

表一 毕业论文（设计）开题报告

论文（设计）题目：基于对数正态分布的快速公交车辆行程时间研究	
<p>（简述选题的目的、思路、方法、相关支持条件及进度安排等）</p> <p>选题目的</p> <p>优先发展公共交通是缓解城市道路交通日渐拥堵的有效手段，而快速公交系统（BRT）由于其易建设、大容量、便捷环保的优势已成为各大中城市公共交通的主力。快速公交车辆行程时间决定了快速公交系统的服务效率，能定性分析其行程时间的影响因素并且定量分析其行程时间的分布规律有助于精确预测规划中的快速公交的行程时间，从而对规划方案提出建设性的意见。</p> <p>思路方法：</p> <p>第一步、读取经过快速公交站台的所有报站数据，排除偶然性数据和缺失数据，计算站间行程时间；</p> <p>第二步、在 Matlab 平台上对站间行程时间数据进行正态分布和对数正态分布拟合，绘制其拟合曲线并记录分布参数；</p> <p>第三步、研究不同线路和不同运行时间段对站间行程时间分布的影响。</p> <p>相关支持条件：</p> <p>一、已有 2013 年 10 月录入的广州市公交 GPS 报站数据；</p> <p>二、Matlab 软件在数学分析和曲线拟合领域占有优势。</p> <p>进度安排：</p> <p>2014 年 3 月上旬，完成相关文献的阅读并设计研究框架；</p> <p>2014 年 3 月中旬，完成车辆报站数据筛选和站间行程时间计算；</p> <p>2014 年 4 月上旬，完成 Matlab 分布检验和曲线拟合；</p> <p>2014 年 4 月下旬，完成站间行程时间影响因素的定性分析；</p> <p>2014 年 4 月底，完成论文的撰写。</p> <p>学生签名：_____ 年 月 日</p>	

指导教师意见：

1、同意开题（√） 2、修改后开题（ ） 3、重新开题（ ）

指导教师签名：

年 月 日

表二 毕业论文（设计）过程检查情况记录表

指导教师分阶段检查论文的进展情况（要求过程检查记录不少于 3 次）：

第 1 次检查

学生总结：

根据已有的 2013 年 10 月录入的广州市公交 GPS 报站数据进行数据初步筛选，得到有效的报站时间数据，并计算出站间行程时间数据，但是筛选过程中仍存在漏洞，筛选考虑情况不全面。

指导教师意见：

分析报站数据的记录过程，思考其可能出现的偶然情况，增加筛选条件以将未考虑的情况考虑进去。

第 2 次检查

学生总结：

已更新了数据筛选条件，但是偶然性数据仍然未被排除，特别是终点站的报站数据易造成错误。

指导教师意见：

可通过增加运行方向来判断终点站处的报站数据是否可以组合计算站间行程时间，若同向便可以组合，若不同向则要排除。

第 3 次检查

学生总结：

已基本排除偶然性数据的干扰，但绘制概率密度分布曲线时发现个别曲线会存在不规则性，数据集中点较多，造成多个峰值。

指导教师意见：

可通过分线路分时间段来绘制曲线，以去除不同线路或不同时间段的干扰。

表三 毕业论文（设计）答辩情况

答辩人	吴馥谧	专 业	交通工程
论文（设计）题目	基于对数正态分布的快速公交行程时间分析		
答辩小组成员	李熙莹、李军、胡继华、熊会元		
<div>答辩记录：</div> <div>1、论文目录需要优化一下</div> <div>2、论文题目需要优化，标题太长</div> <div>3、问：主要是基于什么分布？正态分布还是对数正态分布？ 答：数据分布和对数正态分布更为拟合。</div> <div>4、参考文献要与交通相关，需要增加参考文献数量。</div> <div>5、PPT 的字体颜色太淡，需要加深。</div>			

论文原创性声明内容

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

基于对数正态分布的快速公交车辆行程时间研究

论文设计者：吴馥谧

指导教师：胡继华 讲师

专业：交通工程

论文方向：交通规划与管理

二〇一四年 五 月

摘要

【论文摘要】城市道路交通日渐拥挤及堵塞，这已严重影响城市的可持续性发展。论文以智能交通系统为背景，对公交车辆行程时间进行研究，意在提高公交系统的服务效率。经过分析，得知公交车辆行程时间具有明显的时段分布特征，并且公交车辆是典型的时空过程对象，其运行过程具有状态转移性。

因此，为了得出公交行程时间的分布特征，本研究对现有快速公交报站数据进行筛选，得出现有快速公交行程时间数据，运用正态分布和对数正态分布对其分布进行检验，并根据其分布参数绘制相应的正态分布和对数正态分布曲线，直观表现其分布拟合程度，并计算相应分布的参数记录下来，同时利用累积概率分布函数来计算其行程时间的可靠性。

最后以现有的中山大道的 BRT 站台的报站数据进行带入计算，结果表明，车辆行程时间分布符合这两种分布类型，这种拟合方式是可行的。

【关键词】智能交通系统；快速公交系统；行程时间；正态分布；对数正态分布；

Abstract

【Abstract】 Urban traffic congestion is becoming more and more serious, which has seriously affected the development of cities. Based on Intelligent Transportation System, this article is intended to improve the service of the public transportation system by studying the bus travel time. As we know, bus travel time has obvious period distribution characteristics. Besides, public transport vehicle, with the characteristic of state transferring, is a typical object of space-time process.

Therefore, in order to achieve the rule of travel time, this research use the data of arrival time and depart time in bus rapid transit dataset to calculate the travel time. Normal distribution and lognormal distribution also use in it to fit the distribution of travel time. Meanwhile, the reliability of travel time will be figured out by means of Finally, considering the way that different route and time will affect the travel time, the research separate data according to those factor and analysis.

In the final part, the data of Guangzhou bus rapid transit will be tested to ensure the probability of this methodology and the distribution rule of those data will be figure out.

【Key words】 Intelligent Transportation System(ITS); bus rapid transit; travel time; normal distribution; lognormal distribution

目录

摘要.....	8
Abstract.....	8
1 概述.....	12
1.1 研究背景.....	12
1.2 研究内容.....	13
1.3 研究意义.....	13
1.4 研究思路.....	13
2 快速公交站间行程时间分布及运行可靠性概述	14
2.1 快速公交运行特征.....	14
2.1.1 快速公交行程特性	14
2.1.2 快速公交行程过程影响因素.....	14
2.1.3 快速公交行程时间组成.....	15
2.2 快速公交运行时间可靠性概述	15
3 基于对数正态分布拟合的 BRT 站间行程时间算法	16
3.1 公交 GPS 报站数据采集和处理算法.....	16
3.1.1 算法基础	16
3.1.2 报站数据的读取索引.....	17
3.1.3 报站数据的筛选算法.....	18
3.1.4 行程时间计算	20
3.2 基于正态分布和对数正态分布的公交车辆行程时间 Matlab 曲线拟合	21
3.3 站间行程时间可靠性分析.....	22
3.4 在单一变量的情况下 BRT 行程时间数据对比及定性分析.....	23
4 实例分析	24
4.1 交通信息数据采集及处理.....	24
4.1.1 交通信息数据采集.....	24
4.1.2 交通信息数据处理.....	27
4.2 基于正态分布和对数正态分布的公交车辆行程时间曲线拟合	29
4.2.1 Matlab 数据的处理	29
4.2.2 分布类型判定.....	33
4.2.3 正态分布和对数正态分布检验及参数确定.....	34
4.2.4 正态分布和对数正态分布拟合曲线绘制.....	36
4.3 站间行程时间可靠性分析.....	37
4.4 单一变量的公交站间行程时间数据对比及定性分析.....	39
4.4.1 以线路为变量的公交站间行程时间数据对比及分析.....	39
4.4.2 以运行时间段为变量的公交站间行程时间数据对比及分析...	41
4.4 本章小结.....	43

5 总结与展望	44
参考文献:	45
致谢	46
附录	47

1 概述

1.1 研究背景

城市交通是城市建设和发展的主要推动力。随着城市规日益扩大,人口数量增长,传统的交通方式已不能满足人们的出行需求。众所周知,城市交通是一个城市发展的命脉,糟糕的交通不但影响了城市居民的日常出行,并且严重制约着城市的顺利发展。目前,城市交通面对着种种问题:

①道路容量严重不足。很多城市缓解交通问题的普遍做法是扩建或新建道路。但是,大部分城市土地开发已完全,新建道路不仅会破坏现有开布局,而且还会对居民造成影响,比如拆迁造成的生活不稳定。因此通过扩建或新建道路的方式来缓解交通压力的做法已不适用。

②私有车辆数的飞速增长。^{【1】}2011 年底,汽车保有量达到 10578 万辆,而 2012 年底,汽车保有量已增长到约为 12313 万辆,截止 2013 年底,该数目已增加到 1.37 亿辆,而城市布局基本成型,已难以再开发新道路来满足持续增长的车辆数。

③城市人口不断增加,出行量飞速上升。目前人口分布越来越趋向集中分布,大量人口从乡村涌入城市,而城居人口逐渐外迁,呈现出郊区化的发展趋势,因此通勤出行量较前些年大幅增加,且人均出行距离逐步增大。^{【2】}

所幸,随着公共交通技术的发展,各个城市交通部分开始大力发展公共交通,多种新型公共交通方式的出现为改善交通情况提供了基础。目前,广州大力推行“公交优先”政策,并结合多种公共交通方式,力求改善出行状况,而快速公交就是其中一种新型公共交通方式。

快速公交(Bus Rapid Transit, BRT)系统是一种新型的大容量快速公共汽车运营方式,以其专用道路网络为支撑,结合传统公交车辆技术,吸取轨道交通优点,具有交通运输量大、网络布设灵活、建设周期短、建设成本较低、快捷安全舒适等特点,相对地铁、轻轨系统在经济性上具有一定优势,从而为城市提高公共交通服务水平提供了一种经济快捷的选择方式,并在越来越多的国外城市得到关注和应用,同样快速公交在我国发展迅速,已成为各大中城市解决交通拥堵、方便居民出行、环保低碳的重要交通工具之一。

为了能给出行居民提供便利有效的公共交通服务,吸引更多的出行乘客选择公共交通出行,城市交通管理者致力于发展智能交通系统,^[3]结合交通信息系统和公共交通系统,通过 GPS 卫星实时获取公交车辆的位置,为公民出行提供该车辆的运行状况,方便公民合理安排出行时间。而在这过程中,提供准确的车辆到站时间一直是重点研究方向,其中车辆到站时间又受到站间运行时间和站台停靠时间的影响。准确、可靠的公交车辆行程时间预测技术不仅能够为公交车辆调度技术提供保障,还将为乘客提供可靠的乘车信息,能够很大程度的提高公共交通服务的稳定性,从而有效的增加公共交通满意度以及公共交通的吸引力,缓解城市交通压力。^[4]

1.2 研究内容

本问题的研究目标是通过是现有快速公交站间行程时间进行分布拟合和定性分析,从而得出站间运行时间的分布规律。其主要内容包括:

- (1) 快速公交报站数据的筛选;
- (2) 快速公交站间行程时间的计算;
- (3) 运用 Matlab 对站间行程时间分布的检验和拟合;
- (4) 计算站间行程时间的可靠性;
- (5) 分析不同线路和不同运行时间段对站间行程时间的影响。

1.3 研究意义

- (1) 为快速公交车辆行程时间的预测提供依据
- (2) 为居民出行时间安排提供可靠性分析
- (3) 促进快速公交系统的发展

1.4 研究思路

本次研究主要有三个部分,分别为报站数据的采集和处理、分布拟合和参数计算、控制变量分析规律。

2 快速公交站间行程时间分布及运行可靠性概述

2.1 快速公交运行特征

2.1.1 快速公交行程特性

城市快速公交一般具有安全、快速、准时、舒适的特点。快速公交的特性主要有以下几个方面：^{【5】}

- 1) 快速公交有公交专用道，即拥有路权。
- 2) 快速公交所经交叉口大部分采取公交优先政策，使得快速公交车辆经过交叉口时能优先通过。
- 3) 较传统公交来说，其行车过程中收到周围车流影响较小。
- 4) 由于受周围车流影响较小，快速公交的发车间隔和到站时间间隔相近，较少出现多辆同线路快速公交车辆同时入站的情况。

2.1.2 快速公交行程过程影响因素

快速公交运行时间受多种因素影响，主要有以下几个大方面：^{【6】}

1) 人

驾驶员和乘客的行为都会对快速公交行程时间造成影响。在运行过程中，驾驶员的驾驶习惯，对行车速度的偏好都会影响到车辆行程时间，而在站台停靠过程中，乘客的上车拥挤及下车延误同样也会使车辆不能在预定时间内到站。

2) 车

不同条件的公交车，其行驶性能有所差异，对其运行时间、速度等的影响也不相同。^{【7】}

3) 路

这里除了要考虑道路几何线形、路面条件、容量限制等因素影响外，还考虑道路设置的快速公交车专用道。快速公交专用道上只允许快速公交车行驶，禁止其他社会车辆行驶，有利于快速公交车畅通行驶。^{【8】}

2.1.3 快速公交行程时间组成

快速公交站间行程时间主要有路段行驶时间、交叉口等待时间、外界车辆干扰造成的延误时间等组成。^{【9】}在本次研究中，统一将它们算入快速公交行程时间内，而不单独考虑其影响。

2.2 快速公交运行时间可靠性概述

可靠性是指系统在规定的条件下和规定的时间内达到系统规定的功能的概率。^{【10】}而在本次研究中，主要研究的可靠性为准点性可靠度，^{【11】}该可靠度描述的是公交车能准点到达或驶离公交站点的能力，将其定量化可表示为公交车辆到站时刻与规定时刻的实际偏离值小于限定值的概率。

3 基于对数正态分布拟合的 BRT 站间行程时间算法

本章节将介绍基于正态分布和对数正态分布拟合的 BRT 站间行程时间算法，其过程主要分为三个部分，分别如下图所示：

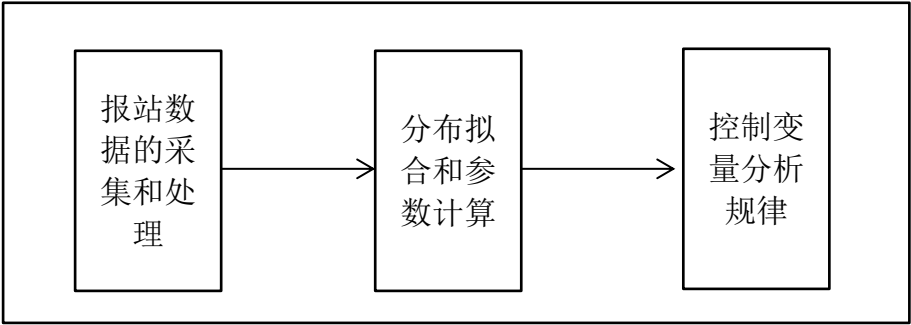


图 3-1 基于正态分布和对数正态分布拟合的 BRT 站间行程时间算法流程图

3.1 公交 GPS 报站数据采集和处理算法

3.1.1 算法基础

目前车辆报站数据的获取方式是通过车辆自带的 GPS 定位器实时获得车辆经纬度坐标，与数据库存有的站台经纬度范围坐标进行比对，以判断车辆进出站状态。其原理图如下：

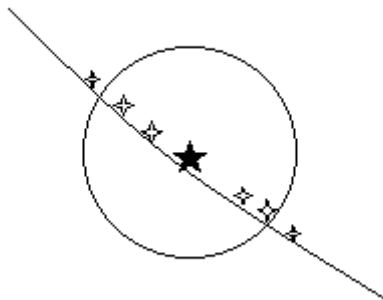


图 3-2 车辆报站数据获取原理图

注：实心五角星代表公交站台，圆圈为该站台在数据库内记录的经纬度坐标范围，线条为公交车实际运行轨迹，空心四角星为 GPS 实时记录的公交车位置

GPS 发射器每间隔一段时间记录一次此时公交车的经纬度坐标，并与数据库内存有的经纬度坐标进行比对。当记录的经纬度坐标首次出现在某站台经纬度坐

标范围内时,视该时间点为进站时间,记录进站数据;当在进站之后一段时间内,首次检测到公交车实时经纬度坐标已不在某站台经纬度坐标范围内时,则视该时间点为出站时间,记录出站数据。【12】

在这种记录情况下,不考虑 GPS 本身定位误差,设其每间隔 t 秒定位一次车辆,则其进出站数据误差范围为 $(0, t)$,站间行程时间和站台停靠时间数据误差范围为 $(0, 2t)$ 。随着 t 值越大,其误差范围越大。

同时,因为数据传送信号不良或者其他因素干扰,会导致两种数据获取失效情况,一是首次检测数据传送延误而造成将其后多次检测的数据当成首次检测数据,从而导致数据冗余,二是数据传送失败而造成数据缺失。要获得具有连续性且较为完整的 GPS 报站数据,就必须排除冗余数据和识别缺失数据。

本节将讲述如何原始的报站数据处理为有连续性的有效报站数据。具体步骤如下:

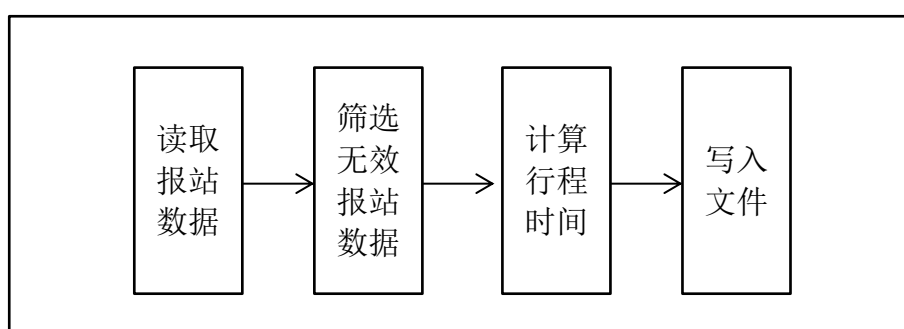


图 3-3 公交 GPS 报站数据采集和处理算法流程图

3.1.2 报站数据的读取索引

广州市 GPS 公交报站数据表里面包括了时间段内所有公交 GPS 报站时间数据,结构如下图所示:

organame	buscode	routecode	service	busstop	adflag	adtime
营运公司名称	车辆编号	线路编号	运行方向指示	站台名字	出入站指示	报站时间

图 3-4 报站时间数据表结构范例

要从其中筛选出经过 BRT 站台的所有公交 GPS 报站时间数据,其思路如下:

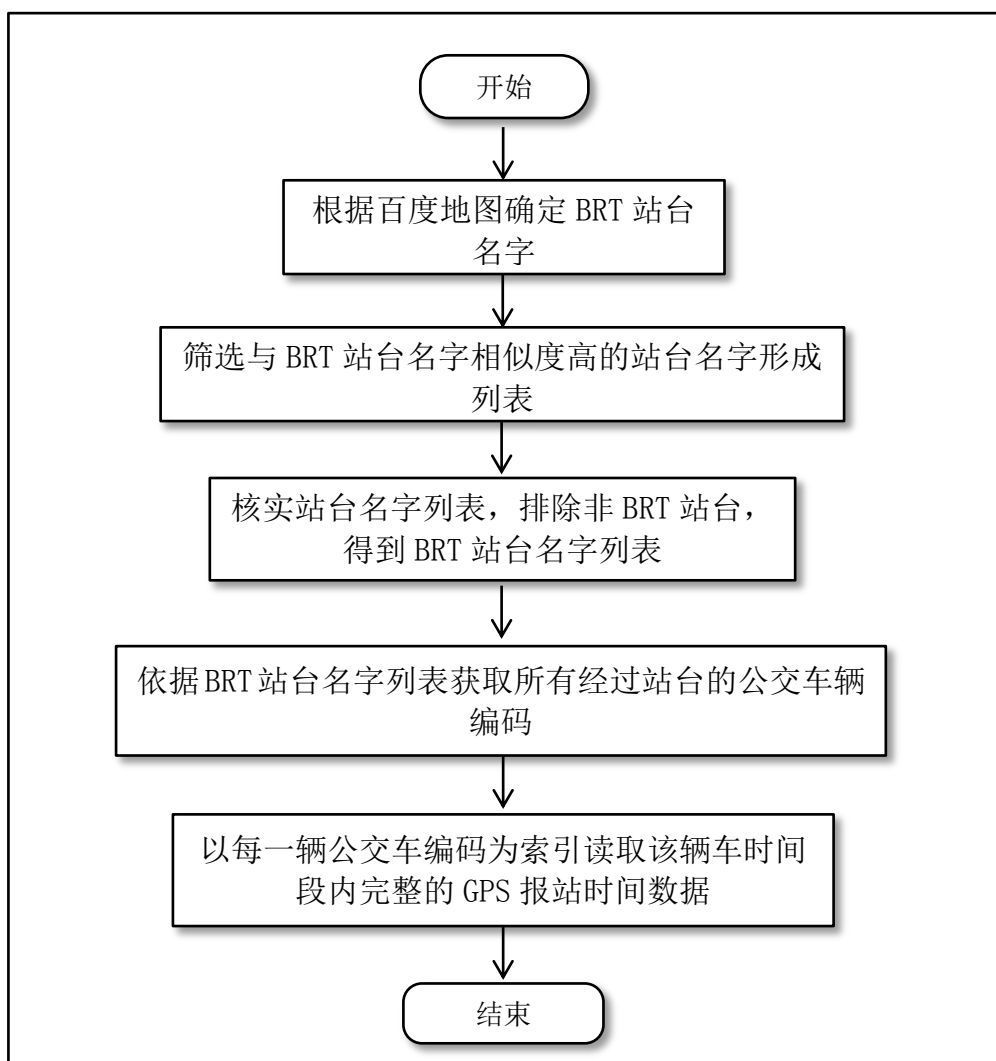


图 3-5 报站数据读取索引流程图

3.1.3 报站数据的筛选算法

由于报站数据的获取方式会导致无效或者冗余数据的出现,对于这些数据筛选出有效数据是本节需要做的事情。

在本节中,以 A 代表上一个站, B 代表下一个站, stopnum 为站台编号,每个站台都有对应编号,不同站台之间编号不同, adflag 为进出站标示, 0 代表进站, 1 代表出站。

在原始报站数据表中,相邻两个报站数据有以下十种组合:

A 进 A 进 (A0A0), A 进 A 出 (A0A1), A 进 B 进 (A0B0), A 进 B 出 (A0B1)

A 出 A 出 (A1A1), A 出 B 进 (A1B0), A 出 B 出 (A1B1)

B 进 B 进 (B0B0), B 进 B 出 (B0B1)

B 出 B 出 (B1B1)

其中，只有 A 进 A 出 (A0A1)、A 出 B 进 (A1B0)、B 进 B 出 (B0B1) 为有效的对应数据组合。

对于以上十种数据组合的筛选流程如下：

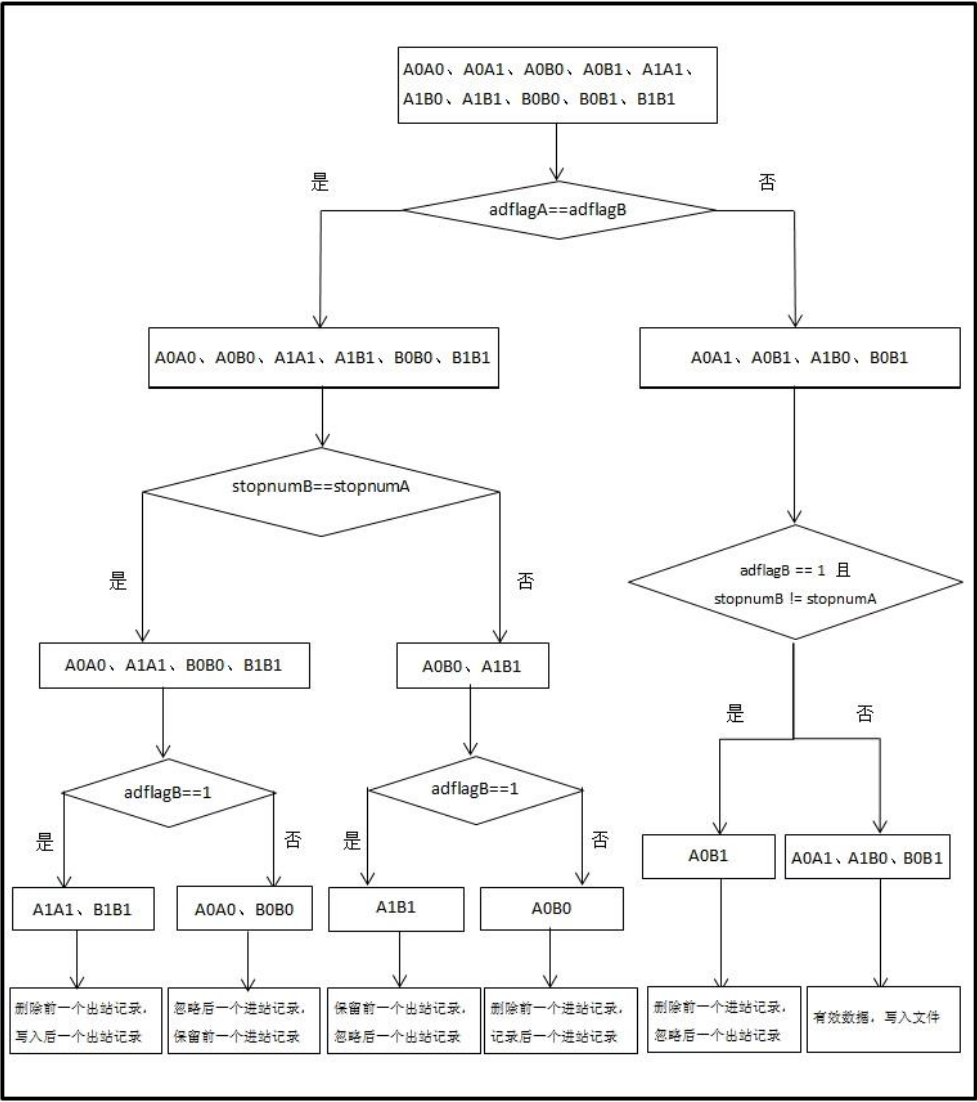


图 3-6 报站数据筛选算法

最后存入文件数据时，仅当某一站台同时具有进站和出站数据对的时候才会被存入数据文件。文件存储格式范例如下图所示：

线路编号	所经站台编号	出入站指示	站间行程时间	原始报站时间数据A	原始报站时间数据B	车辆编号
80170	4,5	10	126	80170 4 1 2013/10/8 6:16:21 00	80170 5 0 2013/10/8 6:18:27 00	802130
80170	4,5	10	120	80170 4 1 2013/10/8 9:19:48 00	80170 5 0 2013/10/8 9:21:48 00	802130
80170	4,5	10	121	80170 4 1 2013/10/8 11:26:37 00	80170 5 0 2013/10/8 11:28:38 00	802130
80170	4,5	10	121	80170 4 1 2013/10/8 12:59:31 00	80170 5 0 2013/10/8 13:01:32 00	802130
80170	4,5	10	97	80170 4 1 2013/10/8 14:24:22 00	80170 5 0 2013/10/8 14:25:59 00	802130
80170	4,5	10	130	80170 4 1 2013/10/8 17:18:59 00	80170 5 0 2013/10/8 17:21:09 00	802130
80170	4,5	10	101	80170 4 1 2013/10/8 20:18:04 00	80170 5 0 2013/10/8 20:19:45 00	802130
80170	4,5	10	113	80170 4 1 2013/10/8 20:53:43 28	80170 5 0 2013/10/8 20:55:36 28	802130
80170	4,5	10	118	80170 4 1 2013/10/9 6:14:55 00	80170 5 0 2013/10/9 6:16:53 00	802130

图 3-7 站间行程时间文件存储格式范例

3.1.4 行程时间计算

行程时间的获取是通过相邻两个有效报站数据相减所得，相减的两个行程时间必须保证同向，且考虑到车辆调度，一辆车可能会在不同线路中运行，除了保持同向，还需保持同线路，该组报站数据才可以用来相减。

由于上述提到公交 GPS 报站时间数据获取的原理，这种方法会出现数据缺失的情况，如下所示：

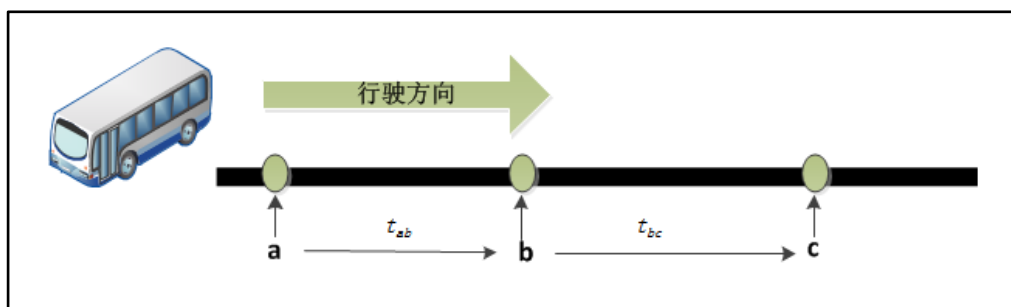


图 3-8 车辆行驶过程图

a, b, c 为 3 个连续的 BRT 站台，a, c 报站时间数据完整，而 b 站台报站时间数据缺失。

当 b 站台缺失进站数据的时候，a 站到 b 站的站间行程时间将无法计算；

当 b 站台缺失出站数据的时候，b 站到 c 站的站间行程时间也将无法计算；

当 b 站台进出站数据都缺失的时候，仅能计算 a 站到 c 站的站间行程时间，而该站间行程时间包含 t_{ab} 和 t_{bc} 两段站间行程时间，其数值远大于正常站间行程时间，会对后续的曲线拟合造成误导。

对于这种数据缺失情况的处理方法是是对站台预先编号，其计算流程如下：

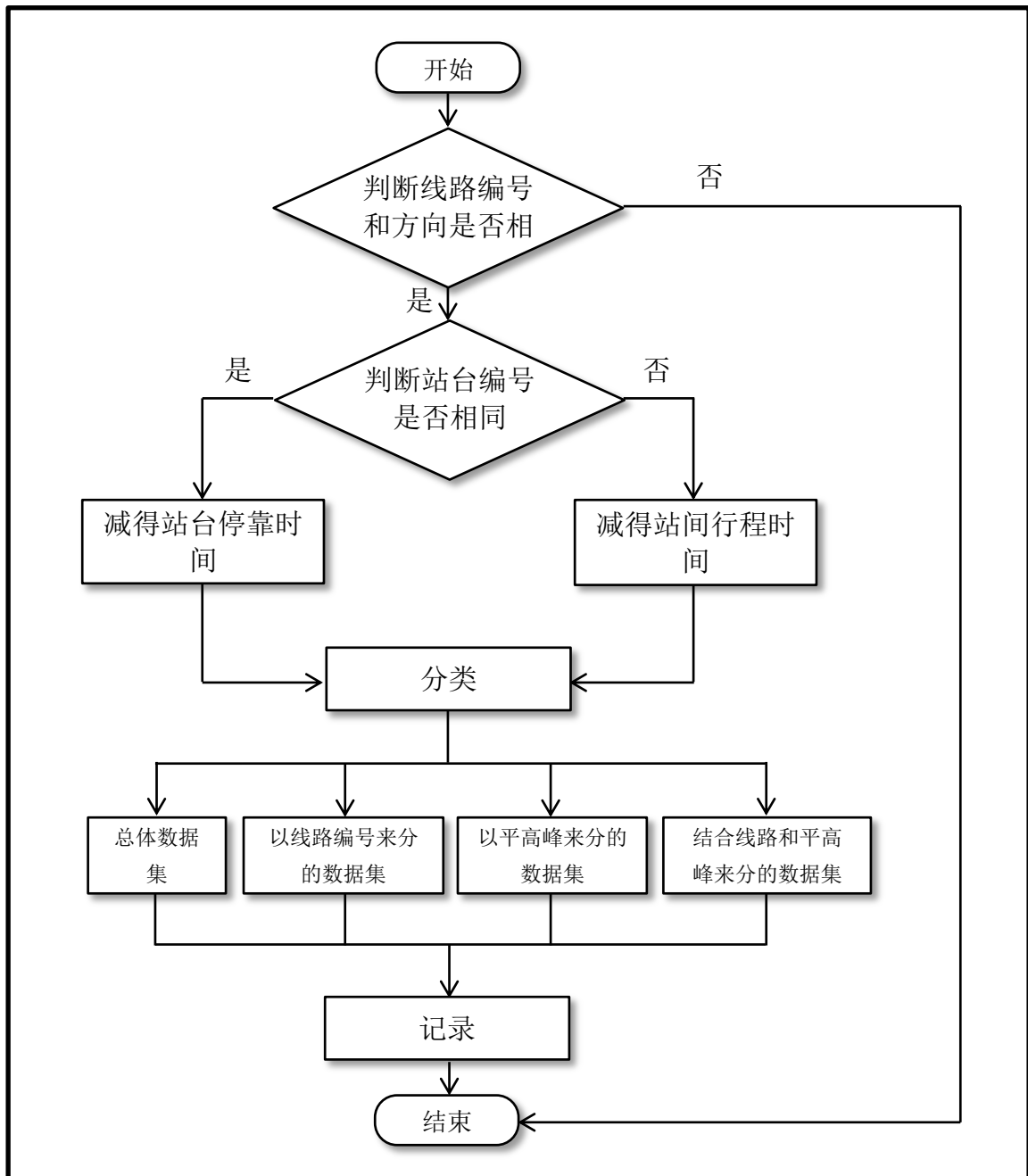


图 3-9 站间行程时间计算流程图

3.2 基于正态分布和对数正态分布的公交车辆行程时间 Matlab 曲线拟合

若要定量分析站间行程时间，需要先确定数据的分布类型，再根据其分布类型算出分布参数。其算法流程图如下：

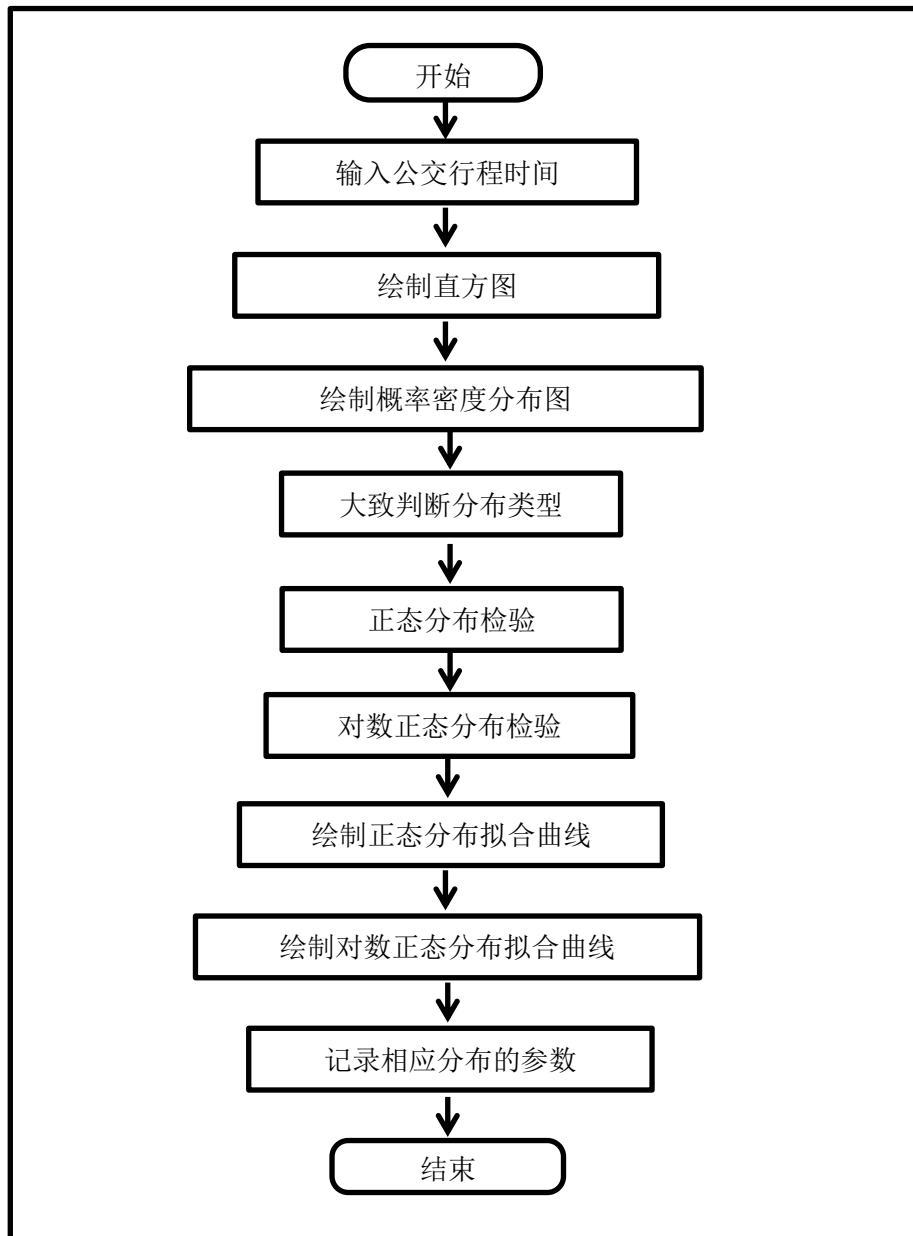


图 3-10 Matlab 平台上分布拟合算法流程图

3.3 站间行程时间可靠性分析

在已知站间行程时间分布参数的情况下，可以推断出车辆站间行程时间落在某段时间范围内的概率，即在该段时间内到达的准点可靠性。具体算法流程图如下：

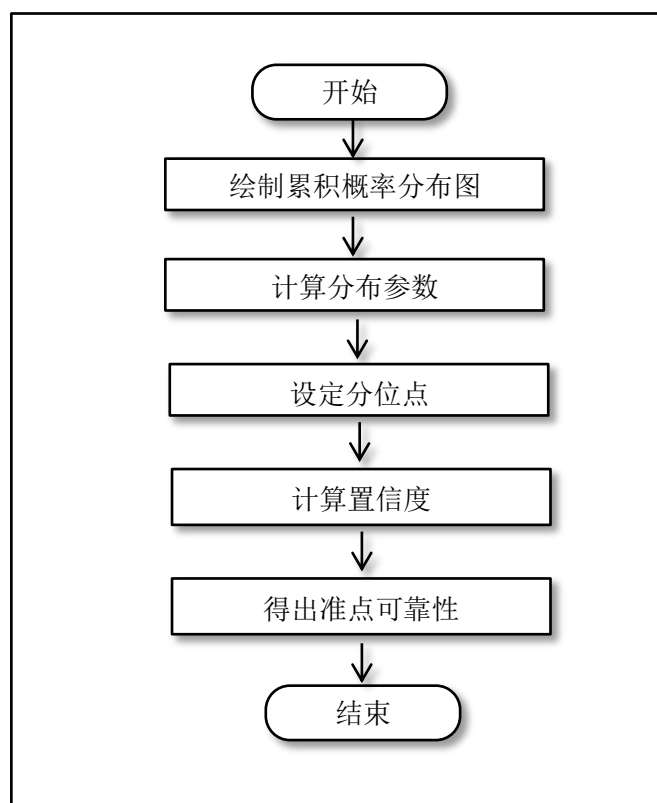


图 3-11 站间行程时间可靠性算法流程图

3.4 在单一变量的情况下 BRT 行程时间数据对比及定性分析

BRT 行程时间数据的主要影响因素来自公交线路分布和公交运行时间段两方面。

本研究中将站间行程时间数据按照线路编号进行分类,单独探讨线路队站间行程时间数据的影响。同时,本次研究还将站间行程时间数据按照其记录时间进行分类,类别有三种:早高峰时段(早上 7 到 9 点)、晚高峰时段(下午 5 点到 7 点)和平峰时间段。在对数据进行细分之后,对小类数据做曲线拟合和计算其均值和方差,^{【13】}以便直观且能定性分析站间行程时间数据分布的影响因素。

4 实例分析

4.1 交通信息数据采集及处理

4.1.1 交通信息数据采集

根据本问题研究的要求，实例分析的数据来自于广州市公交 GPS 数据库里面 BRT 车辆报站数据，主要内容包含车辆编号、线路编号、运行方向、经过站点、出入站指示和 GPS 时间。如下图 4-1 所示：

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME
1	BRT营运公司	00803142	912255	80050	00	天朗明居1	出站	2013/10/8 18:44:04
2	BRT营运公司	0080z187	982265	80160	00	珠江村站1	进站	2013/10/8 18:44:05
3	BRT营运公司	00881787	983953	80280	99	黄埔客运站1	出站	2013/10/8 18:44:06
4	BRT营运公司	0080G796	986103	80260	00	下沙站B	出站	2013/10/8 18:44:06
5	BRT营运公司	00802178	985048	80270	99	东圃3	进站	2013/10/8 18:44:06
6	BRT营运公司	00881701	983867	80020	99	华景新城站1	出站	2013/10/8 18:44:02
7	BRT营运公司	00801143	982167	80160	99	师大暨大站1	出站	2013/10/8 18:44:02
8	BRT营运公司	00801509	986128	80076	00	天朗明居站2	进站	2013/10/8 18:44:02
9	BRT营运公司	00801226	988065	80220	00	东圃镇3	进站	2013/10/8 18:44:03
10	BRT营运公司	0080z229	982262	80160	00	天朗明居站1	进站	2013/10/8 18:44:04
11	BRT营运公司	00803205	912293	80050	99	学院1	出站	2013/10/8 18:43:59
12	BRT营运公司	00801604	986055	80043	00	体育中心	进站	2013/10/8 18:44:00
13	BRT营运公司	00881830	983996	80240	00	茅岗站B	出站	2013/10/8 18:44:00
14	BRT营运公司	N2S92808	985187	80030	99	华景新城3	出站	2013/10/8 18:44:01
15	BRT营运公司	00881802	983968	80280	00	庙头站1	出站	2013/10/8 18:44:01
16	BRT营运公司	00801801	987082	80010	00	岗顶3	出站	2013/10/8 18:43:57
17	BRT营运公司	0080293K	981711	80200	00	上社1	进站	2013/10/8 18:43:57
18	BRT营运公司	00801530	986077	80068	99	棠东站	出站	2013/10/8 18:43:58
19	BRT营运公司	008z1144	982168	80160	99	师大暨大站1	出站	2013/10/8 18:43:58
20	BRT营运公司	00801154	982175	80160	99	东圃镇站1	进站	2013/10/8 18:43:58
21	BRT营运公司	N2S92809	985188	80030	00	黄村1	进站	2013/10/8 18:43:58
22	BRT营运公司	0080452E	981096	80200	99	师大暨大1	进站	2013/10/8 18:43:58

图 4-1 公交 GPS 报站时间数据表

而本次研究涉及的 BRT 站台共有 26 个，分别如图 4-2 所示：

站台名称	编号	站台名称	编号	站台名称	编号
体育中心	1	天朗明居	10	乌冲	19
石牌桥	2	车陂	11	黄埔客运站	20
岗顶	3	东圃镇	12	双岗	21
师大暨大	4	黄村	13	沙浦	22
华景新城	5	珠村	14	南海神庙	23
上社	6	莲溪	15	庙头	24
学院	7	茅岗	16	南湾	25
棠下村	8	珠江村	17	夏园	26
棠东	9	下沙	18		

图 4-2 BRT 站台名字列表

用 SQL 语言从数据库中筛选出相似的站台名字, 通过与百度地图站台名字对比判断是否是 BRT 站台。其筛选语句如下:

```
Select * from tablename where busstop like 'stopname%';
```

%tablename 为查询的表名, stopname 为表 4-1 中 BRT 站台名称。

部分筛选结果如下图 4-3:

体育中心	石牌桥	岗顶	师大暨大	华景新城
体育中心	石牌桥	岗顶	师大暨大	华景新城
体育中心1	石牌桥1	岗顶1	师大暨大1	华景新城1
体育中心2	石牌桥2	岗顶2	师大暨大3	华景新城3
体育中心3	石牌桥3	岗顶3	师大暨大4	华景新城西站
体育中心4	石牌桥站	岗顶站	师大暨大站	华景新城站
体育中心东门	石牌桥站1	岗顶站1	师大暨大站1	华景新城站1
体育中心东门站	石牌桥站2	岗顶站2	师大暨大站2	华景新城站2
体育中心南门		岗顶站3	师大暨大站3	华景新城总站
体育中心站				
体育中心站1				
体育中心站1①				
体育中心站3				
体育中心站4				
体育中心总站				

图 4-3 部分站台名字筛选列表

注：其中阴影部分为非 BRT 站台名字。

根据筛选后的站台名字遍历所有经过该站台的公交 GPS 报站时间数据，不重复记录该数据对应车辆编号，得到车辆编号索引表，并以车辆编号为索引读取该车辆时间段内的完整时空过程且按报站时间先后排序，所得车辆报站时间数据列表如下图 4-4：

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME
1	BRT营运公司	00802130	985128	80170	00	师大暨大	进站	2013/10/8 6:15:49
2	BRT营运公司	00802130	985128	80170	00	师大暨大	出站	2013/10/8 6:16:21
3	BRT营运公司	00802130	985128	80170	00	华景新城3	进站	2013/10/8 6:18:27
4	BRT营运公司	00802130	985128	80170	00	华景新城3	出站	2013/10/8 6:19:18
5	BRT营运公司	00802130	985128	80170	00	上社2	进站	2013/10/8 6:20:12
6	BRT营运公司	00802130	985128	80170	00	棠下村3	出站	2013/10/8 6:25:39
7	BRT营运公司	00802130	985128	80170	00	棠东3	进站	2013/10/8 6:26:18
8	BRT营运公司	00802130	985128	80170	00	棠东3	出站	2013/10/8 6:26:58
9	BRT营运公司	00802130	985128	80170	00	天朗明居3	进站	2013/10/8 6:27:39
10	BRT营运公司	00802130	985128	80170	00	天朗明居3	出站	2013/10/8 6:28:18
11	BRT营运公司	00802130	985128	80170	00	车陂3	进站	2013/10/8 6:29:51
12	BRT营运公司	00802130	985128	80170	00	车陂3	出站	2013/10/8 6:30:37
13	BRT营运公司	00802130	985128	80170	00	东圃3	进站	2013/10/8 6:31:41
14	BRT营运公司	00802130	985128	80170	00	东圃3	出站	2013/10/8 6:33:09
15	BRT营运公司	00802130	985128	80170	00	黄村3	进站	2013/10/8 6:34:41
16	BRT营运公司	00802130	985128	80170	00	黄村3	出站	2013/10/8 6:35:15
17	BRT营运公司	00802130	985128	80170	00	珠江1	进站	2013/10/8 6:36:10
18	BRT营运公司	00802130	985128	80170	00	珠江1	出站	2013/10/8 6:36:50

图 4-4 公交车辆在 BRT 站台部分报站时间数据（以车辆编号为 00802130 的公交车辆为例）

以车辆编号为索引读取的报站时间数据保证了数据与数据之间的相关性和连续性，使其可以两两组合计算。

4.1.2 交通信息数据处理

由原始报站时间数据列表可以看出其存在的三种数据缺失情况和两种数据冗余情况，如图 4-5 和图 4-6 所示。

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME
3	BRT营运公司	...	00802130	985128	80170	00	华景新城3 ...	进站 2013/10/8 6:18:27
4	BRT营运公司	...	00802130	985128	80170	00	华景新城3 ...	出站 2013/10/8 6:19:18
5	BRT营运公司	...	00802130	985128	80170	00	上社2 ...	进站 2013/10/8 6:20:12
6	BRT营运公司	...	00802130	985128	80170	00	棠下村3 ...	出站 2013/10/8 6:25:39
7	BRT营运公司	...	00802130	985128	80170	00	棠东3 ...	进站 2013/10/8 6:26:18
8	BRT营运公司	...	00802130	985128	80170	00	棠东3 ...	出站 2013/10/8 6:26:58

图 4-5 报站时间数据缺失情况（以车辆编号为 00802130 的公交车辆为例）

注：上图中上社站缺失了出站数据，棠下村站缺失了进站数据，而在上社和棠下村之间的学院站则缺失全部报站数据。

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME
2860	BRT营运公司	...	00802130	985128	80170	00	珠村1 ...	进站 2013/10/15 6:39:09
2861	BRT营运公司	...	00802130	985128	80170	00	珠村1 ...	进站 2013/10/15 6:39:09
2862	BRT营运公司	...	00802130	985128	80170	00	珠村1 ...	出站 2013/10/15 6:39:45
2863	BRT营运公司	...	00802130	985128	80170	00	珠村1 ...	出站 2013/10/15 6:39:45
2864	BRT营运公司	...	00802130	985128	80170	00	莲溪1 ...	进站 2013/10/15 6:41:47
2865	BRT营运公司	...	00802130	985128	80170	00	莲溪1 ...	进站 2013/10/15 6:41:47
2866	BRT营运公司	...	00802130	985128	80170	00	莲溪1 ...	出站 2013/10/15 6:42:22
2867	BRT营运公司	...	00802130	985128	80170	00	莲溪1 ...	出站 2013/10/15 6:42:22
2868	BRT营运公司	...	00802130	985128	80170	00	茅岗1 ...	进站 2013/10/15 6:44:02
2869	BRT营运公司	...	00802130	985128	80170	00	茅岗1 ...	进站 2013/10/15 6:44:02
2870	BRT营运公司	...	00802130	985128	80170	00	茅岗1 ...	出站 2013/10/15 6:44:48
2871	BRT营运公司	...	00802130	985128	80170	00	茅岗1 ...	出站 2013/10/15 6:44:48

图 4-6 报站时间数据冗余情况（以车辆编号为 00802130 的公交车辆为例）

注：上图中珠村、莲溪、茅岗站出现了进站冗余和出站冗余。

对于数据缺失的情况，只要该站数据没有完整的进出站报站时间数据的匹配对，则将该站的数据不予保留。对于数据冗余的情况，若冗余的是进站数据，保留第一次读取的进站数据，若冗余的是出站数据，则保留最后一次读取的出站数据。其算法流程见图 3-6。

对照图 4-5 和图 4-6，经过程序筛选后，最后保留的数据如下图 4-7 所示：

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME			
3	BRT营运公司	...	00802130	985128	80170	00	华景新城3	...	进站	2013/10/8 6:18:27	▼
4	BRT营运公司	...	00802130	985128	80170	00	华景新城3	...	出站	2013/10/8 6:19:18	▼
7	BRT营运公司	...	00802130	985128	80170	00	棠东3	...	进站	2013/10/8 6:26:18	▼
8	BRT营运公司	...	00802130	985128	80170	00	棠东3	...	出站	2013/10/8 6:26:58	▼
	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME			
2860	BRT营运公司	...	00802130	985128	80170	00	珠村1	...	进站	2013/10/15 6:39:09	▼
2863	BRT营运公司	...	00802130	985128	80170	00	珠村1	...	出站	2013/10/15 6:39:45	▼
2864	BRT营运公司	...	00802130	985128	80170	00	莲溪1	...	进站	2013/10/15 6:41:47	▼
2867	BRT营运公司	...	00802130	985128	80170	00	莲溪1	...	出站	2013/10/15 6:42:22	▼
2868	BRT营运公司	...	00802130	985128	80170	00	茅岗1	...	进站	2013/10/15 6:44:02	▼
2871	BRT营运公司	...	00802130	985128	80170	00	茅岗1	...	出站	2013/10/15 6:44:48	▼

图 4-7 筛选后报站时间数据样本（对应图 4-5 和图 4-6）

（以车辆编号为 00802130 的公交车辆为例）

在有效车辆报站时间数据将表中连续两个数据两两组合的方式，对组合数据进行计算，得到站间行程时间和站台停靠时间，其数据范例结构如下图 4-8、4-9 所示：

线路编号	所经站台编号	出入站指示	站间行程时间	原始报站时间数据A	原始报站时间数据B	车辆编号
80170	4, 5	10	126	80170 4 1 2013/10/8 6:16:21 00	80170 5 0 2013/10/8 6:18:27 00	802130
80170	4, 5	10	120	80170 4 1 2013/10/8 9:19:48 00	80170 5 0 2013/10/8 9:21:48 00	802130
80170	4, 5	10	121	80170 4 1 2013/10/8 11:26:37 00	80170 5 0 2013/10/8 11:28:38 00	802130
80170	4, 5	10	121	80170 4 1 2013/10/8 12:59:31 00	80170 5 0 2013/10/8 13:01:32 00	802130
80170	4, 5	10	97	80170 4 1 2013/10/8 14:24:22 00	80170 5 0 2013/10/8 14:25:59 00	802130
80170	4, 5	10	130	80170 4 1 2013/10/8 17:18:59 00	80170 5 0 2013/10/8 17:21:09 00	802130
80170	4, 5	10	101	80170 4 1 2013/10/8 20:18:04 00	80170 5 0 2013/10/8 20:19:45 00	802130
80170	4, 5	10	113	80170 4 1 2013/10/8 20:53:43 28	80170 5 0 2013/10/8 20:55:36 28	802130
80170	4, 5	10	118	80170 4 1 2013/10/9 6:14:55 00	80170 5 0 2013/10/9 6:16:53 00	802130
80170	4, 5	10	121	80170 4 1 2013/10/9 9:03:43 00	80170 5 0 2013/10/9 9:05:44 00	802130
80170	4, 5	10	126	80170 4 1 2013/10/9 11:00:50 00	80170 5 0 2013/10/9 11:02:56 00	802130
80170	4, 5	10	118	80170 4 1 2013/10/9 12:24:09 00	80170 5 0 2013/10/9 12:26:07 00	802130
80170	4, 5	10	112	80170 4 1 2013/10/9 13:06:09 00	80170 5 0 2013/10/9 13:08:01 00	802130
80170	4, 5	10	111	80170 4 1 2013/10/9 14:30:02 00	80170 5 0 2013/10/9 14:31:53 00	802130
80170	4, 5	10	109	80170 4 1 2013/10/9 17:28:09 00	80170 5 0 2013/10/9 17:29:58 00	802130
80170	4, 5	10	108	80170 4 1 2013/10/9 20:18:38 00	80170 5 0 2013/10/9 20:20:26 00	802130

图 4-8 站间行程时间列表范例

（以部分线路编码为 80170 车辆编号为 00802130 从师大暨大到华景新城的站间行程时间数据为例）

线路编号	所经站台 编号	出入站 指示	站台停靠 时间	原始报站数据A	原始报站数据B	车辆编号
80170	4, 4	01	28	80170 4 0 2013/10/8 9:19:20 00	80170 4 1 2013/10/8 9:19:48 00	00802130
80170	4, 4	01	26	80170 4 0 2013/10/8 11:26:11 00	80170 4 1 2013/10/8 11:26:37 00	00802130
80170	4, 4	01	34	80170 4 0 2013/10/8 12:58:57 00	80170 4 1 2013/10/8 12:59:31 00	00802130
80170	4, 4	01	24	80170 4 0 2013/10/8 14:23:58 00	80170 4 1 2013/10/8 14:24:22 00	00802130
80170	4, 4	01	48	80170 4 0 2013/10/8 17:18:11 00	80170 4 1 2013/10/8 17:18:59 00	00802130
80170	4, 4	01	26	80170 4 0 2013/10/8 20:17:38 00	80170 4 1 2013/10/8 20:18:04 00	00802130
80170	4, 4	01	27	80170 4 0 2013/10/9 6:14:28 00	80170 4 1 2013/10/9 6:14:55 00	00802130
80170	4, 4	01	36	80170 4 0 2013/10/9 9:03:07 00	80170 4 1 2013/10/9 9:03:43 00	00802130
80170	4, 4	01	27	80170 4 0 2013/10/9 11:00:23 00	80170 4 1 2013/10/9 11:00:50 00	00802130
80170	4, 4	01	25	80170 4 0 2013/10/9 12:23:44 00	80170 4 1 2013/10/9 12:24:09 00	00802130
80170	4, 4	01	29	80170 4 0 2013/10/9 13:05:40 00	80170 4 1 2013/10/9 13:06:09 00	00802130
80170	4, 4	01	31	80170 4 0 2013/10/9 14:29:31 00	80170 4 1 2013/10/9 14:30:02 00	00802130
80170	4, 4	01	30	80170 4 0 2013/10/9 17:27:39 00	80170 4 1 2013/10/9 17:28:09 00	00802130
80170	4, 4	01	32	80170 4 0 2013/10/9 20:18:06 00	80170 4 1 2013/10/9 20:18:38 00	00802130
80170	4, 4	01	61	80170 4 0 2013/10/10 6:45:19 00	80170 4 1 2013/10/10 6:46:20 00	00802130
80170	4, 4	01	29	80170 4 0 2013/10/10 9:43:56 00	80170 4 1 2013/10/10 9:44:25 00	00802130

图 4-9 站台停靠时间列表范例

(以部分线路编码为 80170 车辆编号为 00802130 从师大暨大到华景新城的站台停靠时间数据为例)

当出入站指示为 10 时，代表该数据为这一站的进站数据和上一站出站数据相减，得到两站之间的站间行程时间；当出入站指示为 01 时，代表该数据为这一站出站数据和这一站的进站数据相减，得到该站的站台停靠时间。

同时考虑到原始车辆 GPS 报站时间数据可能出现的偶然数据情况，在计算结果旁边附上相减的一组原始报站时间数据方便对偶然结果的判断和排除。

4.2 基于正态分布和对数正态分布的公交车车辆行程时间曲线拟合

4.2.1 Matlab 数据的处理

由于公交 GPS 报站时间数据受多种因素影响，会出现偶然性数据，通常偶然数据情况有两种，一是由于车辆调度或者车辆班次轮候造成车辆长时间未运行，并且当车辆在长时间未运行后，再次运行时仍然以上一次运行方向运行导致的，具体情况如图 4-10 所示。此时由于方向前后一致，且站台编号连续，会造成程序筛选报站时间数据时无法识别，而误当这两对报站时间数据为连续报站时间来计算。二是由于道路影响、路段交通环境影响、车辆影响、人的影响、突发影响等方面影响因素，^[1]其表现如图 4-11 所示。这种无法具体探讨形成原因的偶然性报站数据，也是筛选程序的漏洞。

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME
7	BRT营运公司	00804432	981073	80090	99	师大暨大4	进站	2013/10/8 7:00:53
8	BRT营运公司	00804432	981073	80090	99	师大暨大4	出站	2013/10/8 7:01:24
9	BRT营运公司	00804432	981073	80090	00	师大暨大4	进站	2013/10/8 8:07:53
10	BRT营运公司	00804432	981073	80090	00	师大暨大4	出站	2013/10/8 8:09:03
11	BRT营运公司	00804432	981073	80090	00	岗顶2	进站	2013/10/8 8:11:39
12	BRT营运公司	00804432	981073	80090	00	岗顶2	出站	2013/10/8 8:12:19
13	BRT营运公司	00804432	981073	80090	00	师大暨大4	进站	2013/10/8 11:12:14
14	BRT营运公司	00804432	981073	80090	00	师大暨大4	出站	2013/10/8 11:12:46
15	BRT营运公司	00804432	981073	80090	00	岗顶2	进站	2013/10/8 11:14:04
16	BRT营运公司	00804432	981073	80090	00	岗顶2	出站	2013/10/8 11:14:43

图 4-10 偶然性数据情况一（以线路编码为 80090 从岗顶到师大暨大的站间行程时间数据为例）

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME
170	BRT营运公司	00804432	981073	80090	00	师大暨大4	进站	2013/10/12 20:09:08
171	BRT营运公司	00804432	981073	80090	00	师大暨大4	出站	2013/10/12 20:09:52
172	BRT营运公司	00804432	981073	80090	00	岗顶2	进站	2013/10/12 20:11:29
173	BRT营运公司	00804432	981073	80090	00	岗顶2	出站	2013/10/12 20:11:43
174	BRT营运公司	00804432	981073	80090	99	岗顶2	进站	2013/10/12 22:15:05
175	BRT营运公司	00804432	981073	80090	99	岗顶2	出站	2013/10/12 22:15:30
176	BRT营运公司	00804432	981073	80090	99	师大暨大4	进站	2013/10/12 22:24:36
177	BRT营运公司	00804432	981073	80090	99	师大暨大4	出站	2013/10/12 22:26:24

图 4-11 偶然性数据情况二（以线路编码为 80090 从岗顶到师大暨大的站间行程时间数据为例）

这种偶然性报站数据直接导致依据其计算的站间行程时间数据和站台停靠时间数据出现偶然性，比如远远偏离常规值。对于这种不符合实际通常情况且并没有出现缺失或冗余的报站时间数据并不能简单通过程序判断或者设定数值限定来排除，本节的研究内容就是在 Matlab 中通过 tabulate 语句计算各个数据的分布概率来排除分布概率极小的偶然值。其算法流程图如下：

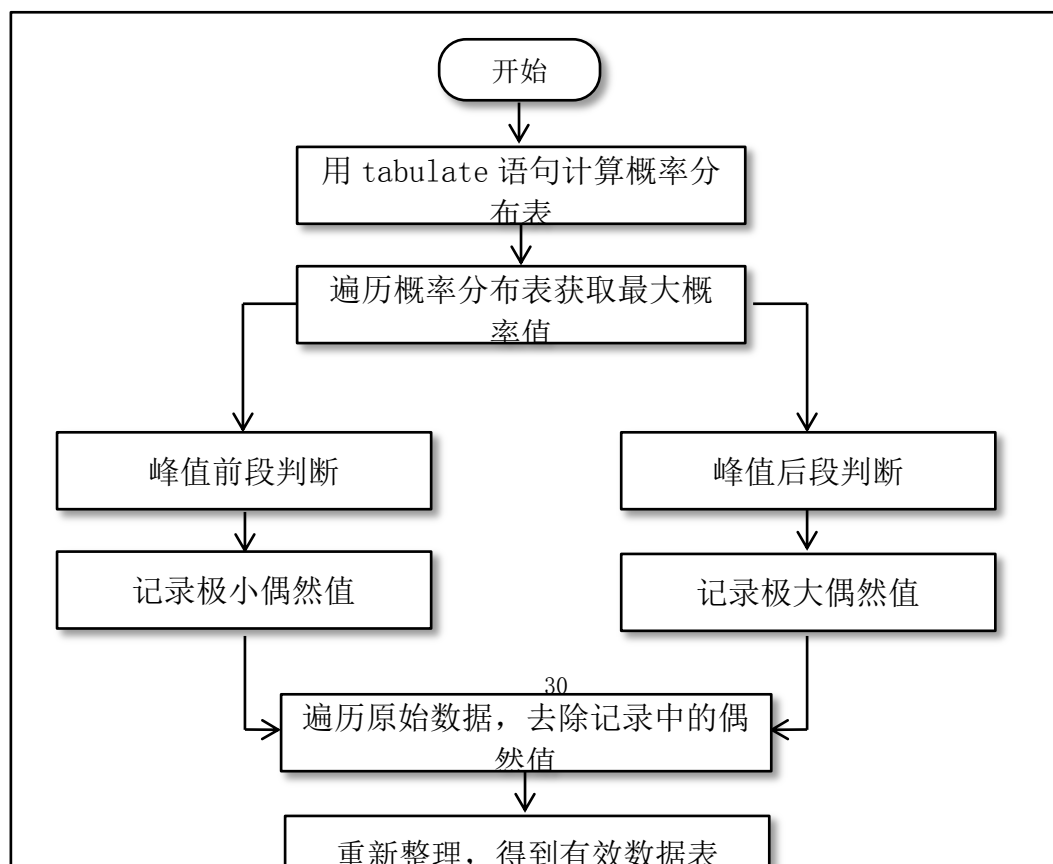


图 4-12 Matlab 数据筛选流程图

Tabulate 语句展示如下：

```
TABLE = tabulate(x);
```

%创建一个记录数据集 X 内各个数据的频率，其结果分为三列，首列是从 0 开始直到 X 内最大数值的无重复数据项，第二列是该数据出现的次数，第三列是该数据的出现的概率(%)。

概率分布表具体如下图 4-13 所示：

数据项	出现次数	出现概率 (%)
97	5	0.27173913
98	5	0.27173913
99	6	0.326086957
100	7	0.380434783
101	3	0.163043478
102	10	0.543478261
103	15	0.815217391
104	10	0.543478261
105	12	0.652173913
106	20	1.086956522
107	15	0.815217391
108	20	1.086956522
109	20	1.086956522
110	27	1.467391304
111	21	1.141304348

图 4-13 数据项概率分布表

(以线路编码为 80090 从岗顶到师大暨大的站间行程时间数据为例)

通过遍历该表，循环判断出现概率极小且偏离分布集中范围较远的数据，将该数值记录下来，进行排除范围的确定，排除前后数据分布范围对比如图 4-14 所示：

原始数据概率分布表			筛选后数据概率分布表的数据		
数据项	出现次数	出现概率 (%)	数据项	出现次数	出现概率 (%)
546	1	0.054348	223	1	0.054348
521	1	0.054348	222	0	0
272	1	0.054348	221	1	0.054348
251	1	0.054348	220	0	0
243	3	0.163043	219	1	0.054348
239	1	0.054348	218	0	0
235	1	0.054348	217	1	0.054348
233	1	0.054348	216	0	0
227	1	0.054348	215	1	0.054348
223	1	0.054348	214	0	0
222	0	0	213	1	0.054348
221	1	0.054348	212	0	0
220	0	0	211	1	0.054348
219	1	0.054348	210	0	0
218	0	0	209	0	0
217	1	0.054348	208	0	0
216	0	0	207	2	0.108696
215	1	0.054348	206	1	0.054348

图 4-14 Matlab 数据范围筛选前后对比

（以线路编码为 80090 从岗顶到师大暨大的站间行程时间数据为例）

注：红线为分割线，经过筛选，红线后的数据被排除，红字为偶然数据值。

由上表可以看出除了延误程度较小的站间行程时间数据之外，还有与正常值范围偏差较大的站间行程时间数据，即红字标注部分。根据其对应的原始报站数据可以简单推测其形成原因。站间行程时间数值 546 和数值 521 的来源如图 4-15 和图 4-16 所示：

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME
382	BRT营运公司	00804432	981073	80090	99	岗顶2	进站	2013/10/12 22:15:05
383	BRT营运公司	00804432	981073	80090	99	岗顶2	出站	2013/10/12 22:15:30
384	BRT营运公司	00804432	981073	80090	99	师大暨大4	进站	2013/10/12 22:24:36
385	BRT营运公司	00804432	981073	80090	99	师大暨大4	出站	2013/10/12 22:26:24

图 4-15 站间行程时间错误数据——数值 546

（以线路编码为 80090 车辆编号为 00804432 从岗顶到师大暨大的站间行程时间数据为例）

	ORGANNAME	BUSCODE	OBUID	ROUTECODE	SERVICE	BUSSTOP	ADFLAG	ADTIME	
411	BRT营运公司	00804423	981071	80090	99	岗顶2	进站	2013/10/12 22:15:44	▼
412	BRT营运公司	00804423	981071	80090	99	岗顶2	出站	2013/10/12 22:16:30	▼
413	BRT营运公司	00804423	981071	80090	99	师大暨大4	进站	2013/10/12 22:25:11	▼
414	BRT营运公司	00804423	981071	80090	99	师大暨大4	出站	2013/10/12 22:29:07	▼

图 4-16 站间行程时间错误数据——数值 521

(以线路编码为 80090 车辆编号为 00804423 从岗顶到师大暨大的站间行程时间数据为例)

由图 4-15 和图 4-16 可看出,以上报站时间数据记录时间相近,且在线路编号为 80090 的线路中,师大暨大为其在方向编号为 99 的行程中经过的最后一个 BRT 站台,与其他站台不同,其所受影响因素更多,出现偶然值的可能性也更大。在此可以推测该偶然数值是因为进站延误引起的。

4.2.2 分布类型判定

本节研究的是如何将已有数据集直观表现出来,并根据其大概分布判断分布类型,主要涉及的两个 Matlab 语句为 hist 语句和 ksdensity 语句,展示如下:

```
hist(Y,nbins);
```

%绘制 Y 数据集的直方图,横轴为数据,纵轴为该数据对应的频数,nbins 是个标量,指定数据集 Y 所分的小区间数。

在本程序中,为了能精确定位到每一个数据项出现次数,该小区间数为数据集 Y 中的最大数值减去其最小数值。

```
[f,xi] = ksdensity(x);
```

%计算 X 数据集的概率密度分布,将 x 的范围均分一百份,计算每一份 X 数据集内数据的概率密度,返回值 xi 为均分后 X 的其中一份的均值,f 为 xi 的概率密度

其绘制结果如下图 4-17 所示:

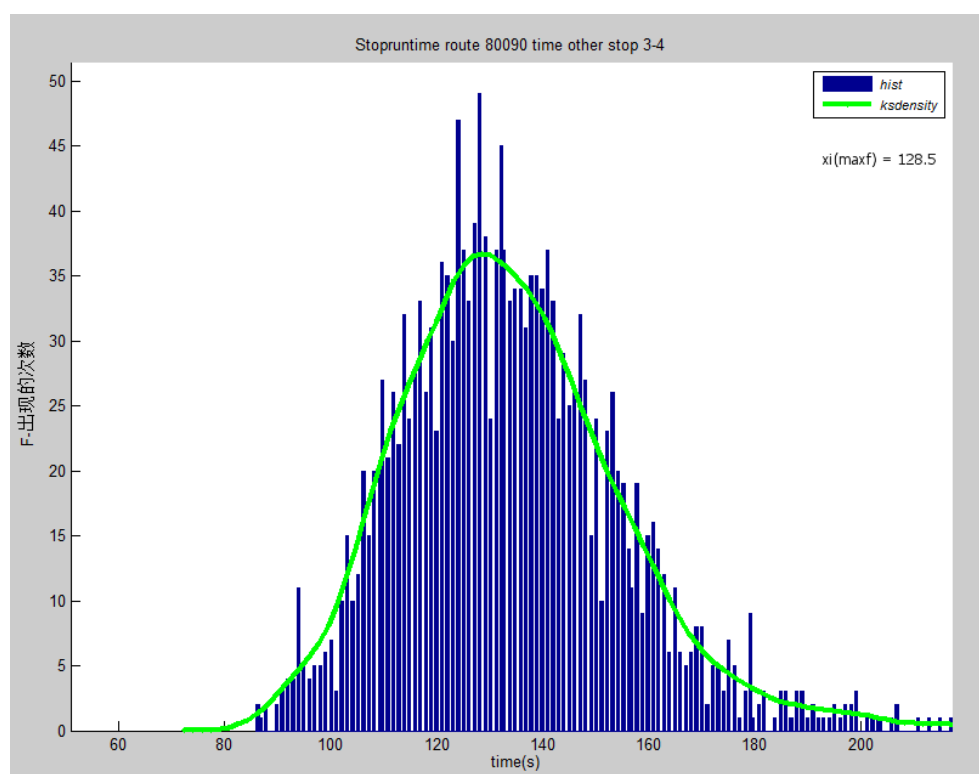


图 4-17 站间行程时间分布直方图和概率密度分布曲线(以线路编码为 80090 且在平峰时间段从岗顶到师大暨大的站间行程时间数据为例)

注：其中 $xi(maxf)$ 为当频数最大时对应的 X 数值。

由上图可直观判断车辆行程时间分布近似正态分布和对数正态分布。

4.2.3 正态分布和对数正态分布检验及参数确定

本节研究应用 Matlab 里面自带语句 `fit` 语句进行分布拟合并计算分布参数，并根据分布参数生成相应的累积分布模型，运用 `kstest` 语句检验实际数据与生成模型是否相符合，并结合 `ttest` 均值判断大致估计实际数据更接近两种分布的哪一种。同时利用 `probplot` 函数来直观表现这两种分布成立的可能性。

以下为 Matlab 语句展示：

```
[mu, sigma]=normfit(A);
```

%对 A 数据集进行正态分布拟合，计算其均值 μ 和标准差 σ

```
parmhat=lognfit(A, alpha);
```

%对 A 数据集进行正态分布拟合，计算其均值 μ 和标准差 σ

```
p=normcdf(A, mu, sigma);
```

% 累计正态分布函数

```
p=logncdf(A, parmhat(1), parmhat(2));
```

% 累计对数正态分布函数

```
[h,p,ksstat,cv] = kstest(x,CDF,alpha);
```

%检验数据集 x 属于某种分布类型的假设是否成立, **CDF** 函数为某类型的累积分布函数, **alpha** 为显著性水平, 默认值为 0.05, 可根据需要适当放宽其范围, 返回值 **h** 为一个布尔值, $h=1$ 表示可以拒绝假设, $h=0$ 表示不可以拒绝假设, **P** 为原假设成立的概率, **ksstat** 为测试统计量的值, **cv** 为是否接受假设的临界值, 当 **ksstat** 小于 **cv** 的时候可以接受原假设; [3]

```
[h, sig, ci] = ttest(x, m, alpha, tail);
```

%检验数据 x 的关于均值的某一假设是否成立, 其中 **alpha** 为显著性水平, 究竟检验什么假设取决于 **tail** 的取值: $\text{tail} = 0$, 检验假设“ x 的均值等于 m ”, $\text{tail} = 1$, 检验假设“ x 的均值大于 m ”, $\text{tail} = -1$, 检验假设“ x 的均值小于 m ”, **tail** 的缺省值为 0, **alpha** 的缺省值为 0.05. 返回值 **h** 为一个布尔值, $h=1$ 表示可以拒绝假设, $h=0$ 表示不可以拒绝假设, **sig** 为假设成立的概率, **ci** 为均值的 $1-\alpha$ 置信区间。当 $\text{xi}(\text{maxf}) < \mu(x)$ 的时候, 数据会更接近对数正态分布。在此处, m 的取值为分布概率最大的 xi 值, 即 $\text{xi}(\text{maxf})$ 。

```
probplot(distribution,Y);
```

%表现 **Y** 数据集符合某种分布的可能性, **distribution** 指定分布类型

具体参数记录格式如下图 4-18 所示:

				kstest								ttest		
type	route code	stopnum	time	normal distribution				lognormal distrubution				tail	sig	ci
				μ	σ	h	p	μ	σ	h	p			
stoprun time	80090	3-4	7-9am	126.0617	16.18343	0	0.051703	4.828852	0.125102651	0	0.381076	-1	1.14E-291	[124.5786,Inf]
			5-7pm	151.9520	26.23691	1	0.008949	5.0094	0.167012838	0	0.170988	-1	4.84E-194	[149.2124,Inf]
			other	134.0455	21.02856	1	0.000185	4.886291	0.153468016	0	0.285835	-1	0.00E+00	[133.2350,Inf]

图 4-18 正态分布和对数正态分布的参数及 **kstest** 和 **ttest** 的返回值

(以线路编码为 80090 且在不同时间段从岗顶到师大暨大的站间行程时间数据为例)

其可能性图如下：

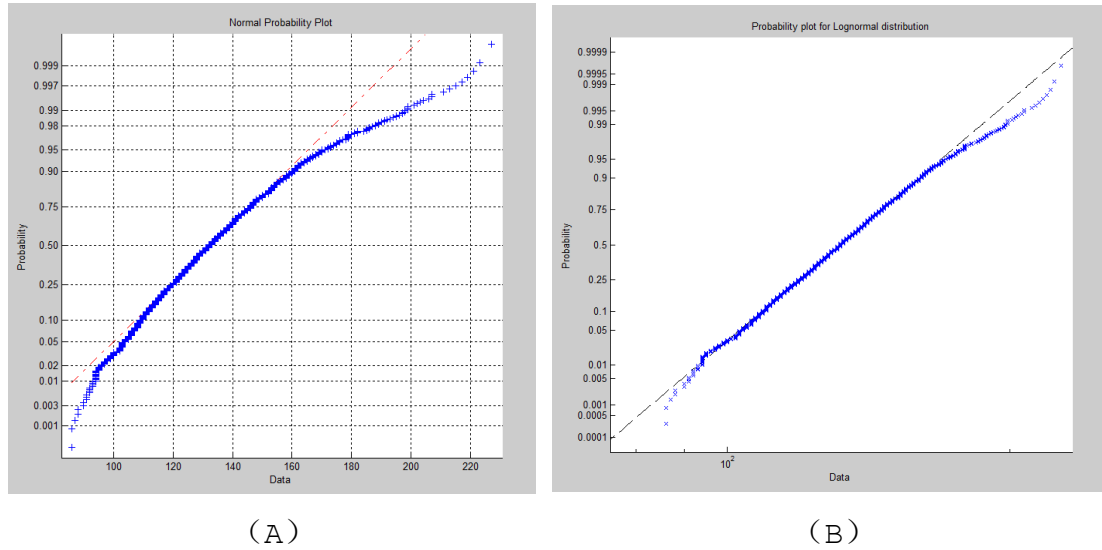


图 4-19 分布符合可能性图

(以线路编码为 80090 且在平峰时间段从岗顶到师大暨大的站间行程时间数据为例)

注：A 图基于正态分布，B 图基于对数正态分布

在可能性图中，当数据散点排列得越接近既定直线，则符合该分布的可能性就越大。由上图可以看出，基于对数正态分布可能性图中，散点更为接近既定直线。在 $ttest$ 中，其成立假设为正态分布均值大于分布概率最大的 x_i 值，说明该数据分布属于右偏态分布，而对数正态分布就为右偏态分布的一种，且在 $kstest$ 的返回值 p 中也可看出，正态分布检验的 p 值仅为 0.000185，而对数正态分布检验的 p 值为 0.285835，以默认值为 0.05 判断，正态分布检验的 p 值远小于默认值，而对数正态分布检验的 p 值大于默认值，由此也可看出符合对数正态分布的可能性更大。

4.2.4 正态分布和对数正态分布拟合曲线绘制

本节主要根据上一节所算出的正态分布和对数正态分布的参数来绘制对应的正态分布曲线和对数正态分布曲线，并将其与 $ksdensity$ 所绘制的曲线进行比较。Matlab 中计算分布曲线各点的函数如下所示：

```
Y = normpdf(X, mu, sigma);
```

%对于给定的 μ , σ , $\text{normpdf}(X, \mu, \sigma)$ 计算正态概率密度在 X 处的值。

$Y = \text{lognpdf}(X, \mu, \sigma);$

%对于给定的 μ , σ , $\text{lognpdf}(X, \mu, \sigma)$ 计算对数正态概率密度在 X 处的值。

其中正态分布的 μ 和 σ 由 normfit 函数算得, 对数正态分布的 μ 和 σ 由 lognfit 算得。

具体图像如下图 4-20 所示:

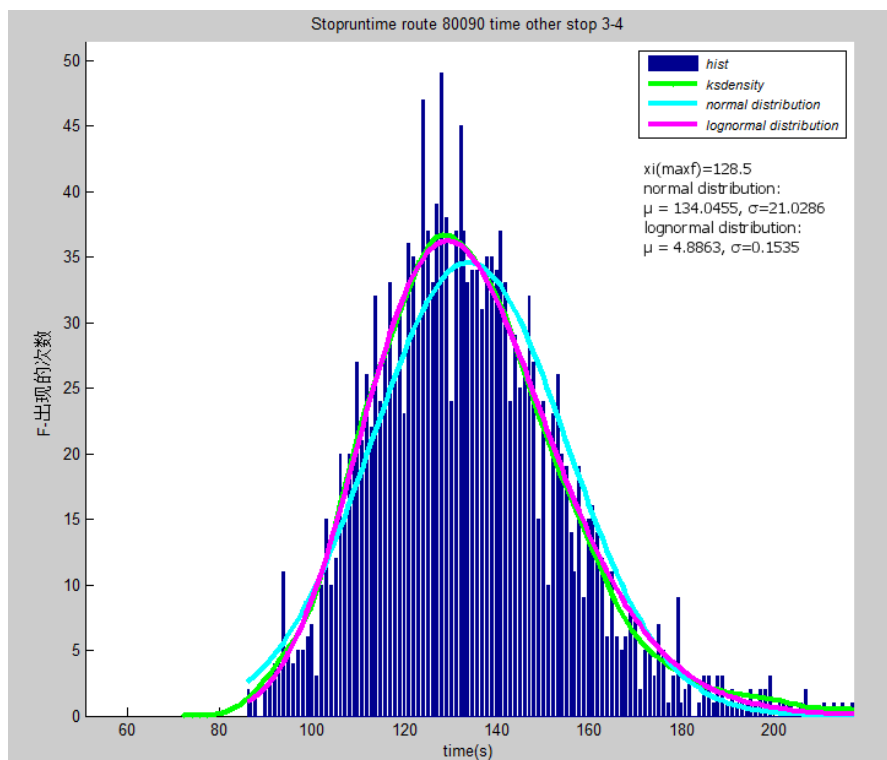


图 4-20 正态分布和对数正态分布曲线图

(以线路编码为 80090 且在平峰时间段从岗顶到师大暨大的站间行程时间数据为例)

由上图可以看出, 对数正态分布的曲线十分接近概率密度分布曲线, 且具有右偏态特性, 与检验结果相符合。

4.3 站间行程时间可靠性分析

本节主要通过累积概率分布函数获取截止各数据点的累积概率, 通过不同数据点累积概率相减得出行程时间落在两个数据点范围内的概率。数据点上下限的

设置主要考虑了乘客出行心理，在预计出行时间正负百分之二十的范围内都视为准点。主要涉及的 matlab 函数如下：

```
Y = cdf(name, X, A);
```

```
%累积概率分布函数
```

```
mu=mean(X);
```

```
%均值计算
```

绘制的累计概率分布曲线图如下：

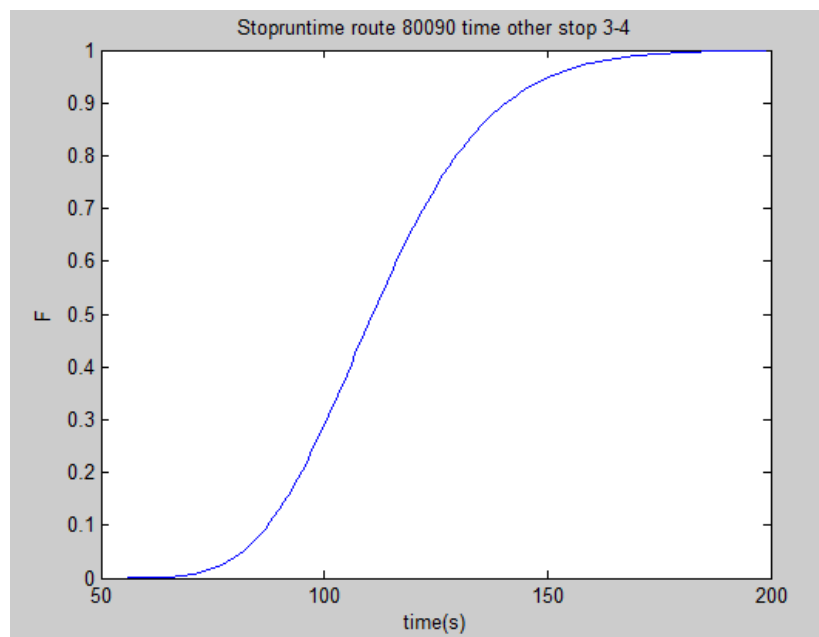


图 4-21 累积概率分布图

设定的百分之二十的波动范围如黑线所截止的范围：

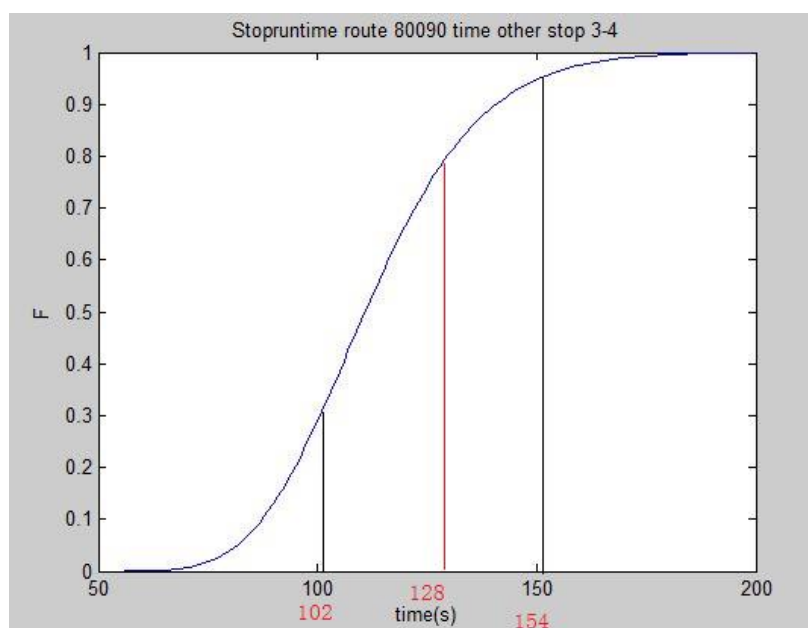


图 4-22 准点可靠性的分位点

通过当数值为 154 的累积概率减去数值为 102 的累积概率可以得到该站间运行时间的准点可靠性为 72.02%。

4.4 单一变量的公交站间行程时间数据对比及定性分析

公交行程途中会受多种因素的影响，比如车辆自身情况、道路情况、交通拥挤和天气等，还有行车途中红绿灯等待时间的影响，车辆在同两个公交站间行程时间会有很大变化。^[14]如果将各个线路各个时间段的站间行程时间杂糅在一起分析，会掩盖实际影响因素及其分布规律。本节主要研究内容是以线路、运行时间段为变量，将站间行程时间数据进行分类，逐个分析并对比，得出数据不同之处，结合实际道路情况、线路区别和运行时间段的影响来分析其不同之处。

4.4.1 以线路为变量的公交站间行程时间数据对比及分析

本小节线路编号为控制变量来分析相同两个 BRT 站间且在相同时间段内不同线路的站间运行时间分布，通过绘制概率密度分布曲线来直观比较不同线路之间的区别，同时获取其峰值和计算均值，通过这两个数值的偏差来大致推测影响因素。

以下为线路的概率密度分布曲线图范例：

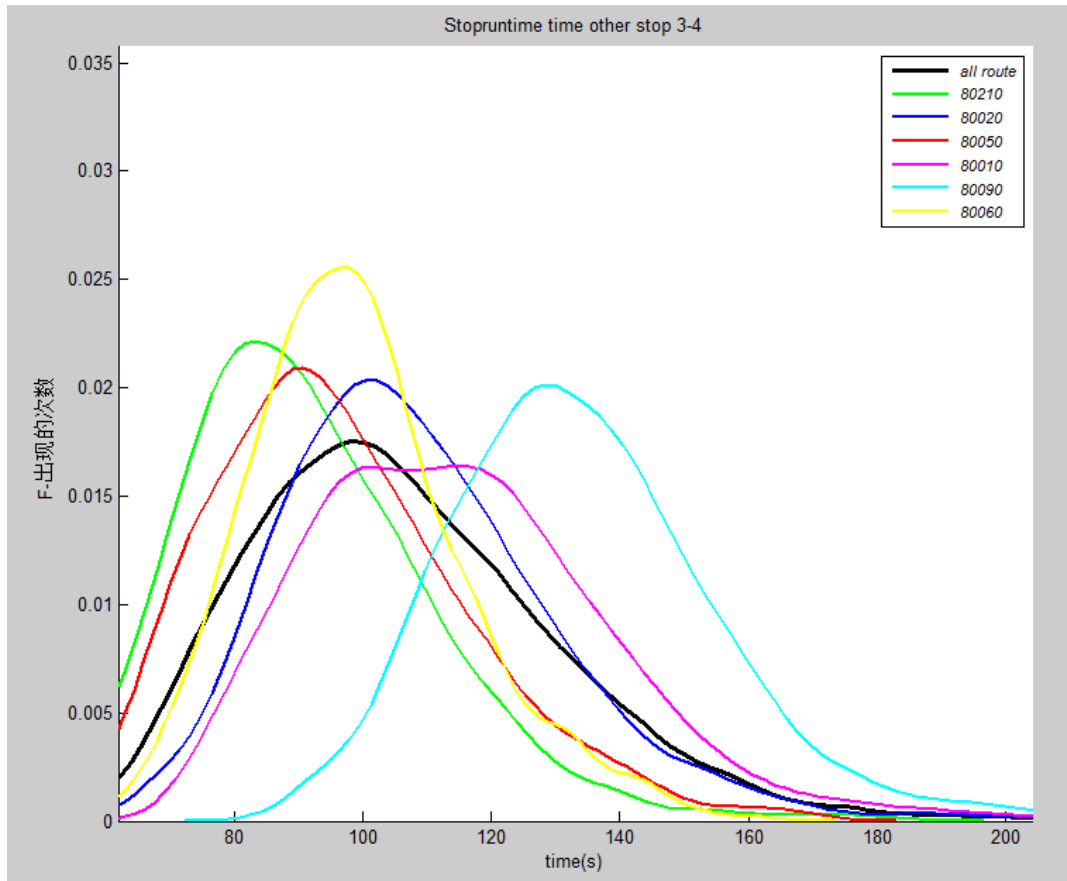


图 4-223 同站点同时间段不同线路概率密度分布图

注：黑线为所有在平峰时间段从岗顶到师大暨大的站间行程时间数据的概率密度分布曲线（以在平峰时间段从岗顶到师大暨大的站间行程时间数据，取线路编号 80210、80020、80050、80010、80090、80060 为例）

其计算数值如下图所示：

route code	μ	σ	$x_i(\max f)$
80210	91.4588	19.8037	92
80020	108.6094	21.4124	95
80050	96.1460	20.3614	98
80010	114.7843	22.9461	103
80090	134.0455	21.0286	128
80060	99.1424	16.6918	100
all	106.0571	24.1380	98

图 4-24 不同线路同站点同时间段的车辆站间行程时间分布参数及峰值

注： $x_i(\max f)$ 为当分布概率最高时对应的 x 值

（以在平峰时间段从岗顶到师大暨大的站间行程时间数据，取线路编号 80210、80020、80050、80010、80090、80060 为例）

从上表可以看出，站间行程时间分布并不全是右偏态分布，各线路的均值会在总体均值附近波动。

由于 BRT 有专用车道，其受车流影响较小，且路况相似，受红绿灯影响的几率相近，但是各线路的站间行程时间仍存在较大的偏差，说明站间行程时间会受到车辆本身和线路分布影响。

4.3.2 以运行时间段为变量的公交站间行程时间数据对比及分析

本小节将根据不同运行时间段划分同线路同站台的站间行程时间数据，其划分为早上 7 点到 9 点之间的早高峰、下午 5 点到 7 点之间的晚高峰和除这两段时间外的平峰时间段三段时间段。

其概率密度分布图如下：

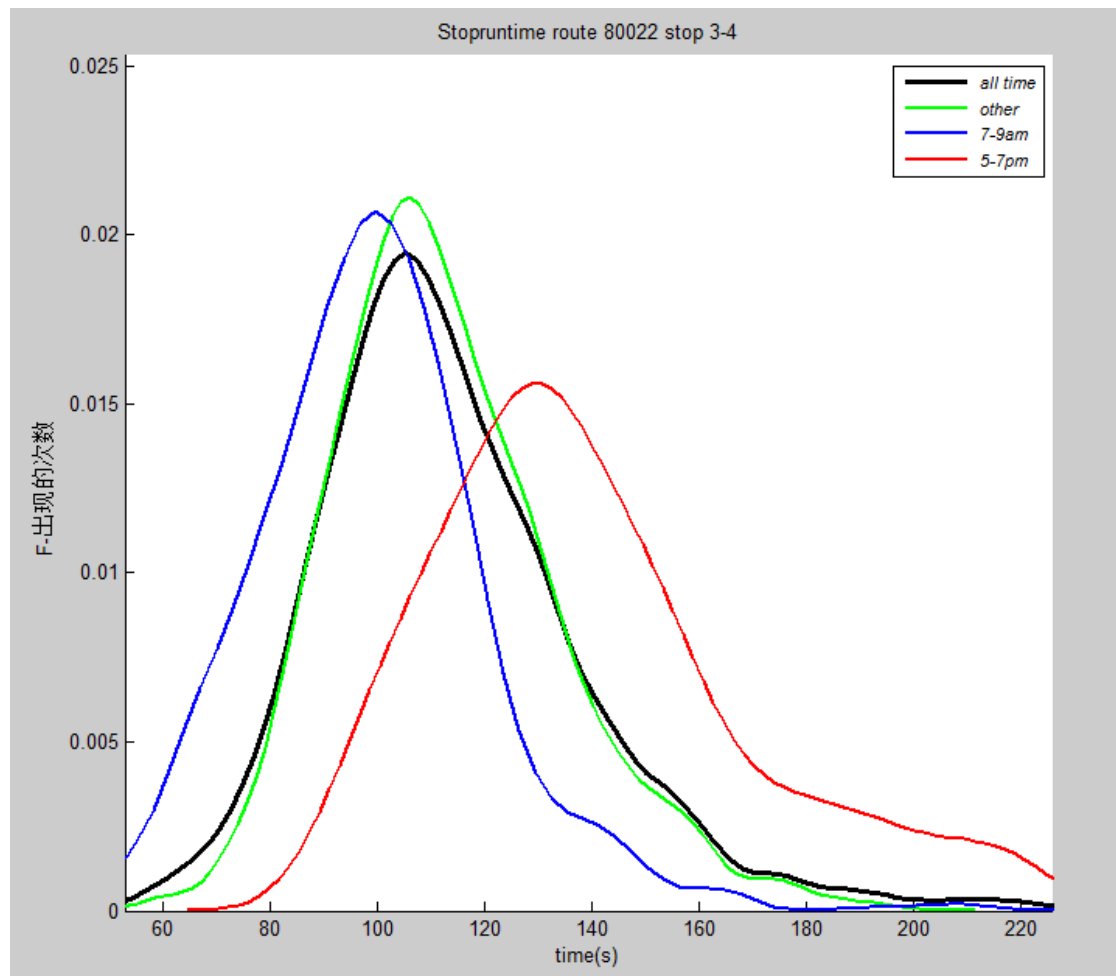


图 4-25 同站点同线路不同时间段车辆站间行程时间概率密度分布图

注：黑线为所有时间段数据总和的概率密度分布曲线

（以线路编号为 80022 从岗顶到师大暨大的站间行程时间数据为例）

其计算数值如下图所示：

time	μ	σ	xi(maxf)
other	112.8066	21.2962	105
7-9am	100.2443	25.5403	91
5-7pm	139.0503	29.4493	131
all	113.7596	25.0493	105

图 4-26 同站点同线路不同时间段车辆站间行程时间分布参数及峰值

（以线路编号为 80022 从岗顶到师大暨大的站间行程时间数据为例）

由图可以看出，其早高峰的站间行程时间均值较总体站间行程时间均值小，而平峰时间段的均值与总体时间的均值相近，而晚高峰的均值较总体时间均值大。可以看出上述站间行程时间数据分布都是符合右偏态分布，其分布也在总体时间均值的附近波动，说明运行时间段会影响到站间行程时间。

4.4 本章小结

本章主要针对上一章节提出的研究思路流程进行实例分析，其中包括数据筛选的算法验证、数据分布曲线拟合和在控制变量下的数据分析。结果表明，站间行程时间数据用正态分布和对数正态分布拟合是可行的，且站间行程时间的影响因素主要来源于 BRT 线路分布情况和运行时间段。

5 总结与展望

虽然正态分布和对数正态分布的应用领域已相当广泛,但其在对交通行程时间拟合方面的应用还比较少,而利用正态分布和对数正态分布来预测公交车辆站间行程时间是一个可行分方向,有助于为公交车辆站间行程时间预测提供依据,有助于提高公交系统的服务水平。

同时,类似的研究可以用于研究站台停靠时间,将站台停靠时间加入多个站之间的站间行程时间内,使得多站情况下站间行程时间更为精确。

参考文献:

- [1] 亓秀贞, 常规公交高峰时段运行可靠性评价方法研究[D]. 重庆交通大学, 2013, 6
- [2] 陈璇, 城市快速公共交通(BRT)综合评价方法研究[D]. 北京交通大学, 2012. 6
- [3] 高联雄, 智能公交系统数据挖掘研究与应用[D]. 中南大学, 2011. 6
- [4] 朱丽颖, 公交车辆行程时间预测方法研究[D]. 北京交通大学, 2010. 6
- [5] 丁亚民, 柳波, 考虑行程时间约束的快速公交网络可靠性[J]. 交通信息与安全, 2013, 2, 31:55-58
- [6] 龙佳, 公交车到站时间分布规律及其运行可靠性分析[D]. 中南大学, 2012, 5
- [7] 交通规划理论与方法[M] 北京:清华大学出版社, 2006.
- [8] 交通规划[M] 北京:人民交通出版社, 2007.
- [9] 刘汶菠, 基于行程时间可靠性的公交专用道线网优化决策研究[D]. 西南交通大学, 2012. 5
- [10] 刘锐, 严宝杰, 黄志鹏等. 可靠性理论在公交网络分析中的应用[J]. 公路交通科技, 2008, 25(4):122-126
- [11] 高桂凤, 魏华, 严宝杰. 城市公交服务质量可靠性评价研究[J]. 武汉理工大学学报(交通科学与工程版). 2007, 31(1): 140-14
- [12] 许连华, 李学庆, 基于 GPS 的公交车自动报站系统[J]. 计算机工程, 2005, 12, 31:191-192
- [13] 李玉辉, 戈春珍, 胡金成, 快速公交系统在我国的发展研究[J]. 辽宁交通科技, 2006, 1: 75-78
- [14] 陆奇志, 公共交通运行时间可靠性评价方法研究[D]. 新疆农业大学: 机械交通学院, 2006, 6

致谢

虽然这一部分放在了最后，但却不是最后写的，因为那种感激的心情让人按捺不住。

在此，首先要感谢导师胡继华对我的毕业设计的指导。从一开始的一头雾水到慢慢有了轮廓，再到朝着明确的方向进行，这其中的变化都是老师您耐心的指导形成的。在此谨向老师致以诚挚的谢意和崇高的敬意！

感谢程智锋师兄和李国源师兄的细心解答。当我还是一个小白，很简单的问题都需要问的时候，是他们不厌其烦地跟我解说，并鼓励我去克服更多的问题。特别是在程序的编写上，李国源师兄指导了我很多，让我编程的过程如此顺利。

感谢我的舍友和同学，在毕设进行到紧张阶段的时候，为我分担，一次次提醒我、鼓励我，让我能如期完成工作。

感谢中山大学，我衷心地为能进到如此氛围的学校而庆幸，您为我提供的平台和资源都让我成长。

最后感谢我的父母，有些话真的不知道怎么说出口，因为你们也不需要，我也拙于表达，但是那一点一滴心里自然有数。

回首大学四年，有舍不得，有遗憾，有哭有笑，但无悔。

附录

A. c#编写的报站数据筛选程序（数据筛选部分）

```
if (m == 0) //确保数据的第一条是正确且有效数据，逐条读取，直到读取到第一个进站数据
{
    if (adflag == 0) //第一条记录是进站记录
    {
        IOTool.WriteFile(filenamebusadtime, str);
        List<string> busadm = IOTool.ReadText(filenamebusadtime);
        m = busadm.Count - 1;
        adflagA = adflag;
        busstopnumA = busstopnum;
    }
    else //如果是出站，则忽略读取下一条，
    {
        m = 0;
        continue; //回到 Line72
    }
}
else //不是第一条记录的情况下
{
    adflagB = adflag;
    busstopnumB = busstopnum;

    if (adflagB != adflagA) //进出站连续的情况
    {
        if (adflagB == 1 && busstopnumB != busstopnumA) //排除 A 进 B 出的情况，处理是删掉 A，不记录 B
        {
            List<string> busadterror = IOTool.ReadText(filenamebusadtime);
            string errorline = Convert.ToString(busadterror.Count-1);
            IOTool.DeleteLine(filenamebusadtime, errorline);
            List<string> busadm = IOTool.ReadText(filenamebusadtime);
            m = busadm.Count - 1;

            if(m == 0) //如果是只有一条记录，删除了回到 m=0
            {
                m = 0;
                continue; //回到 Line72
            }

            List<string> busadreread = IOTool.ReadText(filenamebusadtime); //重新读取 adflagA 和 busstopnumA
```

```

string[] rereadlastline = busadreread[busadreread.Count-1].Split(' ');
busstopnumA=Convert.ToInt32(rereadlastline[1]);
adflagA=Convert.ToInt32(rereadlastline[2]);
continue;//回到 Line72
}
else//剩余 3 种情况, A 进 A 出, B 进 B 出, A 出 B 进, 非错误记录, 判断是否
重复, 记录
{

/*****写入数据段*****/
List<string> busadtime = IOTool.ReadText(filenamebusadtime);
bool bp = true;
for (int k = 0; k < busadtime.Count; k++)
{
    string[] adcheck = busadtime[k].Split(' ');
    string[] wstr = str.Split(' ');
    if (adcheck[4].Equals(wstr[4]) && adcheck[3].Equals(wstr[3]))//时间判
断是否重复
    {
        bp = false;
        break;
    }
}

if (bp == true)//不重复记录
{
    adflagA=adflagB;//更新判断标志
    busstopnumA=busstopnumB;
    IOTool.WriteFile(filenamebusadtime, str);
    List<string> busadm = IOTool.ReadText(filenamebusadtime);
    m = busadm.Count - 1;
    string buscode = Buscode[i];

    Calculation.Calculate(filenamebusadtime, m, str, buscode);
    continue;

}

/*****/
}
}
else//连续进站或者连续出站
{
    if (busstopnumB==busstopnumA)//A 进 A 进、A 出 A 出、B 进 B 进、B 出 B 出
    {

```



```

if(adflagB==1)//情况 A 出 A 出, B 出 B 出, 取后者, 删前一记录, 记录后一数据
{
    List<string> busadterror = IOTool.ReadText(filenamebusadtime);
    string errorline = Convert.ToString(busadterror.Count-1);
    IOTool.DeleteLine(filenamebusadtime, errorline);

    List<string> busadtime = IOTool.ReadText(filenamebusadtime);
    bool bp = true;
    for (int k = 0; k < busadtime.Count; k++)
    {
        string[] adcheck = busadtime[k].Split(' ');
        string[] wstr = str.Split(' ');
        if (adcheck[4].Equals(wstr[4]) && adcheck[3].Equals(wstr[3]))//如果已记录, 则不再记录, 若无, 则记录
        {
            bp = false;
            break;
        }
        if (bp == true)
        {
            adflagA=adflagB;
            busstopnumA=busstopnumB;
            IOTool.WriteFile(filenamebusadtime, str);
            List<string> busadm = IOTool.ReadText(filenamebusadtime);
            m = busadm.Count - 1;
            string buscode = Buscode[i];

            Calculation.Calculate(filenamebusadtime, m, str, buscode);
            continue;
        }
    }
    else //A 进 A 进、B 进 B 进, 不记录
    {
        continue;
    }

}
else //A 进 B 进、A 出 B 出
{
    if (adflagB == 0 && busstopnumB != busstopnumA)//排除 A 进 B 进, 删 A 进, 留 B 进
    {

```

```

List<string> busadterror = IOTool.ReadText(filenamebusadtime);
string errorline = Convert.ToString(busadterror.Count - 1);
IOTool.DeleteLine(filenamebusadtime, errorline);

adflagA = adflagB;
busstopnumA = busstopnumB;
List<string> busadtime = IOTool.ReadText(filenamebusadtime);
bool bp = true;
for (int k = 0; k < busadtime.Count; k++)
{
string[] adcheck = busadtime[k].Split(' ');
string[] wstr = str.Split(' ');
if (adcheck[4].Equals(wstr[4]) && adcheck[3].Equals(wstr[3]))//如果已
记录, 则不再记录, 若无, 则记录
{
bp = false;
break;
}
}
if (bp == true)
{
IOTool.WriteFile(filenamebusadtime, str);
List<string> busadm = IOTool.ReadText(filenamebusadtime);
m = busadm.Count - 1;
string buscode = Buscode[i];

Calculation.Calculate(filenamebusadtime, m, str, buscode);
continue;
}
}
else
{
continue;//A 出 B 出, 不记录 B 出
}
}
}
}
}
}
}
}

```

B. c#编写的站间行程时间计算程序（数据计算部分）

```

/*****计算数据段*****/
    if (m > 1)
    {
        List<string> busadreread = IOTool.ReadText(filenamebusadtime);

        string lstline = busadreread[busadreread.Count - 2];
        string[] lastline = busadreread[busadreread.Count - 2].Split(' ');
        string lastad = lastline[3] + " " + lastline[4];
        DateTime ladtime = Convert.ToDateTime(lastad);
        string lservice = Convert.ToString(lastline[5]);
        int lstopnum = Convert.ToInt32(lastline[1]);
        int ladflag = Convert.ToInt32(lastline[2]);

        string[] adcontent = busadreread[busadreread.Count - 1].Split(' ');
        string contentad = adcontent[3] + " " + adcontent[4];
        DateTime cadtime = Convert.ToDateTime(contentad);
        string cservice = Convert.ToString(adcontent[5]);
        int cstopnum = Convert.ToInt32(adcontent[1]);
        int cadflag = Convert.ToInt32(adcontent[2]);

        TimeSpan DeltaT = cadtime - ladtime;
        int DeltaStop = Math.Abs(lstopnum - cstopnum);

        if (lservice.Equals(cservice) && adcontent[0].Equals(lastline[0]))//
        判断是否同向且是否同线路，如果不同向，则两个数据无法计算
        {
            /*****全数据分析*****/
            if (DeltaStop == 0 && ladflag < cadflag && DeltaT.TotalSeconds < 700)//A
            进 A 出、B 进 B 出的情况
            {
                string filenamewt = "Stopwaittime_" + lastline[1] + "_" + cservice +
                ".txt";
                string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
                lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
                str + "" + buscode;
                IOTool.WriteFile(filenamewt, strd);
            }
            else if (DeltaStop == 1 && DeltaT.TotalSeconds < 1000)
            {
                string filename3 = "Stopruntime_" + lastline[1] + "_" + cstopnum + ".txt";
                string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
                lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
                str + "" + buscode;
            }
        }
    }

```

```

IOTool.WriteFile(filename3, strd);
}
/*****

/*****时间段分析*****/
if (DeltaStop == 0 && ladflag < cadflag && DeltaT.TotalSeconds < 700)//A
进 A 出、B 进 B 出的情况
{
if (cadtime.Hour > 6 && cadtime.Hour < 9)
{
string filenamewt = "Stopwaittime_" + "time_" + "7-9am_" + lastline[1]
+ "_" + cservice + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + Istline + "" +
str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
else if (cadtime.Hour > 16 && cadtime.Hour < 19)
{
string filenamewt = "Stopwaittime_" + "time_" + "5-7pm_" + lastline[1]
+ "_" + cservice + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + Istline + "" +
str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
else
{
string filenamewt = "Stopwaittime_" + "time_" + "other_" + lastline[1]
+ "_" + cservice + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + Istline + "" +
str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
}
else if (DeltaStop == 1 && ladflag > cadflag && DeltaT.TotalSeconds < 1000)
{
if (cadtime.Hour > 6 && cadtime.Hour < 9)
{
string filenamewt = "Stopruntime_" + "time_" + "7-9am_" + lastline[1] +
 "_" + cstopnum + ".txt";

```

```

string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenameewt, strd);
}
else if (cadtime.Hour > 16 && cadtime.Hour < 19)
{
string filenameewt = "Stopruntime_" + "time_" + "5-7pm_" + lastline[1] +
"_" + cstopnum + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenameewt, strd);
}
else
{
string filenameewt = "Stopruntime_" + "time_" + "other_" + lastline[1] +
"_" + cstopnum + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenameewt, strd);
}
}
/*****

/*****线路分析*****/

if (DeltaStop == 0 && ladflag < cadflag && DeltaT.TotalSeconds < 700)//A
进 A 出、B 进 B 出的情况
{
string filenameewt = "Stopwaittime_" + "route_" + adcontent[0] + "_" +
lastline[1] + "_" + cservice + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenameewt, strd);
}
else if (DeltaStop == 1 && DeltaT.TotalSeconds < 1000)
{
string filenameewt = "Stopruntime_" + "route_" + adcontent[0] + "_" +
lastline[1] + "_" + cstopnum + ".txt";

```

```

string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filename3, strd);
}

/*****

/*****线路时间分析*****/

if (DeltaStop == 0 && ladflag < cadflag && DeltaT.TotalSeconds < 700)//A
进 A 出、B 进 B 出的情况
{
if (cadtime.Hour > 6 && cadtime.Hour < 9)
{
string filenamewt = "Stopwaittime_" + "route_" + adcontent[0] + "_time_"
+ "7-9am_" + lastline[1] + "_" + cservice + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;IOTool.WriteFile(filenamewt, strd);
}
else if (cadtime.Hour > 16 && cadtime.Hour < 19)
{
string filenamewt = "Stopwaittime_" + "route_" + adcontent[0] + "_time_"
+ "5-7pm_" + lastline[1] + "_" + cservice + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
else
{
string filenamewt = "Stopwaittime_" + "route_" + adcontent[0] + "_time_"
+ "other_" + lastline[1] + "_" + cservice + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
}
else if (DeltaStop == 1 && DeltaT.TotalSeconds < 1000)
{
if (cadtime.Hour > 6 && cadtime.Hour < 9)
{

```

```

string filenamewt = "Stopruntime_" + "route_" + adcontent[0] + "_time_"
+ "7-9am_" + lastline[1] + "_" + cstopnum + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
else if (cadtime.Hour > 16 && cadtime.Hour < 19)
{
string filenamewt = "Stopruntime_" + "route_" + adcontent[0] + "_time_"
+ "5-7pm_" + lastline[1] + "_" + cstopnum + ".txt";
    string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + "
" + lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline +
"" + str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
else
{
string filenamewt = "Stopruntime_" + "route_" + adcontent[0] + "_time_"
+ "other_" + lastline[1] + "_" + cstopnum + ".txt";
string strd = adcontent[0] + " " + lastline[1] + "," + cstopnum + " " +
lastline[2] + cadflag + " " + DeltaT.TotalSeconds + "" + lstline + "" +
str + "" + buscode;
IOTool.WriteFile(filenamewt, strd);
}
}

/*****/

}

}

/*****/

```

C. Matlab 数据筛选、分布检验和曲线绘制程序段

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%错误或偶然数据的排除
```

```
table=tabulate(b(:));
```

```
%获取峰值
```

```
k=0;
```

```
s=0;
```

```
r=0;
```

```
maxt=max(table(:,3));
```

```
maxtime=max(table(:,1));
```

```
for t=1:maxtime
```

```
    if table(t,3) == maxt
```

```
        break;
```

```
    end
```

```
end
```

```
%峰值前半段判断
```

```
for j=1:t
```

```
    p=table(j,3);
```

```
    if p < maxt
```

```
        if p > 0
```

```
            s=s+1;
```

```
            if s > 5
```

```
                break;
```

```
            end
```

```
        elseif p == 0 && s>=0
```

```
            s=s-1;
```

```
        end
```

```
        continue;
```

```
    end
```

```
end
```

```
j=j-s;
```

```
%峰值后半段无效数据的排除
```

```
for i=t:maxtime
```

```
    q=table(i,3);
```

```
    if q < 0.02
```

```
        k=k+1;
```

```
        if k>5
```

```
            break;
```

```
        end
```

```
    elseif p > 0 && k>0
```

```
        k=k-1;
```



```

        end
    end
    i=i-k;

    %对数据重新整理，去除错误的数据
    newtable=table(j:i,:);
    maxst=max(newtable(:,1));
    minst=min(newtable(:,1));
    sortb=sort(b);
    sizeb=size(b);
    for ti=1:sizeb(1)
        if sortb(ti) == minst
            sortb(1:ti) = [];
            break;
        end
    end
    for tii=1:sizeb(1)-ti
        if sortb(tii)-maxst > 5
            sortb(tii:sizeb(1)-ti) = [];
            break;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%重新用有效数据计算
[f,xi]=ksdensity(sortb);
maxf=max(f);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%样本方差均值的计算
Delta=var(sortb);%方差
Mean=mean(sortb);%均值

%正态分布拟合判断
alpha=0.05;
[mu,sigma]=normfit(sortb);
p1=normcdf(sortb,mu,sigma);
[H1,p1,k1,c1]=kstest(sortb,[sortb,p1],alpha);%kstest
n=length(sortb);
if H1==0
    disp('该数据源服从正态分布。')
else

```

```

disp('该数据源不服从正态分布。')
end

for it=-1:1
    [hi, sig, ci] = ttest(sortb, maxf, 0.05, it); %ttest
    if hi==0 && it== -1
        disp('其均值比峰值大')
        disp(it);
        disp(hi);
        disp(sig);
        disp(ci);
    elseif hi == 0 && it == 0
        disp('其均值和峰值相等')
        disp(it);
        disp(hi);
        disp(sig);
        disp(ci);
    elseif hi == 0 && it == 1
        disp('其均值比峰值小')
        disp(it);
        disp(hi);
        disp(sig);
        disp(ci);
    end
end
disp(p1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

parmhat=lognfit(sortb, alpha);
p3=logncdf(sortb, parmhat(1), parmhat(2));
[H3, p3, k3, c3]=kstest(sortb, [sortb, p3], alpha);
if H3==0
    disp('该数据源服从对数正态分布。')
else
    disp('该数据源不服从对数正态分布。')
end
disp(p3)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%probability
figure(1)
normplot(sortb); %正态

```

```

figure(2)
probplot('lognormal', sortb); %对数正态
axis([minst-10, maxst+10, -4, 4]);

figure(3);
%» æ Ĩ ¼
sb1=size(sortb);
f=f*sb1(1);
maxfn=max(f);
%hist
hist(sortb, (maxst-minst)*2);
hold on;
%ksdensity
plot(xi, f, 'g.-', 'LineWidth', 3);
hold on;
%正态分布
fnorm=normpdf(sortb, mu, sigma);
fnorm=sb1(1)*fnorm;
plot(sortb, fnorm, 'c', 'LineWidth', 3);
hold on;
%对数正态分布
flogn=lognpdf(sortb, parmhat(1), parmhat(2));
flogn=sb1(1)*flogn;
plot(sortb, flogn, 'm', 'LineWidth', 3);
hold on;
axis([mu-3*sigma-20, mu+3*sigma+20, 0, maxfn*1.4]);
box off
title('Stopruntime route 80090 time other stop 3-4');
xlabel('time(s)')
ylabel('F-3 ö Ĩ Ö μ Ä ' Î Ê ý ')
legend('\fontsize{8}\it hist', '\fontsize{8}\it
ksdensity', '\fontsize{8}\it normal distribution', '\fontsize{8}\it
lognormal distribution')

```