_day_freq = zeros(181,20);
end_station_day_freq = zeros(181,20);
for i_day = 1:20
    day_start_station = start_station(find(start_day == i_day));
    day_end_station = end_station(find(end_day == i_day));
    for i_station = 1:181
        start_station_day_freq(i_station,i_day) = sum(day_start_station ==
i_station);
        end_station_day_freq(i_station,i_day) = sum(day_end_station == i_station);
    end
end
start_station_total_freq = sum(start_station_day_freq,2);
end_station_total_freq = sum(end_station_day_freq,2);

[start_sort_freq start_sort_index] = sort(start_station_total_freq,'descend');
[end_sort_freq end_sort_index] = sort(end_station_total_freq,'descend');

save frequency_anlyz.mat start_station_day_freq end_station_day_freq
start_station_total_freq ...
    end_station_total_freq start_sort_freq start_sort_index end_sort_freq
end_sort_index

fid = fopen('freq_outcome.txt','w');
fprintf(fid,'站点 & ');
for j = 1:10
    fprintf(fid,' $a_{i%d}$ & ',j);
end
fprintf(fid,' 累计 \\\ \r\n');
for i = 1:20
    fprintf(fid,'%d & ',i);
    for j = 1:10
        fprintf(fid,'%d & ',start_station_day_freq(i,j));
    end
    fprintf(fid,'%d \\\ \r\n',start_station_total_freq(i));
end

fprintf(fid,' \r\n\r\n\r\n');
fprintf(fid,'站点 & ');
for j = 1:10
    fprintf(fid,' $b_{i%d}$ & ',j);

```



```

end
fprintf(fid,'累计 \\\ \r\n');
for i = 1:20
    fprintf(fid,'%d & ',i);
    for j = 1:10
        fprintf(fid,'%d & ',end_station_day_freq(i,j));
    end
    fprintf(fid,'%d \\\ \r\n',end_station_total_freq(i));
end

fprintf(fid,' \r\n\r\n\r\n');

fprintf(fid,'排序 & 借车站点 & 借车频次 & 排序 & 还车站点 & 还车频次 \\\ \r\n');
for i = 1:20
    fprintf(fid,'%d & %d & %d & %d & %d & %d \\\ \r\n',...
        i, start_sort_index(i), start_sort_freq(i),...
        i, end_sort_index(i), end_sort_freq(i));
end

fclose(fid);

% duration_distribution_anlyz
clear all;
load bike_record_corrected.mat
min_duration = min(duration_float);
max_duration = max(duration_float);
min_duration_sec = min_duration * 60;
max_duration_sec = max_duration * 60;
max_duration_day = floor(max_duration_sec / 24 / 3600);
max_duration_sec = mod(max_duration_sec, 24 * 3600);
max_duration_hour = floor(max_duration_sec / 3600);
max_duration_sec = mod(max_duration_sec, 3600);
max_duration_min = floor(max_duration_sec / 60);
max_duration_sec = mod(max_duration_sec, 60);
coarse_x = 50:100:max_duration;
coarse_hist = hist(duration_float, coarse_x);
coarse_hist_freq = coarse_hist / length(duration);
coarse_hist_freq_cum = cumsum(coarse_hist_freq);

duration2 = duration_float(find(duration_float < 100 & duration_float >= 1));
fine_x = 1.5:1:100;
fine_hist = hist(duration2, fine_x);
fine_hist_freq = fine_hist / length(duration2);

```

```

duration2_std = std(duration2);
mean_duration2 = mean(duration2);

figure;
bar(fine_x, fine_hist);
xlabel('用户用车时间 (分钟)');
xlim([1 100]);
ylabel('频次');

lambda = 1/100;
d = fine_x;
pdf_d = lambda * exp(- lambda * d);
pdf_d = pdf_d / sum(pdf_d);
figure;
plot(d, pdf_d);
xlabel('用户需要出行的时间 (分钟)');
ylabel('密度函数p(T=t)');

pdf_y = fine_hist_freq;
pdf_y_d = pdf_y ./ pdf_d;
% figure;
% plot(d, pdf_y_d);

n = 0.01:0.01:1000;
% n = 0:0.01:5;
cdf_n = normcdf(n);
figure;
plot(cdf_n, n);
xlabel('x');
ylabel('正态累计分布的反函数Q^{-1}(x)');

u_d = zeros(1, 99);

for i = 1:99;
    [min_error index] = min(abs(cdf_n - pdf_y_d(i)));
    u_d(i) = index / 100;
end
figure;
plot(d, u_d);
xlabel('用户用车时间 (分钟)');
ylabel('效用函数u_0(t)');

% indv_freq_anlyz
clear all;

```

```

load bike_record_corrected.mat

start_time_vec = datevec(start_time);
start_day = start_time_vec(:,3);
end_time_vec = datevec(end_time);
end_day = end_time_vec(:,3);

day_user_amount = zeros(1,20);
for i = 1:20;
    user_sn_day = [user_sn(find(start_day == i))
        user_sn(find(end_day == i))];
    day_user_amount(i) = length(unique(user_sn_day));
end

user_sn_sort = sort(user_sn);
user_sn_sort_diff = diff(user_sn_sort);
change_i = find(user_sn_sort_diff~=0);
indv_freq = diff([0;change_i]);
coarse_x = 25:50:639;
coarse_hist = hist(indv_freq,coarse_x);
coarse_hist_freq = coarse_hist / length(indv_freq);
coarse_hist_freq_cum = cumsum(coarse_hist_freq);

indv_freq2 = indv_freq(find(indv_freq<100));
fine_x = 0.5:1:100;
fine_hist = hist(indv_freq2,fine_x);
fine_hist_freq = fine_hist / length(indv_freq2);

figure;
bar(fine_x, fine_hist);
xlabel('借车卡借车次数');
xlim([1 100]);
ylabel('频次');

station_coor = [
    1488    803
    1514    772
    704 712
    828 454
    450 168
    1566    685
    1462    681
    1507    678

```

682 428
796 208
647 717
993 644
689 686
1000 772
843 918
840 768
651 313
665 313
793 375
1014 690
776 169
1138 888
1052 722
1115 741
1255 665
1126 762
1325 832
1565 842
1392 632
1402 688
1195 801
1054 368
591 361
586 329
556 252
512 195
606 401
729 213
642 207
683 134
601 152
896 366
660 433
769 341
852 610
873 586
876 495
864 221
965 459
1380 609
1594 627
1180 690

778 505
685 616
831 305
671 349
903 397
1687 664
1041 409
856 296
945 213
757 426
1009 552
788 281
704 503
1083 600
1676 793
1200 879
1173 644
1192 756
799 414
1692 775
1757 644
808 667
894 835
1620 633
1158 595
997 606
1066 773
760 808
1198 917
781 124
1067 901
1116 824
1389 784
691 686
381 496
440 395
556 439
1084 191
920 145
1054 262
1179 221
1336 368
1330 438
1491 405

1499 320
1240 596
1320 476
1329 553
1228 399
1458 415
1159 334
904 192
1342 596
1234 548
1122 382
1445 601
1496 557
1564 502
1479 464
1692 562
1864 574
1881 498
1678 604
1776 562
1172 279
1396 402
1676 503
1811 321
555 644
589 638
551 604
548 572
1157 550
1147 501
1228 498
713 540
685 451
585 558
452 613
429 670
496 446
524 530
614 446
445 477
293 307
289 275
481 551
407 548

407 548
376 428
377 289
357 247
365 211
1425 501
1594 469
1523 611
1146 446
1322 515
1107 268
1489 806
1484 808
904 356
316 194
81 194
86 217
229 396
248 332
260 328
135 313
234 212
520 420
444 312
570 760
898 776
1048 828
1022 485
287 210
737 624
547 330
939 280
564 272
564 272
416 207
749 591
1089 488
673 289
994 367
1015 247
413 372];

```
station_coor_m = station_coor * 5.4530;  
station_coor_g = station_coor;
```

```

station_coor_g(:,1) = 1.2623 * (station_coor(:,1) - 373) + 468;
station_coor_g(:,2) = 1.3890 * (station_coor(:,2) - 340) + 343;
station_coor_g = round(station_coor_g);
station_coor_f2 = station_coor;

save station_coor.mat station_coor_f2 station_coor_m station_coor_g

load station_coor.mat
% station_coor(:,1)=station_coor(:,1)-1000;
station_coor = station_coor_g;

map_google = imread('lucheng_map.bmp');
figure;
imshow(map_google);

% map_cdata = imread('map.jpg');
% figure;
% imshow(map_cdata);

% imshow(map_cdata(:,1000:end,:));
fontsize = 10;
hold on
for i = 1:180
%     switch(mod(i,4))
%         case 0,
%             plot(station_coor(i,1),station_coor(i,2),'r*');
%
%             text(station_coor(i,1)+5,station_coor(i,2)+5,num2str(i),'FontSize',fontsize,'Color', 'red');
%         case 1,
%             plot(station_coor(i,1),station_coor(i,2),'b*');
%
%             text(station_coor(i,1)+5,station_coor(i,2)+5,num2str(i),'FontSize',fontsize,'Color', 'blue');
%         case 2,
%             plot(station_coor(i,1),station_coor(i,2),'g*');
%
%             text(station_coor(i,1)+5,station_coor(i,2)+5,num2str(i),'FontSize',fontsize,'Color', 'green');
%         case 3,
%             plot(station_coor(i,1),station_coor(i,2),'k*');
%
%             text(station_coor(i,1)+5,station_coor(i,2)+5,num2str(i),'FontSize',fontsize,'Color

```



```

','black');
%     end
    plot(station_coor(i,1),station_coor(i,2),'k*');

text(station_coor(i,1)+5,station_coor(i,2)+5,num2str(i),'FontSize',fontsize,'Color',
','black');
end
hold off

% busy_day_data
clear all;
load bike_record_corrected.mat
load frequency_anlyz.mat
ss_station_sum = sum(start_station_day_freq,1);
es_station_sum = sum(end_station_day_freq,1);
day_data = ss_station_sum + es_station_sum;
[day_data_sort index] = sort(day_data,'descend');
busy_day = index(1);

start_time_vec = datevec(start_time);
start_day = start_time_vec(:,3);
end_time_vec = datevec(end_time);
end_day = end_time_vec(:,3);
start_station_day_freq = zeros(181,20);
end_station_day_freq = zeros(181,20);

item_no = item_no(find(start_day == busy_day & end_day == busy_day));
bike_sn = bike_sn(find(start_day == busy_day & end_day == busy_day));
start_station = start_station(find(start_day == busy_day & end_day == busy_day));
start_time = start_time(find(start_day == busy_day & end_day == busy_day));
end_station = end_station(find(start_day == busy_day & end_day == busy_day));
end_time = end_time(find(start_day == busy_day & end_day == busy_day));
duration = duration(find(start_day == busy_day & end_day == busy_day));
user_sn = user_sn(find(start_day == busy_day & end_day == busy_day));
start_pos = start_pos(find(start_day == busy_day & end_day == busy_day));
end_pos = end_pos(find(start_day == busy_day & end_day == busy_day));
duration_float = duration_float(find(start_day == busy_day & end_day == busy_day));

save bike_record_busy_day.mat item_no bike_sn start_station ...
    start_time end_station end_time duration user_sn ...
    start_pos end_pos duration_float

```

```

% distance_calc
clear all;
load bike_record_busy_day.mat
load station_coor.mat

station_coor = station_coor_m;

distance_mat_norm2 = zeros(181,181);
distance_mat_norm1 = zeros(181,181);
for i = 1:181
    for j = 1:181
        distance_mat_norm2(i,j) = norm(station_coor(i,:)-station_coor(j,:));
        distance_mat_norm1(i,j) = norm(station_coor(i,:)-station_coor(j,:),1);
    end
end

distance_mat = distance_mat_norm1;
[farest_0 farest_index] = max(distance_mat);

start_station0 = start_station(find(duration_float >=1));
end_station0 = end_station(find(duration_float >=1));
item_length = length(start_station0);

station_connected = zeros(181,181);
for i = 1:item_length
    i_start = start_station0(i);
    i_end = end_station0(i);
    station_connected(i_start,i_end) = station_connected(i_start,i_end) + 1;
end

distance_mat_path = distance_mat.*(station_connected>0);
[farest_real farest_real_index] = max(distance_mat_path,[],2);
nearest_real = zeros(181,1);
nearest_real_index = zeros(181,1);
for i = 1:181
    distance_mat_path_row = distance_mat_path(i,:);
    i_nz = find(distance_mat_path_row~=0);
    nz = distance_mat_path_row(i_nz);
    [nearest_real0 nearest_real_index0] = min(nz);
    if ~isempty(nearest_real0)
        nearest_real(i) = nearest_real0;
        nearest_real_index(i) = i_nz(nearest_real_index0);
    end
end

```

```

end

% time_ditribution_anlyz
clear all;
load bike_record_busy_day.mat
load frequency_anlyz.mat

start_station_total_freq = start_station_day_freq(:,20);
end_station_total_freq = end_station_day_freq(:,20);

[start_sort_freq start_sort_index] = sort(start_station_total_freq,'descend');
[end_sort_freq end_sort_index] = sort(end_station_total_freq,'descend');

max_start = start_sort_index(1);
max_end = end_sort_index(1);

start_time0 = start_time(find(start_station == max_start));
duration_ms = duration_float(find(start_station == max_start));

end_time0 = end_time(find(end_station == max_end));
duration_me = duration_float(find(end_station == max_end));

start_time0 = mod(start_time0,1) * 24;
time_x = 0:0.2:24;
start_time_hist = hist(start_time0,time_x);
figure;
plot(time_x,start_time_hist);
xlim([0 24]);
xlabel('时间（小时）');
ylabel('借车频次');
title('最大借车站点（42号）的借车频次分布')

end_time0 = mod(end_time0,1) * 24;
time_x = 0:0.2:24;
end_time_hist = hist(end_time0,time_x);
figure;
plot(time_x,end_time_hist);
xlim([0 24]);
xlabel('时间（小时）');
ylabel('还车频次');
title('最大还车站点（56号）的还车频次分布')

duration_ms0 = duration_ms(find(duration_ms<100 & duration_ms>=1));

```

```

duration_x = 1.5:1:100;
duration_ms0_hist = hist(duration_ms0,duration_x);
duration_ms0_hist_freq = duration_ms0_hist / length(duration_ms0);
figure;
bar(duration_x,duration_ms0_hist);
xlabel('用户用车时间（分钟）');
xlim([1 100]);
ylabel('频次');
title('最大借车站点（42号）的用户用车时间分布');

duration_me0 = duration_me(find(duration_me<100 & duration_me>=1));
duration_x = 1.5:1:100;
duration_me0_hist = hist(duration_me0,duration_x);
duration_me0_hist_freq = duration_me0_hist / length(duration_me0);
figure;
bar(duration_x,duration_me0_hist);
xlabel('用户用车时间（分钟）');
xlim([1 100]);
ylabel('频次');
title('最大还车站点（56号）的用户用车时间分布');

% start ud calc
lambda = 1/100;
d = duration_x;
pdf_d = lambda * exp( - lambda * d);
pdf_d = pdf_d / sum(pdf_d);
% figure;
% plot(d, pdf_d);

pdf_y = duration_ms0_hist_freq;
pdf_y_d = pdf_y ./ pdf_d;
% figure;
% plot(d, pdf_y_d);

n = 0.01:0.01:100;
cdf_n = normcdf(n);
% figure;
% plot(n, cdf_n);

u_d = zeros(1,99);

for i = 1:99;
    [min_error index] = min(abs(cdf_n - pdf_y_d(i)));

```

```

        u_d(i) = index / 100;
    end
    figure;
    plot(d,u_d);
    xlabel('用户用车时间（分钟）');
    ylabel('效用函数u_0(t)');
    title('最大借车站点（42号）的用户用车时间效用函数');

% end ud calc
lambda = 1/100;
d = duration_x;
pdf_d = lambda * exp( - lambda * d);
pdf_d = pdf_d / sum(pdf_d);
% figure;
% plot(d, pdf_d);

pdf_y = duration_me0_hist_freq;
pdf_y_d = pdf_y ./ pdf_d;
% figure;
% plot(d, pdf_y_d);

n = 0.01:0.01:100;
cdf_n = normcdf(n);
% figure;
% plot(n,cdf_n);

u_d = zeros(1,99);

for i = 1:99;
    [min_error index] = min(abs(cdf_n - pdf_y_d(i)));
    u_d(i) = index / 100;
end
figure;
plot(d,u_d);
xlabel('用户用车时间（分钟）');
ylabel('效用函数u_0(t)');
title('最大还车站点（56号）的用户用车时间效用函数');

% peak_cluster
clear all
load busy_peak.mat

% 计算等效峰值向量

```

```

n_epeak = 20;
s_epeak_mat = zeros(181,n_epeak);
% e_epeak_mat = zeros(181,n_epeak);
for i = 1:181
    peak_time = s_peak_time{i};
    peak_index = s_peak_index{i};
    st_hist_mean_f = start_time_hist_mat(i,:);
    if length(peak_time) ~= 0
        peak_x = st_hist_mean_f(peak_index);
        peak_x = peak_x /sum(peak_x) * n_epeak;
        peak_x_cum = cumsum(peak_x);
        peak_x_cum = round(peak_x_cum);
        i_peak = 1;
        for j = 1:n_epeak
            if j <= peak_x_cum(i_peak)
                s_epeak_mat(i,j) = peak_time(i_peak);
            else
                i_peak = i_peak + 1;
                s_epeak_mat(i,j) = peak_time(i_peak);
            end
        end
    end
end
end

e_epeak_mat = zeros(181,n_epeak);
for i = 1:181
    peak_time = e_peak_time{i};
    peak_index = e_peak_index{i};
    et_hist_mean_f = end_time_hist_mat(i,:);
    if length(peak_time) ~= 0
        peak_x = et_hist_mean_f(peak_index);
        peak_x = peak_x /sum(peak_x) * n_epeak;
        peak_x_cum = cumsum(peak_x);
        peak_x_cum = round(peak_x_cum);
        i_peak = 1;
        for j = 1:n_epeak
            if j <= peak_x_cum(i_peak)
                e_epeak_mat(i,j) = peak_time(i_peak);
            else
                i_peak = i_peak + 1;
                e_epeak_mat(i,j) = peak_time(i_peak);
            end
        end
    end
end
end

```

```

end

idx = kmeans(s_epeak_mat,5);
load station_coor.mat
station_coor = station_coor_g;

map_google = imread('lucheng_map_light.bmp');
figure;
imshow(map_google);
MarkerSize = 40;

hold on
plot(station_coor(idx==1,1),station_coor(idx==1,2),'r.','MarkerSize',MarkerSize);
plot(station_coor(idx==2,1),station_coor(idx==2,2),'g.','MarkerSize',MarkerSize);
plot(station_coor(idx==3,1),station_coor(idx==3,2),'b.','MarkerSize',MarkerSize);
plot(station_coor(idx==4,1),station_coor(idx==4,2),'k.','MarkerSize',MarkerSize);
plot(station_coor(idx==5,1),station_coor(idx==5,2),'c.','MarkerSize',MarkerSize);
hold off

idx = kmeans(e_epeak_mat,5);
load station_coor.mat
station_coor = station_coor_g;

map_google = imread('lucheng_map_light.bmp');
figure;
imshow(map_google);
MarkerSize = 40;

hold on
plot(station_coor(idx==1,1),station_coor(idx==1,2),'r.','MarkerSize',MarkerSize);
plot(station_coor(idx==2,1),station_coor(idx==2,2),'g.','MarkerSize',MarkerSize);
plot(station_coor(idx==3,1),station_coor(idx==3,2),'b.','MarkerSize',MarkerSize);
plot(station_coor(idx==4,1),station_coor(idx==4,2),'k.','MarkerSize',MarkerSize);
plot(station_coor(idx==5,1),station_coor(idx==5,2),'c.','MarkerSize',MarkerSize);
hold off
% peak_display
clear all
load busy_peak.mat
fid = fopen('peak_display.txt','w');
for i=1:181
    peak_time = s_peak_time{i};
    if s_peak_amount(i) == 0
        fprintf(fid,'\r\n');
    end
end

```

```

    for j = 1:s_peak_amount(i)
        fprintf(fid,'%d:%02d',floor(peak_time(j)),floor(60 * mod(peak_time(j),1)));
        if j == s_peak_amount(i)
            fprintf(fid,'\r\n');
        else
            fprintf(fid,', ');
        end
    end
end
fprintf(fid,'\r\n');
fprintf(fid,'\r\n');

for i=1:181
    peak_time = e_peak_time{i};
    if e_peak_amount(i) == 0
        fprintf(fid,'\r\n');
    end
    for j = 1:e_peak_amount(i)
        fprintf(fid,'%d:%02d',floor(peak_time(j)),floor(60 * mod(peak_time(j),1)));
        if j == e_peak_amount(i)
            fprintf(fid,'\r\n');
            break;
        else
            fprintf(fid,', ');
        end
    end
end
end

fclose(fid);

function st_hist_mean_f = mean_filter(start_time_hist,mean_filter_radiu)

st_hist_mean_f = start_time_hist;
% mean_filter_radiu = 1;
for j = 1:length(start_time_hist)
    if j-mean_filter_radiu < 1
        mean_range = 1:j+mean_filter_radiu;
    elseif j+mean_filter_radiu > length(start_time_hist)
        mean_range = j-mean_filter_radiu:length(start_time_hist);
    else
        mean_range = j-mean_filter_radiu:j+mean_filter_radiu;
    end
    st_hist_mean_f(j) = mean(start_time_hist(mean_range));
end

```



```

function peak = find_peak(x)
peak0 = [];
peak_cnt = 0;
x_diff = diff(x);
n = length(x);
x_mid = min(x) + 0.6 * (max(x)-min(x));
for i = 2:n-1
    if x_diff(i-1) >= 0 && x_diff(i) <= 0 && ...
        x_diff(i-1)^2 + x_diff(i)^2 ~=0 && x(i) > x_mid
        peak_cnt = peak_cnt + 1;
        peak0 = [peak0, i];
    end
end
if peak_cnt <= 1
    peak = peak0;
else
    peak_actual_cnt = 1;
    peak = peak0(1);
    peak_set_cnt = 1;
%    new_peak_flag = 0;
    i_peak = 2;
    while i_peak <= peak_cnt
        if abs(peak0(i_peak) - peak(peak_actual_cnt)) < 3
            peak(peak_actual_cnt) = (peak_set_cnt * peak(peak_actual_cnt) +
peak0(i_peak) ) / (peak_set_cnt + 1);
            peak_set_cnt = peak_set_cnt + 1;
        else
            peak_actual_cnt = peak_actual_cnt + 1;
            peak(peak_actual_cnt) = peak0(i_peak);
            peak_set_cnt = 1;
        end
        i_peak = i_peak + 1;
    end
    peak = round(peak);
end
end

```