同濟大學
TONGJI UNIVERSITY

交通数据分析---期中作业

同济大学交通运输与工程学院

综合交通信息与控制工程系

1854084 徐晨龙

2021.04

Part 01

数据说明

本数据取自 2017 年 2 月 6 日（星期一）至 2017 年 2 月 10 日（星期五）每天 0:00-12:00 的快速路微波车辆检测器数据。数据集内共包含 5 个检测器点位，检测范围内包括 2 个快速路入口匝道和 1 个快速路出口匝道。其具体布设位置如以下简图所示：

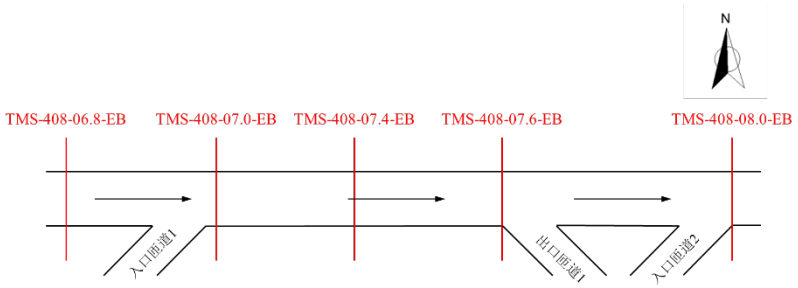

图 1：快速路微波的路面实际布置图

观察所给数据可以看出各检测线圈检测到的车道为：

$TMS-408-06.8-EB$: $line01$, $line02$, $line\ 03$

$TMS-408-07.0-EB$: $line01$, $line02$, $line\ 03$, $line04$

$TMS-408-07.4-EB$: $line01$, $line02$, $line\ 03$

$TMS-408-07.6-EB$: $line01$, $line02$, $line\ 03$, $line04$

$TMS-408-08.0-EB$: $line01$, $line02$, $line\ 03$, $Rmp-line04$

理论上完整的采集数据量为：

$$N = 5 * (12 * 60 + 1) * 18 = 64890$$

处理要求

(1) 分别采用最小值—最大值规范化法、$z-score$规范化法（标准偏差）对检测数据速度进行标准化处理。并分别提供均值与标准差。

(2) 分别运用阈值法的五分位数法、3 倍标准差法，剔除异常检测数据（速度、流量）。并给出异常数据的样本集

(3) 分别使用时间序列法、或基于历史数据的修补方法、或基于空间位置的修补方法，修复缺失检测数据（包括运用阈值法剔除异常数据后产生的空缺数据）。给出填补的数据样本集

数据分析结果

(1) 由于不同车道交通流参数分布的不同，因此首先将不同车道的数据分离，从数据和位置图之间的关系可以看出不同车道主要有：

$lane1, lane2, lane3, ramp1, ramp2, ramp3$

之后分别利用最小值-最大值规范化法、$z-score$规范化法对不同车道的数据进

行处理，得到对应数据处理前后的均值和标准差如下表所示：

表格 1：处理后的不同车道的均值和标准差

	原始数据		最小值-最大值规范化		$z-score$规范化	
	$mean$	sd	$mean$	sd	$mean$	sd
$lane1$	67.8	6.95	0.7	0.05	0	1.6
$lane2$	63.09	5.92	0.3	0.03	0	1.59
$lane3$	61.6	6.02	0.3	0.038	0	1.54
$ramp1$	49.98	6.23	0.55	0.098	0	1.34
$ramp2$	53.18	5.72	0.46	0.096	0	1.37
$ramp3$	45.8	7.14	0.48	0.12	0	1.3

处理方式是首先提取所选取的六条车道的信息，之后使用函数$data_standard = funciton(data)$，其作用对输入数据返回原始数据的均值和标准差、最小值-最大值规范后的均值和标准差、$z-score$的均值和标准差。

（2） 本案例中异常数据可以分为缺失数据、异常数据、重复冗余数据。由于数据中存在大量的缺失数据、冗余数据、异常数据。因此我们首先生成完整的表格（理论上 64890 行）

由于$date$、$tiemstamp$、$lane_id$可以唯一确定一个样本数据，因此首先生成完整地三行对应数据，之后新建的$New_midterm_data$中$speed$、$volume$、$occupancy$三列向量对应的默认值为-1，之后根据循环判断语句将对应的数据加到新建的数据框中

$$logic = (Midterm\_assignment\$date = \\ = data\_date[n]\&Midterm\_assignment\$timestamp = \\ = data\_time[n]\&Midterm\_assignment\$lane\_id == data\_lane[n])$$

寻找对应数据的逻辑语句如上式表示，其中重复数据值中逻辑下标值会存在多个，因此赋值是采用下式可以解决重复数据：

$$New\_Midterm\_data[n, c(5,6,7)] \\ = apply(Midterm\_assignment[logic, c(5,6,7)], 2, sum)/sum(logic)$$

这里仅采用“独立判断+联合判断+有效车长判断”来识别故障数据，对应函数$judge1 = function(data)$：

a. 独立判断，即根据流量、平均速度和占有率的取值范围来判断异常值，根据交通经验设置采取的取值范围如下

$$0 \le q \le 51(pcu/min)$$
$$0 \le v \le 80 * 1.4(km/h)$$
$$0 \le o \le 95$$

b. 利用流量、速度、占有率之间的逻辑关系进行判别，可能的情况有：

情况 1. 流量、速度、占有率均不为 0

情况 2. 流量、速度、占有率均为 0

情况 3. 流量、密度为 0，占有率超过 95%

情况 4. 流量超过 5pcu/min，车速不为 0，占有率为 0

c. 根据联合车长判断，根据单位换算$v(km/h)0(\%),q(pcu/min)$

$$Avel = 1000 * V * O/Q = v * o/6q$$

其中平均车长的合理范围(有部分数据车长约为 1.9m 也认为正常，因此将 2m 改为 1.5m)为：

$$1.5m < Avel < 22m$$

独立判断可以根据不同变量的阈值范围来单独判断是否出错，因此如果出错会单独设置对应的标记位。对于联合判断和有效车长判断的情况，由于其对应的三个之间对应的关系，因此如果出错会将三者的标志位均记为 1

经过计算可得，故障数据的样本集为：

	date	timestamp	detector_id	lane_id	type	speed	s_ab	volume	v_ab	occupancy	o_ab	error
1	2017-02-10	1899-12-31 05:51:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Rmp-Lane04	Ramp	37	1	1	1	4	1	3
2	2017-02-10	1899-12-31 09:01:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Rmp-Lane04	Ramp	45	1	2	1	6	1	3
3	2017-02-09	1899-12-31 11:01:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane01	Mainline	58	1	1	1	6	1	3
4	2017-02-09	1899-12-31 11:01:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane03	Mainline	51	1	5	1	30	1	3
5	2017-02-08	1899-12-31 00:38:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane04	Ramp	53	1	1	1	3	1	3
6	2017-02-08	1899-12-31 00:39:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane03	Mainline	59	1	1	1	4	1	3
7	2017-02-08	1899-12-31 00:42:00	TMS-408-07.6-EB	TMS-408-07.6-EB-Lane04	Ramp	43	1	1	1	4	1	3
8	2017-02-08	1899-12-31 00:47:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane02	Mainline	53	1	2	1	9	1	3
9	2017-02-08	1899-12-31 00:47:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane02	Mainline	55	1	2	1	9	1	3
10	2017-02-08	1899-12-31 01:25:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Lane03	Mainline	72	1	1	1	2	1	3
11	2017-02-08	1899-12-31 07:50:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane01	Mainline	0	1	30	1	17	1	3
12	2017-02-06	1899-12-31 05:11:00	TMS-408-07.4-EB	TMS-408-07.4-EB-Lane03	Mainline	70	1	1	1	2	1	3
13	2017-02-10	1899-12-31 07:22:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Lane01	Mainline	64	0	30	0	114	1	1
14	2017-02-09	1899-12-31 08:18:00	TMS-408-07.6-EB	TMS-408-07.6-EB-Lane03	Mainline	170	1	29	0	12	0	1
15	2017-02-09	1899-12-31 08:21:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane02	Mainline	180	1	29	0	14	0	1
16	2017-02-08	1899-12-31 07:05:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Lane01	Mainline	65	0	30	0	120	1	1
17	2017-02-08	1899-12-31 07:13:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Lane01	Mainline	63	0	30	0	-30	1	1
18	2017-02-08	1899-12-31 08:35:00	TMS-408-07.4-EB	TMS-408-07.4-EB-Lane02	Mainline	65	0	200	1	15	0	1
19	2017-02-07	1899-12-31 07:14:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane01	Mainline	-30	1	30	0	14	0	1
20	2017-02-07	1899-12-31 07:30:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane01	Mainline	60	0	-20	1	17	0	1
21	2017-02-06	1899-12-31 07:25:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane01	Mainline	34	0	60	1	32	0	1

图 2：异常数据样本集

	date	timestamp	detector_id	lane_id	speed	s_ab	volume	v_ab	occupancy	o_ab	error
8	2017-02-10	1899-12-31 00:00:00	TMS-408-07.4-EB	TMS-408-07.4-EB-Lane01	-1	1	-1	1	-1	1	3
18	2017-02-10	1899-12-31 00:00:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Rmp-Lane04	-1	1	-1	1	-1	1	3
32	2017-02-10	1899-12-31 00:01:00	TMS-408-07.6-EB	TMS-408-07.6-EB-Lane04	-1	1	-1	1	-1	1	3
37	2017-02-10	1899-12-31 00:02:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane01	-1	1	-1	1	-1	1	3
40	2017-02-10	1899-12-31 00:02:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane01	-1	1	-1	1	-1	1	3
44	2017-02-10	1899-12-31 00:02:00	TMS-408-07.4-EB	TMS-408-07.4-EB-Lane01	-1	1	-1	1	-1	1	3
54	2017-02-10	1899-12-31 00:02:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Rmp-Lane04	-1	1	-1	1	-1	1	3
55	2017-02-10	1899-12-31 00:03:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane01	-1	1	-1	1	-1	1	3
61	2017-02-10	1899-12-31 00:03:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane04	-1	1	-1	1	-1	1	3
62	2017-02-10	1899-12-31 00:03:00	TMS-408-07.4-EB	TMS-408-07.4-EB-Lane01	-1	1	-1	1	-1	1	3
65	2017-02-10	1899-12-31 00:03:00	TMS-408-07.6-EB	TMS-408-07.6-EB-Lane01	-1	1	-1	1	-1	1	3
73	2017-02-10	1899-12-31 00:04:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane01	-1	1	-1	1	-1	1	3
74	2017-02-10	1899-12-31 00:04:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane02	-1	1	-1	1	-1	1	3
76	2017-02-10	1899-12-31 00:04:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane01	-1	1	-1	1	-1	1	3
79	2017-02-10	1899-12-31 00:04:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane04	-1	1	-1	1	-1	1	3
80	2017-02-10	1899-12-31 00:04:00	TMS-408-07.4-EB	TMS-408-07.4-EB-Lane01	-1	1	-1	1	-1	1	3
83	2017-02-10	1899-12-31 00:04:00	TMS-408-07.6-EB	TMS-408-07.6-EB-Lane01	-1	1	-1	1	-1	1	3
86	2017-02-10	1899-12-31 00:04:00	TMS-408-07.6-EB	TMS-408-07.6-EB-Lane04	-1	1	-1	1	-1	1	3
87	2017-02-10	1899-12-31 00:04:00	TMS-408-08.0-EB	TMS-408-08.0-EB-Lane01	-1	1	-1	1	-1	1	3
93	2017-02-10	1899-12-31 00:05:00	TMS-408-06.8-EB	TMS-408-06.8-EB-Lane03	-1	1	-1	1	-1	1	3
94	2017-02-10	1899-12-31 00:05:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane01	-1	1	-1	1	-1	1	3
96	2017-02-10	1899-12-31 00:05:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane03	-1	1	-1	1	-1	1	3
97	2017-02-10	1899-12-31 00:05:00	TMS-408-07.0-EB	TMS-408-07.0-EB-Lane04	-1	1	-1	1	-1	1	3
98	2017-02-10	1899-12-31 00:05:00	TMS-408-07.4-EB	TMS-408-07.4-EB-Lane01	-1	1	-1	1	-1	1	3

图 3：缺失数据样本集

（3）对于缺失值的填补

表格中出现缺失值的情况主要有两种原因：

a. 第二问中，由于检测器异常导致删除的异常值，造成数据的缺失。

b. 由于检测自身的误差导致数据缺失，存在的异常值

修补方式为采用相同车道在时间或者空间上最接近的两点的正常值之和的平均值来进行修复。设置函数$data_correct$与$data_find$

$data\_correct$:寻找靠近错误数据的最近的两个正常值的平均值来代替错误数据

$data\_find$:用于进行修正

其中$data_correct$的逻辑判断过程为:

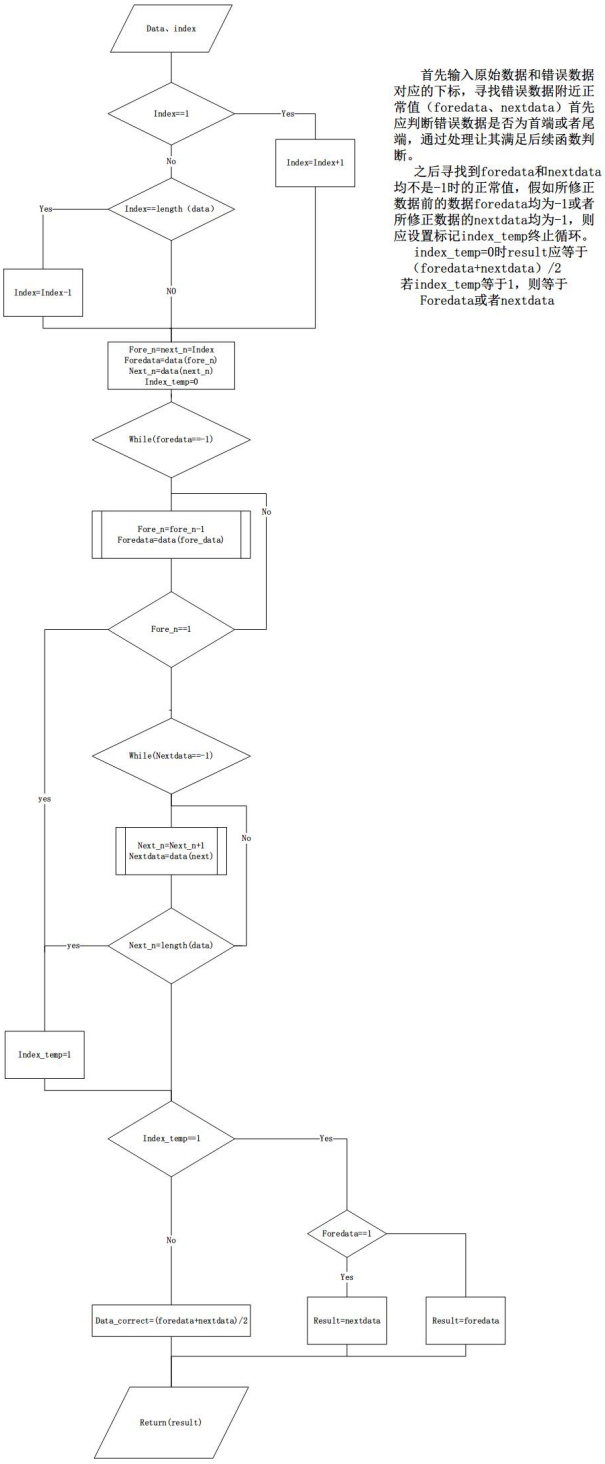

图 4：数据修正中$data_correct$函数思路

	date	timestamp	detector_i	lane_id	type	speed	volume	occupanc
1	2017/2/10	1899-12-31 07:22:00	TMS-408-	TMS-408-	Mainline	64	30	11.5
2	2017/2/9	1899-12-31 11:01:00	TMS-408-	TMS-408-	Mainline	71	7.5	2.5
3	2017/2/8	1899-12-31 07:05:00	TMS-408-	TMS-408-	Mainline	65	30	10
4	2017/2/8	1899-12-31 07:13:00	TMS-408-	TMS-408-	Mainline	63	30	14
5	2017/2/8	1899-12-31 07:50:00	TMS-408-	TMS-408-	Mainline	56.5	30	16
6	2017/2/7	1899-12-31 07:14:00	TMS-408-	TMS-408-	Mainline	64	30	14
7	2017/2/7	1899-12-31 07:30:00	TMS-408-	TMS-408-	Mainline	60	30	17
8	2017/2/6	1899-12-31 07:25:00	TMS-408-	TMS-408-	Mainline	34	28	32
9	2017/2/9	1899-12-31 08:21:00	TMS-408-	TMS-408-	Mainline	46	29	14
10	2017/2/8	1899-12-31 00:47:00	TMS-408-	TMS-408-	Mainline	58	2	5.5
11	2017/2/8	1899-12-31 00:47:00	TMS-408-	TMS-408-	Mainline	56.5	2.5	5
12	2017/2/8	1899-12-31 08:35:00	TMS-408-	TMS-408-	Mainline	65	27	15
13	2017/2/9	1899-12-31 08:18:00	TMS-408-	TMS-408-	Mainline	54	29	12
14	2017/2/9	1899-12-31 11:01:00	TMS-408-	TMS-408-	Mainline	61.5	14	8
15	2017/2/8	1899-12-31 00:39:00	TMS-408-	TMS-408-	Mainline	54.5	1	1
16	2017/2/8	1899-12-31 01:25:00	TMS-408-	TMS-408-	Mainline	61.5	1.5	0
17	2017/2/6	1899-12-31 05:11:00	TMS-408-	TMS-408-	Mainline	61	2.5	1.5
18	2017/2/8	1899-12-31 00:38:00	TMS-408-	TMS-408-	Ramp	36	2	2
19	2017/2/8	1899-12-31 00:42:00	TMS-408-	TMS-408-	Ramp	45	1.5	2.5
20	2017/2/10	1899-12-31 05:51:00	TMS-408-	TMS-408-	Ramp	52	1.5	0.5
21	2017/2/10	1899-12-31 09:01:00	TMS-408-	TMS-408-	Ramp	44	8.5	7.5

图 5:a 类数据修正

B 类数据由于数据量过多不在此展示，两者错误数据修正的结果写入$Midterm_data_error.csv$中

修正之后的完整样本集输出$Midterm_data_correct.csv$中

Part 02

数据说明

本数据取自一段快速路 3 年的事故相关数据, 包含 case(事件是否是事故, 1=crash, 0=non-crash), rain(事件发生时是否下雨， 1=rain, 0=non-rain), speed(事件的速度)。

处理要求

完成事故-下雨-速度，三者之间的两两相关性分析，给出具体的过程，计算检验值，并判断相关性是否显著。不建议采用程序自动计算的方法, 可以通过具体的公式, 一步一步计算。

处理过程

2.1 判断是否发生事故和是否下雨之间的相关性分析

$H0$: *假设发生事故和事件发生时下雨相互独立*

$H1$: *假设发生事故和事件发生时下雨有关联*

利用卡方检验来分析:

是否发生事故	事件是否下雨		合计
	1（是）	0（否）	
1（是）	30	95	125
0（否）	45	567	612
	75	662	737

此时可以考虑利用卡方检验，公式如下

$$\chi^2 = \Sigma_i^j \frac{(E_{ij} - T_{ij})^2}{E_{ij}}$$

是否发生事故	事件是否下雨		合计
	1（是）	0（否）	
1（是）	12.72	112.28	125
0（否）	62.27	549.72	612
	75	662	737

计算可得

$$\chi^2 = 31.4670$$

查表可得，$p_{value} < 0.00001$时，$\chi^2 = 29.6743$，因此有理由相信发生事故和发生事件下雨存在相关性关系。

2.2 判断是否发生事故和速度之间是否存在相关性

	Speed		Speed	
Value	Cash	Noncash	Rain	Non_rain
Mean	23.1	32.12	28.6	30.82
SD	10.76	13.87	12.55	13.94
Min	1.73	5.73	5.73	1.73
Max	67	85	70	85
t-value	8.09			

$$N_{cash} = 125\ N_{noncash} = 612$$

先做假设:

$$H0: \textit{发生事故和未发生事故的速度相互独立}$$

$$H1: \textit{发生事故和未发生事故的速度相互关联}$$

首先需要进行方差齐性检验:

$$H0: \sigma_1^2 = \sigma_2^2$$

$$H1: \sigma_1^2 \neq \sigma_2^2$$

$$F = \frac{\sigma_1^2}{\sigma_2^2} = 1.6616$$

$$F(611,124)_{threshold} = 1.2712$$

（利用$matlab: finv(0.95,611,124)$）

说明两组数据存在显著性差异，即方差不相等.之后进行 t 检验:

$$t_{value} = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2})}} = 8.09$$

其自由度为:

$$dt = \frac{\left(\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}\right)^2}{\frac{\left(\frac{S_1^2}{N_1}\right)^2}{N_1 - 1} + \frac{\left(\frac{S_2^2}{N_2}\right)^2}{N_2 - 1}} = 217$$

查表可得双侧置信度为 95%时

$$t(217)_{threshold} = 1.971$$

(利用$matlab: tinv(0.975,217)$

因此可以认为发生事故和速度存在关系。

2.3 判断是否下雨和事件的速度是否有关系

$$N_{rain} = 75 \quad N_{nonrain} = 662$$

同 2.2 先做方差齐性检验

$$F' = 1.23 < F(661,74)_{threshold} = 1.3575$$

(利用$matlab: finc(0,95,661,74)$)

说明两组数据并不存在显著性差异，即满足方差相等

$$S_p = 190.616 \qquad t_{value} = 1.31977$$

其自由度

$$N = N_1 + N_2 - 2 = 735$$

查表可得双侧置信度为 95%时

$$t(735)_{threshold} = 1.9632$$

（利用$matlab: tinv(0.975,735)$)

说明不可以认为是否下雨和速度存在关系

附录:

Midterm_assginment.xlsx 原始数据

New_Midterm_data.csv 包含错误和缺失的原始数据

Midterm_data_errort.csv 错误和缺失数据样本集

Midterm_data_correct.csv 改正后的完整样本集

Midterm_correct.R 第一问相关代码

```
library(readxl)
library(stringr)
Midterm_assignment <- read_excel("Midterm_assignment.xlsx")
Midterm_assignment=Midterm_assignment[,-5]#type 无用，为了后续处理方便删除type
unique(Midterm_assignment$lane_id)#观察有多少车道

##part01
#1.1 读取数据，分*_Lane01\*_Lane02\*_lane03\*07.0-EB-Lane04\*07.6-EB-Lane04\*08.0-EB-Rmp-Lane04
data_lane01=Midterm_assignment[endsWith(Midterm_assignment$lane_id,'Lane01'),]
```

```
data_lane02=Midterm_assignment[endsWith(Midterm_assignment$lane_id,'Lane02'),]
data_lane03=Midterm_assignment[endsWith(Midterm_assignment$lane_id,'Lane03'),]
data_Ramp1=Midterm_assignment[endsWith(Midterm_assignment$lane_id,'07.0-EB-Lane04'),]
data_Ramp2=Midterm_assignment[endsWith(Midterm_assignment$lane_id,'07.6-EB-Lane04'),]
data_Ramp3=Midterm_assignment[endsWith(Midterm_assignment$lane_id,'08.0-EB-Rmp-Lane04'),]
data_standard=function(data){
  standard1_data=(data-min(data))/(max(data)-min(data))

  sigma_data=sum(abs(data-mean(data)))/length(data)
  standard2_data=(data-mean(data))/sigma_data

  result=vector()
  result[1]=mean(data)
  result[2]=sd(data)
  result[3]=mean(standard1_data)
  result[4]=sd(standard1_data)
  result[5]=mean(standard2_data)
  result[6]=sd(standard2_data)
  return(result)
}
#计算均值
data_standard(data_lane01$speed)
data_standard(data_lane02$speed)
data_standard(data_lane03$speed)
data_standard(data_Ramp1$speed)
data_standard(data_Ramp2$speed)
data_standard(data_Ramp3$speed)

#1.2 异常值的处理（对不同车道实行阈值法）

#生成完整的没有缺失数据的数据框
data_date=rep(unique(Midterm_assignment$date),each=721*18)

temp_time=unique(Midterm_assignment$timestamp)
temptime=temp_time[c(1:100,721,101:720)]
data_time=rep(temptime,times=5,each=18)

temp_lane=unique(Midterm_assignment$lane_id)
```

```
temp_lane=temp_lane[c(1,2,3,4,5,6,7,17,8,9,10,11,12,13,14,15,16,18)]#调整顺序
data_lane=rep(temp_lane,times=721*5)

data_detector=str_sub(data_lane,1,15)
data_speed=rep(-1,times=5*721*18)
data_volume=rep(-1,times=5*721*18)
data_occupancy=rep(-1,times=5*721*18)

New_Midterm_data=data.frame(date=data_date,timestamp=data_time,detector_id=data_detector,
                            lane_id=data_lane,speed=data_speed,volume=data_volume,occupancy=data_occupancy)

for(n in 1:(5*721*18)){
  #将现有的数据加到新建的数据框中
  logic=(Midterm_assignment$date==data_date[n]&Midterm_assignment$timestamp==data_time[n]&Midterm_assignment$lane_id==data_lane[n])
  if(sum(logic)==1){
    New_Midterm_data[n,]=Midterm_assignment[logic,]
  }
  print(n)
}

judge1=function(data){
  #五分位数和三标准差法判断范围过大，采用单独的判断、联合判断和有效车长来进行判断，只有第一种可以区分三个参数中唯一出错的
  s_ab=(data$speed<0|data$speed>112)
  v_ab=(data$volume<0|data$volume>51)
  o_ab=(data$occupancy<0|data$occupancy>95)

  temp1=data$speed==0
  temp2=data$volume==0
  temp3=data$occupancy==0
  temp=temp1+temp2+temp3

  add_01=(((temp==3)|(temp==0)|data$occupancy>95|(data$volume<5&data$occupancy==0))|((s_ab+v_ab+o_ab)!=0))==0

  avel=data$speed*data$occupancy/(6*data$volume)
  add_02=(avel<1.5|avel>22)&((data$volume<5&data$occupancy==0)==0)&((s_ab+v_ab+o_ab+add_01)==0)

  s_ab=s_ab+add_01+add_02
```

```
  v_ab=v_ab+add_01+add_02
  o_ab=o_ab+add_01+add_02

  data['s_ab']=as.integer(s_ab!=0)
  data['v_ab']=as.integer(v_ab!=0)
  data['o_ab']=as.integer(o_ab!=0)
  data['error']=s_ab+v_ab+o_ab

  return(data)
}
New_Midterm_data=judge1(New_Midterm_data)
New_Midterm_data=New_Midterm_data[c(1,2,3,4,5,8,6,9,7,10,11)]#调整参数
write.csv(New_Midterm_data,file='New_Midterm_data.csv')

#1.3 缺失值的填补
data_find=function(data,n){
  #用来寻找异常值或者缺失值时空最接近点上正常的数

  #首先确定 n 是否为第一位或者最后以为
  if(n==1){
    n=n+1
  }
  if(n==length(data)){
    n=n-1
  }
  #初始化前值和后值，以及初始下标和标记位
  foredata=data[n-1]
  nextdata=data[n+1]
  fore_n=n
  next_n=n
  temp_fore=0
  temp_next=0
  #函数主体循环，说明
  while(foredata==-1){
    fore_n=fore_n -1
    #说明前面没有正确的值
    if(length(fore_n)==0){
      temp_fore=1
      break
    }
    foredata=data[fore_n]
    if(length(foredata)==0){
      foredata=0.01
    }
```

```
  }
  while(nextdata==-1){
    next_n=next_n+1
    if(nextdata>length(data)){
      temp_next=1
      break
    }
    nextdata=data[next_n]
    if(length(nextdata)==0){
      nextdata=0.01
    }
  }
  if((temp_fore!=1)&(temp_next!=1)){
    result=(foredata+nextdata)/2
  }else{
    if((temp_fore==1)&(temp_next!=1)){
      result=nextdata
    }else{
      if((temp_fore!=1)&(temp_next==1)){
        result=foredata
      }
    }
  }
  return(result)
}

data_correct=function(data){
  #不同进行修正
  index_speed=which(data$s_ab==1)
  index_volume=which(data$v_ab==1)
  index_occupancy=which(data$o_ab==1)
  for(index in index_speed)
  data$speed[index]=data_find(data$speed,index)
  for(index in index_volume)
    data$volume[index]=data_find(data$volume,index)
  for(index in index_occupancy)
    data$occupancy[index]=data_find(data$occupancy,index)
  return(data)
}

data_lane01=data_correct(New_Midterm_data[endsWith(New_Midterm_data$lane_id
,'Lane01'),])
data_lane02=data_correct(New_Midterm_data[endsWith(New_Midterm_data$lane_id
,'Lane02'),])
```

```
data_lane03=data_correct(New_Midterm_data[endsWith(New_Midterm_data$lane_id
,'Lane03'),])
data_Ramp1=data_correct(New_Midterm_data[endsWith(New_Midterm_data$lane_id,
'07.0-EB-Lane04'),])
data_Ramp2=data_correct(New_Midterm_data[endsWith(New_Midterm_data$lane_id,
'07.6-EB-Lane04'),])
data_Ramp3=data_correct(New_Midterm_data[endsWith(New_Midterm_data$lane_id,
'08.0-EB-Rmp-Lane04'),])

data_after_correct=rbind(data_lane01,data_lane02,data_lane03,data_Ramp1,dat
a_Ramp2,data_Ramp3)
data_error=data_after_correct[data_after_correct$error!=0,]
  write.csv(data_error,file='Midterm_data_error.csv')
```