

校园智能供水系统数据的深度挖掘

摘要

本题数据量较大，因此本文综合利用 Python 和 excel，在对数据进行整合与分割的基础上，完成了指标校准、异常值查找与分析、缺失值查找与补齐。

针对问题一：一方面，统计各水表数据的变化规律：基于各水表时间线上用量趋势变化的观察，对原始的波士顿矩阵进行坐标与内容的修正，并按照修正的矩阵分类对 92 个水表进行整理，逐类分析其变化规律。另一方面，分析功能区的用水特征：根据“水表名”，将水表逐个归入“宿舍、教学楼、办公楼、食堂、科研和生活”六个功能区，通过分析同一功能区内水表数据的变化曲线，得到对应的用水特征。如：宿舍区 2 月的用量较 1 月均出现下降趋势。

针对问题二：首先，水表用量的年度汇总数据与层级数据进行匹配，得到带年总用量的层级数据表。然后，根据不同层级的特征进行误差判断。无下级的水表数据，无需进行用量比对，为正常供水数据；有下级的水表数据，比对上级用量与下级用量之和，若上级用量大，则为正常供水数据，否则定义为不正常供水。最后，针对不正常供水的数据进行误差分析。其表计编码为 403X、405X、20133X 的误差率分别为 7.88%、48.98%、20.00%。

针对问题三：第一步：利用层级关系模型修正总表的不正常供水数据。第二步：用量数据存在读数重复性的问题，易导致漏水量的重复计量，采用层级逆推法处理，可得到无重复读数用量的数据表。第三步：根据实际的生活规律，用夜间最小用量的产生和持续来表示漏水是合理的估算方法，本文基于此构建输水管漏水模型。夜间漏水的计算，需先筛选 2:00 至 5:00 非零用量的数据，再统计水表名和时间下的数据频次，后按照连续不断失水的要求（“频次=11”）筛选出漏水数据。分析数据结果可知，出现漏水的水表有 74 个，即较多水表或其所对应段的水管可能存在漏水现象；漏水水表的平均漏水频次高达 67 次，意味着有些水表或水管存在老化失修的现象。

由于爆管漏水是常见的短时间大量漏失现象，构建爆漏可能性指数，选择孤立森林的异常值聚类模型进行爆漏分析。

针对问题四：夜间漏水近似估算的漏水，含明漏和暗漏。根据国标定义，暗漏为地下管道不易发现的漏水，而明漏为地上管道易发现的漏水。对漏水数据做时间连续性的要求可得到暗漏数据。利用该数据夜间内 15 分钟用量的最小值可近似估算 24 小时的暗漏量，汇总得年暗漏总量为 134936.30。而明漏的处理方式类似暗漏，但是按照白天工人正常上班时间，可估算 8 个小时的明漏量，汇总后得年明漏量为 11230.15。

针对问题五：维修管网会引起维修成本的增加与漏损水费的减少，且在维修后的较长一段时间内，管网漏损量不会增大。因此，扩大时间跨度分析维修成本与漏损水费减少值的变化关系。查找文献了解到水价与维修成本，在分析维修总成本构成的基础上，参照现金持有量成本模型，建立管网维修总成本模型。参照目标现金持有量模型，建立日最佳用水量动态控制模型。若最佳用水量小于实际用水量，且当天漏损量与最佳用水量的比值大于 10%时，依据模型计算得到最优维修决策方案所需的成本，反之不维修。

关键词：层级逆推法 夜间漏水 近似算法 爆漏可能性指数 成本性态分析控制

一、问题重述

校园供水系统使用了智能水表，获得了大量的实时供水系统运行数据。后勤部门希望基于这些数据，通过数学建模和数据挖掘及时发现和解决供水系统中存在的问题，提高校园服务和管理水平。附件是某校区水表层级关系以及所有水表四个季度的读数与相应的用水数据。请利用这些信息和数据，建立数学模型，讨论以下问题：

1. 统计、分析各个水表数据的变化规律，并给出校园内不同功能区的用水特征。
2. 结合校区水表层级关系，建立水表数据之间的关系模型，并利用已有数据分析模型误差。
3. 输水管网的漏损是一个严重问题。资料显示，在维护良好的公共供水网络中，平均失水在 5%左右；而在比较老旧的管网中，失水则会更多。请利用附件提供的数据，建立数学模型，分析该校园供水管网的漏损情况。
4. 地下水管暗漏不容易被发现，需要花费大量人力对供水管道的漏损进行检测及定位，如果能够从水表的实时数据及时发现并确定发生漏损的位置，将极为有益。请帮助学校解决这个问题。
5. 管网维修需要一定的人工费和材料费，但同时可以降低管网漏损程度。请根据以上结果和你了解的水价及维修成本确定管网漏损的最优维修决策方案。

二、问题分析

2.1 数据预处理

根据本题数据特点，数据需要进行整合、切割、指标校准、异常值和缺失值处理的操作。

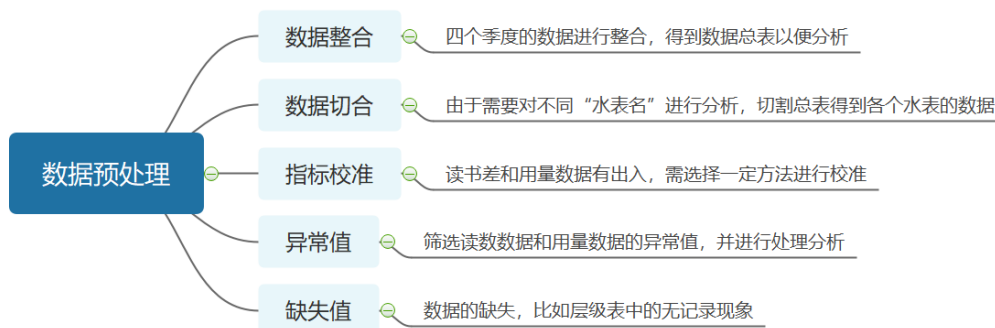


图 1 水表数据预处理的思维导图

利用 python 和 excel 软件进行上述情况进行分析，并选择合适的处理方法。

2.2 问题一分析

本题选择两个比较有针对性的方面进行数据统计描述和分析。一方面，明确各个水表数据的变化趋势类型。参考波士顿矩阵分类的方法，结合各水表数据变化特征，定义符合本题数据的波士顿矩阵，并将 92 个水表按照矩阵分类进行整理。另一方面，构建并分析校园功能区的用水特征。根据“水表名”所指示的功能类型，将水表归入不同的校园功能区内。为了进行比对，选择校园各功能区的具体用处不同，选择同级的水表指标进行功能区内的对比分析，探索功能区的数据特征。

2.3 问题二分析

根据水表层级关系的定义，水表的上级和下级的用量一般存在“产>销”的产销关系，即说明水表的上级要大于对应的下级总和。明确层级关系，并分析其误

差的步骤大致如下。

首先，基于层级数据表和水表数据年度汇总结果的匹配，得到带有年度总用量数据的层级数据表。**然后**，分析不同层级的特征，发现层级可分为两种：有下级的层级和无下级的层级。**其次**，针对无下级的水表数据，无需进行用量处理。针对有下级的水表数据，通过层级倒推计算得到相应的用量。如果不满足水表层级关系的要求（上级大于对应下级的总和），则定义为不正常供水数据。**最后**，针对不正常供水的数据进行误差分析。

2.4 问题三分析

查阅大量输水管漏水的文献之后，发现大部分输水管漏水的计算方法和定位技术都需要大量的指标支持，如管道压力和音听检漏法等。但是本题中所给有效计量指标只是用量，很难进行输水管漏水的准确计量。因此，结合相关文献^[4]与生活的实际规律，夜间最小流量虽然并不能代表真实的输水管漏水，但是在没有足够指标来计算漏水的情况下，用夜间最小流量的产生和持续来近似表示输水管漏水是一个比较合理的估算方法。

在利用夜间漏水近似估算漏水情况的基础上，本文期望构建精准的分类模型去推断爆漏的情况。

2.5 问题四分析

基于问题三中夜间漏水对漏水的估计算法，可以得到的是输水管漏水的近似值。根据相关文献^[6]，输水管漏水的数据中包括：明漏和暗漏，明漏是地面上易被发现的漏水，而暗漏指地面下难以被发现的漏水。

延续夜间漏水的计算思路，对夜间漏水的时间数据进行进一步分析。基于水表名和采集时间的排序结果，如果时间线上相邻两条夜间漏水数据是连续的，则表示漏水在中间的白天未被发现检修，可定义为暗漏，否则定义为明漏。本题主要根据“相邻时间差值=1”的判断准则，判定和分析暗漏的清理。

2.6 问题五分析

管网维修可以降低管网漏损程度，但同时会产生维修费用，即维修管网会引起维修成本的增加、漏损水费的减少。可参考维修成本与漏损水费两者间的平衡关系，判断是否需要进行管网维修。由于管网维修后，管网漏损在之后的长时间内都有改善。因此，应当扩大时间跨度，考虑维修成本与管网漏损费用的变化关系。

分析维修成本的组成，查阅文献数据计算可得相应的维修成本。依据维修成本、漏损水费与固定的管网管理费用，得到管网维修总成本模型。参照目标现金持有量模型的确定原理，结合水表数据可以求出日最佳用水量，同时建立日最佳用水量动态控制模型。从损耗量、用水量与最佳用水量三者间的关系出发，判断是否需要进行维修。通过维修总成本的关系模型，计算得到维修方案的资金预算。

三、模型假设

- 1.假设所有水表读数刻度精准；
- 2.假设 CPI 每年长 2%；
- 3.假设同一水表的每次维修费用均相等；
- 4.假设智能抄表系统可能存在重复抄表，但本题不给予考虑；
- 5.假设智能抄表系统不存在水管跨区域混接现象；
- 6.假设无法得到实际实时用量。

四、符号说明

| 符号 | 说明 | 符号 | 说明 |
|----------|---------|---------------------|------------------------|
| A | 实际供水量 | $\theta_{(i,name)}$ | 第 i 级某 $name$ 水管进户流量 |
| B | 二级供给水量 | $\delta_{(i,name)}$ | 第 i 级某 $name$ 水管漏水量 |
| C | 三级供给水量 | $L\ day_i$ | 第 i 天漏水 |
| D | 四级用水量 | $Y\ day_i$ | 第 i 天夜间漏水 |
| δ | 总体漏水量 | Q | 日最佳用水量 |
| θ | 水管总进户流量 | K | CPI 变化率 |

备注：模型内的参数说明见下文中的具体对应模块。

五、数据预处理

5.1 数据的整合和分割

将各个季度所有水表读数导入 Python，采用 concat 函数将四张季度水表读数的数据进行简单整合，得到一张数据总表。

表 1 数据总表各数据量

| 数据表名 | 季度 1 | 季度 2 | 季度 3 | 季度 4 | 总表 |
|------|--------|--------|--------|--------|---------|
| 数据量 | 729283 | 778195 | 791844 | 787466 | 3086788 |

备注：总表数据过大无法打开，处理总表数据皆应用 python 软件

利用 loc 函数的筛选功能，将总表按照“水表名”进行查找，导出各个水表的实时数据表，共 91 张。

5.2 用量数据的校准

附件中各水表的用水数据包括水表上次读数、当前读数和用量三类数据，三者关系中，水表当前读数与上次读数的差值应当等于用量，但实际情况并不全是如此。通过 excel 作差得到水表当前读数与上次读数的差值，对比相应的用量，得到图 1。

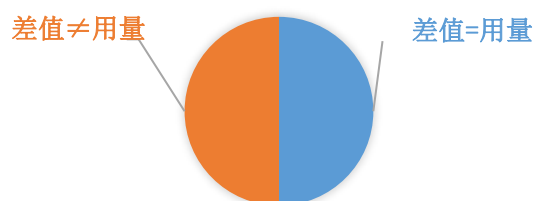


图 2 水表读数差值与用量的对比结果

(1) 水表差值不等于用量的数据占比为 43.18%，其中差值大于用量的占 21.43%，差值小于用量的占 21.75%。部分差值数据在用量数据上下波动，由于缺少用量数据的指标计算说明，因此无法解释这个情况。

(2) 根据差值和用量的做差的数据情况，发现只有一个数据是大于 0.01 的，其他数据皆为 ± 0.01 。

(3) 从上述两点，本文判定用量数据存在计算异常问题，用水表差值对用量进行校准。下文所出现的用量均为校准后的用量。

5.3 异常值的查找

5.3.1 四分位法查找用水数据的异常值

四分位法^[1]是分析数据集分布特征的主要方法,其原理是将全部数据按大小顺序依次排列并分为四等份,超出上四分位差距或下四分位差距则视为异常值。

上次读数、当前读数和用量三类数据中,当前读数和上次读数错行一致,因此选择上次读数和用量进行四分位法异常分析。采用 `describe` 函数,得到上次读数、用量的最大值、最小值、标准差以及四分位数。以用量的异常为例,图 1 显示了部分水表的异常情况。

表 2 部分异常情况

| 水表名 | 用量 25% | 用量 50% | 用量 75% | 用量 max | 异常情况 |
|--------|--------|--------|-------------|--------------|------------------|
| 老六楼 | 0.01 | 0.05 | 0.1 | 418.6 | 最大值超出四分位点很多,存在异常 |
| 第一学生宿舍 | 0.22 | 0.4 | 0.53 | 101.5 | 最大值超出四分位点很多,存在异常 |
| 航空航天 | 0 | 0 | 0 | 261.2 | 最大值超出四分位点很多,存在异常 |
| 校医院南+ | 0.25 | 0.75 | 1 | 11.68 | 最大值超出四分位点很多,存在异常 |

5.3.2 异常值分析

(1) 一段时间用量为零后出现大用量

以 XX 图书馆、XX 第一学生宿舍为例,水表的水表读数长时间未发生改变且用量一直未零,在某一时刻突然出现大用量,然后保持稳定增长。这种异常值情况存在较多。结合实际情况进行合理推断,该异常值出现的原因因为前期存在水表损坏而无法读数的现象,检修人员发现后进行检修,并根据之前实际用量调整了水表读数。

表 3 XX 图书馆用量异常值

| 水表名 | 采集时间 | 用量 | 水表名 | 采集时间 | 用量 |
|--------|----------------|-------|-----------|-----------------|-------|
| XX 图书馆 | 2019/7/2 15:45 | 0 | XX 第一学生宿舍 | 2019/2/27 15:00 | 0 |
| XX 图书馆 | | | XX 第一学生宿舍 | | |
| XX 图书馆 | 2019/7/11 0:00 | 0 | XX 第一学生宿舍 | 2019/3/5 8:00 | 0 |
| XX 图书馆 | 2019/7/11 0:15 | 25.3 | XX 第一学生宿舍 | 2019/3/5 8:15 | 101.5 |

(2) 水表读数的位数异常

第一季度“老六楼”的水表读数均为五位数,第二季度的水表读数为四位数。第一季度仅有 17 天的数据值,但参照这 17 天的水表读数可大致推断第一季度末的读数,结合第二季度初水表读数,合理推断出第一季度出现五位数的原因因为读取值的小数点向右偏移了一位。因此,此处异常值的处理方法为将第一季度的小数点向左移动一位。

表 4 水表读数的位数错误

| 水表名 | 水表号 | 采集时间 | 上次读数 | 当前读数 | 用量 |
|---------|------------|--------------------|------------|------------|-------|
| XXX 老六楼 | 3480300100 | 2019/1/1 00:15:00 | 21540.3400 | 21540.3400 | 0 |
| | | | | | |
| XXX 老六楼 | 3480300100 | 2019/1/17 09:00:00 | 21959.0000 | 21959.0000 | 0 |
| XXX 老六楼 | 3480300100 | 2019/4/10 13:30:00 | 2564.7000 | 2564.7000 | 0 |
| | | | | | |

(3) 水表读数增大下降继续增大

以 XXX 第二学生宿舍为例,水表读数增长达到一定数值后,突然降低至某一数值,然后继续增大。这种异常值情况存在较多,结合例子,合理推断水表读数不连续的转折原因为水表读数上升到一个值后水表“清零”。

表 5 水表读数存在清零

| 水表名 | 采集时间 | 上次读数 | 当前读数 |
|------------|------------------|---------|---------|
| XXX 第二学生宿舍 | | | |
| XXX 第二学生宿舍 | 2019/10/20 11:45 | 70323.7 | 70323.7 |
| XXX 第二学生宿舍 | 2019/10/20 12:00 | 611.4 | 611.4 |
| XXX 第二学生宿舍 | | | |

(4) 一段时间内用量远超常值

以“XXX 花圃+”为例，水表数据保持持续稳定增长，但一段时间内用量突然激增，远超长值。结合存在该种情况的水表名，推断该种情况为合理的特殊用水需求。

表 6 一段时间内用量远超常值

| 水表名 | 采集时间 | 上次读数 | 当前读数 | 用量 |
|---------|----------------|----------|----------|--------|
| XXX 花圃+ | 2019/1/19 3:45 | 16982.5 | 16983.75 | 1.25 |
| XXX 花圃+ | 2019/1/19 4:00 | 16983.75 | 16985 | 1.25 |
| XXX 花圃+ | 2019/1/19 5:00 | 16985 | 17486 | 501 |
| XXX 花圃+ | 2019/1/19 5:15 | 17486 | 17611.25 | 125.25 |
| XXX 花圃+ | 2019/1/19 5:30 | 17611.25 | 17736.5 | 125.25 |

5.3.3 水表的表计编码异常值

观察水表的表计编码的编码规则，可以发现下一级编码的数字段是采用“上一级编码的数字段+两位序数”的形式构成。但“高配房+”所属的二级表计编码为“40105T”，但其三级表计编码为“4010401T”。因此，水表名为“高配房+”的三级表计编码存在错误。在此，将“高配房+”的三级表计编码修正为“4010501T”。

5.4 缺失值的统计

5.4.1 读数缺失

所有水表四个季度读数的数据采集均以 15 分钟为间隔，定义不同水表在相应采集时间下无采集数据的情况为存在缺失值。在数据总表的基础上，查找缺失值，汇总统计得到存在缺失值 136800 条，附件完整采集数据共 3086788 条。水表读数中缺失值仅表示在应采集时点未对水表数据进行采集，缺失值的存在不影响之后水表读数与用水量。因此，本文不对缺失值进行处理。

5.4.2 水表层级的记录缺失

校区水表层级关系的附件中，存在两个一级表计编码下数据的缺失。对此，本文结合其余水表口径与用水信息，对缺失一级水表口径进行拟合求解。具体处理及拟合步骤见问题二。

5.5 重复值的筛选

结合数据情况，定义“不同水表在对应采集时点下水表当前读数完全相同”的数据为重复值。参照上述定义，对附件数据进行筛选与匹配，查找结果为附件数据中不存在重复值。

5.6 “消防”数据的处理

消防不在层级列表中，且用量频次和总和皆少，因此不纳入下文模型的计算。

六、问题一的模型建立与求解

6.1 修正波士顿矩阵^[2]

波士顿矩阵，又称四象限法。它认为一般情况下市场引力与企业实力是决

定产品结构的基本因素。为探究各水表数据的变化规律，本文结合已有数据对波士顿矩阵进行修正。

1.对横、纵坐标的重新定义

在波士顿矩阵中，销售增长率为纵坐标，市场占有率为横坐标。结合题中各水表当前读数的变化特性，令增长段的增长率变化为横坐标、水表当前读数的变化趋势有无转折点为纵坐标。

2.对矩阵内容的重新定义

金牛类：水表当前读数的变化趋势呈现持续缓慢增长，整体趋势无转折；
明星类：水表当前读数的变化趋势呈现持续快速增长，整体趋势无转折；
问题类：水表当前读数的变化趋势由缓慢增长转为快速增长，整体变化趋势由明显转折。

6.2 各水表数据的变化规律

依据各水表的当前读数，绘制其随时间的变化曲线。结合上文对水表读数变化趋势所作的分类，逐个匹配所有水表的类别（完整匹配结果见附录）。

表 7 匹配结果矩阵

| 问题类 | 明星类 |
|-------------------|---------------------------|
| XXXX 馆、教育超市+、浴室…… | XXX 花圃+、XXXX 酒店、XXX 干训楼…… |
| 瘦狗类 | 金牛类 |
| 无 | XXX 教学大楼总表、XXX 校医院…… |

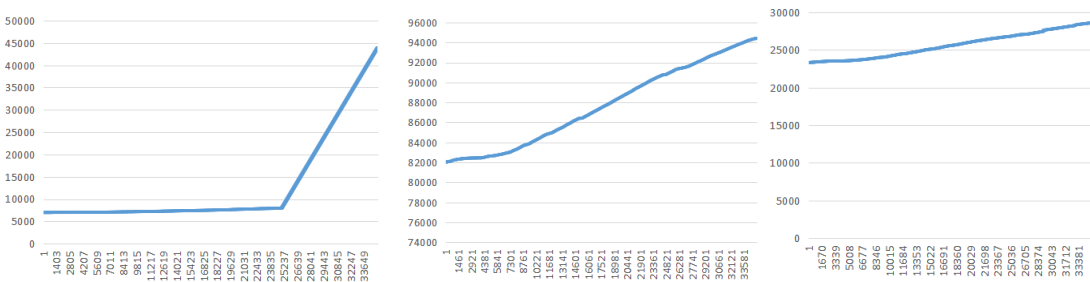


图 3 分别以 XXXS 馆、XXX 教学大楼总表、XXX 干训馆为代表

6.3 不同功能区的用水特征

将校园分为宿舍、教学楼、办公楼、食堂、科研和生活六个功能区，各个功能区均具有各自的功能与特点。在同一功能区内，各水表的用水数据应存在共有的特性。为探究不同功能区的用水特征，参照各水表名称以及水表层级关系，将各水表分别归入相应功能区。

| | | |
|---|--|---|
| 宿舍 宿舍、热泵 | 办公楼 东大门、后勤楼、传达室 | 食堂 食堂、锅炉房 |
| 教学楼 花圃、成教院、田径场、XX 楼、体育馆、干训楼、茶园、养殖队（馆）、养鱼组 | 科研 危险品仓库、纳米楼、航空航天、研究所、种子楼、试验站、高配房、鉴定中心 | 生活 医院、车队、活动室、XX 馆、酒店、污水处理、超市、理发店、游泳（水）池、平房、物业 |

图 4 功能区划分

由于各功能区下水表的水表层级不相等，直接相比时数据存在较大差距。因此，将相同功能区下水表层级关系相同的水表归为一类分析。进而，分析对应功能区的用水特征。

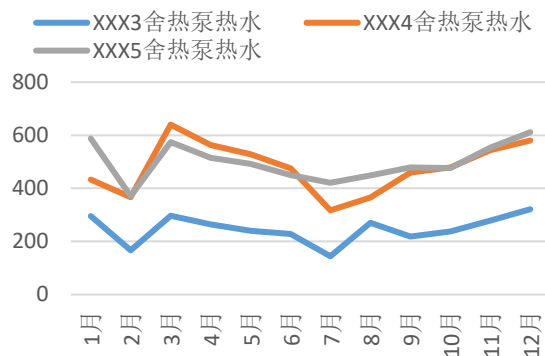


图 5 宿舍（四级）

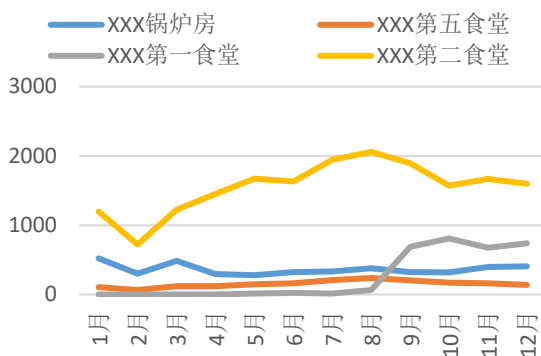


图 6 食堂（二级）

图 6 的数据来源为校园宿舍功能区内水表层级都为四级的水表用水量数据，采用数据透视表可以汇总得到各水表各月的用量。绘制折线图可以发现此类水表 2 月、7 月的用水数据明显下降，总体呈现波动趋势。联系题意与实际，合理推断 2 月、7 月用水数据下降的原因是寒暑假在校内减少。

图 7 的数据来源为校园食堂功能区内水表层级为二级的水表用水量数据，采用数据透视表汇总得到各水表的各月用量，观察发现各水表的用水数据存在较大差异。联系实际，合理推断用水量最大的第二食堂为学校的主食堂，第一食堂上半年无用水的原因可能是该食堂未开张。

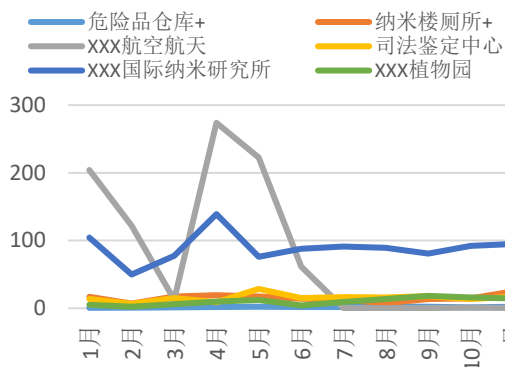


图 7 科研（二级）

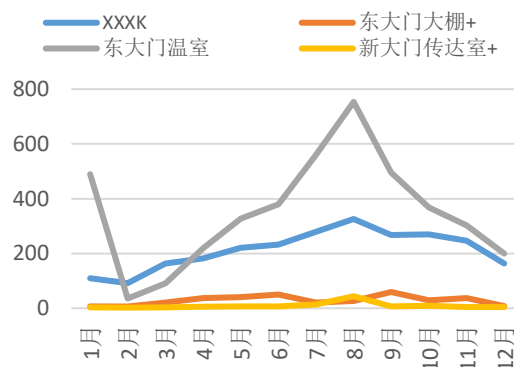


图 8 办公楼（二级）

采用数据透视表分别汇总校园内科研、办公楼功能区内二级水表的各月用水量数据，得到图 8、图 9。观察发现部分的水表用水量数据相差不大，仅个别水表用水量数据大。两个功能区内用水量最大的水表分别为航空航天、东大门温室，且它们全年各月的用水量变化较大。

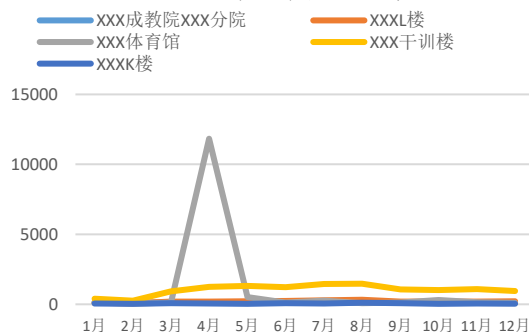


图 9 教学楼（二级）

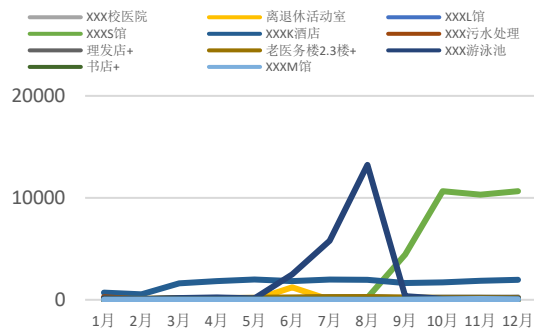


图 10 生活（二级）

采用数据透视表分别汇总校园内教学楼、生活功能区内二级水表的各月用水量数据，得到图 10、图 11。教学楼功能区内，XXX 体育馆 4 月的用水数据激增，其余月份 XXX 干训楼的用水数据较大，所有水表数据在 2 月均有下

降。生活功能区内，XXX 游泳池在 7 月、8 月的用水量激增，XXXS 馆在 10 月—12 月保持高用水量。

七、问题二的模型建立与求解

7.1 水表层级关系的定义

参照《供水系统漏损量组份的分析研究》^[3]，对该校园智能供水系统的水表层级进行分析：一级水管计量为一般指在前端水厂出水进入校园供水管网的供水总量；二级水管计量为从一级供水总量剩余量分配至各学院或区块的消费水量；三级水管计量为从二级剩余供水量分配至各教学楼、寝室楼等整栋消费水量；四级水管计量为从三级剩余供水量分配至末端户内计量水表。

以一级水表“64397 副表”（403X）为例分析具体流向关系。

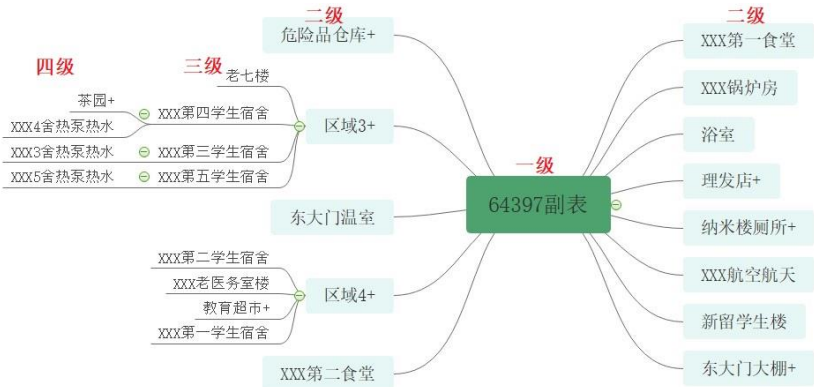


图 11 403X-64397 副表水表层级

从图 13 中可以看出，在所有端口都需水的时候，总水流从一级水表“64397 副表”流入，再流入“XXX 第一食堂”、“XXX 锅炉房”和“浴室”等二级水表。而水流经过“区域 3+”和“区域 4+”的二级水表，会分别流入“老七楼”等和“XXX 第二学生宿舍”等三级水表。最终，通过“xxx 第四学生宿舍”、“xxx 第四学生宿舍”和“xxx 第五学生宿舍”的三级水表，流入最底层的四级水表。

7.2 模型准备

(1) 数据匹配

为了对水表的层级关系进行分析，需要进行数据匹配。首先，通过 Python 将总表的数据按照“水表名”进行用量的汇总，得到年度的水表用量数据表；其次，匹配水表层级附件和年度的水表用量数据表，得到带有年用量的层级表。

表 8 以 416X 为例

| | | |
|---------------------------|----------------|----------------------|
| 一级编码及名称 416X 校医院南+ | 一级供水量 24145 | 二级汇总到一级消耗量 2027.3 |
| 二级编码及名称 41601X XXX 校医院 | 二级原始 2027.3 | 三级汇总到二级消耗量 413.8 |
| 三级编码及名称 4160101T 车队+ | 三级原始 413.8 | 三级汇总到二级消耗量 - |

(2) 补全带有年用量的层级表的关键缺失数据

观察带有年用量的层级表，发现缺失了两个一级指标的信息数据，分别为 402X 和 404X。首先，为方便检索，将其水表名定义为“无记录 1”和“无记录 2 补上”。其次，用下属水表的用水量之和补上其年用量。最后，根据其他水表的年用量和口径的拟合结果，确定缺失的口径数据。

其中，年用量和口径的拟合方程为 $y = 2.89 \times 10^{-13}x^3 - 7.72 \times 10^{-8}x^2 + 0.005376x + 11.2$ ，该方程拟合优度为 0.9551，拟合效果良好，图像如下。

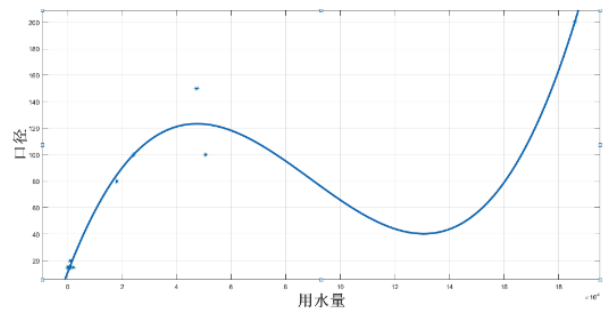


图 12 口径与用水量非线性关系拟合图

参照拟合结果与国家标准管道直径参数表（见附录），进而修正水管口径，修正结果如下表所示。

表 9 修正水管口径结果

| 一级编号 | 无进户流量供水量 | 预测水管口径 | 修正水管口径 |
|------|-----------|-------------|--------|
| 402X | 53433.308 | 122.1617067 | 120 |
| 404X | 20561.71 | 91.64315781 | 100 |

(3) 水表层级关系初想法

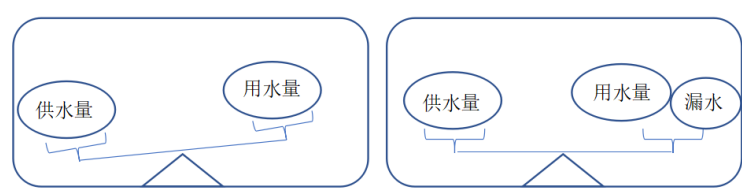


图 13 供产销关系

基于供产销关系，得到不等式：供水量 \geq 用水量。其中，只有在没有漏水与其他损耗的情况下，供水量才会真正等于用水量。因此，在不等式中加入“漏水量”来平衡一般不等式关系下的供销关系。但是，在本题所给的数据中不存在供水量数据。根据水表间的层级关系，上一级水表的用量，可以理解成下一级所有水表的供给量，即：上一级水表的用量 = 下一级所有水表的用量 + 漏水量

(4) 基于题目数据特征的水表层级关系修正

表 10 数据特征说明表

| 一级表计编码 | 二级表计编码 | 三级表计编码 | 四级表计编码 | 水表名 | 用量 |
|--------|--------|----------|--------|---------|----------|
| 416X | | | | 校医院南+ | 24145.00 |
| | 41601X | | | XXX 校医院 | 2027.30 |
| | | 4160101T | | 车队+ | 413.80 |

水流从一级表 416X 到二级表 41601X 过程中的漏水量高达 24145.00-2027.30=22117.7，占一级表流量的 91.60%；水流从二级表 41601X 到三级表 4160101T 过程中的漏水量高达 2027.30-413.80=1613.50，占二级表流量的 79.59%。这个供求关系明显存在不合理性，结合实际的水表名和应用情况，认为存在“进户流量”，即校医院南自身需要用水，不会都流入下级。因此，新的产销平衡公式为：上一级水表的用量=下一级水表的用量和+上一级的进户流量+漏水量。

7.3 水表层级关系模型

令一级水管供水量为A，二级水管供水量B、三级供水量C、四级水管进户量D。各级水管总进户流量 θ 、漏水量 δ 具体符号含义如下：

表 11 符号含义

| 主线 | | 干线 | |
|----|--------|---------------------|----------------------------------|
| A | 实际供水量 | $\theta_{(i,name)}$ | 第 <i>i</i> 级某 <i>name</i> 水管进户流量 |
| B | 二级供给水量 | $\delta_{(i,name)}$ | 第 <i>i</i> 级某 <i>name</i> 水管漏水量 |
| C | 三级供给水量 | δ | 总体漏水量 |
| D | 四级用水量 | θ | 水管总进户流量 |

按照水表层级得出以下关系：

$$s. t. \begin{cases} B \geq C + \sum_1^n \delta_{(3,name)} + \sum_1^m \delta_{(2,name)} + \sum_1^m \theta_{(2,name)} & m \in [1,53]; n \in [1,25] \\ C \geq D + \sum_1^n \delta_{(4,name)} + \sum_1^m \delta_{(3,name)} + \sum_1^m \theta_{(3,name)} & m \in [1,25]; n \in [1,5] \\ A \geq B + \sum_1^n \delta_{(2,name)} + \sum_1^m \delta_{(1,name)} + \sum_1^m \theta_{(1,name)} & m \in [1,13]; n \in [1,53] \\ \sum_1^n \delta_{(i,name)} = \delta & i = \{1,2,3\}; n \in [1,93] \\ \sum_1^n \theta_{(i,name)} = \theta & i = \{1,2,3\}; n \in [1,93] \end{cases} \quad (1)$$

结合模型需要测算的变量分析附件数据发现，难以精确计算出 δ 、 θ 。为了便于研究后续漏水情况，进一步改进模型，将 θ 赋值为零，即不考虑各级水管总进户流量。

7.4 水表层级关系模型的模型误差分析

由于带有年用量的层级表的数据样式比较原始，并不能直接代入水表层级关系模型，所以误差分析分为以下几步来实现。

第一步：观察水表层级数据，可分为两类：无下级数据和存在下级的数据。

第二步：针对无下级的数据，无需进行下级汇总，直接进行摘录处理。

第三步：针对存在下级的数据，需要将上次作为“产”用量，下级的汇总作为“销”用量，进行比对分析。

第四步：产销差应为进户流量和漏水量的和，因此“产”>“销”，为正常供水。而“产”<“销”为不正常供水，汇总统计不正常供水情况，即误差结果。

利用 python 软件按照上述步骤撰写代码进行求解，结果见表 13。

表 12 误差结果

| 表计编码 | 用户名 | 用水量 | 下级汇总 | 误差值 | 结果 | 误差率 |
|--------|-----------|---------|-----------|----------|-------|--------|
| 403X | 东大门大棚（副表） | 185989 | 201907.24 | 15918.24 | 不正常供水 | 7.88% |
| 405X | XXX 花圃 | 47341 | 92793.57 | 45452.57 | 不正常供水 | 48.98% |
| 40133X | 区域 1（西） | 15894.2 | 19866.73 | 3972.53 | 不正常供水 | 20.00% |

发生不良供水现象的可能原因：

- （1）读表异常，包括水表损坏、水表故障出现的读数错误等；
- （2）水表层级接错或者标错；
- （3）数据传输被调换；
- （4）管道混乱（某级下属管道水量可能来自非本上级供给）。

八、问题三的模型建立与求解

8.1 基于夜间漏水统计的输水管漏水模型

8.1.1 基于层级数据误差修正的模型准备

根据问题二，层级关系之间存在“不正常供水”现象，具体表现为某一级的用量数据小于其下级的用量之和。结合实际情况和修正的可操作性，本文将数据中存在的“不正常供水”的原因归咎于水表的读数误差。因此，针对不正常供水的水

表，通过对指标用量数据的上浮调整来修正读数误差，上浮量为下级用量和与该级用量之间的差值。其公式可以定义为：

$$\text{某一级的用量} = \begin{cases} \text{该级用量的数据, 正常供水} \\ \text{下级用量之和, 不正常供水} \end{cases} \quad (2)$$

选择包含总表数据，划分为有下级和无下级的数据块。有下级的数据块利用 python 软件的筛选和分组统计功能，进行层级误差的统计和是否正常供水的判定，并结合上述公式进行不正常供水的用量修正。无下级的数据块，不存在层级误差，确定为正常供水，无需修正。两块数据进行汇总就可以得到基于误差修正的总表数据，部分表格展示如下。

表 13 基于误差修正的部分数据展示

| 采集时间 | 水表名 | 用量 | 下级和上级与下级之和的差值 | 是否正常供水 | 修正的用量 | |
|---------------|----------------|------|---------------|--------|-------|------|
| 2019/1/1 3:15 | XXX 成教院 XXX 分院 | 0.02 | 0 | 0.02 | 正常供水 | 0.02 |
| 2019/1/1 5:15 | XXX 成教院 XXX 分院 | 0.02 | 0.02 | 0 | 正常供水 | 0.02 |
| 2019/1/1 8:00 | XXX 成教院 XXX 分院 | 0.02 | 0.05 | -0.03 | 不正常供水 | 0.05 |

8.2 基于夜间漏水统计的输水管漏水模型

从一天中不同的用水时刻出发，观察对比用量数据，不难发现夜间的用水量相较白天降低很多。这种数据特点也印证了，人们的一般生活规律。因此夜间最小流量虽然并不能代表真实的输水管漏水，但是在没有足够指标来计算漏水的情况下，用夜间最小流量的产生和持续来近似表示输水管漏水是一个比较合理的估算方法。

8.2.1 夜间数据的判定和获取

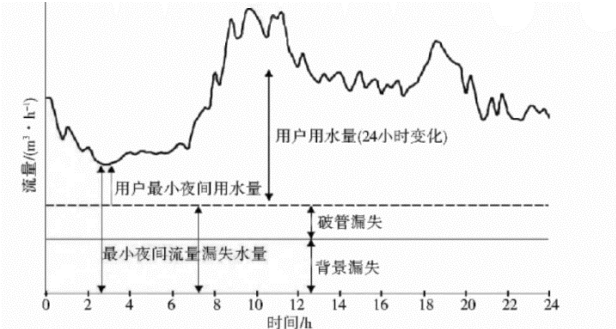


图 14 每天水量变化图^[4]

结合上图，夜间用水判定的时间段应选择 2：00 至 5：00。因此，对“采集时间”进行筛选，得到所有水表夜间 2：00 至 5：00 的用水数据。

8.1.2 无重复读数数据的判定和获取

(1) 读表重复性

各水表读数均反映各处的流量，依据水表层级关系，下属水表的用量均经过对应一级水表。发生在某低级水表的水表数据，必先发生在其上所有级别的水表上。比如，当四级水表下使用 0.01 用量的水，水须从一级表进入，且流经对应的二级表和三级表，因此此时一级二级三级水表都会重复四级水表的读数增量。

(2) 层级逆推法（以 403X 为例）

由于存在读表重复性，因此依据水表的层级关系，采用层级逆推法得到无重复读数的用量数据。下文将以 403X 层级为例，撰写具体步骤说明层级逆推法。

Step1：得到“无下级水表”，及其用量。

无下级水表是指不向下产生分支的水表，其在读表重复性问题的处理中有非常重要的意义（下图中通过对无下级水表加蓝框的形式，进行区别显示）。

无下级水表，不会因为产生下级用水而重复读数，所以无下级水表的用量数据直接有效。

$$\text{无重复读数用量}_{\text{无下级水表}} = \text{水表用量} \tag{3}$$

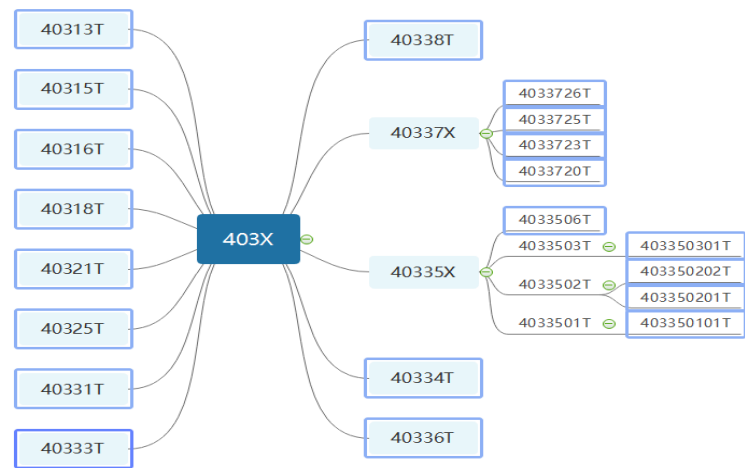


图 15 无下级水表的示意图

Step2：分析“有下级水表”的层级结构，并得到其无重复用量。

分析三级实际用量（以 4033502T 为例）：4033502T 的无重复读数用水量=4033502T 的用量-（403350202T 的用量+403350201T 的用量）

分析二级实际用量（以 40337X 为例）：40337X 的无重复读数用量=40337X 的读表用量-（4033726T 的用量+4033725T 的用量+4033723T 的用量+4033720T 的用量）

分析一级实际用量（以 403X 为例）：403X 的无重复读数用量=403X 的读表用量-上述所有二级标题的用量总和。

因此，“有下级水表”的无重复读数用量的计算公式为：

$$\text{无重复读数用量}_{\text{有下级水表}} = \text{水表用量} - \text{对应下级水表的用量和} \tag{4}$$

备注：上级与下级用量和的差=入户流量和漏水量之和，但考虑到本题的数据实际情况，无法分割右侧两者。为了简化处理，在层级逆推法的计算中，漏水量假设为 0。

Step3：汇总所有水表的无重复读数用量。

根据层级逆推法可得无重复读数的新总表。由于无重复读数的新总表和原总表的差异主要出现在有下级部分，因此展示部分有下级的无重复读数用量。

表 14 无重复读数用量数据的部分展示

| 采集时间 | 水表名 | 修正的用量 | 下级和 | 是否为正常供水 | 无重复读数的用量 |
|----------------|---------|-------|------|---------|----------|
| 2019/1/3 16:15 | XXX 花圃+ | 0.98 | 0.98 | 非正常供水 | 0 |
| 2019/1/3 17:45 | XXX 花圃+ | 2.75 | 0.9 | 正常供水 | 1.85 |
| 2019/1/3 18:00 | XXX 花圃+ | 2.75 | 0.99 | 正常供水 | 1.76 |

8.1.3 夜间漏水的判定和数据获取

（1）基于所有水表 2：00 至 5：00 的无重复读数用量的数据，以“用量>0”为筛选标准，得到夜间用水的数据表。

（2）基于夜间用水数据表，以“日”为时间计量维度，统计各个水表的夜间用水频次。

（3）夜间漏水因休息不容易被发现，（假设夜间漏水时，检修人员无法及时检修）漏水情况一旦成立，就意味着在 2：00 至 5：00 持续失水，频次达到 11 次。因此，以“频次=11”来判定夜间漏水的情况。

综上，夜间漏水表部分结果展示如下：

表 15 夜间漏水表的部分展示

| 水表名 | 夜间三小时内的漏水频次 | 漏水时间 |
|------------|-------------|-----------|
| 64397 副表 | 11 | 2019/1/13 |
| 64397 副表 | 11 | 2019/2/13 |
| ... | ... | ... |
| XXX3 舍热泵热水 | 11 | 2019/1/8 |
| ... | ... | ... |

8.2.4 夜间漏水情况的结果分析

下表展示了部分夜间漏水的数据，完整数据结果见数据附录。

表 16 部分夜间漏水数据的展示

| 水表名 | 漏水频次 | 最早发现时间 | 水表名 | 漏水频次 | 最早发现时间 |
|----------|------|-----------|------------|------|----------|
| 64397 副表 | 19 | 2019/1/13 | XXXT 馆后平房 | 82 | 2019/1/8 |
| XXXS 宾馆 | 167 | 2019/1/8 | XXX 第五食堂 | 33 | 2019/1/8 |
| XXXS 馆 | 127 | 2019/1/8 | XXX 第五学生宿舍 | 36 | 2019/1/2 |

对夜间漏水的数据总表进行统计，可以得到以下结论：出现漏水现象的水表有 74 个，较多水表或其所对应段的水管可能存在漏水现象，需要数据预警以后进行检修和维护；漏水水表的平均漏水频次高达 67 次，意味着有些水表或水管存在老化失修的现象。

8.3 基于爆漏可能性指数统计的爆漏模型

8.3.1 爆漏可能性指数的定义

爆漏指的是因水管破裂导致的大量水流失。由于本题中只提供了读数和用量指标，因此在该爆漏模型中，本文自定义了爆漏可能性指数来进行爆漏问题的诠释。爆漏可能性指数的定义可分为以下几步：

第一步：每个水表的相关参数和使用情况皆有不同，因此本文将总表依据“水表名”划分为 91 个水表数据，分别针对不同水表进行爆管可能性的统计。

第二步：统计某一水表时刻上的用量均值。均值代表该水表用量的一般水平，而爆管指较突出的激增用量，因此这个均值可作为爆漏可能的下限。

第三步：针对这一水表，筛选出超过爆漏可能下限的时刻数据，并对超过部分数据计算“离差绝对和”（定义方式类似离差绝对和，不同时刻数据减去均值下限，再求和）。该离差绝对和可代表超过下限的部分的整体高度水平，离差绝对和越大，则表示越可能爆漏，因此将其定义为爆漏可能性指数。

第四步：利用数据透视表，汇总得到各水表每日的爆管可能性指数表。

根据上述步骤，得到的爆管可能性指数表的部分展示如下。

表 17 爆漏可能性指数的部分展示

| 水表名 | 时间 | 爆管可能性指数 |
|----------|----------|-------------|
| 64397 副表 | 2019/1/3 | 19.80645161 |
| 64397 副表 | 2019/1/4 | 55.72916668 |
| ... | ... | ... |
| XXX 东大楼 | 2019/6/3 | 0.815625 |
| ... | ... | ... |

7.3.2 基于爆漏可能性指数的聚类模型

由于爆漏数据属于异常大的数据类型，而孤立森林聚类对异常值有比较好

的识别效果。因此，本文基于每个水表的日度数据，利用 python 软件，以爆漏可能性指数为指标，选择该聚类模型，分别对 91 个水表的数据进行分类分析。

在孤立森林聚类算法中，python 所运行的类别结果有两种：1 和-1。其中，1 代表正常值，-1 代表异常值。以“XXX 游泳池”的数据为例，得到以下分类结果，仅展示前面部分的表格。

表 18 “XXX 游泳池”聚类结果的部分展示

| 异常分类标签 | 时间 | 爆漏可能性指数 | 异常分类标签 | 时间 | 爆漏可能性指数 |
|--------|-----------|---------|--------|-----------|---------|
| -1 | 2019/2/25 | 10.68 | -1 | 2019/4/28 | 18.25 |
| -1 | 2019/1/21 | 10.80 | -1 | 2019/6/20 | 24.06 |
| -1 | 2019/5/23 | 11.00 | -1 | 2019/7/12 | 29.40 |

从 91 张水表的整体情况来看，有些水表聚类效果较好，有些水表聚类效果较差。具体较差的程度体现在爆漏可能性指数的大小上，即有些较小的指数也被划分到异常分类中，存在一定的不合理性。

九、问题四的模型建立与求解

9.1 暗漏计算模型

9.1.1 暗漏定义说明

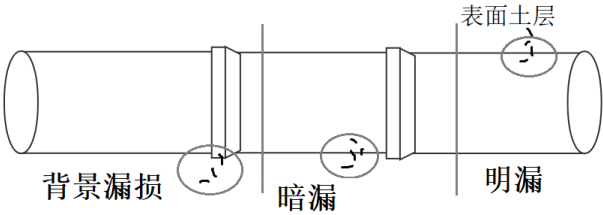


图 16 漏损类型

在《CJJ92-2018 城镇供水管网漏损控制及评定标准》^[5]中，国家明确定义“暗漏是在地面以下检测到的管网漏点的漏失水量”。《供水管网暗漏漏点流量的估算方法》^[6]说明了暗漏的计量方法、检测方法等。但是结合本题实际的数据情况，无法得到管道压力、漏损系数、管材等指标，也无法用这类方法去实践探测。因此，本文按照暗漏的含义，利用仅有的用量数据，进行暗漏计算方法的自定义。

结合问题三中采用夜间流量计算的方法，来挖掘数据中存在输水管漏水的信号且上文将漏水频次=11（次）视为漏水。观察和分析数据结果，发现夜间漏水的数据在时间线上存在两种情况——连续和不连续。因此，根据该特点区分明漏和暗漏。表 19 分别给出了一个时间线上连续和不连续的例子。

表 19 时间线上连续和不连续的例子

| | 时间 | 夜间漏水频次 | 漏水时间差 | 连续性 | 类型 |
|----------|------------|--------|-------|-----|----|
| 64397 副表 | 2019/2/14 | 11 | 3 | 非连续 | 明漏 |
| | 2019/2/17 | 11 | | | |
| XXXL 楼 | 2019/11/13 | 11 | 1 | 连续 | 暗漏 |
| | 2019/11/14 | 11 | | | |

从表 19 中可以看出：

（1）“64397 副表”的 2019 年 2 月 14 日有夜漏现象，2019 年 2 月 15 日没有夜漏现象，即说明白天漏水被发现检修，这种情况理解为是地面上易发现的**明漏**。因此，夜漏计算中的明漏的判断公式为：

$$L \text{ day}_{i+1} - L \text{ day}_i > 1 \quad (5)$$

其中： $L \text{ day}_i$ ——第*i*天漏水。

(2) “XXXL 楼”的 2019 年 11 月 13 日和 2019 年 11 月 14 日皆有夜漏现象，即 2019 年 11 月 13 日的陵城检测到的了漏水，但是 2019 年 11 月 13 日的白天未被发现检修，才出现 2019 年 11 月 14 日夜间的继续漏水现象。这种情况理解为不是明漏，而是在地下管道难以发现的暗漏。因此，夜漏计算中的暗漏的判断公式为：

$$Y \text{ day}_{i+1} - Y \text{ day}_i = 1 \quad (6)$$

其中： $Y \text{ day}_i$ ——第*i*天夜间漏水。

9.1.2 暗漏计算模型

暗漏计算的思路主要分为以下几步：

第一步：根据问题三，得到夜间漏水的数据表。

第二步：按照“水表名”和“日期”进行双下表排序，并得到各水表漏水数据的上下时间差。

第三步：根据 4.1.1 中的暗漏公式，从时间线是否连续的角度判定漏水类型，如果连续则为暗漏，否则为明漏。

第四步：筛选得到所有水表的暗漏数据。

第五步：基于暗漏数据，由于漏水可以叠加所有正常用水产生，所以将 15 分钟的暗漏漏水量近似估计为夜间 2: 00 至 5: 00 每 15 分钟产生的最小用量，其公式为：

$$AL_{(name,t)}^{15} \approx \text{Min}(AL_{(name,t)}^{180}) \quad (7)$$

其中： $AL_{(name,t)}^{15}$ ——第*name*水管第*t*天 15 分钟的暗漏漏水量；

$\text{Min}(AL_{(name,t)}^{180})$ ——第*name*水管第*t*天（夜间 2: 00 至 5: 00 每 15 分钟）180 分钟的暗漏漏水量的最小值。

第六步：暗漏当天未被检修，全天会产生暗漏用量（24 小时共有 96 个 15 分钟，公式如下），汇总可得到所检测的暗漏水表的每日暗漏总量。对各个水表进行汇总，即可得到暗漏总量。

$$WAL_{(name,t)} = AL_{(name,t)}^{15} \times 96 \quad (8)$$

其中： $AL_{(name,t)}^{15}$ ——第*name*水管第*t*天 15 分钟的暗漏漏水量；

$WAL_{(name,t)}$ ——每日第*name*水管第*t*天的暗漏总量。

9.2 暗漏计算结果

9.2.1 暗漏结果展示和分析

通过数据透视表，计量不同水表名的暗漏频次和最早一次的暗漏时间，完整结果见附录。

表 20 暗漏结果部分展示

| 水表名 | 暗漏频次 | 最早一次漏水时间 |
|------------|-------|-----------|
| 64397 副表 | 6 | 2019/4/10 |
| XXX3 舍热泵热水 | 16 | 2019/1/8 |
| XXX4 舍热泵热水 | 21 | 2019/2/9 |
| XXX5 舍热泵热水 | 192 | 2019/1/1 |
| XXX8 舍热泵 | 23 | 2019/1/8 |
| | | |

9.2.2 暗漏漏水量的近似估算结果

表 21 部分暗漏漏水量的近似估算结果

| 水表名 | 漏水时间 | 估计漏水量 |
|------------|------------|-----------|
| 64397 副表 | 2019/4/10 | 29.958816 |
| 64397 副表 | 2019/4/11 | 40.518816 |
| 64397 副表 | 2019/4/12 | 33.798912 |
| ... | ... | ... |
| XXX3 舍热泵热水 | 2019/11/21 | 8.64 |
| XXX3 舍热泵热水 | 2019/11/22 | 8.64 |
| ... | ... | ... |
| 暗漏总量 | | 134936.30 |

9.3 明漏的计算结果

利用夜间 2:00 至 5:00 的数据计算而得的漏水情况中包括明漏。其处理方式与暗漏基本相同,不同的地方有两点。

(1) 暗漏是时间上的连续漏水,时间差为 1;明漏是时间上的非连续漏水,时间差大于 1。

(2) 每日暗漏总量是将 15 分钟的暗漏量乘以 96,表达全天漏水的意思。但是明漏在夜间漏水后,会在当天被检修。假设维修工人 9 点正常上班,10 点可以完成检修,则每日的明漏总量是 15 分钟的明漏量与 32(共八小时)的乘积,公式如下:

$$WAL_{(name,t)} = AL_{(name,t)}^{15} \times 32 \quad (9)$$

9.3.1 明漏的结果展示与分析

通过数据透视表,计量不同水表名的明漏频次和最早一次的明漏时间,完整结果展示见附录。

表 22 明漏的部分结果展示

| 水表名 | 最小值项:采集时间 2 | 计数项:采集时间 | 最早漏水时间 |
|------------|-------------|----------|-----------|
| 64397 副表 | 14 | 13 | 2019/1/13 |
| XXX3 舍热泵热水 | 40 | 15 | 2019/2/8 |
| XXX4 舍热泵热水 | 50 | 24 | 2019/2/18 |
| XXX5 舍热泵热水 | 19 | 24 | 2019/1/18 |
| XXX8 舍热泵 | 17 | 34 | 2019/1/16 |
| | | | |

9.3.2 明漏漏水量的近似估算结果

表 23 部分明漏漏水量的近似估算结果

| 水表名 | 估计漏水量 | 漏水时间 |
|------------|-------|-----------|
| 64397 副表 | 2.88 | 2019/1/13 |
| 64397 副表 | 2.88 | 2019/2/13 |
| 64397 副表 | 6.72 | 2019/2/16 |
| ... | ... | ... |
| XXX3 舍热泵热水 | 5.76 | 2019/2/8 |
| ... | ... | ... |
| 明漏总量 | | 11230.15 |

十、问题五的模型建立与求解

10.1 管网维修总成本模型

为保障校园供水系统的正常运行，学校投入了一定的人力、物力和财力。因此，不管是否对管网漏损进行维修，成本中均存在一定的管理费用。管网漏损的维修成本包括维修人员维修管网得到的提成和购买材料的成本，由于用水量大小对管网的损耗程度不同，因此管网漏损的概率随着用水量的增加而增大。

若不对管网漏损进行维修，则不产生维修成本，但漏损情况未得到改善，即漏损量随着用量的增加而增加，从而引起漏损水费的增加。若对管网漏损进行维修，则产生维修成本，同时漏损情况得到改善，从而漏损水费得到减少。

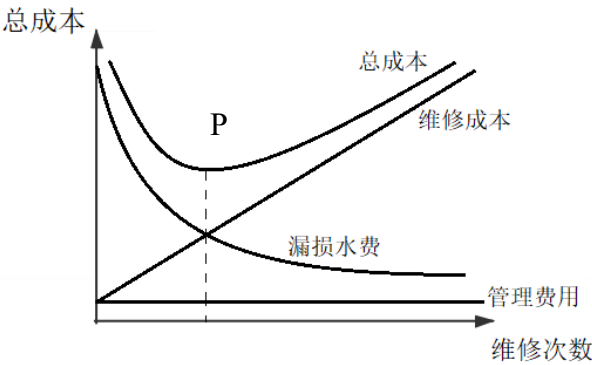


图 17 管网维修成本模型

为探究管网漏损维修成本与漏损水费间的关系，引入现金持有量成本模型。考虑管网漏损维修发生的收支，求得管网漏损维修的成本曲线，进而得到管网维修的最优决策方案。其中：

总成本 = 维修成本 + 漏损水费 + 管理费用 。

结合模型原理与图像，可知漏损水费与维修成本图像相交时总成本最低。

10.2 维修成本

1.更换水管维修费

以重庆市物价局修正的收费标准^[7]为例，结合上述管网维修成本模型和日最佳用水量动态控制模型，分析该校园管道漏损的最佳维修决策方案。参照 2001 年不同口径水管新装或改进的收费，结合年 CPI 变化率推导得到 2020 年对应的收费（具体见下表）。

表 24 维修收费参考值

| 口径 | 2001 新装或改进 | 2020 新装或改进 | 口径 | 2001 新装或改进 | 2020 新装或改进 |
|----|------------|------------|-----|------------|------------|
| 15 | 80 | 98 | 80 | 1400 | 1707 |
| 20 | 140 | 171 | 100 | 2000 | 2438 |
| 25 | 220 | 268 | 150 | 4000 | 4876 |
| 40 | 400 | 488 | 200 | 7500 | 9142 |
| 50 | 600 | 731 | - | - | - |

2.其他费用

查勘费：水表口径为 DN15—DN20 的每只表收取 122 元；水表口径为 DN25—DN100 的每只表收取 366 元；

人工费：水表口径在 DN100 以下的每工作日按 55 元计收；水表口径在 DN100 以上（含 DN100）的按国家当时颁布的市政工程造价定额标准计收；

材料费：自来水安装和维修所需材料价格按供水企业实际进价或均价加 15% 综合差率作价收取；等等。

10.3 日最佳用水量动态控制模型

结合目标现金持有量的确定模型，采纳该模型的计算思路，将指标值进行替换，得到日最佳用水量的控制模型。具体公式为：

$$Q = \sqrt[3]{\frac{3F\sigma^2}{4K}} + L \tag{10}$$

其中：Q——日最佳用水量；
F——每月平均水价；
 σ^2 ——每日用水量变化的方差；
L——假期最低用水量；
K——CPI 变化率。

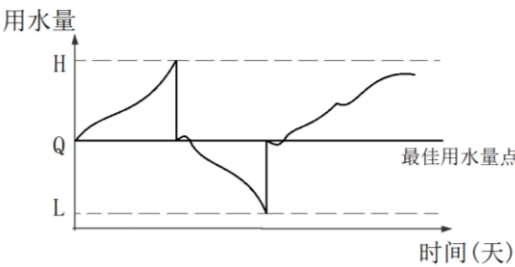


图 18 日最佳用水量动态控制模型

10.4 最优维修决策方案

结合各水表用水数据，通过日最佳用水量动态控制模型求出该水表的日最佳用水量。若最佳用水量大于实际用水量，则不进行管道维修。

若最佳用水量小于实际用水量，则计算水表当天漏损量与最佳用水量的比值。若比值大于 10%，则参照漏损原因判断是否进行管道维修。将漏损量代入管网维修总成本模型得到维修成本，即最优维修决策方案所需的成本。即：

$$\text{维修总成本} = 2 \times \text{漏水量}_{(\text{距离上一次检修})} \times \text{水价}_{\text{月平均水平}} + \text{管理费用}。$$

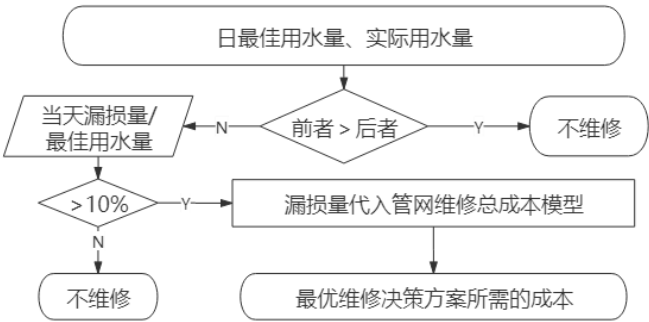


图 19 最优维修决策方案流程

十一、模型评价与拓展

11.1 模型评价

11.1.1 模型优点

- (1) 充分利用软件进行数据处理和挖掘，减少了手工工作量，实现高度自动化。
- (2) 模型可操作性强，在可用数据匮乏背景下，充分量化数据图像的趋势特征，反馈数据内涵，对比现实生活，得出合乎情理结论。
- (3) 模型基于传统又突破弊端，提出了最质朴而又较为准确的搜索类、预警类算法思路及其结果，有较大的参考价值。

11.1.2 模型缺点

全文多个模型是近似算法，由于题目数据资源匮乏、挖掘深度有限。例如进户流量和漏水量数据无法分离和精确计算。

11.2 模型拓展

11.2.1 层级误差和漏水信息的挖掘模型，可拓展构建学校漏水的实时警报系统

(1) 根据层级误差模型，构造读数异常的警报系统。当检测到不正常供水的情况，即对系统使用方进行异常读数数据的位置提示和警告。

(2) 根据暗漏数据模型，构造暗漏警示的警报系统。当检测到连续的夜间漏水后，即对系统使用方进行暗漏定位的提示和警告。

(3) 根据明漏数据模型，不仅可以提示明漏的位置，而且当出现高频次明漏水表时，可对系统使用方进行老化管道和水表的检修提示。预警系统图见附录。

11.2.2 最优维修方案的策划思路，可拓展到其他降本增效的经营决策中。

在资源分配最优化的背景下，运用第五题计算校园最佳维修经费预算的思路，可帮助制定降本增效的经营决策，可用以下案例演示。

假设该校面勘查完成后决定更替二级的N15、N20、N25、N100五个口径的W个水管，可供替换的原材料有M个。假设上述模型测算出修N次最佳所需预算P元；人工费、勘察费、材料费均按上文收费标准进行计价。且人工费、勘察费、材料费和水费的各支出占比15%、20%、15%。可以建立资源分配模型。设N15、N20、N25、N100损坏个数为 x_1 、 x_2 、 x_3 、 x_4 、 x_5 。

$$\begin{aligned} \max &= \sum_{i=1}^5 X_i & (11) \\ s.t. & \begin{cases} \sum_{i=1}^5 V_i * X_i = 20\%P \\ \sum_{i=1}^3 55 * X_i + 15\%(\Delta T_5 * X_5) = 15\%P \\ \sum_{i=1}^5 G_i * X_i = 15\%P \\ M \geq \sum_{i=1}^5 X_i \geq 1 \end{cases} \end{aligned}$$

在保证至少修一个二级口径水管且原材料数量有限的情况下使替换的个数尽可能多，才能保证漏水得到有效治理的决策。当然约束条件可以复杂得多。可用于后续研究。

11.2.3 孤立森林聚类方法，可拓展应用到多指标的系统聚类中。在比较完善的供水系统中，能够获得较多的用量外的指标，比如压力、阻力系数、实际供水量等等，这这类系统中孤立森林聚类方法在一定程度上能够有效提高漏水判断的准确性。

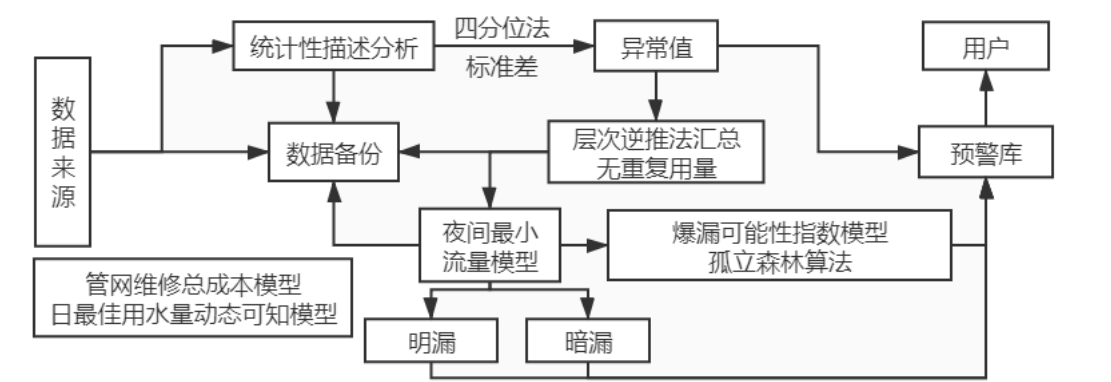
十二、参考文献

- [1]杨锡运,刘玉奇,李建林.基于四分位法的含储能光伏电站可靠性置信区间计算方法[J].电工技术学报,2017,32(15):136-144.
- [2]徐丛春,宋维玲,李双建.基于波士顿矩阵的广东省海洋产业竞争力评价研究[J].特区经济,2011(02):35-37.
- [3] 陈鹏. 供水系统漏损量组份的分析研究[D].北京工业大学,2016.
- [4]郝志萍,侯煜堃,崔昌.计量小区(DMA)夜间最小流量解析方法探讨与案例研究[J].南水北调与水利科技,2013,11(04):124-127+195.
- [5]<https://wenku.baidu.com/view/cb34f25eab8271fe910ef12d2af90242a895ab24.html>
- [6]翟参. 供水管道小孔漏损分析和数值模拟研究[D].郑州大学,2019.
- [7]http://fzggw.cq.gov.cn/zwxw_167/bmdt/202002/t20200212_5205042.html

十三、附录

13.1 结果附录

13.1.0 模型拓展



附图 1 预警系统图

13.1.1 问题一

附表 1 波斯顿矩阵数据总表

| 金牛类 | 问题类 | 明星类 | 瘦狗类 |
|---|---|---|-----|
| XXX 教学大楼总表、 XXX 国际纳米、 留学生楼（新）、 XXX 校医院、XXX 种子楼 XXX 成教院 XXX 分院、 XXX 田径场厕所、 XXXL 馆、XXX8 舍热泵、 XXXXK 楼、XXX 第五食堂、 XXX 污水处理、 东大门大棚+、老七楼、 XXX5 舍热泵热水、 XXX3 舍热泵热水、 XXX4 舍热泵热水、 危险品仓库+、 东大门温室、XXX 体育馆、 XXX 中心大楼泵房、 XXX4 舍热泵热水、 XXX 教学大楼总表、 XXX 锅炉房、东大门温室、 XXX 科学楼、XXX 图 | XXXXS 馆、 XXX 第一食堂、 XXX 游泳池、 教育超市+、 浴室 | XXX 花园+、区域 4+、 XXXXS 宾馆、XXXXK 酒店、 XXX 干训楼、64397 副表、 区域 3+、XXX 第二食堂、 养殖队 6721 副表+、 XXX 第五学生宿舍、 XXX 第四学生宿舍、 校医院南+、区域 2、 区域 1（西）、 XXX 第一学生宿舍、 留学生楼（新）、 XXX 第三学生宿舍、 XXX 西大楼、XXX 老五楼、 XXX 第八学生宿舍、 XXX 第七学生宿舍、 XXX 第九学生宿舍、 XXXX5 舍热泵热水、 XXX 第七学生宿舍 | 无 |

| | | | |
|--|--|--|--|
| 书馆、 XXXK、XXXL 楼、 老医务楼 2.3 楼+、 XXXT 馆后平房、 农业试验站大棚+、 XXXL 馆、危险品仓库 +、 XXX 大楼厕所东、 养殖馆+、高配房+、 XXX 大楼厕所东、 | | | |
|--|--|--|--|

13.1.2 (国家标准管道直径参数表)

一般家用水管规格有：DN15（4 分管）、DN20（6 分管）、DN25（1 寸管）、DN32（1 寸 2 管）、DN40（1 寸半管）、DN50（2 寸管）、DN65（2 寸半管）、DN80（3 寸管）、DN100（4 寸管）、DN125（5 寸管）、DN150（6 寸管）、DN200（8 寸管）、DN250（10 寸管）等。

13.1.2 问题三

附表 2 夜间漏水数据

| 水表名 | 漏水频次 | 最小值项:采集时间 2 | 最早一次的发现时间 |
|----------------|------|-------------|-----------|
| 64397 副表 | 19 | 14 | 2019/1/13 |
| XXX3 舍热泵热水 | 31 | 9 | 2019/1/8 |
| XXX4 舍热泵热水 | 45 | 41 | 2019/2/9 |
| XXX5 舍热泵热水 | 216 | 2 | 2019/1/1 |
| XXX8 舍热泵 | 57 | 9 | 2019/1/8 |
| XXXX | 43 | 9 | 2019/1/8 |
| XXXX 酒店 | 121 | 8 | 2019/1/7 |
| XXXX 楼 | 60 | 9 | 2019/1/8 |
| XXXL 馆 | 50 | 9 | 2019/1/8 |
| XXXL 楼 | 59 | 9 | 2019/1/8 |
| XXXM 馆 | 12 | 313 | 2019/11/8 |
| XXXS 宾馆 | 167 | 9 | 2019/1/8 |
| XXXS 馆 | 127 | 9 | 2019/1/8 |
| XXXT 馆后平房 | 82 | 9 | 2019/1/8 |
| XXX 成教院 XXX 分院 | 23 | 63 | 2019/3/3 |
| XXX 大楼厕所东 | 53 | 9 | 2019/1/8 |
| XXX 大楼厕所西 | 100 | 8 | 2019/1/7 |
| XXX 第八学生宿舍 | 30 | 8 | 2019/1/7 |
| XXX 第二食堂 | 85 | 3 | 2019/1/2 |
| XXX 第二学生宿舍 | 156 | 3 | 2019/1/2 |
| XXX 第九学生宿舍 | 48 | 8 | 2019/1/7 |
| XXX 第七学生宿舍 | 32 | 9 | 2019/1/8 |
| XXX 第三学生宿舍 | 51 | 9 | 2019/1/8 |
| XXX 第四学生宿舍 | 27 | 9 | 2019/1/8 |

| | | | |
|-------------|-----|-----|-----------|
| XXX 第五食堂 | 33 | 9 | 2019/1/8 |
| XXX 第五学生宿舍 | 36 | 3 | 2019/1/2 |
| XXX 第一食堂 | 25 | 152 | 2019/5/31 |
| XXX 第一学生宿舍 | 314 | 2 | 2019/1/1 |
| XXX 东大楼 | 49 | 9 | 2019/1/8 |
| XXX 毒物研究所 | 7 | 220 | 2019/8/7 |
| XXX 干训楼 | 70 | 25 | 2019/1/24 |
| XXX 锅炉房 | 56 | 7 | 2019/1/6 |
| XXX 国际纳米研究所 | 41 | 9 | 2019/1/8 |
| XXX 航空航天 | 9 | 9 | 2019/1/8 |
| XXX 后勤楼 | 31 | 62 | 2019/3/2 |
| XXX 教学大楼总表 | 65 | 2 | 2019/1/1 |
| XXX 科学楼 | 282 | 4 | 2019/1/3 |
| XXX 老六楼 | 52 | 107 | 2019/4/16 |
| XXX 老五楼 | 40 | 9 | 2019/1/8 |
| XXX 老医务室楼 | 60 | 9 | 2019/1/8 |
| XXX 体育馆 | 36 | 8 | 2019/1/7 |
| XXX 田径场厕所 | 42 | 9 | 2019/1/8 |
| XXX 图书馆 | 72 | 9 | 2019/1/8 |
| XXX 污水处理 | 40 | 16 | 2019/1/15 |
| XXX 西大楼 | 49 | 9 | 2019/1/8 |
| XXX 校医院 | 44 | 61 | 2019/3/1 |
| XXX 游泳池 | 89 | 9 | 2019/1/8 |
| XXX 中心大楼泵房 | 23 | 9 | 2019/1/8 |
| XXX 种子楼 | 39 | 8 | 2019/1/7 |
| 茶园+ | 8 | 82 | 2019/3/22 |
| 东大门传达室+ | 128 | 2 | 2019/1/1 |
| 东大门大棚+ | 2 | 244 | 2019/8/31 |
| 东大门温室 | 214 | 2 | 2019/1/1 |
| 高配房+ | 95 | 39 | 2019/2/7 |
| 教育超市+ | 25 | 35 | 2019/2/3 |
| 老七楼 | 41 | 8 | 2019/1/7 |
| 老医务楼 2.3 楼+ | 1 | 219 | 2019/8/6 |
| 离退休活动室 | 7 | 178 | 2019/6/26 |
| 留学生楼（新） | 353 | 3 | 2019/1/2 |
| 农业试验站大棚+ | 120 | 31 | 2019/1/30 |
| 区域 1（西） | 33 | 39 | 2019/2/7 |
| 区域 2 | 51 | 9 | 2019/1/8 |
| 区域 3+ | 8 | 36 | 2019/2/4 |
| 区域 4+ | 237 | 6 | 2019/1/5 |
| 司法鉴定中心 | 27 | 123 | 2019/5/2 |
| 校管中心种子楼东+ | 12 | 62 | 2019/3/2 |

| | | | |
|--------------|-----|-----|------------|
| 校医院南+ | 25 | 2 | 2019/1/1 |
| 新留学生楼 | 1 | 337 | 2019/12/2 |
| 养鱼组厕所+ | 37 | 39 | 2019/2/7 |
| 养鱼组临工宿舍+ | 20 | 123 | 2019/5/2 |
| 养殖队 6721 副表+ | 121 | 19 | 2019/1/18 |
| 养殖馆+ | 1 | 348 | 2019/12/13 |
| 养殖馆附房保卫处宿舍+ | 4 | 10 | 2019/1/9 |
| 浴室 | 70 | 9 | 2019/1/8 |

附表 3 漏水数据分析表

| 水表名 | 漏水 频次 | 最早一次的发 现时间 | 水表名 | 漏水 频次 | 最早一次的发 现时间 |
|----------------|----------|---------------|----------------|----------|---------------|
| 64397 副表 | 19 | 43478 | XXX 老六楼 | 52 | 43571 |
| XXX3 舍热泵 热水 | 31 | 43473 | XXX 老五楼 | 40 | 43473 |
| XXX4 舍热泵 热水 | 45 | 43505 | XXX 老医务室 楼 | 60 | 43473 |
| XXX5 舍热泵 热水 | 216 | 43466 | XXX 体育馆 | 36 | 43472 |
| XXX8 舍热泵 | 57 | 43473 | XXX 田径场厕 所 | 42 | 43473 |
| XXXX | 43 | 43473 | XXX 图书馆 | 72 | 43473 |
| XXXX 酒店 | 121 | 43472 | XXX 污水处理 | 40 | 43480 |
| XXXX 楼 | 60 | 43473 | XXX 西大楼 | 49 | 43473 |
| XXXL 馆 | 50 | 43473 | XXX 校医院 | 44 | 43525 |
| XXXL 楼 | 59 | 43473 | XXX 游泳池 | 89 | 43473 |
| XXXM 馆 | 12 | 43777 | XXX 中心大楼 泵房 | 23 | 43473 |
| XXXS 宾馆 | 167 | 43473 | XXX 种子楼 | 39 | 43472 |
| XXXS 馆 | 127 | 43473 | 茶园+ | 8 | 43546 |
| XXXT 馆后平 房 | 82 | 43473 | 东大门传达室+ | 128 | 43466 |
| XXX 成教院 | | | 东大门大棚+ | 2 | 43708 |
| XXX 分院 | 23 | 43527 | 东大门温室 | 214 | 43466 |
| XXX 大楼厕所 东 | 53 | 43473 | 高配房+ | 95 | 43503 |
| XXX 大楼厕所 西 | 100 | 43472 | 教育超市+ | 25 | 43499 |
| XXX 第八学生 宿舍 | 30 | 43472 | 老七楼 | 41 | 43472 |
| XXX 第二食堂 | 85 | 43467 | 老医务楼 2.3 楼 | | |
| XXX 第二学生 宿舍 | 156 | 43467 | + | 1 | 43683 |

| | | | | | |
|-----------------|-----|-------|--------------------|-----|-------|
| XXX 第九学生 宿舍 | 48 | 43472 | 离退休活动室 | 7 | 43642 |
| XXX 第七学生 宿舍 | 32 | 43473 | 留学生楼（新） 农业试验站大棚 | 353 | 43467 |
| XXX 第三学生 宿舍 | 51 | 43473 | + | 120 | 43495 |
| XXX 第四学生 宿舍 | 27 | 43473 | 区域 1（西） | 33 | 43503 |
| XXX 第五食堂 | 33 | 43473 | 区域 2 | 51 | 43473 |
| XXX 第五学生 宿舍 | 36 | 43467 | 区域 3+ | 8 | 43500 |
| XXX 第一食堂 | 25 | 43616 | 区域 4+ | 237 | 43470 |
| XXX 第一学生 宿舍 | 314 | 43466 | 司法鉴定中心 校管中心种子楼 | 27 | 43587 |
| XXX 东大楼 | 49 | 43473 | 东+ | 12 | 43526 |
| XXX 毒物研究 所 | 7 | 43684 | 校医院南+ | 25 | 43466 |
| XXX 干训楼 | 70 | 43489 | 新留学生楼 | 1 | 43801 |
| XXX 锅炉房 | 56 | 43471 | 养鱼组厕所+ | 37 | 43503 |
| XXX 国际纳米 研究所 | 41 | 43473 | 养鱼组临工宿舍 + | 20 | 43587 |
| XXX 航空航天 | 9 | 43473 | 养殖队 6721 副 表+ | 121 | 43483 |
| XXX 后勤楼 | 31 | 43526 | 养殖馆+ | 1 | 43812 |
| XXX 教学大楼 | | | 养殖馆附房保卫 | | |
| 总表 | 65 | 43466 | 处宿舍+ | 4 | 43474 |
| XXX 科学楼 | 282 | 43468 | 浴室 | 70 | 43473 |

13.1.3 问题四

附表 4 暗漏结果展示

| 水表名 | 暗漏频次 | 最早一次漏水时间 |
|------------|------|-----------|
| 64397 副表 | 6 | 2019/4/10 |
| XXX3 舍热泵热水 | 16 | 2019/1/8 |
| XXX4 舍热泵热水 | 21 | 2019/2/9 |
| XXX5 舍热泵热水 | 192 | 2019/1/1 |
| XXX8 舍热泵 | 23 | 2019/1/8 |
| XXXXK | 24 | 2019/1/8 |
| XXXXK 酒店 | 74 | 2019/1/8 |
| XXXXK 楼 | 31 | 2019/1/8 |
| XXXL 馆 | 32 | 2019/1/8 |
| XXXL 楼 | 40 | 2019/1/8 |
| XXXM 馆 | 7 | 2019/11/8 |
| XXXS 宾馆 | 134 | 2019/1/8 |

| | | |
|----------------|-----|-----------|
| XXXXS 馆 | 117 | 2019/1/8 |
| XXXXT 馆后平房 | 37 | 2019/1/8 |
| XXX 成教院 XXX 分院 | 9 | 2019/7/5 |
| XXX 大楼厕所东 | 27 | 2019/1/8 |
| XXX 大楼厕所西 | 88 | 2019/1/7 |
| XXX 第八学生宿舍 | 8 | 2019/1/8 |
| XXX 第二食堂 | 58 | 2019/1/5 |
| XXX 第二学生宿舍 | 94 | 2019/1/3 |
| XXX 第九学生宿舍 | 26 | 2019/1/8 |
| XXX 第七学生宿舍 | 17 | 2019/1/8 |
| XXX 第三学生宿舍 | 36 | 2019/1/8 |
| XXX 第四学生宿舍 | 15 | 2019/1/8 |
| XXX 第五食堂 | 24 | 2019/1/8 |
| XXX 第五学生宿舍 | 22 | 2019/1/3 |
| XXX 第一食堂 | 15 | 2019/5/31 |
| XXX 第一学生宿舍 | 304 | 2019/1/1 |
| XXX 东大楼 | 39 | 2019/1/8 |
| XXX 毒物研究所 | 7 | 2019/8/7 |
| XXX 干训楼 | 34 | 2019/3/2 |
| XXX 锅炉房 | 35 | 2019/1/7 |
| XXX 国际纳米研究所 | 26 | 2019/1/8 |
| XXX 航空航天 | 4 | 2019/1/8 |
| XXX 后勤楼 | 17 | 2019/3/2 |
| XXX 教学大楼总表 | 54 | 2019/1/1 |
| XXX 科学楼 | 228 | 2019/1/3 |
| XXX 老六楼 | 28 | 2019/4/16 |
| XXX 老五楼 | 23 | 2019/1/8 |
| XXX 老医务室楼 | 35 | 2019/1/8 |
| XXX 体育馆 | 18 | 2019/1/8 |
| XXX 田径场厕所 | 26 | 2019/1/8 |
| XXX 图书馆 | 41 | 2019/1/8 |
| XXX 污水处理 | 26 | 2019/1/16 |
| XXX 西大楼 | 32 | 2019/1/8 |
| XXX 校医院 | 30 | 2019/3/2 |
| XXX 游泳池 | 71 | 2019/1/8 |
| XXX 中心大楼泵房 | 14 | 2019/1/8 |
| XXX 种子楼 | 21 | 2019/1/8 |
| 茶园+ | 5 | 2019/3/23 |
| 东大门传达室+ | 72 | 2019/1/1 |
| 东大门大棚+ | 1 | 2019/9/1 |
| 东大门温室 | 195 | 2019/1/1 |
| 高配房+ | 47 | 2019/3/4 |

| | | |
|--------------|-----|-----------|
| 教育超市+ | 25 | 2019/2/3 |
| 老七楼 | 24 | 2019/1/7 |
| 离退休活动室 | 5 | 2019/11/9 |
| 留学生楼（新） | 346 | 2019/1/2 |
| 农业试验站大棚+ | 63 | 2019/1/30 |
| 区域 1（西） | 12 | 2019/8/2 |
| 区域 2 | 25 | 2019/1/8 |
| 区域 3+ | 4 | 2019/2/4 |
| 区域 4+ | 184 | 2019/1/6 |
| 司法鉴定中心 | 24 | 2019/5/2 |
| 校管中心种子楼东+ | 6 | 2019/3/2 |
| 校医院南+ | 7 | 2019/1/1 |
| 新留学生楼 | 1 | 2019/12/2 |
| 养鱼组厕所+ | 22 | 2019/2/15 |
| 养鱼组临工宿舍+ | 5 | 2019/5/3 |
| 养殖队 6721 副表+ | 49 | 2019/1/28 |
| 养殖馆附房保卫处宿舍+ | 1 | 2019/1/10 |
| 浴室 | 38 | 2019/1/8 |

附表 5 明漏的结果展示与分析

| 水表名 | 最小值项:采集时间 2 | 计数项:采集时间 | 最早漏水时间 |
|----------------|-------------|----------|------------|
| 64397 副表 | 14 | 13 | 2019/1/13 |
| XXX3 舍热泵热水 | 40 | 15 | 2019/2/8 |
| XXX4 舍热泵热水 | 50 | 24 | 2019/2/18 |
| XXX5 舍热泵热水 | 19 | 24 | 2019/1/18 |
| XXX8 舍热泵 | 17 | 34 | 2019/1/16 |
| XXXX | 40 | 19 | 2019/2/8 |
| XXXX 酒店 | 8 | 47 | 2019/1/7 |
| XXXX 楼 | 61 | 29 | 2019/3/1 |
| XXXL 馆 | 40 | 18 | 2019/2/8 |
| XXXL 楼 | 40 | 19 | 2019/2/8 |
| XXXM 馆 | 315 | 5 | 2019/11/10 |
| XXXS 宾馆 | 12 | 33 | 2019/1/11 |
| XXXS 馆 | 40 | 10 | 2019/2/8 |
| XXXT 馆后平房 | 33 | 45 | 2019/2/1 |
| XXX 成教院 XXX 分院 | 63 | 14 | 2019/3/3 |
| XXX 大楼厕所东 | 61 | 26 | 2019/3/1 |
| XXX 大楼厕所西 | 40 | 12 | 2019/2/8 |
| XXX 第八学生宿舍 | 8 | 22 | 2019/1/7 |
| XXX 第二食堂 | 3 | 27 | 2019/1/2 |
| XXX 第二学生宿舍 | 3 | 62 | 2019/1/2 |
| XXX 第九学生宿舍 | 8 | 22 | 2019/1/7 |

| | | | |
|-------------|-----|----|-----------|
| XXX 第七学生宿舍 | 40 | 15 | 2019/2/8 |
| XXX 第三学生宿舍 | 40 | 15 | 2019/2/8 |
| XXX 第四学生宿舍 | 40 | 12 | 2019/2/8 |
| XXX 第五食堂 | 61 | 9 | 2019/3/1 |
| XXX 第五学生宿舍 | 3 | 14 | 2019/1/2 |
| XXX 第一食堂 | 239 | 10 | 2019/8/26 |
| XXX 第一学生宿舍 | 5 | 10 | 2019/1/4 |
| XXX 东大楼 | 50 | 10 | 2019/2/18 |
| XXX 干训楼 | 25 | 36 | 2019/1/24 |
| XXX 锅炉房 | 7 | 21 | 2019/1/6 |
| XXX 国际纳米研究所 | 61 | 15 | 2019/3/1 |
| XXX 航空航天 | 40 | 5 | 2019/2/8 |
| XXX 后勤楼 | 142 | 14 | 2019/5/21 |
| XXX 教学大楼总表 | 61 | 11 | 2019/3/1 |
| XXX 科学楼 | 15 | 54 | 2019/1/14 |
| XXX 老六楼 | 129 | 24 | 2019/5/8 |
| XXX 老五楼 | 13 | 17 | 2019/1/12 |
| XXX 老医务室楼 | 40 | 25 | 2019/2/8 |
| XXX 体育馆 | 8 | 18 | 2019/1/7 |
| XXX 田径场厕所 | 27 | 16 | 2019/1/26 |
| XXX 图书馆 | 45 | 31 | 2019/2/13 |
| XXX 污水处理 | 16 | 14 | 2019/1/15 |
| XXX 西大楼 | 40 | 17 | 2019/2/8 |
| XXX 校医院 | 61 | 14 | 2019/3/1 |
| XXX 游泳池 | 142 | 18 | 2019/5/21 |
| XXX 中心大楼泵房 | 40 | 9 | 2019/2/8 |
| XXX 种子楼 | 8 | 18 | 2019/1/7 |
| 茶园+ | 82 | 3 | 2019/3/22 |
| 东大门传达室+ | 8 | 56 | 2019/1/7 |
| 东大门大棚+ | 244 | 1 | 2019/8/31 |
| 东大门温室 | 53 | 19 | 2019/2/21 |
| 高配房+ | 39 | 48 | 2019/2/7 |
| 老七楼 | 23 | 17 | 2019/1/22 |
| 老医务楼 2.3 楼+ | 219 | 1 | 2019/8/6 |
| 离退休活动室 | 178 | 2 | 2019/6/26 |
| 留学生楼（新） | 57 | 7 | 2019/2/25 |
| 农业试验站大棚+ | 69 | 57 | 2019/3/9 |
| 区域 1（西） | 39 | 21 | 2019/2/7 |
| 区域 2 | 40 | 26 | 2019/2/8 |
| 区域 3+ | 39 | 4 | 2019/2/7 |
| 区域 4+ | 6 | 53 | 2019/1/5 |
| 司法鉴定中心 | 126 | 3 | 2019/5/5 |

| | | | |
|--------------|-----|----|------------|
| 校管中心种子楼东+ | 70 | 6 | 2019/3/10 |
| 校医院南+ | 5 | 18 | 2019/1/4 |
| 养鱼组厕所+ | 39 | 15 | 2019/2/7 |
| 养鱼组临工宿舍+ | 123 | 15 | 2019/5/2 |
| 养殖队 6721 副表+ | 19 | 72 | 2019/1/18 |
| 养殖馆+ | 348 | 1 | 2019/12/13 |
| 养殖馆附房保卫处宿舍+ | 10 | 3 | 2019/1/9 |
| 浴室 | 89 | 32 | 2019/3/29 |

13.2 程序附录

说明：当应用 **python** 软件时，这一部分代码表示加载分析常用包，以下每个程序都需先撰写这一块，此后问题中的 **python** 代码将不做重复附录。

```

1. import pandas as pd
2. import numpy as np
3. import pandas as pd
4. import pandas as pd
5. import numpy as np
6. import matplotlib.pyplot as plt
7. from pandas import DataFrame, Series
8. import seaborn as sns
9. import matplotlib.pyplot as plt
10. from sklearn.linear_model import LinearRegression
11. import statsmodels.api as sm
12. from sklearn import preprocessing

```

13.2.1 数据预处理

```

1. # 定义 画图中文异常
2. plt.rcParams['font.sans-serif'] = ['SimHei'] # 中文字体设置-黑体
3. plt.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号 '-' 显示为方块的问题
4. sns.set(font='SimHei') # 解决 Seaborn 中文显示问题

```

13.2.2 问题一

```

1. #读表
2. cc1=pd.read_csv("附件_一季度.csv",encoding="gbk")
3. cc2=pd.read_csv("附件_二季度.csv",encoding="gbk")
4. cc3=pd.read_csv("附件_三季度.csv",encoding="gbk")
5. cc4=pd.read_csv("附件_四季度.csv",encoding="gbk")
6. #对表进行叠加
7. df = pd.concat([cc1, cc2,cc3,cc4])
8. #数据筛选
9. cc=df.loc[df["水表名"]=="东大门传达室+"]
10. cc.to_csv("东大门传达室+.csv")
11. #数据 描述性分析
12. cc5=df.groupby("水表名").describe()
13. cc6=cc4.groupby("水表名").sum()

```



```

14. #导出结果
15. cc5.to_csv("描述.csv")
16. cc6.to_csv("总用水 4.csv")
17. #判断读数差和用量直接的因果关系
18. df["cha"]=(df["当前读数"]-df["上次读数"]-df["用量"])
19. len(df["cai"].loc[(df["cha"]>0) & (df["cha"]<0.01)])
20. #Out[192]: 661040
21. # 727156 -0.01 ~ 0 总共 130W
22.
23. len(df["cha"].loc[(df["cha"]>0.01) & (df["cha"]<0.011)])
24. #Out[195]: 66116
25. #统计
26. df["ss"]=df["当前读数"]-df["上次读数"]
27. df["ss"]=df["ss"]-df["用量"]
28. aa=df.loc[df["ss"]!=0]
29. #计算用量差
30. cc=df.loc[df["水表名"]=="校医院南+"]
31. bb=df.loc[df["水表名"]=="XXX 校医院"]
32. aa=df.loc[df["水表名"]=="车队+"]
33. ff=pd.merge(cc,bb,on="采集时间")
34. gg=pd.merge(ff,aa,on="采集时间")
35. gg["差"]=gg["用量_x"]-gg["用量_y"]
36. len(gg["差"].loc[gg["差"]<0])
37. cc=df.loc[df["水表名"]=="XXX 成教院 XXX 分院"]
38. bb=df.loc[df["水表名"]=="离退休活动室"]
39. aa=df.loc[df["水表名"]=="XXXL 馆"]
40. zz=df.loc[df["水表名"]=="XXX 花圃+"]
41. aa=pd.merge(zz,aa,on="采集时间")
42. aa=pd.merge(aa,bb,on="采集时间")
43. aa=pd.merge(aa,cc,on="采集时间")
44. aa["差"]=aa["用量_x"]-aa["用量_y"]
45. len(aa["差"].loc[aa["差"]<0])
46. #获得不重复的表名
47. ming=df["水表名"].unique()
48. #折线图 波士顿
49. for i in ming:
50.     aa=df.loc[df["水表名"]==i]
51.     aa=aa[["当前读数","采集时间"]]
52.     plt.plot(aa["当前读数"],aa["采集时间"],color='red',linewidth=2.0,linestyle='--')
53.     plt.show()

```

13.2.3 问题二

```

1. b1=pd.read_csv("b1.csv",encoding="gbk") #一级表
2. b2=pd.read_csv("b2.csv",encoding="gbk") #二级表

```

```

3. b3=pd.read_csv("b3.csv",encoding="gbk") #三级表
4. b4=pd.read_csv("b4.csv",encoding="gbk") #四级表
5. year_sum=pd.read_csv("niansum.csv",encoding="gbk") #年汇总数据
6.
7. #数据匹配
8. b11=pd.merge(b1,year_sum)
9. b22=pd.merge(b2,year_sum)
10. b33=pd.merge(b3,year_sum)
11. b44=pd.merge(b4,year_sum)
12. b11.to_csv("b11.csv")
13. b22.to_csv("b22.csv")
14. b33.to_csv("b33.csv")
15. b44.to_csv("b44.csv")
16.
17. #遍历
18. #确定二级求和结果
19. b5=[]
20. for i in b1.values:
21.     b5=b22[(b22["b2"]>int(i)*100)&(b22["b2"]<int(i)*100+99)]
22.     print(b5["cha"].sum())
23.
24. b5=[]
25. #确定三级求和结果
26. for i in b2.values:
27.     b5=b33[(b33["b3"]>int(i)*100)&(b33["b3"]<int(i)*100+99)]
28.     print(b5["cha"].sum())
29.
30. b5=[]
31. #确定四级求和结果
32. for i in b3.values:
33.     b5=b44[(b44["b4"]>int(i)*100)&(b44["b4"]<int(i)*100+99)]
34.     print(b5["cha"].sum())

```

13.2.4 问题三

(1) 层级误差的数据修正

```

1. cc1=pd.read_csv("一季度.csv",encoding="gbk")
2. cc2=pd.read_csv("二季度.csv",encoding="gbk")
3. cc3=pd.read_csv("三季度.csv",encoding="gbk")
4. cc4=pd.read_csv("四季度.csv",encoding="gbk")
5. df = pd.concat([cc1, cc2,cc3,cc4])
6. df["cha"]=df["当前读数"]-df["上次读数"]
7. cc=cc5[cc5.水表名.isin(['XXX 花圃+', 'XXX 成教院 XXX 分院','XXX 田径场厕所',"离退休活动室","XXXX 馆","XXXX 楼","XXS
  馆","XXXX","XXXX 酒店","XXX 体育馆","XXX 干训楼"])]
8. cc=cc5[cc5.水表名.isin(['校医南',"XXX 校医院","车队+"))]

```

```

9. cc=cc5[cc5.水表号.isin(["3320100600","183671860","3320100700","3200200100","3390100400","3180700100"])]
10. cc=cc5[cc5.水表号.isin(["3620300200","0","3315400100","3620302700","3620302600","3000000001","3290100300","3422000100","3620302500","3280100100","3161100100","3160100100","3000000000","3421300300","3421900100","3421300200","3170200100","3170100600","3620302000","3620301200","3620301000","3620300900","3620300800","3620300700","3620300600","3620301100","3400500100","3421300100","3422200100","3315400200","3160300300","3620301300"])]
11. cc=cc5[cc5.水表号.isin(["3620301800","3480300100","3480200100","3320100900","3450200100","3320100800"])]
12. cc=cc5[cc5.水表号.isin(["3620300300","3160200300","3620301400","3320100200","3480100100","3620301700","3320100100","3620302100","3620304000","3480400100","3320100500","1836718629","3320100400","3620301500","1836718625","3320100300","1836718633","3620302400","3620302300","3620301900","3390100500","3520300100","3360300100","3390100200","3390100300","3421400100"])]
13. # 倒推
14. t401=pd.read_csv("t401.csv",encoding="gbk")
15. t402=pd.read_csv("t402.csv",encoding="gbk")
16. t403=pd.read_csv("t403.csv",encoding="gbk")
17. t404=pd.read_csv("t404.csv",encoding="gbk")
18. t405=pd.read_csv("t405.csv",encoding="gbk")
19. t416=pd.read_csv("t416.csv",encoding="gbk")
20. t401["cha"]=t401["当前读数"]-t401["上次读数"]
21. t402["cha"]=t402["当前读数"]-t402["上次读数"]
22. t403["cha"]=t403["当前读数"]-t403["上次读数"]
23. t404["cha"]=t404["当前读数"]-t404["上次读数"]
24. t405["cha"]=t405["当前读数"]-t405["上次读数"]
25. t416["cha"]=t416["当前读数"]-t416["上次读数"]
26. # 探究 层级之间 是否截流, 和对用量进行修正
27. #405
28. #三进 2
29. cc=t405[t405.水表名.isin(["XXX 田径场厕所"])].reset_index(drop=True)
30. cc.set_index(["采集时间 2"], inplace=True)
31. ccc=t405[t405.水表名.isin(["XXX 成教院 XXX 分院"])].reset_index(drop=True)
32. ccc.set_index(["采集时间 2"], inplace=True)
33. cccc=ccc["用量"]-cc["用量"]
34. cccc.to_csv("cccc.csv")
35. #2 进 1
36. cc=t405[t405.水表名.isin(["XXX 成教院 XXX 分院","离退休活动室","XXXL 馆","XXXL 楼","XXXS 馆","XXXK","XXXX 酒店","XXX 体育馆","XXX 干训楼"])].reset_index(drop=True)
37. cc.set_index(["采集时间 2"], inplace=True)
38. ccc=t405[t405.水表名.isin(["XXX 花园+"])].
39. cc=cc.groupby("采集时间 2")["用量"].sum()
40. ccc.set_index(["采集时间 2"], inplace=True)
41. cccc=ccc["用量"]-cc
42. cccc.to_csv("cccc.csv")
43. #401
44. #3jin 2

```

```

45. cc=t401[t401.水表名.isin(["XXXT 馆后平房","XXX 后勤楼","校管中心种子楼东","XXX 图书馆","XXX 毒物研究所","XXX 种子楼"])]
46. cc.set_index(["采集时间 2"], inplace=True)
47. ccc=t401[t401.水表名.isin(["区域 2"])]
48. ccc.set_index(["采集时间 2"], inplace=True)
49. cc=cc.groupby("采集时间 2")["用量"].sum()
50. cccc=ccc["用量"]-cc
51. cccc.to_csv("cccc.csv")
52. #3jin2
53. cc=t401[t401.水表名.isin(["纳米楼 4.5 楼+","纳米楼 3 楼+"])]
54. cc.set_index(["采集时间 2"], inplace=True)
55. ccc=t401[t401.水表号.isin(["3315400100"])]
56. ccc.set_index(["采集时间 2"], inplace=True)
57.
58. cc=cc.groupby("采集时间 2")["用量"].sum()
59.
60. cccc=ccc["用量"]-cc
61. cccc.to_csv("cccc.csv")
62. #3jin2
63. cc=t401[t401.水表名.isin(["XXX 大楼厕所西","XXX 科学楼","XXX 大楼厕所东","XXX 中心水池","XXX 西大楼"])]
64. cc.set_index(["采集时间 2"], inplace=True)
65. ccc=t401[t401.水表名.isin(["区域 1（西）"])]
66. ccc.set_index(["采集时间 2"], inplace=True)
67. cc=cc.groupby("采集时间 2")["用量"].sum()
68. cccc=ccc["用量"]-cc
69. cccc.to_csv("cccc.csv")
70. #3jin2
71. cc=t401[t401.水表名.isin(["高配房+"])]
72. cc.set_index(["采集时间 2"], inplace=True)
73. ccc=t401[t401.水表名.isin(["XXX 植物园"])]
74. ccc.set_index(["采集时间 2"], inplace=True)
75. cc=cc.groupby("采集时间 2")["用量"].sum()
76. cccc=ccc["用量"]-cc
77. cccc.to_csv("cccc.csv")
78. # 2jin 1
79. cc=t401[t401.水表名.isin(["司法鉴定中心","XXX 国际纳米研究所","区域 2","区域 1（西）","书店+","新大门传达室+","养殖馆附房保卫处宿舍+","养殖馆公共厕所+","养殖馆附房二楼厕所+","养殖馆附房一楼厕所+","养殖馆+","养殖队+","XXX 教学大楼总表","XXX 中心大楼泵房","XXX 东大楼","XXXM 馆","XXX 植物园"])]
80. cc.set_index(["采集时间 2"], inplace=True)
81. ccc=t401[t401.水表名.isin(["养殖队 6721 副表+"])]
82. ccc.set_index(["采集时间 2"], inplace=True)
83. cc=cc.groupby("采集时间 2")["cha"].sum()
84. cccc=ccc["cha"]-cc
85. cccc.to_csv("cccc.csv")

```

```

86. #404
87. #3jin2
88. cc=t404[t404.水表号.isin(["183671860"])]
89. cc.set_index(["采集时间 2"], inplace=True)
90. ccc=t404[t404.水表名.isin(["XXX 第八学生宿舍"])]
91. ccc.set_index(["采集时间 2"], inplace=True)
92. cc=cc.groupby("采集时间 2")["cha"].sum()
93. cccc=ccc["cha"]-cc
94. cccc.to_csv("cccc.csv")
95. #416
96. #3jin 2
97. cc=t416[t416.水表号.isin(["3620303100"])]
98. cc.set_index(["采集时间 2"], inplace=True)
99. ccc=t416[t416.水表名.isin(["XXX 校医院"])]
100. ccc.set_index(["采集时间 2"], inplace=True)
101. cc=cc.groupby("采集时间 2")["cha"].sum()
102. cccc=ccc["cha"]-cc
103. cccc.to_csv("cccc.csv")
104. #2jin 1
105. cc=t416[t416.水表号.isin(["3290100100"])]
106. cc.set_index(["采集时间 2"], inplace=True)
107. ccc=t416[t416.水表号.isin(["3620300500"])]
108. ccc.set_index(["采集时间 2"], inplace=True)
109. cc=cc.groupby("采集时间 2")["cha"].sum()
110. cccc=ccc["cha"]-cc
111. cccc.to_csv("cccc.csv")
112. #403
113. #4jin3
114. cc=t403[t403.水表号.isin(["1836718629"])]
115. cc.set_index(["采集时间 2"], inplace=True)
116. ccc=t403[t403.水表号.isin(["3320100500"])]
117. ccc.set_index(["采集时间 2"], inplace=True)
118. cc=cc.groupby("采集时间 2")["cha"].sum()
119. cccc=ccc["cha"]-cc
120. cccc.to_csv("cccc.csv")
121. #4 jin 3
122. cc=t403[t403.水表号.isin(["3620301500","1836718625"])]
123. cc.set_index(["采集时间 2"], inplace=True)
124. ccc=t403[t403.水表号.isin(["3320100400"])]
125. ccc.set_index(["采集时间 2"], inplace=True)
126. cc=cc.groupby("采集时间 2")["cha"].sum()
127. cccc=ccc["cha"]-cc
128. cccc.to_csv("cccc.csv")
129. #4 jin 3

```

```

130. cc=t403[t403.水表号.isin(["1836718633"])]
131. cc.set_index(["采集时间 2"], inplace=True)
132. ccc=t403[t403.水表号.isin(["3320100300"])]
133. ccc.set_index(["采集时间 2"], inplace=True)
134. cc=cc.groupby(["采集时间 2"])[["cha"].sum()
135. cccc=ccc["cha"]-cc
136. cccc.to_csv("cccc.csv")
137. #3jin 2
138. cc=t403[t403.水表号.isin(["3320100200","3480100100","3620301700","3320100100"])]
139. cc.set_index(["采集时间 2"], inplace=True)
140. ccc=t403[t403.水表号.isin(["3620301400"])]
141. ccc.set_index(["采集时间 2"], inplace=True)
142.
143. cc=cc.groupby(["采集时间 2"])[["cha"].sum()
144. cccc=ccc["cha"]-cc
145. cccc.to_csv("cccc.csv")
146. #3jin 2
147. cc=t403[t403.水表号.isin(["3480400100","3320100500","3320100400","3320100300"])]
148. cc.set_index(["采集时间 2"], inplace=True)
149. ccc=t403[t403.水表号.isin(["3620300400"])]
150. ccc.set_index(["采集时间 2"], inplace=True)
151. cc=cc.groupby(["采集时间 2"])[["cha"].sum()
152. cccc=ccc["cha"]-cc
153. cccc.to_csv("cccc.csv")
154. #2jin 1
155. cc=t403[t403.
                                水
                                表
                                号.isin(["3160200300","3620301400","3620302100","3620300400","3620302400","3620302300","3620301900","3390100500","352030
                                0100","3360300100","3390100200","3390100300","3421400100"])]
156. cc.set_index(["采集时间 2"], inplace=True)
157.
158. ccc=t403[t403.水表号.isin(["3620300300"])]
159. ccc.set_index(["采集时间 2"], inplace=True)
160.
161. cc=cc.groupby(["采集时间 2"])[["cha"].sum()
162.
163. cccc=ccc["cha"]-cc
164. cccc.to_csv("cccc.csv")
165. #频次
166. cc5=pd.read_excel("无流量重复（总结）.xlsx")
167. cc6=cc5.groupby(["水表","时间"])[["cha"].value_counts()
168. cc6.to_csv("chuchong .csv")
169. #后续在 excel 中操作
170.

```

(2) 爆漏模型 (DBSCAN)

```
1. # DBSCAN
2. from sklearn.cluster import DBSCAN
3. # 定义模型
4. cc99=pd.read_csv("水表层级.csv",encoding="gbk")
5. cc1=pd.read_csv("附件_一季度.csv",encoding="gbk")
6. cc2=pd.read_csv("附件_二季度.csv",encoding="gbk")
7. cc3=pd.read_csv("附件_三季度.csv",encoding="gbk")
8. cc4=pd.read_csv("附件_四季度.csv",encoding="gbk")
9. cc1["cha"]=cc1["当前读数"]-cc1["上次读数"]
10. cc2["cha"]=cc2["当前读数"]-cc2["上次读数"]
11. cc3["cha"]=cc3["当前读数"]-cc3["上次读数"]
12. cc4["cha"]=cc4["当前读数"]-cc4["上次读数"]
13. df= pd.concat([cc1, cc2,cc3,cc4])
14. cc100=pd.read_csv("水表层级(1).csv",encoding="gbk")
15. cc111=pd.merge(df,cc100,on="水表名")
16. cc111=cc111[["水表名","水表号_x","采集时间","上次读数","当前读数","用量","口径"]]
17. cc110=cc111.groupby(["水表名","采集时间"])[["用量"].sum()
18. cc110.to_csv("日用水合.csv")
19. cc112=pd.read_csv("日用水合.csv",encoding="gbk")
20. cc113=pd.merge(cc112,cc100,on="水表名")
21. cc113=cc113[["水表名","采集时间","用量","口径"]]
22. X =cc113[["用量","口径"]]
23. model = DBSCAN(eps=10, min_samples=2)
24. # 模型拟合与聚类预测
25. yhat = model.fit_predict(X)
26. labels = model.labels_
27. cc113['cluster_db'] = labels # 在数据集最后一列加上经过 DBSCAN 聚类后的结果
28. cc113.to_csv("白天.csv")
29. # 爆管处理
30. cc1=pd.read_csv("一季度.csv",encoding="gbk")
31. cc2=pd.read_csv("二季度.csv",encoding="gbk")
32. cc3=pd.read_csv("三季度.csv",encoding="gbk")
33. cc4=pd.read_csv("四季度.csv",encoding="gbk")
34. df= pd.concat([cc1, cc2,cc3,cc4])
35. df["cha"]=df["当前读数"]-df["上次读数"]
36. cc=df.groupby(["水表名","采集时间"])[["cha"].mean()
37. ccq=df.groupby(["水表名","采集时间"])[["cha"].value_counts()
38. ccw=ccq/cc
39. cc.to_csv("cc.csv")
40. ccq.to_csv("ccq.csv")
41. ccw.to_csv("ccw.csv")
42. cc=pd.read_csv("cc.csv",encoding="gbk")
43. ccq=pd.read_csv("ccq.csv",encoding="gbk")
```



```

44. cce=pd.merge(cc,ccq,on=["时间","表"])
45. cce.to_csv("cce.csv")
46. cce=pd.read_csv("cce.csv",encoding="gbk")
47. ccr=cce.groupby(["表","时间"])["结果"].sum()
48. ccr.to_csv("ccr.csv")
49. ccr=pd.read_csv("ccr.csv",encoding="gbk")
50. #孤立森林
51. from sklearn.ensemble import IsolationForest
52. import numpy as np
53.
54. ming=ccr["表"].unique()
55. model_isof = IsolationForest(n_estimators=20)
56. #养殖馆+
57. cc22=pd.DataFrame(columns=['表', '时间', '结果'])
58. for i in ming:
59.     cc21=ccr[ccr.表.isin([i])].reset_index(drop=True)
60.     X=np.array(cc21["结果"]).reshape(-1,1)
61.     outlier_label = model_isof.fit_predict(X)
62.     cc21["outlier_pd"]=outlier_label
63.     cc22=pd.concat([cc21,cc22])

```

13.2.5 问题四

```

1. #问题四 暗漏 明漏数据采集和预处理
2. #405
3. #三进 2
4. cc=df[df.水表名.isin(["XXX 田径场厕所"])].reset_index(drop=True)
5. cc.set_index(["采集时间"], inplace=True)
6. ccc=df[df.水表名.isin(["XXX 成教院 XXX 分院"])].reset_index(drop=True)
7. cc=cc.groupby("采集时间")["cha"].sum()
8. ccc.set_index(["采集时间"], inplace=True)
9. ccc["下级合"]=cc
10. ccc["上下级差"]=ccc["cha"]-cc
11. ccc.to_csv("cccc.csv")
12. #2 进 1
13. cc=df[df.水表名.isin(["XXX 成教院 XXX 分院","离退休活动室","XXXL 馆","XXXL 楼","XXXS 馆","XXXK","XXXK 酒店","XXX 体育馆","XXX 干训楼"])].reset_index(drop=True)
14. cc.set_index(["采集时间"], inplace=True)
15. ccc=df[df.水表名.isin(["XXX 花圃+"])
16. cc=cc.groupby("采集时间")["cha"].sum()
17. ccc.set_index(["采集时间"], inplace=True)
18. ccc["下级合"]=cc
19. ccc["上下级差"]=ccc["cha"]-cc
20. ccc.to_csv("cccc.csv")
21. #401

```

```

22. #3jin 2
23. cc=df[df.水表名.isin(["XXXT 馆后平房","XXX 后勤楼","校管中心种子楼东+","XXX 图书馆","XXX 毒物研究所","XXX 种子楼"])]
24. cc.set_index(["采集时间"], inplace=True)
25. ccc=df[df.水表名.isin(["区域 2"])]
26. ccc.set_index(["采集时间"], inplace=True)
27. cc=cc.groupby("采集时间")["cha"].sum()
28. ccc["下级合"]=cc
29. ccc["上下级差"]=ccc["cha"]-cc
30. ccc.to_csv("cccc.csv")
31. #3jin2
32. cc=df[df.水表名.isin(["纳米楼 4.5 楼+","纳米楼 3 楼+"])]
33. cc.set_index(["采集时间"], inplace=True)
34. ccc=df[df.水表号.isin(["3315400100"])]
35. ccc.set_index(["采集时间"], inplace=True)
36. cc=cc.groupby("采集时间")["cha"].sum()
37. ccc["下级合"]=cc
38. ccc["上下级差"]=ccc["cha"]-cc
39. ccc.to_csv("cccc.csv")
40. #3jin2
41. cc=df[df.水表名.isin(["XXX 大楼厕所西","XXX 科学楼","XXX 大楼厕所东","XXX 中心水池","XXX 西大楼"])]
42. cc.set_index(["采集时间"], inplace=True)
43. ccc=df[df.水表名.isin(["区域 1（西）"])]
44. ccc.set_index(["采集时间"], inplace=True)
45. cc=cc.groupby("采集时间")["cha"].sum()
46. ccc["下级合"]=cc
47. ccc["上下级差"]=ccc["cha"]-cc
48. ccc.to_csv("cccc.csv")
49. #3jin2
50. cc=df[df.水表名.isin(["高配房+"])]
51. cc.set_index(["采集时间"], inplace=True)
52. ccc=df[df.水表名.isin(["XXX 植物园"])]
53. ccc.set_index(["采集时间"], inplace=True)
54. cc=cc.groupby("采集时间")["cha"].sum()
55. ccc["下级合"]=cc
56. ccc["上下级差"]=ccc["cha"]-cc
57. ccc.to_csv("cccc.csv")
58. #2jin 1
59. cc=df[df.水表名.isin(["司法鉴定中心","XXX 国际纳米研究所","区域 2","区域 1（西）","书店+","新大门传达室+","养殖馆附房保卫
处宿舍+","养殖馆公共厕所+","养殖馆附房二楼厕所+","养殖馆附房一楼厕所+","养殖馆+","养殖队+","XXX 教学大楼总表","XXX
中心大楼泵房","XXX 东大楼","XXXM 馆","XXX 植物园"])]
60. cc.set_index(["采集时间"], inplace=True)
61. ccc=df[df.水表名.isin(["养殖队 6721 副表+"])]
62. ccc.set_index(["采集时间"], inplace=True)
63. cc=cc.groupby("采集时间")["cha"].sum()

```

```

64. ccc["下级合"]=cc
65. ccc["上下级差"]=ccc["cha"]-cc
66. ccc.to_csv("cccc.csv")
67. #404
68. #3jin2
69. cc=df[df.水表号.isin(["183671860"])]
70. cc.set_index(["采集时间"], inplace=True)
71. ccc=df[df.水表名.isin(["XXX 第八学生宿舍"])]
72. ccc.set_index(["采集时间"], inplace=True)
73. cc=cc.groupby("采集时间")["cha"].sum()
74. ccc["下级合"]=cc
75. ccc["上下级差"]=ccc["cha"]-cc
76. ccc.to_csv("cccc.csv")
77. #416
78. #3jin 2
79. cc=df[df.水表号.isin(["3620303100"])]
80. cc.set_index(["采集时间"], inplace=True)
81.
82. ccc=df[df.水表名.isin(["XXX 校医院"])]
83. ccc.set_index(["采集时间"], inplace=True)
84. cc=cc.groupby("采集时间")["cha"].sum()
85. ccc["下级合"]=cc
86. ccc["上下级差"]=ccc["cha"]-cc
87. ccc.to_csv("cccc.csv")
88. #2jin 1
89. cc=df[df.水表号.isin(["3290100100"])]
90. cc.set_index(["采集时间"], inplace=True)
91. ccc=df[df.水表号.isin(["3620300500"])]
92. ccc.set_index(["采集时间"], inplace=True)
93. cc=cc.groupby("采集时间")["cha"].sum()
94. ccc["下级合"]=cc
95. ccc["上下级差"]=ccc["cha"]-cc
96. ccc.to_csv("cccc.csv")
97. #403
98. #4jin3
99. cc=df[df.水表号.isin(["1836718629"])]
100. cc.set_index(["采集时间"], inplace=True)
101. ccc=df[df.水表号.isin(["3320100500"])]
102. ccc.set_index(["采集时间"], inplace=True)
103. cc=cc.groupby("采集时间")["cha"].sum()
104. ccc["下级合"]=cc
105. ccc["上下级差"]=ccc["cha"]-cc
106. ccc.to_csv("cccc.csv")
107. #4 jin 3

```

```

108. cc=df[df.水表号.isin(["3620301500","1836718625"])]
109. cc.set_index(["采集时间"], inplace=True)
110. ccc=df[df.水表号.isin(["3320100400"])]
111. ccc.set_index(["采集时间"], inplace=True)
112. cc=cc.groupby("采集时间")["cha"].sum()
113. ccc["下级合"]=cc
114. ccc["上下级差"]=ccc["cha"]-cc
115. ccc.to_csv("cccc.csv")
116. #4 jin 3
117. cc=df[df.水表号.isin(["1836718633"])]
118. cc.set_index(["采集时间"], inplace=True)
119. ccc=df[df.水表号.isin(["3320100300"])]
120. ccc.set_index(["采集时间"], inplace=True)
121. cc=cc.groupby("采集时间")["cha"].sum()
122. ccc["下级合"]=cc
123. ccc["上下级差"]=ccc["cha"]-cc
124. ccc.to_csv("cccc.csv")
125. #3jin 2
126. cc=df[df.水表号.isin(["3320100200","3480100100","3620301700","3320100100"])]
127. cc.set_index(["采集时间"], inplace=True)
128. ccc=df[df.水表号.isin(["3620301400"])]
129. ccc.set_index(["采集时间"], inplace=True)
130. cc=cc.groupby("采集时间")["cha"].sum()
131. ccc["下级合"]=cc
132. ccc["上下级差"]=ccc["cha"]-cc
133. ccc.to_csv("cccc.csv")
134. #3jin 2
135. cc=df[df.水表号.isin(["3480400100","3320100500","3320100400","3320100300"])]
136. cc.set_index(["采集时间"], inplace=True)
137. ccc=df[df.水表号.isin(["3620300400"])]
138. ccc.set_index(["采集时间"], inplace=True)
139. cc=cc.groupby("采集时间")["cha"].sum()
140. ccc["下级合"]=cc
141. ccc["上下级差"]=ccc["cha"]-cc
142. ccc.to_csv("cccc.csv")
143. #2jin 1
144. cc=df[df.水表号.isin(["3160200300","3620301400","3620302100","3620300400","3620302400","3620302300","3620301900","3390100500","352030100","3360300100","3390100200","3390100300","3421400100"])]
145. cc.set_index(["采集时间"], inplace=True)
146. ccc=df[df.水表号.isin(["3620300300"])]
147. ccc.set_index(["采集时间"], inplace=True)
148. cc=cc.groupby("采集时间")["cha"].sum()
149. ccc["下级合"]=cc

```

```

150. ccc["上下级差"]=ccc["cha"]-cc
151. ccc.to_csv("cccc.csv")
152. #统计 无下级
153. cc=df[df.                                     水                                     表
    号.isin(["3620303100","3421200300","3363000100","3313800500","3370100100","3313200200","3370300100","3370200100","303010
    0100","3210100100","183671860","3320100700","3200200100","3390100400","3180700100","3160200300","3320100200","34801001
    00","3620301700","3320100100","3620302100","3480400100","1836718629"])]
154. cccc=df[df.                                     水                                     表
    号.isin(["3620301500","1836718625","1836718633","3620302400","3620302300","3620301900","3390100500","3520300100","336030
    0100","3390100200","3390100300","3421400100","3620301800","3480300100","3480200100","3320100900","3450200100","3320100
    800","0","3620302700","3620302600","3290100300"])]
155. ccccc=df[df.                                     水                                     表
    号.isin(["3422000100","3620302500","3280100100","3161100100","3160100100","3421300300","3421900100","3421300200","317020
    0100","3170100600","3620302000","3620301200","3620301000","3620300900","3620300800","3620300700","3620300600","3620301
    100","3400500100","3421300100","3422200100"])]
156. cccccc=df[df.                                     水                                     表
    号.isin(["3315400200","3620301300","3620303000","3620302900","3620302800","3030100102","3030100101","3312800100","362030
    2200"])]
157. #拆分小数据集 供 excel 打开 注( excel 最多打开 100W 行数据 而 统计后有将近 280W 数据)
158. cc.to_csv("无下级.csv")
159. cccc.to_csv("无下级 1.csv")
160. ccccc.to_csv("无下级 2.csv")
161. cccccc.to_csv("无下级 3.csv")
162. cc12=pd.concat([cc,cccc,ccccc,ccccc])
163.
164. #导入已经处理后的层次截流关系
165. cc11=pd.read_csv("lv9.14.csv",encoding="gbk")
166. #叠加
167. cc13=pd.concat([cc11,cc12])
168. #筛选夜间
169. cc14=cc13.loc[(cc13["时间"]>2) & (cc13["时间"]<5)]
170. cc15=cc14.groupby(["水表名","采集时间"])["cha"].value_counts()
171. cc15.to_excel("cc15.xlsx")

```