

校园供水系统智能管理

摘要

智能水表的使用伴随着人工智能与科技的发展，变得越来越广泛，它的使用极大的方便了管理人员对用水情况的管理，它产生的数据也可以帮助其进行合理的数据估计，从而合理规划维修，本文基于某校园智能水表产生的数据，进行了挖掘和分析，具体如下：

对问题一，首先借助于 Python 中的 pandas 库对大规模、多种数据类型共存数据处理的优势进行数据的操作和处理，描述分析了宿舍、教学楼、办公楼、食堂的用水特征；

对问题二，在问题一的基础上，首先利用水表层级关系附件，建立各级表的数据关系拓扑图，结合实际情况分析得出，一级水表中 401X、403X、405X 处的数据与二级水表相比误差率分别为 11%、11%、4%；其他一级水表按照优质水表 $\pm 0.5\%- \pm 3\%$ ；二级以上水表按照普通水表-5%-3%来标记误差；

对于问题三，首先利用时间点左右各 5 个点对其本身进行样条插值平滑预测，利用预测值与原始值的差值估计漏损量，总漏损量为 32215，总漏损率为 9.8%，该校园漏损率低于全国的平均漏损率，此外还分析评估了各水表漏损情况；

对于问题四，根据上一问的估计结果，以 9.8% 为阈值，漏损率超出阈值则标记一次漏损事故，结合各水表处的漏水量情况，按照漏损次数对水表进行漏损等级评定，得到 40118T/40121T/40331T 等水表点的漏水可能性较高；

对于问题五，通过查看相关文献资料，获得平均水价范围 5.5 元/吨-6.5 元/吨，单次维修费用 500 元/次-1500 元/次，建立了每天是否维修的 0-1 规划模型，并取平均水价 6 元/吨以及单次维修成本均价 1000 元/次分析维修次数、损失金额与水价和维修成本的关系，最终的漏水维修 33 次，总损失金额为 174630 元。

本文由数据驱动，建立模型挖掘校园用水的特性，对于供热网、高速路网等问题都具有良好的推广应用价值。

关键字：数据挖掘 样条插值平滑预测 漏损检测 0-1 规划



关注数学模型
获取更多资讯

一 问题重述

1.1 问题背景

水是生命之源，自来水通过供水管道进入千家万户，随着科学技术的不断发展，在校园中，学校建立了一套基于智能水表的供水情况监测系统，目的是实时监测供水管道中各种运行数据，例如管道内的水压、用水量的记录。

供水管道的检漏、维护和修补，这需要耗费巨大的人力、物力、财力，是否可以通过智能水表提供的准确、及时的数据，分析管道中潜在的漏水问题，并进行及时的处理修补，减少由于水资源浪费造成的经济损失以及检漏、维护成本支出。

1.2 已知条件

1. 水表的层级关系
2. 水表四个季度所有的用水数据记录（每十五分钟记录一次）

1.3 解决问题

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（宿舍、教学楼、办公楼、食堂等）的用水特征。
2. 结合水表层级关系，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. 输水管网漏损严重，平均失水在 5% 左右；较老旧的管网，失水则更为严重。利用已有数据，建立数学模型，分析该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，从水表的实时数据发现并确定发生漏损的位置。
5. 管网维修需要人工费和材料费，但同时可以降低管网漏损程度。根据上述结论和查询到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

对于整体五个问题我们的分析流程见下图：



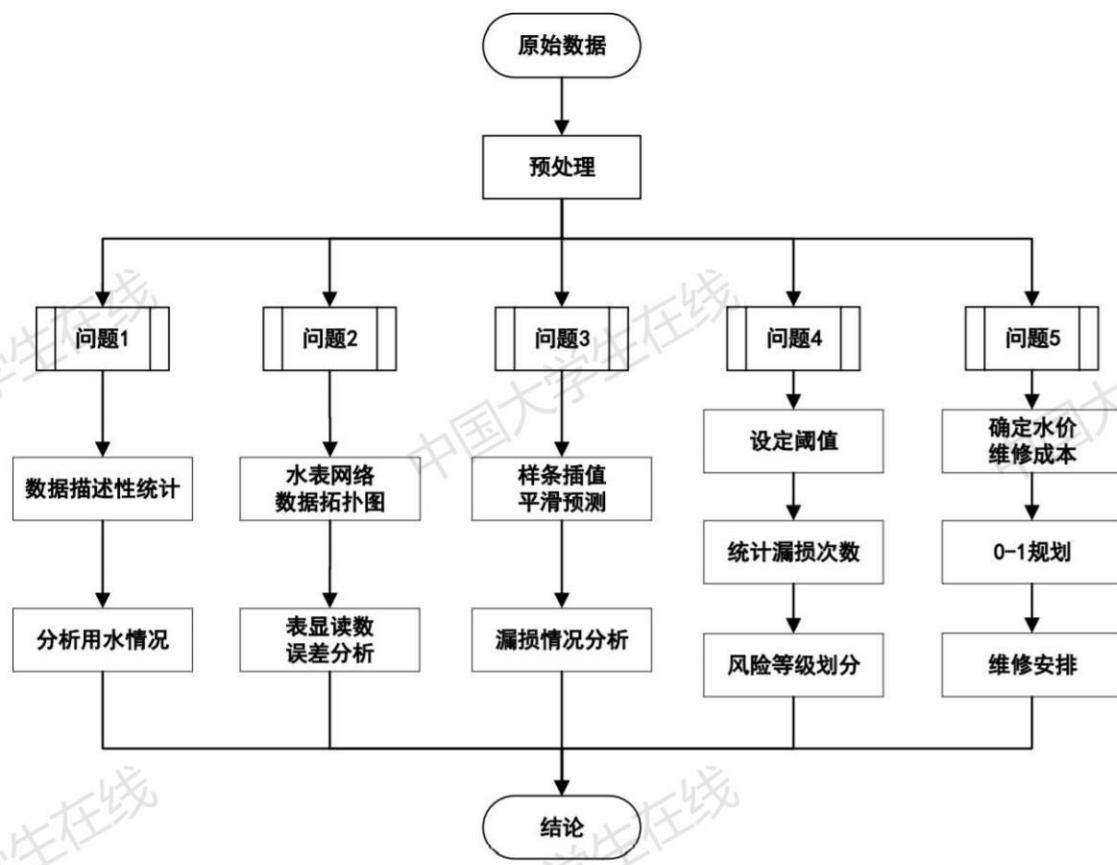


图1. 问题分析流程图

2.1 问题一分析

问题需要根据已有季度用水表(附件)、水表层次(附件)的数据分析宿舍、教学楼、办公楼、食堂的用水情况特性。为此需要有以下几个分析步骤。

1. 对数据进行描述性统计，获得缺失值、异常值，以及校园用水的总体特征。
2. 需要对数据进行缺失值、异常值的处理，并对水表进行功能区划分，建立用水情况的水表层级关系表。
3. 对宿舍功能区、教学楼功能区、办公楼功能区、食堂功能区进行用水分析。

2.2 问题二分析

问题需要根据附件中提供的水表层级关系，建立不同层级间水表的关系模型，利用已知数据分析模型误差。

水表的合理使用：

由资料^[1]电磁水表的误差率较小，而机械速度式水表可检测的最小水流量与管道直径成正比，在直径较大的骨干管道中，采用机械速度式水表难以检测弱水流，误差较大，

见下图：



图2. 高精度电磁水表和传统水表

所以：

1. 一级水表采用精度更高的电磁水表；
2. 二级至四级均采用已广泛使用的机械速度式水表，

我们采用拓扑图将水表间的层级关系可视化；

计算上级水表表显读数和下属水表表显读数之和的差值，将误差分为三类；

1. 一级水表误差按电磁水表统计，二级至四级水表按照机械速度式水表统计
2. 二级水表按照实际误差来计算，利用其表显数据与一级表数据做差得出误差；

2.3 问题三分析

问题需要根据各级水表的显示数据分析出校园水网的具体漏损情况，文献^[2]采用时间序列神经网络（NAR 神经网络）的预测方法，基于当前各水表读数时间序列，对每个读数进行预测，表显读数与预测值波动超过一定阈值的则视为该时刻发生漏水，为解决本问题提供了思路，但是根据附件的数据，各水表的年用水情况各不相同，如若进行时间序列分析，需分水表、分时间段进行稳定性、白噪声以及参数预置的讨论，所有情况过于繁琐复杂，本问题并不合适。

有鉴于此，我们参考其预测方法，对时间序列 t 时刻的值进行样条插值平滑预测，从而进行漏损估计，具体的步骤如下：

1. 首先对数据进行预处理工作。对一级水表按照超过一定误差的阈值的异常数据进行置均值的平滑处理；
2. 其次对一级水表的时间序列数据进行样条插值平滑预测；



- 利用表显数据与预测值之差进行漏损量的估计。

2.4 问题四分析

问题需要从水表的实时数据中发现供水管网的漏损位置：

我们基于问题三的样条插值平滑预测模型：

- 对二级水表及下属水表进行漏损偏差统计；
- 计算漏损率：

$$\text{漏损率} = \frac{\text{原始值} - \text{预测值}}{\text{预测值}}$$

- 对漏损率超过总体漏损率的时间点和地点判定其发生漏损现象，根据偏差统计划分风险等级，给出供水管网中，可能出现漏损的位置及概率。

2.5 问题五分析

问题五需要根据查询资料，结合前面所得到的漏水数据，做出最佳维修策略，为此应首先查阅国家相关水务数据^[3]，劳务数据以及材料费用数据^[4]，从而确定水价范围和单次维修价格的范围。

以总损失金额最小作为规则目标函数，以每天是否维修作为决策变量，建立 0-1 规划模型，从而得到维修决策。

三 模型假设

- 假设一级水表采用测量精度较高的电磁水表；
- 假设二级至四级采用均已广泛使用机械速度式水表，并有 2%-5% 的相对误差。
- 无人蓄意破坏供水管道；
- 各水表工作在理想状态下，不受强磁等外界因素干扰；
- 水质良好，无明显杂质，不对机械速度式水表产生过大影响；



四 符号说明

序号	符号	符号表示含义
1	$x_{\text{漏损}}$	各一级水表时间点漏损量
2	$x_{\text{原始}}$	一级水表的原始数据
3	$x_{\text{预测}}$	一级水表的预测漏损量
4	$p_{\text{漏损}}$	水表的漏损率
5	$b_{\text{维修}}$	单次维修费用
6	$w_{\text{维修}i}$	第 <i>i</i> 天是否维修

五 模型建立与求解

5.1 问题一模型建立与求解

5.1.1 整体数据描述

首先对数据进行描述性统计如下（部分数据，详细数据见附件）：

表1. 描述性统计

	有效 个案数	缺失 个案数	平均值	中位数	标准偏差	最小值	最大值
416X	365.00	0.00	66.15	68.75	18.47	0.00	114.00
405X	363.00	2.00	130.41	139.16	81.02	0.80	1011.74
403X	363.00	2.00	512.37	474.75	176.38	104.50	1201.25
401X	363.00	2.00	139.45	139.25	50.24	18.16	347.91
419T	338.00	27.00	2.80	2.89	0.82	0.27	5.03
418T	338.00	27.00	2.18	1.71	1.68	0.36	16.30
417T	338.00	27.00	5.96	4.83	4.11	0.00	21.58

当缺失值不影响数据统计时，不做替换操作，但是对于异常数据，例如二级水表



40337X 中 113-115 行数据大于平均值的 5 倍以上，影响整体数据的光滑程度，不利于数据分析，因此对其进行均值替换处理。根据实际情况我们认为一级水表采用的是误差率较低的水表，而二级以上水表采用的是误差率较高的传统机械式水表，最终统计出一级表显校园总用水量：328232.24，二级表显总用水量：369307，误差率 12.5%。

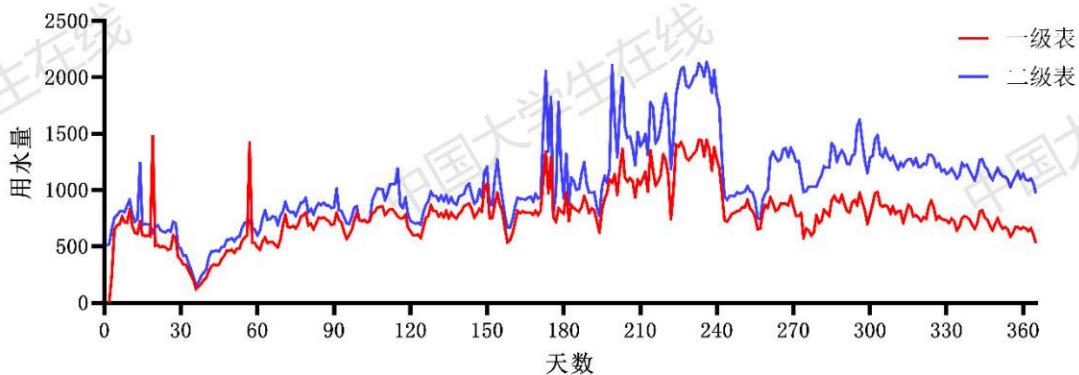


图3. 一、二级水表表显用水量（年）

根据水表层级附件，我们将二级以上的水表进行了功能区划分如下表：

表2. 功能区分类

	水表号	水表名		水表号	水表名
宿 舍	3320100600	XXX第八学生宿舍		3210100200	XXX成教院XXX分院
	183671860	XXX8舍热泵		3313800500	XXXL馆
	3320100700	XXX第七学生宿舍		3370100100	XXXL楼
	3320100200	XXX第二学生宿舍		3313200200	XXXS馆
	3320100100	XXX第一学生宿舍		3370300100	XXXK
	3320100500	XXX第五学生宿舍		3200200100	XXXK楼
	1836718629	XXX5舍热泵热水		3480400100	老七楼
	3320100400	XXX第四学生宿舍		3360300100	新留学生楼
	1836718625	XXX4舍热泵热水		3480300100	XXX老六楼
	3320100300	XXX第三学生宿舍		3480200100	XXX老五楼
	1836718633	XXX3舍热泵热水		3320100900	留学生楼（新）
	3320100800	XXX第九学生宿舍		3421300300	XXX大楼厕所西
	3620301000	养殖馆附房保卫处宿舍+		3421300200	XXX大楼厕所东
教学 楼	3210100100	XXX干训楼		3400500100	XXX教学大楼总表
	3170100600	XXX西大楼		3421300100	XXX中心大楼泵房
	3422200100	XXX东大楼		3315400200	XXXM馆
办 公 楼	3210100100	XXX干训楼		3390100400	XXX第五食堂
	3170100600	XXX西大楼		3160200300	XXX第一食堂
	3422200100	XXX东大楼		3390100500	XXX第二食堂
食 堂	3210100100	XXX干训楼		3390100400	XXX第五食堂
	3170100600	XXX西大楼		3160200300	XXX第一食堂
	3422200100	XXX东大楼		3390100500	XXX第二食堂

5.1.2 功能区用水情况（全年）分析

在 Python 中利用 pandas 库的数据处理优势对各功能区进行分类汇总，获得各功能区全年的用水情况如下图：



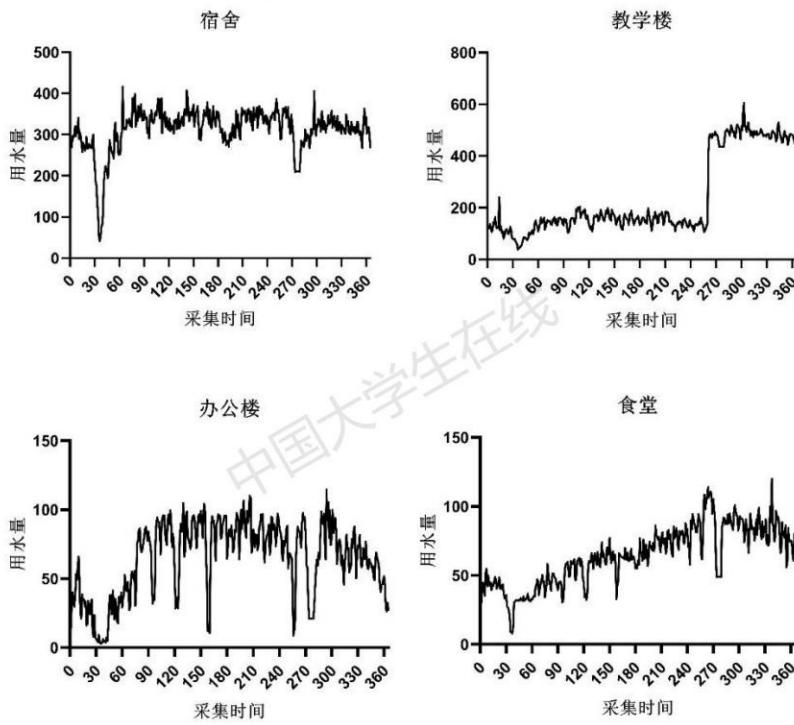


图4. 各功能区全年用水情况

由上图可以看出，各功能区用水情况有以下特点：1.整体数据都呈现周期性波动，每周用水量按照工作日和非工作日有峰谷的变化；2.各功能区在1-3月份均处于用水量较少的位置，这个时间段是农历新年所在的寒假期间；3.教学楼和食堂数据在9-10月份有明显的增加，这一时期正是新生报到的时间，因此引起用水量的变化；4.办公楼和宿舍的用水数据在寒假之外的时间段没有明显的上升下降变化。

5.1.3 功能区用水情况（每天）分析

将不同功能区按照工作日、周六日和节假日三种类型进行数据统计分析得到其每日的用水概况图如下：

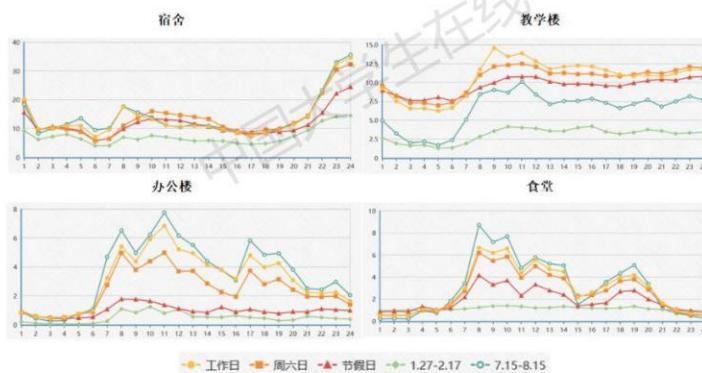


图5. 功能区按三种日期类型每日用水统计



功能区每日用水分析

从上图可以看出各功能区每日用水情况有以下特点：1.所有功能区 00:00-06:00 时间段内的用水量普遍较少；2.宿舍和教学楼的用水时间正好相反，白天学生大部分在教室上课或者学习，夜晚学生回到宿舍，此时宿舍用水处于高峰；3.办公楼用水高峰期正好是每天上午和下午的工作时间；4.食堂的用水高峰分别是 07:00/12:00/18:00 三个时间左右；5.寒假期间各功能区用水量均明显小于全年其他时间。

5.2 问题二模型建立与求解

5.2.1 水表数据关系

首先根据水表层级附件绘制出水网的拓扑结构图，显示出各水表之间的层级关系。

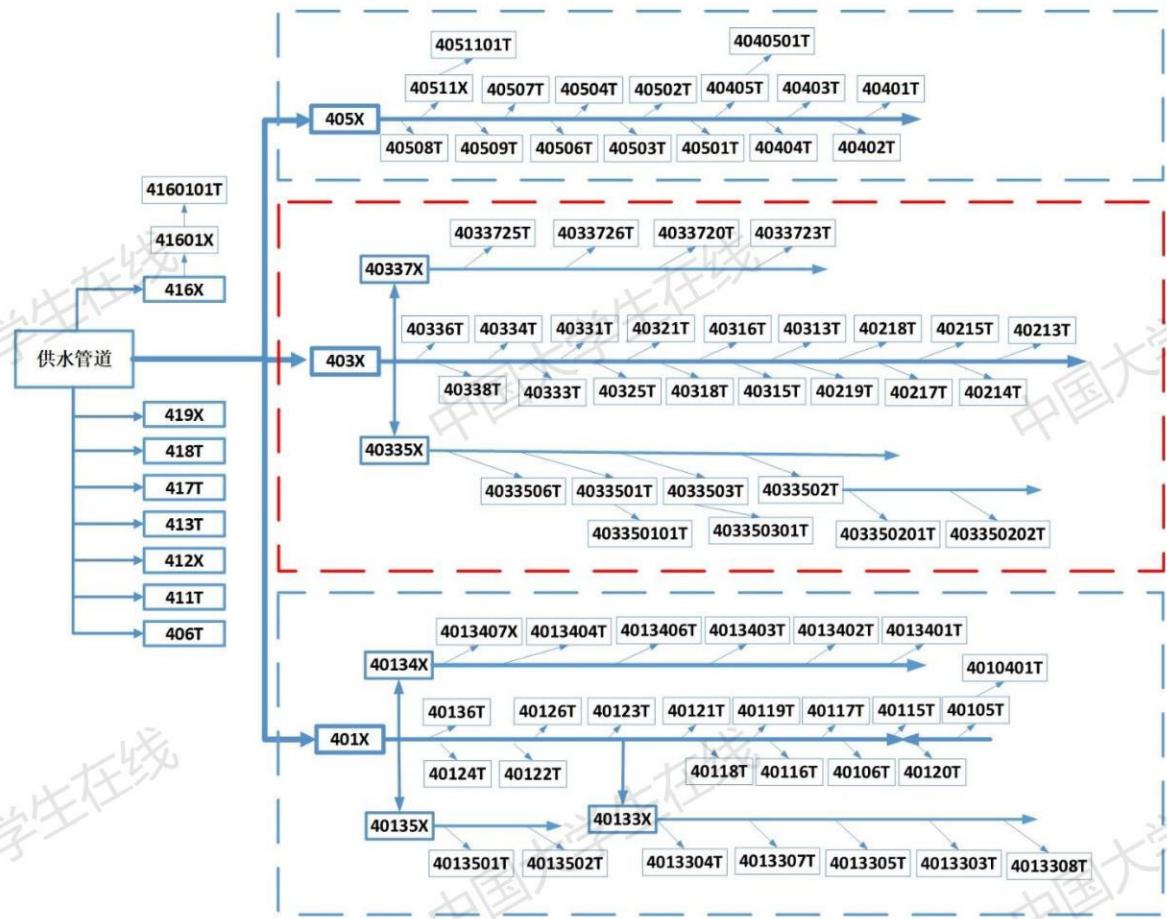


图6. 水表数据关联拓扑图

并得到他们之间的数据关系为：1.一级水表 401X、403X、405X 等表显数据应为其二级水表的数据之和；2.一级水表 416X 的显示数据包含其二级子水表的数据，但两者并不相等；3.二级水表 40335X/40337X 等所处的位置如区域 1、区域 2、区域 3、区域 4 等本身也会用水，所以二级水表数据包含其三级子水表的数据，但是两者并不相等。

5.2.2 误差分析

关联数据平均相对误差 MRE 的计算方法为：

$$RE_{ij} = \frac{|\sum x_{\text{二级}} - x_{\text{一级}}|}{x_{\text{一级}}}$$

$$MRE = \frac{1}{n} \sum RE_{ij}$$

我们将数据的误差分为三类：

首先一级水表 401X、403X、405X 与其二级子水表之间的数据理论上应该是相等的，
经过 pandas 库数据处理分析得到其平均相对误差为

表3. 一级水表数据误差

一级水表	误差	一级水表	误差	一级水表	误差
401X	11%	403X	4%	405X	11%

对于 406T、411T、412T 等一级水表，根据资料^[1]按精度最高的电磁水表来考虑，
其显示数据误差为 $\pm 0.5\% - 3\%$ ；

对于二级、三级、四级水表，根据资料^[1]我们按通用的机械水表来考虑，其显示数
据误差为：-5%-3%。

5.3 问题三模型建立和求解

5.3.1 数据预处理

首先在 pandas 中对一级水表的显示数据进行按天的汇总，数据见附件，其波动变化
图如下：



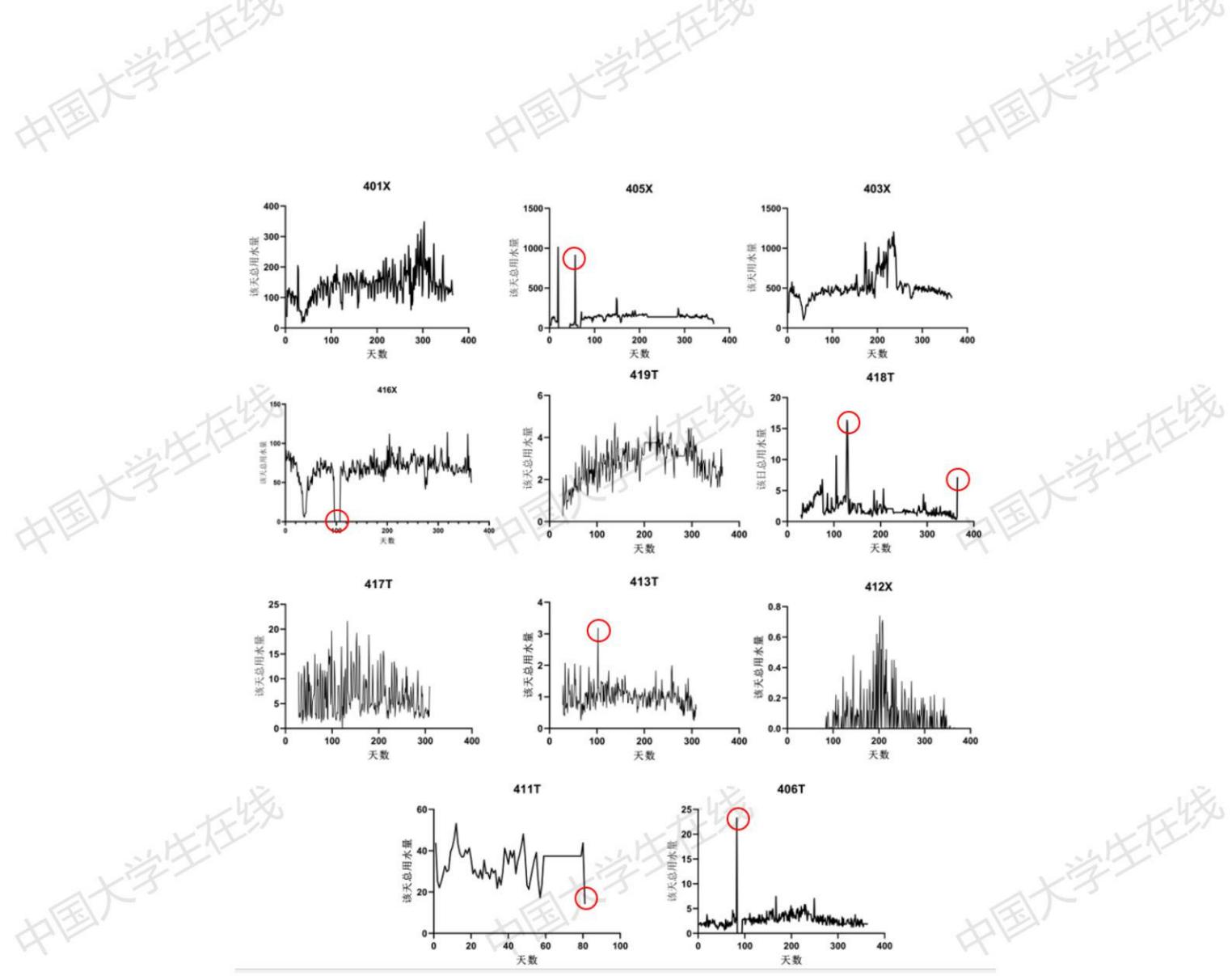


图7. 一级水表全年用水时间序列图

针对每个一级水表的数据的不同特性，我们把红圈中的异常值进行均值替换处理，从而让数据更加平滑稳定。

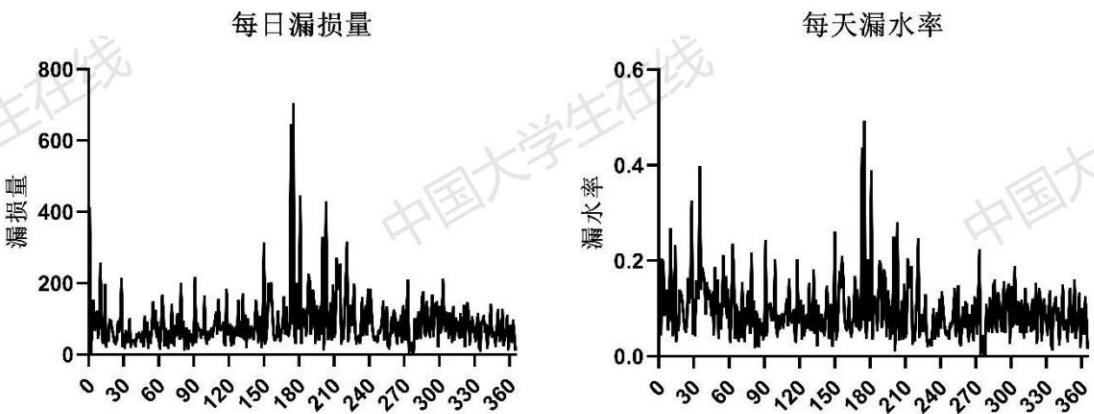


图8. 全年整体的漏损量和漏损率时间序列图

由上图可看出，全年的漏水量和漏水频率是平稳波动的，日均漏水量为 88，日均漏水率为 8.8%，在年中的 7、8 月份，漏水量与漏水率均有较大幅度的波动，需重点进行防范和检修。

5.3.2 样条插值平滑预测

样条插值的基本原理如下：

设给定区间 $[a, b]$ 的一个分划：

$$\Delta: a = x_0 < x_1 < \dots < x_n = b$$

若函数 $s(x)$ ，满足条件：

在每一个单独区间段内 $[x_{i-1}, x_i]$, ($i = 1, 2, \dots, n$)，是 k 次多项式；

$s(x)$ 及直到 $k - 1$ 阶导数在给定区间 $[a, b]$ 上连续。

则称函数 $s(x)$ 是关于分划 Δ 的 k 次多项式样条函数， x_0, x_1, \dots, x_n 称为样条节点，样条函数记作 $S_p(\Delta, k)$ 。 k 次多项式样条函数的一般形式为：

$$s_k(x) = \sum_{i=0}^k \frac{\alpha_i x^i}{i!} + \sum_{j=1}^{n-1} \frac{\beta_j}{k!} (x - x_j)_+^k$$

其中 α_i ($i = 0, 1, \dots, k$) 和 β_j ($i = 1, 2, \dots, n - 1$) 均为任意常数

这里我们采用三次样条插值对一级水表的每个时间序列数据进行预测。

在数据 x_{ij} 附近向左向右各选 5 个数据共 10 个，即 $[x_{i,j-5} \dots x_{i,j-1} x_{i,j+1} \dots x_{i,j+5}]$ ，通过这十个数据对 x_{ij} 进行插值预测，得到的预测数据（具体见附件）。

5.3.3 漏损情况分析

各一级水表时间点漏损数据 $x_{\text{漏损}}$ 计算方法为：

$$x_{\text{漏损}} = x_{\text{原始}} - x_{\text{预测}}$$

漏损率 $p_{\text{漏损}}$ 为：

$$p_{\text{漏损}} = \frac{x_{\text{漏损}}}{x_{\text{预测}}}$$

通过借助于 Python 数据处理，得到各一级水表的具体数据，文献 [2] 显示我国城市漏水率一般在 15% 以上，北方地区甚至达到了 40%，在此基础上统计得出各水表的漏水次数和漏水量为（具体数据见附件）：



表4. 各水表漏损量和漏损率

水表号	412X	40336T	40403T	40105T	40502T	40334T	4010401T	403350202T	40126T	40118T
漏水量	34.8%	25.8%	25.2%	20.2%	20.0%	19.2%	18.9%	18.6%	17.9%	17.7%
漏水量	6.33	76.22	163.90	10.53	228.90	0.97	292.14	10.87	0.92	45.43
水表号	40124T	417T	40316T	4013403T	40333T	40318T	4013407X	40121T	40331T	40120T
漏水量	17.2%	16.6%	16.6%	16.0%	15.5%	15.5%	15.2%	14.9%	14.6%	12.8%
漏水量	14.15	335.40	89.36	474.39	26.09	10.32	347.28	27.74	13.54	17.26
水表号	40119T	4160101T	40136T	40214T	40115T	40321T	40135X	40509T	4013404T	413T
漏水量	12.6%	12.2%	11.9%	11.8%	11.7%	10.4%	10.3%	9.9%	9.7%	9.5%
漏水量	90.18	51.17	25.28	1115.73	89.82	440.32	106.94	33.17	91.73	26.69
水表号	4013501T	4051101T	4013307T	40511X	4013401T	40106T	4013308T	40508T	40133X	40405T
漏水量	9.4%	9.2%	9.2%	9.1%	9.1%	8.8%	8.5%	8.3%	8.2%	8.2%
漏水量	39.86	148.22	285.15	140.48	99.98	6.98	441.09	161.01	1308.17	733.24
水表号	4013305T	406T	4013406T	41601X	40313T	418T	40116T	40402T	40401T	40123T
漏水量	7.9%	7.8%	7.6%	7.5%	7.4%	7.4%	6.9%	6.8%	6.7%	6.7%
漏水量	88.29	75.24	43.03	150.84	233.82	52.82	398.88	124.85	65.72	39.73
水表号	4013303T	40122T	40134X	4033726T	40338T	40504T	401X	4033506T	4040501T	419T
漏水量	6.6%	6.5%	6.4%	6.3%	6.1%	6.1%	5.7%	5.6%	5.3%	5.0%
漏水量	684.46	9.51	1032.60	645.72	184.70	154.50	2872.94	106.29	89.74	47.77
水表号	40218T	40507T	40315T	4013502T	4033725T	403350301T	403350101T	416X	40117T	4033723T
漏水量	5.0%	4.8%	4.7%	4.3%	4.1%	4.1%	4.0%	3.8%	3.7%	3.7%
漏水量	95.09	113.88	203.48	10.19	115.34	235.91	115.72	907.76	191.78	74.13
水表号	40404T	403350201T	40219T	40501T	4013402T	40337X	4033720T	40217T	40325T	411T
漏水量	3.7%	3.7%	3.7%	3.5%	3.4%	3.4%	3.0%	3.0%	2.9%	2.9%
漏水量	273.55	212.11	87.40	429.83	7.90	3006.97	463.68	184.87	544.23	490.07
水表号	4033502T	40503T	405X	40335X	40215T	403X	4033501T	40213T	4033503T	4013304T
漏水量	2.8%	2.7%	2.7%	2.7%	2.6%	2.5%	2.5%	2.3%	1.9%	1.2%
漏水量	524.61	539.53	1227.30	1525.04	343.88	4724.68	282.11	153.52	353.49	0.33
										88.49

上表显示一级水表的漏损量估计为 10766，二级以上水表的漏损量估计为：21488，总漏损率估计为：9.8%，

同时数据显示在一级水表中，403X 处的漏损量是最多的，412X 处的漏水率是最高的，平均漏损量为 978；在二级以上的水表中，40337X 的漏损量是最多的，40336T 处的漏损率是最高的，平均漏损量为 268。

5.4 问题四模型建立与求解

基于第三问的漏损率数据，我们设置预报发生漏损的漏损率阈值为 9.8%，

统计得出各水表的漏损频次排序如图：

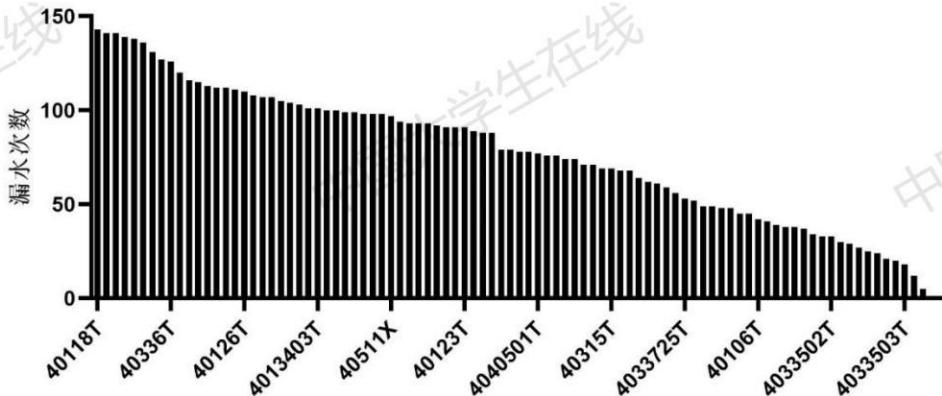


图9. 水表漏水频次



从上图可以看出 40118T/40121T/40331T 的漏损次数排名前三，我们按照每隔 30 次一个漏损发生预报等级，对各水表位置进行漏损等级预报如下图：

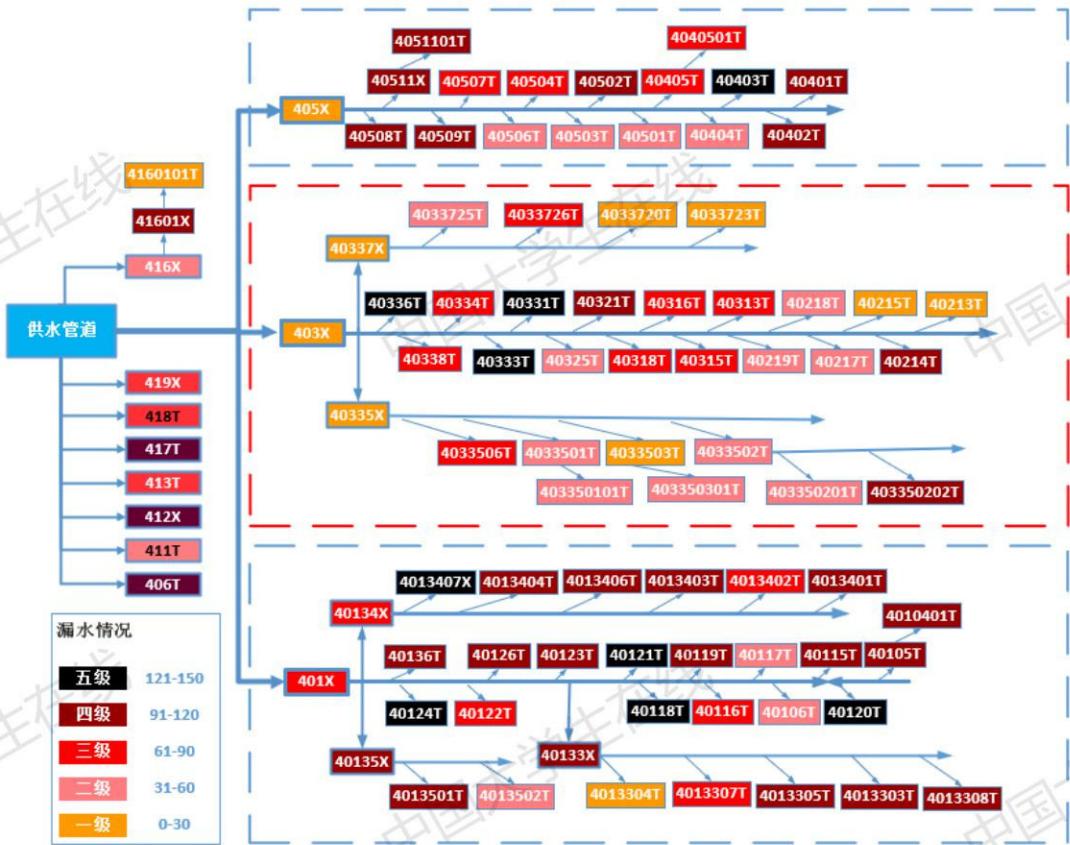


图10. 漏损预报等级

根据漏损预报等级显示：40118T/40121T/40331T 等 9 处水表位置预警等级最高，这些水表点应该重点注意。4010401T/4013501T/40135X 等 32 处预警等级次之需要提高警惕。

5.5 问题五模型建立与求解

5.5.1 0-1 规划模型建立

依据全国水价网数据资料^[3]，公共事业单位用水均价为 5.5-6.5 元/吨，另据资料^[4]显示供水地下暗管价格为 300 元/m，单次维修人数 3 人，人工费为 200 元/人，单次维修总费用为 $b_{\text{维修}} = 500 \text{ 元}-1500 \text{ 元}$ ，由第三、四问可知全年每天的漏水量是不固定的，因此以总损失金额最少建立 0-1 规划模型：

$$\min \sum_{i=1}^{365} (b_{\text{维修}} w_{\text{维修}_i} + b_{\text{漏损}} (1 - w_{\text{维修}_i}))$$

$$s.t. \quad w_{\text{维修}_i} = 0, 1$$



5.5.2 模型求解

在 matlab 中运用 intlinprog 求解线性 0-1 规划得到维修次数和损失金额关于水价和维修成本的三维曲面图如下：

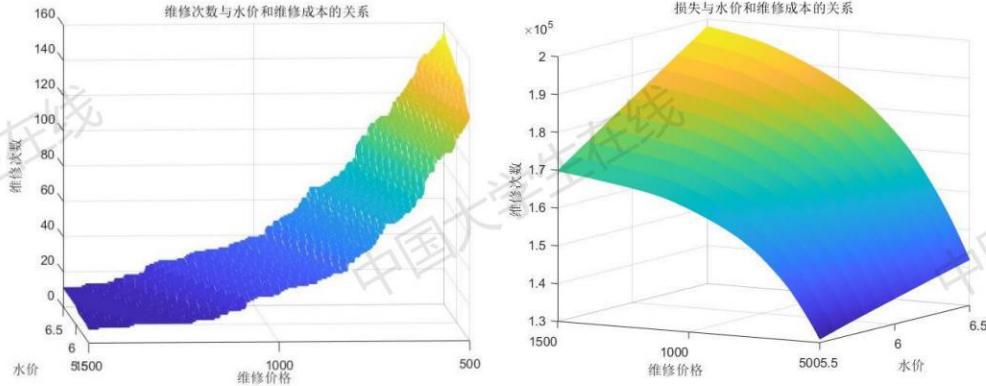


图11. 维修次数和损失金额与水价和维修成本的关系图

可以看出当水价为 5.5 元/吨，维修成本为 500 元/次时总的损失金额是最少的；当水价为 6.5 元/吨，维修成本为 1500 元/次时，总的维修次数是最少的。

我们取水价为 6 元/吨，维修成本为 1000 元/次，具体维修安排如下：

01/01	01/02	01/03	01/04	01/05	01/06	01/07	01/08	01/09	01/10	01/11	01/12	01/13	01/14	01/15	01/16	01/17	01/18	01/19
01/20	01/21	01/22	01/23	01/24	01/25	01/26	01/27	01/28	01/29	01/30	01/31	02/01	02/02	02/03	02/04	02/05	02/06	02/07
02/08	02/09	02/10	02/11	02/12	02/13	02/14	02/15	02/16	02/17	02/18	02/19	02/20	02/21	02/22	02/23	02/24	02/25	02/26
02/27	02/28	03/01	03/02	03/03	03/04	03/05	03/06	03/07	03/08	03/09	03/10	03/11	03/12	03/13	03/14	03/15	03/16	03/17
03/18	03/19	03/20	03/21	03/22	03/23	03/24	03/25	03/26	03/27	03/28	03/29	03/30	03/31	04/01	04/02	04/03	04/04	04/05
04/06	04/07	04/08	04/09	04/10	04/11	04/12	04/13	04/14	04/15	04/16	04/17	04/18	04/19	04/20	04/21	04/22	04/23	04/24
04/25	04/26	04/27	04/28	04/29	04/30	05/01	05/02	05/03	05/04	05/05	05/06	05/07	05/08	05/09	05/10	05/11	05/12	05/13
05/14	05/15	05/16	05/17	05/18	05/19	05/20	05/21	05/22	05/23	05/24	05/25	05/26	05/27	05/28	05/29	05/30	05/31	06/01
06/02	06/03	06/04	06/05	06/06	06/07	06/08	06/09	06/10	06/11	06/12	06/13	06/14	06/15	06/16	06/17	06/18	06/19	06/20
06/21	06/22	06/23	06/24	06/25	06/26	06/27	06/28	06/29	06/30	07/01	07/02	07/03	07/04	07/05	07/06	07/07	07/08	07/09
07/10	07/11	07/12	07/13	07/14	07/15	07/16	07/17	07/18	07/19	07/20	07/21	07/22	07/23	07/24	07/25	07/26	07/27	07/28
07/29	07/30	07/31	08/01	08/02	08/03	08/04	08/05	08/06	08/07	08/08	08/09	08/10	08/11	08/12	08/13	08/14	08/15	08/16
08/17	08/18	08/19	08/20	08/21	08/22	08/23	08/24	08/25	08/26	08/27	08/28	08/29	08/30	08/31	09/01	09/02	09/03	09/04
09/05	09/06	09/07	09/08	09/09	09/10	09/11	09/12	09/13	09/14	09/15	09/16	09/17	09/18	09/19	09/20	09/21	09/22	09/23
09/24	09/25	09/26	09/27	09/28	09/29	09/30	10/01	10/02	10/03	10/04	10/05	10/06	10/07	10/08	10/09	10/10	10/11	10/12
10/13	10/14	10/15	10/16	10/17	10/18	10/19	10/20	10/21	10/22	10/23	10/24	10/25	10/26	10/27	10/28	10/29	10/30	10/31
11/01	11/02	11/03	11/04	11/05	11/06	11/07	11/08	11/09	11/10	11/11	11/12	11/13	11/14	11/15	11/16	11/17	11/18	11/19
11/20	11/21	11/22	11/23	11/24	11/25	11/26	11/27	11/28	11/29	11/30	12/01	12/02	12/03	12/04	12/05	12/06	12/07	12/08
12/09	12/10	12/11	12/12	12/13	12/14	12/15	12/16	12/17	12/18	12/19	12/20	12/21	12/22	12/23	12/24	12/25	12/26	12/27
12/28	12/29	12/30	12/31															

图12. 维修安排（蓝色代表不维修，红色表示维修）

从上表结合模型求解结果可以得出，全年总共维修 33 次，具体的安排如表所示，全年中 6-8 月份维修频率较多，其原因可能是因为夏季雨水增多，造成地下水管的腐蚀，



产生管网沉积物，引起供水管破裂，因此 6-8 月份漏水量增大，需检修次数相应增加。

与之相反的是在 11-12 月份的维修次数相对较少，这可能是由于降水量减少，供水管路运行良好，漏水量降低的原因。

总的损失金额为 174,630 元

六 模型检验

6.1 问题三和问题四的模型灵敏度检验

针我们将时序点向前向后的差值点数由前后各 5 个改为前后 3 个和前后 7 个，分别通过样条插值得到其漏损量和漏损率比较见下表：

表5. 插值点数量的灵敏度检验

向前向后插值点数	3 个点	5 个点	7 个点
漏损量	32255	32255	32255
漏损率	9.8%	9.8%	9.8%

从上表可以看出当改变插值点的个数时，对于漏损量和漏损率均没有的改变，一方面是因为样条插值预测的特性，另一方面也说明我们的模型健壮性较好。

6.2 问题五的模型灵敏度检验

我们将水价进一步拓展为 5 元/吨-7 元/吨，维修成本进一步拓展为 200 元/次-2000 元/次，可以看出总维修次数随着水价和维修成本均单调递减，总损失金额随水价和维修成本单调递增。

七 模型改进

7.1 问题三和问题四的模型改进

我们可以参考文献 [2] 中的时间序列神经网络对数据进行平滑预测，得到更精确的预测值作为计算漏损量和漏损率的参考数据，并以此进行漏损检验分析，分析发生漏损的位置。

7.2 问题五模型改进

参考文献 5 发生漏损的点位数量在时间 $[T, T + \tau]$ 内服从泊松分布，我们把全年的漏损数据看成是一个完整的泊松过程，这个过程也是一个马尔科夫链，用随机过程分析方法来对维修决策进行重新规划，可以得到更好的维修安排决策。

此外在考虑约束条件时，也可以加入维修均衡度等条件。



八 模型评价

8.1 优点

- 对于问题给出的数据量较大、多种数据格式类型共存、缺失值异常值较多的数据特点，我们采用 Python 中的 pandas 库，快捷方便的对数据进行了挖掘处理；
- 我们利用样条插值的光滑预测特性，对漏损情况进行分析，得到较好的漏损位置预测结果；
- 我们利用结合实际的水价和维修价格不断波动的特点，建立了动态的 0-1 规划模型，较好的给出了维修决策；

综合以上，我们的模型不但可以推广到其他的供水网络数据分析中，还可以进一步应用到供热数据分析、网络流量数据分析、高速路网数据分析当中，具有较好的推广价值。

8.2 缺点

- 预测数据模型可以考虑更多的因素，获得更加精确的预测值，以此来建立模型；
- 维修规划时可以考虑维修的均衡度等因素，进一步优化模型。

九 参考文献

- [1] “你好，查水表，” [联机]. Available: <https://www.bilibili.com/video/av77960277/>.
- [2] 朱乃富,《基于数据驱动的供水管网独立计量分区漏损检测和定位方法研究》: 硕士学位论文, 2019 年 2 月.
- [3] “中国水网水价,” [联机]. Available: <http://www.h2o-china.com/price/>.
- [4] “双高筋增强聚乙烯 (HDPE) 缠绕管 , ” [联机]. Available: http://www.jt1688.cn/photo_show.aspx?id=206.
- [5] 卓严报,《基于支持向量机的城市给水管网故障诊断研究》: 硕士学位论文, 2013 年 6 月.
- [6] 谢金星、薛毅,《优化建模与 LINDO/LINGO 软件》, 清华大学出版社, 2006 年.
- [7] 吴小刚、张土乔,“《城市给水网系统的故障风险评价决策技术》,”《自然灾害学报》, 2006 年 4 月.
- [8] 王凌,《维修决策模型和方法的理论与应用研究》, 博士学位论文, 2007 年 6 月.
- [9] 司守奎、孙兆亮,《数学建模算法与应用》, 第二版 编辑, 国防工业出版社, 2011 年.
- [10] 苗志宏、马金强, MATLAB 面向对象程序设计, 电子工业出版社, 2014 年.
- [11] 吕自荟,《水泵系统状态监测与故障诊断》, 硕士学位论文, 2014 年 6 月.



- [12] 梁建文、肖笛、赵新华、张宏伟, “《给水网故障实时诊断方法》,”《水利学报》, 2001 年 12 月.
- [13] 梁建文、肖笛、张宏伟、赵新华, “《给水管网故障实时诊断方法 (II)》,”《自然科学进展》, 2002 年 9 月.
- [14] 李花、梁辉、于宁, Excel 高级数据处理及分析, 电子工业出版社, 2015 年.
- [15] 李航, 《统计学习方法》, 清华大学出版社, 2013 年.
- [16] 郭明乐、黄旭东, 《概率论与数理统计》, 中国科学技术大学出版社, 2011 年.



十 附 录

10.1 支撑材料目录

问题一	一二级水表表显用水量(年).xls 各功能区全年用水情况.xlsx 功能区按三种日期类型每日用水统计.xls Code_1.py
问题二	一级水表数据误差.xlsx Code_2.py
问题三	一级水表全年用水时间序列图.xlsx 全年整体的漏损量和漏损率时间序列图.xlsx 各水表漏水量和漏损率.xls Code_3.py
问题四	水表漏水频次.xls Code_4.py
问题	cs.xls sunshi.xlsx suihua.m

10.2 代码、操作及结果

1. 问题一和问题二

```
import numpy as np
import pandas as pd
import os
import xlrd
from pandas import to_datetime
import matplotlib.pyplot as plt
import xlwt
import datetime,time
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False

inputdir=r'D:\主桌面\E题\水表数据'
df_empty=pd.DataFrame(columns=['水表名','水表号','采集时间','上次读数','当前读数','用量'])
for parents, dirnames, filenames in os.walk(inputdir):
    for filename in filenames:
        df=pd.read_excel(os.path.join(parents,filename))
        df_empty=df_empty.append(df,ignore_index=True)
```



```

data_index=pd.read_excel('D:\\主桌面\\E 题\\附件_水表层级(1).xlsx')
data_index.dropna(axis=0,thresh=7,inplace=True)
data=df_empty.copy()

data_all=pd.merge(data,data_index,how="outer")
data_all.to_csv('D:\\主桌面\\E 题\\功能区关联表.csv')

# 用量是否为表显数据差
data_all['表显差值']=data_all['当前读数']-data_all['上次读数']
data_all['差值']=data_all['表显差值']-data_all['用量']
data_all['差值'].abs().sum()
data_all_tem=data_all[data_all['差值'].abs()==0.01]
# 各水表总用水量统计
sbm_sl=data_all['用量'].groupby(by=data_all['序号']).sum()
# 12 个工作区总用水量
gnq_sl=data_all['用量'].groupby(by=data_all['功能分类']).sum()

data_all['采集时间']=to_datetime(data_all.采集时间,format='%Y/%m/%d %H:%M:%S')
gnq_sl.to_excel('D:\\主桌面\\E 题\\功能区总水量.xls')

# 12 个功能区按天统计用水量
gnq_day=pd.DataFrame(columns=['总表','宿舍','教学楼','办公楼','食堂','科研楼','运动场所','后勤保卫','校医院','澡堂','酒店','图书馆','其他'])
for i in np.arange(13):
    temp=data_all[data_all['功能分类']==i]
    temp.set_index('采集时间',inplace=True)
    gnq_day.iloc[:,i-1]=temp['用量'].resample('D').sum()
gnq_day.to_excel('D:\\主桌面\\E 题\\功能区每天水量.xls')

# 12 个功能区按月度统计用水量
gnq_month=pd.DataFrame(columns=['总表','宿舍','教学楼','办公楼','食堂','科研楼','运动场所','后勤保卫','校医院','澡堂','酒店','图书馆','其他'])
for i in np.arange(13):
    temp=data_all[data_all['功能分类']==i]
    temp.set_index('采集时间',inplace=True)
    gnq_month.iloc[:,i-1]=temp['用量'].resample('M').sum()
gnq_month.to_excel('D:\\主桌面\\E 题\\功能区每月水量.xls')

# 12 个功能区按小时统计用水量
gnq_hour=pd.DataFrame(columns=['总表','宿舍','教学楼','办公楼','食堂','科研楼','运动场所','后勤保卫','校医院','澡堂','酒店','图书馆','其他'])
for i in np.arange(13):

```



```

temp=data_all[data_all['功能分类']==i]
temp.set_index('采集时间',inplace=True)
gnq_hour.iloc[:,i-1]=temp['用量'].resample('H').sum()
gnq_hour.to_excel('D:\主桌面\E题\功能区每小时.xls')

gnq_15min=pd.DataFrame(columns=['总表','宿舍','教学楼','办公楼','食堂','科研楼','运动场所','后勤保卫','校医院','澡堂','酒店','图书馆','其他'])
for i in np.arange(13):
    temp=data_all[data_all['功能分类']==i]
    temp.set_index('采集时间',inplace=True)
    gnq_15min.iloc[:,i-1]=temp['用量'].resample('15min').sum()
gnq_15min.to_excel('D:\主桌面\E题\功能区 15min.xls')

gnq_ht=gnq_15min.loc['2019-04-01T00:00:00.000000000':'2019-04-30T23:45:00.000000000']
plt.figure()
for i in np.arange(4):
    plt.subplot(4,1,i+1)
    plt.subplots_adjust(hspace=1)
    temp=gnq_ht.iloc[:,i+1]
    plt.plot(np.arange(len(temp[temp<30])),temp[temp<30])
    plt.xlabel(gnq_ht.columns[i+1])
    plt.ylabel('用水量(方)')
plt.savefig('D:\主桌面\E题\用水量 4月.png')
plt.show()

gnq_ht10=gnq_15min.loc['2019-10-01T00:00:00.000000000':'2019-10-31T23:45:00.000000000']
plt.figure()
for i in np.arange(4):
    plt.subplot(4,1,i+1)
    plt.subplots_adjust(hspace=1)
    temp=gnq_ht10.iloc[:,i+1]
    plt.plot(np.arange(len(temp[temp<30])),temp[temp<30])
    plt.xlabel(gnq_ht10.columns[i+1])
    plt.ylabel('用水量(方)')
plt.savefig('D:\工作\2020 数学建模\E题\用水量 10月.png')
plt.show()

gnq_htsj=gnq_15min.loc['2019-07-15T00:00:00.000000000':'2019-08-15T23:45:00.000000000']
plt.figure()

```



```

for i in np.arange(4):
    plt.subplot(4,1,i+1)
    plt.subplots_adjust(hspace=1)
    temp=gnq_htsj.iloc[:,i+1]
    plt.plot(np.arange(len(temp[temp<30])),temp[temp<30])
    plt.xlabel(gnq_htsj.columns[i+1])
    plt.ylabel('用水量(方)')
plt.savefig('D:\\工作\\2020 数学建模\\E 题\\用水量暑假.png')
plt.show()

gnq_ht['时间']=gnq_ht.index.tolist()
gnq_ht['时间']=gnq_ht['时间'].astype('str')
gnq_ht_sd=pd.DataFrame(columns=['02:00','06:00','10:00','14:00','18:00','22:00'])
for i in gnq_ht_sd.columns:
    temp=gnq_ht[gnq_ht['时间'].str.contains(i)]
    temp=temp.set_index(np.arange(len(temp['时间'])))
    gnq_ht_sd.loc[:,i]=temp.iloc[:,1]
    gnq_ht_sd.set_index(np.arange(len(temp['时间'])))

plt.figure()
for i in np.arange(len(gnq_ht_sd.columns)):
    plt.subplot(6,1,i+1)
    plt.subplots_adjust(top=1.5,bottom=0)
    plt.plot(gnq_ht_sd.index,gnq_ht_sd.iloc[:,i])
    plt.xlabel(gnq_ht_sd.columns[i])
    plt.ylabel('用水量 (方)')
plt.savefig('D:\\工作\\2020 数学建模\\E 题\\时间点用水量.png')
plt.show()

all_sum=pd.DataFrame(columns=np.arange(91)+1)
for i in np.arange(91)+1:
    temp=data_all[data_all['序号']==i]
    temp.set_index('采集时间',inplace=True)
    all_sum.iloc[:,i-1]=temp['用量'].resample('D').sum()

print(all_sum.iloc[9].sum(),all_sum.iloc[9].sum()-all_sum.iloc[9,20])
all_sum.to_excel('D:\\工作\\2020 数学建模\\E 题\\all_sum.xls')

```



```

# 分级统计水量
# 一级水表
sbm_1=data_all[data_all['一级表计编码'].notnull()]
sbm_1_yl=pd.DataFrame(columns=sbm_1['一级表计编码'].unique())
sbm_1_yl15=pd.DataFrame(columns=sbm_1['一级表计编码'].unique())
for i in sbm_1['一级表计编码'].unique():
    temp=sbm_1[sbm_1['一级表计编码']==i]
    temp.set_index('采集时间',inplace=True)
    sbm_1_yl[i]=temp['用量'].resample('D').sum()
    sbm_1_yl15[i]=temp['用量'].resample('15min').sum()
# 一级总用水量
for i in sbm_1['一级表计编码'].unique():
    temp=sbm_1_yl[i]
    temp[temp>5*temp.mean()]=temp.mean()
    sbm_1_yl[i]=temp
sbm_1_yl=sbm_1_yl.fillna(0)
zysl_yj=(sbm_1_yl.sum(axis=0)).sum()

# 二级水表
sbm_2=data_all[data_all['二级表计编码'].notnull()]
sbm_2_yl=pd.DataFrame(columns=sbm_2['二级表计编码'].unique())
sbm_2_yl15=pd.DataFrame(columns=sbm_2['二级表计编码'].unique())
for i in sbm_2['二级表计编码'].unique():
    temp=sbm_2[sbm_2['二级表计编码']==i]
    temp.set_index('采集时间',inplace=True)
    sbm_2_yl[i]=temp['用量'].resample('D').sum()
    sbm_2_yl15[i]=temp['用量'].resample('15min').sum()

# 二级表显用水量
for i in sbm_2['二级表计编码'].unique():
    temp=sbm_2_yl[i]
    temp[temp>5*temp.mean()]=temp.mean()
    sbm_2_yl[i]=temp
sbm_2_yl=sbm_2_yl.fillna(0)
zysl_ej=(sbm_2_yl.sum(axis=0)).sum()

# 三级水表
sbm_3=data_all[data_all['三级表计编码'].notnull()]
sbm_3_yl=pd.DataFrame(columns=sbm_3['三级表计编码'].unique())
sbm_3_yl15=pd.DataFrame(columns=sbm_3['三级表计编码'].unique())
for i in sbm_3['三级表计编码'].unique():

```



```

temp=sbm_3[sbm_3['三级表计编码']==i]
temp.set_index('采集时间',inplace=True)
sbm_3_yl[i]=temp['用量'].resample('D').sum()
sbm_3_yl15[i] = temp['用量'].resample('15min').sum()

# 四级水表
sbm_4=data_all[data_all['四级表计编码'].notnull()]
sbm_4_yl=pd.DataFrame(columns=sbm_4['四级表计编码'].unique())
sbm_4_yl15=pd.DataFrame(columns=sbm_4['四级表计编码'].unique())
for i in sbm_4['四级表计编码'].unique():
    temp=sbm_4[sbm_4['四级表计编码']==i]
    temp.set_index('采集时间',inplace=True)
    sbm_4_yl[i]=temp['用量'].resample('D').sum()
    sbm_4_yl15[i] = temp['用量'].resample('15min').sum()

```

```

sbm_yl=pd.concat([sbm_1_yl,sbm_2_yl],axis=1)
sbm_yl=pd.concat([sbm_yl,sbm_3_yl],axis=1)
sbm_yl=pd.concat([sbm_yl,sbm_4_yl],axis=1)

sbm_yl15=pd.concat([sbm_2_yl15,sbm_3_yl15],axis=1)
sbm_yl15=pd.concat([sbm_yl15,sbm_4_yl15],axis=1)

sbm_yl.to_excel('D:\\工作\\2020 数学建模\\E 题\\各级用量.xls')

```

```

for i in np.arange(11):
    plt.figure()
    temp=sbm_yl.iloc[:,i]
    temp=temp[temp.notnull()]
    # temp[temp>100]=temp.mean()
    plt.plot(np.arange(len(temp)),temp)
    plt.xlabel(sbm_yl.columns[i])
    plt.ylabel('用水量')
    plt.savefig('D:\\工作\\2020 数学建模\\E 题\\%d.png'%(i+1))

```

2. 问题三和问题四

```

# 数据插值 以每个点向前向后各 5 个点，共十个点作为插值数据进行预测
# data_cz=pd.DataFrame([])
# sbm_yl_pro=pd.DataFrame([])
# for i in np.arange(91):
#     temp=pd.DataFrame(sbm_yl.iloc[:,i].dropna(),columns=[sbm_yl.columns[i]])

```



```

#
temp[sbm_yl.columns[i]][temp[sbm_yl.columns[i]]>5*temp[sbm_yl.columns[i]].mean()]
]=temp[sbm_yl.columns[i]].mean()
#      sbm_yl_pro = sbm_yl_pro.join(temp, how='outer')
#      for j in np.arange(len(temp)):
#          if j-5<0:
#              temp1=temp.iloc[0:j+6,:].copy()
#              temp1.iloc[j,:]=np.nan
#              temp1=temp1.interpolate(method='pchip',limit_direction='both')
#              temp.iloc[j,:]=temp1.iloc[j,:]
#          elif j+5>len(temp)-1:
#              temp1=temp[j-5:len(temp)].copy()
#              temp1.iloc[5,:]=np.nan
#              temp1=temp1.interpolate(method='pchip',limit_direction='both')
#              temp.iloc[j,:]=temp1.iloc[5,:]
#          else:
#              temp1 = temp.iloc[j - 5:j + 6,:].copy()
#              temp1.iloc[5,:]= np.nan
#              temp1=temp1.interpolate(method='pchip',limit_direction='both')
#              temp.iloc[j,:]= temp1.iloc[5,:]
#      data_cz=data_cz.join(temp,how='outer')

```

```

# 前后各 3 各数据进行插值
data_cz=pd.DataFrame([])
sbm_yl_pro=pd.DataFrame([])
for i in np.arange(91):
    temp=pd.DataFrame(sbm_yl.iloc[:,i].dropna(),columns=[sbm_yl.columns[i]])

    temp[sbm_yl.columns[i]][temp[sbm_yl.columns[i]]>5*temp[sbm_yl.columns[i]].mean()]
]=temp[sbm_yl.columns[i]].mean()
    sbm_yl_pro = sbm_yl_pro.join(temp, how='outer')
    for j in np.arange(len(temp)):
        if j-3<0:
            temp1=temp.iloc[0:j+4,:].copy()
            temp1.iloc[j,:]=np.nan
            temp1=temp1.interpolate(method='pchip',limit_direction='both')
            temp.iloc[j,:]=temp1.iloc[j,:]
        elif j+3>len(temp)-1:
            temp1=temp[j-3:len(temp)].copy()
            temp1.iloc[3,:]=np.nan
            temp1=temp1.interpolate(method='pchip',limit_direction='both')
            temp.iloc[j,:]=temp1.iloc[3,:]

```



```

else:
    temp1 = temp.iloc[j - 3:j + 4,:].copy()
    temp1.iloc[3,:] = np.nan
    temp1=temp1.interpolate(method='pchip',limit_direction='both')
    temp.iloc[j,:]=temp1.iloc[3,:]
    data_cz=data_cz.join(temp,how='outer')
data_cz.to_excel('D:\\主桌面\\E 题\\插值结果.xls')

# 前后各 7 各数据进行插值
# data_cz=pd.DataFrame([])
# sbm_yl_pro=pd.DataFrame([])
# for i in np.arange(91):
#     temp=pd.DataFrame(sbm_yl.iloc[:,i].dropna(),columns=[sbm_yl.columns[i]])
#
#     temp[sbm_yl.columns[i]][temp[sbm_yl.columns[i]]>5*temp[sbm_yl.columns[i]].mean()]=temp[sbm_yl.columns[i]].mean()
#     sbm_yl_pro = sbm_yl_pro.join(temp, how='outer')
#     for j in np.arange(len(temp)):
#         if j-7<0:
#             temp1=temp.iloc[0:j+8,:].copy()
#             temp1.iloc[j,:]=np.nan
#             temp1=temp1.interpolate(method='pchip',limit_direction='both')
#             temp.iloc[j,:]=temp1.iloc[j,:]
#         elif j+7>len(temp)-1:
#             temp1=temp[j-7:len(temp)].copy()
#             temp1.iloc[7,:]=np.nan
#             temp1=temp1.interpolate(method='pchip',limit_direction='both')
#             temp.iloc[j,:]=temp1.iloc[7,:]
#         else:
#             temp1 = temp.iloc[j - 7:j + 8,:].copy()
#             temp1.iloc[7,:]= np.nan
#             temp1=temp1.interpolate(method='pchip',limit_direction='both')
#             temp.iloc[j,:]=temp1.iloc[7,:]
#     data_cz=data_cz.join(temp,how='outer')

# 预测误差
data_error = sbm_yl_pro - data_cz
data_error_sum = data_error.sum(axis=1)
data_yll = sbm_yl.iloc[:, 0:11].sum(axis=1)
data_yl = data_yll

```



```
data_lslv = data_error_sum / data_yl # 每天的漏损率估计
```

```
# data_lslv.to_excel('D:\\主桌面\\E 题\\漏水率（每天）.xls')
```

```
# data_error_sum.to_excel('D:\\主桌面\\E 题\\漏水量（每天）.xls')
```

```
data_error[data_error<0]=0
```

```
data_error_prob=data_error/data_cz
```

```
# data_error_prob[data_cz<2]=0
```

```
data_error_prob=data_error_prob.fillna(0)
```

```
error_num=(data_error_prob>0.098).astype(int).sum()
```

```
error_num.to_excel('D:\\主桌面\\E 题\\漏水次数（每天）.xls') # 每个水表位置的漏水次数统计
```

```
# 计算漏损流量
```

```
# data_error_prob[data_error_prob<0.2]=0
```

```
# data_ls=data_error_prob*data_cz
```

```
data_ls=(data_error).sum() # 每天总漏水量
```

```
sbm_yl_sum=sbm_yl.sum() # 每天总用水量
```

```
sbm_lslv=data_ls/sbm_yl_sum
```

```
sbm_lslv.to_excel('D:\\主桌面\\E 题\\漏水率.xls')
```

```
data_ls.to_csv('D:\\主桌面\\E 题\\漏水量.csv')
```

```
zlsl=data_ls.sum() # 总漏水量
```

```
# 总漏水率
```

```
zlslv=zlsl/zysl_yj # 总漏水率
```

3. 问题五 (Matlab):

```
clear
```

```
clc
```

```
data = xlsread('F:\\数学建模\\E 题\\漏水量（每天）.xls');
```

```
data = data(2 : end);
```

```
sj = 5.5 : 0.1 : 6.5;
```

```
wxj = 500 : 1500;
```

```
cs = [];
```

```
sunshi = [];
```

```
for i = 1 : length(sj)
```

```
    for j = 1 : length(wxj)
```

```
        lsje = data * sj(i);
```

```
        wxje = wxj(j) * ones(365, 1);
```

```
        lsje_sum = sum(lsje);
```



关注数学模型

获取更多资讯

```

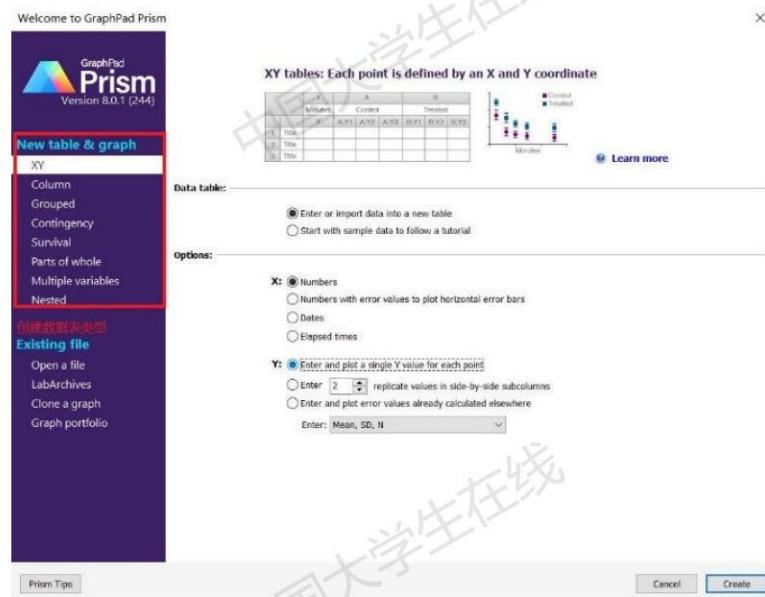
A = wxje - lsje;
lb = zeros(365, 1);
ub = ones(365, 1);
intcon = 1 : 365;
x0 = zeros(365, 1);
[x, f] = intlinprog(A, intcon, [], [], [], lb, ub, x0);
cs(i, j) = sum(x);
sunshi(i, j) = f + lsje_sum;
end
end
xlswrite('F:\数学建模\cs.xlsx', cs)
xlswrite('F:\数学建模\sunshi.xlsx', sunshi)
% % [X,Y]=mesh(sj,wxj);
plot3(sj, wxj, cs)

```

Graphpad

作图流程：以每天总漏水量为例：

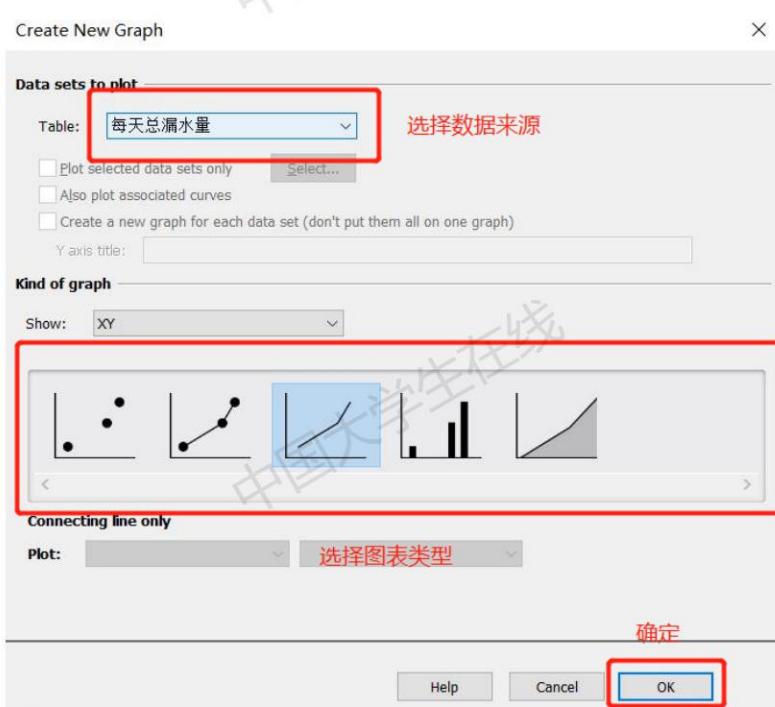
1. 打开 Graphpad,
2. 创建数据集：



3. 将 Excel 数据粘贴至 Graphpad:

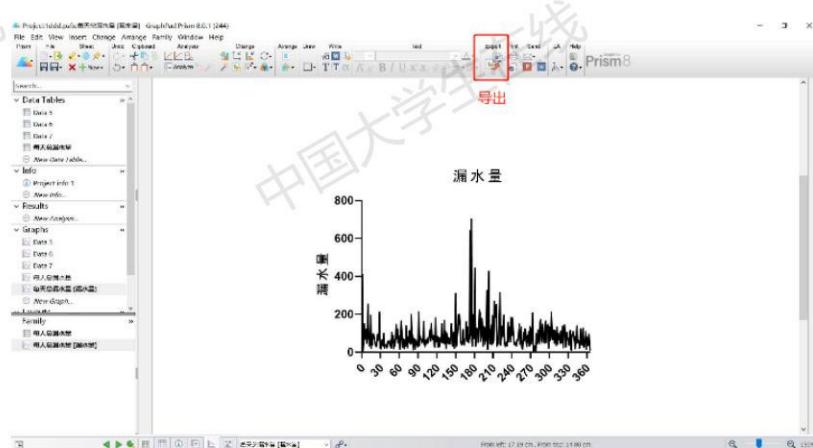


4. 创建图表:



5. 随后调节字体、大小、线条宽度和坐标轴等参数

6. 输出图片:





7. 导出完成



Excel

1、先利用 python 中的 pandas 库对数进行了以功能区按小时的求和的数据，并导入到了 excel 表格中。

The screenshot shows a Microsoft Excel spreadsheet titled "功能区每小时.xls [兼容模式] - Excel". The table has 16 columns labeled A through P. Column A is "采集时间" (Collection Time) and column B is "总表" (Total Table). The data starts from row 2, showing various values for different categories over time. The table is styled with green headers and yellow borders.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
采集时间	总表	宿舍	教学楼	办公楼	食堂	科研楼	运动场所	后勤保卫	校医院	课堂	酒店	图书馆	其他		
2019-01-01 00:00:00	9.72	2.19	0.3	0.51	0.12	0	1.14	0.03	0	2.31	0	5.23	3		
2019-01-01 01:00:00	10.88	2.92	0.4	0.68	0.19	0	1.52	0.04	0	3.08	0	4.32	1		
2019-01-01 02:00:00	10.46	2.92	0.4	0.68	0.17	0	1.52	0.04	0	3.08	0	3.48	6		
2019-01-01 03:00:00	12.67	2.23	0.29	0.67	0.04	0	1.03	0.01	0	1.77	0	2.86	8		
2019-01-01 04:00:00	12	1.96	0.08	1	0.16	0.08	0.96	0.08	0	2.44	0.04	2.92	7.25		
2019-01-01 05:00:00	12	1.96	0.08	1	0.16	0.08	0.96	0.08	0	2.44	0.04	2.92	1.75		
2019-01-01 06:00:00	12	1.96	0.08	1	0.31	0.08	0.96	0.08	0	2.44	0.04	6.1	2.25		
2019-01-01 07:00:00	7.84	2.49	0.91	2.45	0.8	0.52	1.04	0.12	0	1.71	0.01	11.36	2.25		
2019-01-01 08:00:00	9.84	5	0.92	2.48	0.85	0.36	1.44	0.84	0	3.36	0.2	14.64	4.25		
2019-01-01 09:00:00	10.04	5	0.92	2.48	0.93	0.36	1.56	0.84	0	3.36	0.2	12.93	1.25		
2019-01-01 10:00:00	9.64	5	0.92	2.48	0.78	0.36	1.48	0.84	0	3.36	0.1	9.93	7.5		
2019-01-01 11:00:00	13.02	6.19	0.33	1.83	3.36	0.39	1.26	0.71	0	4.64	0	11.36	4.75		
2019-01-01 12:00:00	10.32	5.48	0.8	1.24	1.79	0.4	1.84	0.36	0	4.28	0.08	9.6	3		
2019-01-01 13:00:00	10.32	5.48	0.8	1.24	1.72	0.4	1.84	0.36	0	4.28	0.08	8.97	0.75		
2019-01-01 14:00:00	10.32	5.48	0.8	1.24	1.83	0.4	1.84	0.36	0	4.28	0.08	8.76	4.5		
2019-01-01 15:00:00	7.35	6.71	1.6	0.81	0.89	0.4	2.11	0.39	0	3.15	0.02	7.43	3.75		
2019-01-01 16:00:00	8.38	5.66	0.52	1.6	0.71	0.28	2.57	0.52	0.04	3.44	0.1	6.89	3		
2019-01-01 17:00:00	8.18	5.66	0.52	1.6	0.75	0.28	2.56	0.52	0.04	3.44	0	7.4	0.75		
2019-01-01 18:00:00	8.18	5.56	0.52	1.6	0.64	0.28	2.56	0.52	0.04	3.44	0.1	9.64	5.25		
2019-01-01 19:00:00	10.06	5.93	0.93	1.4	0.47	0.37	2.27	0.43	0.01	2.76	0.1	10.83	4		
2019-01-01 20:00:00	18.96	8.2	0.44	0.64	0.74	0.4	3.32	0.32	0	2.6	0.08	11.86	3.75		
2019-01-01 21:00:00	18.96	8.2	0.44	0.64	0.69	0.4	3.32	0.32	0	2.6	0.08	13.14	1		
2019-01-01 22:00:00	18.96	8.2	0.44	0.64	0.68	0.4	3.32	0.32	0	2.6	0.08	17.31	7.5		
2019-01-01 23:00:00	22.15	8.09	0.11	0.16	0.57	0.1	2.35	0.28	0	1.34	0.12	14.82	5.5		
2019-01-02 00:00:00	13.28	2.44	0.04	0.08	0.36	0	1.36	0.04	0	0.32	0.1	8.3	2.5		
2019-01-02 01:00:00	13.28	2.44	0.04	0.08	0.39	0	1.36	0.04	0	0.32	0	4.32	0.5		
2019-01-02 02:00:00	12.38	2.44	0.04	0.08	0.37	0	1.36	0.04	0	0.32	0	3.58	4.5		
2019-01-02 03:00:00	11.74	1.55	0.11	0.11	0.29	0	1.04	0.01	0	0.49	0	2.86	4.5		
2019-01-02 04:00:00	11.8	1.6	0.24	1.32	0.44	0.08	1.12	0.08	0	0.12	0.04	3.16	1.75		
2019-01-02 05:00:00	11.8	1.6	0.24	1.32	0.44	0.08	1.12	0.08	0	0.12	0.04	3.16	0.25		
2019-01-02 06:00:00	11.8	1.6	0.24	1.32	0.59	0.08	1.12	0.08	0	0.12	0.04	6.34	4.5		

2、通过函数填充出各时间段的具体日期

The screenshot shows a Microsoft Excel spreadsheet titled "功能区每小时 (图表).xls [兼容模式] - Excel". The table has 16 columns labeled A through P. Column A is "采集时间" (Collection Time) and column B is "时间" (Time). The data starts from row 26, showing various times and their corresponding day of the week. The formula =TEXT(A26,"aaaa") is highlighted in red in cell D26.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
采集时间	时间	数学桂	工作日	周六	节假日	1. 27-2.1	7. 15-8.1								
2019-01-02 00:00:00	0:00:00	2.44	星期三												
2019-01-02 00:00:00	1:00:00	2.44	星期三												
2019-01-02 00:00:00	2:00:00	2.44	星期三												
2019-01-02 00:00:00	3:00:00	1.55	星期三												
2019-01-02 00:00:00	4:00:00	1.6	星期三												
2019-01-02 00:00:00	5:00:00	1.6	星期三												
2019-01-02 00:00:00	6:00:00	1.6	星期三												
2019-01-02 00:00:00	7:00:00	4.34	星期三												
2019-01-02 00:00:00	8:00:00	6.64	星期三												
2019-01-02 00:00:00	9:00:00	6.64	星期三												
2019-01-02 00:00:00	10:00:00	6.64	星期三												
2019-01-02 00:00:00	11:00:00	7.3	星期三												
2019-01-02 00:00:00	12:00:00	6.48	星期三												
2019-01-02 00:00:00	13:00:00	6.48	星期三												
2019-01-02 00:00:00	14:00:00	6.48	星期三												
2019-01-02 00:00:00	15:00:00	6.95	星期三												
2019-01-02 00:00:00	16:00:00	5.92	星期三												
2019-01-02 00:00:00	17:00:00	5.52	星期三												
2019-01-02 00:00:00	18:00:00	6.52	星期三												
2019-01-02 00:00:00	19:00:00	6.52	星期三												



功能区每小时 (图表).xls [兼容模式] - Excel

开始 插入 页面布局 公式 数据 审阅 视图 帮助 Inquire Power Pivot pv 操作说明搜索

粘贴 复制 格式刷 剪贴板

字体 Arial 10pt 文本 对齐方式 合并后居中 数字

E98 ✓ fx =TEXT(A98,"aaaa")

A	B	C	D	E	F	G	H	I	J	K
采集时间	时间	教学楼	工作日	周六	节假日	1.27-2.1	7.15-8.1			
2019-01-05 00:00:00	0:00:00	3.04		星期六						
2019-01-05 00:00:00	1:00:00	3.04		星期六						
2019-01-05 00:00:00	2:00:00	3.04		星期六						
2019-01-05 00:00:00	3:00:00	2.1		星期六						
2019-01-05 00:00:00	4:00:00	1.6		星期六						
2019-01-05 00:00:00	5:00:00	1.6		星期六						
2019-01-05 00:00:00	6:00:00	1.6		星期六						
2019-01-05 00:00:00	7:00:00	3.84		星期六						
2019-01-05 00:00:00	8:00:00	5.48		星期六						
2019-01-05 00:00:00	9:00:00	5.48		星期六						
2019-01-05 00:00:00	10:00:00	5.48		星期六						
2019-01-05 00:00:00	11:00:00	5.51		星期六						
2019-01-05 00:00:00	12:00:00	5.56		星期六						
2019-01-05 00:00:00	13:00:00	5.56		星期六						
2019-01-05 00:00:00	14:00:00	5.56		星期六						
2019-01-05 00:00:00	15:00:00	4.53		星期六						
2019-01-05 00:00:00	16:00:00	5.02		星期六						
2019-01-05 00:00:00	17:00:00	5.12		星期六						
2019-01-05 00:00:00	18:00:00	5.02		星期六						
2019-01-05 00:00:00	19:00:00	6.12		星期六						
2019-01-05 00:00:00	20:00:00	5.72		星期六						
2019-01-05 00:00:00	21:00:00	5.72		星期六						
2019-01-05 00:00:00	22:00:00	5.72		星期六						
2019-01-05 00:00:00	23:00:00	6.27		星期六						
2019-01-06 00:00:00	0:00:00	3.32		星期日						
2019-01-06 00:00:00	1:00:00	3.32		星期日						

3、选择 D-H 列的全部数据——点击筛选

功能区每小时 (图表).xls [兼容模式] - Excel

开始 插入 页面布局 公式 数据 审阅 视图 帮助 Inquire Power Pivot pv 操作说明搜索

取数 从文 自自 最近使 现有 居、本/CSV 网站 格/区域 用的源 连接

全部刷新 属性 排序 筛选 高级

获取和转换数据

A	B	C	D	E	F	G	H	I
采集时间	时间	教学楼	工作日	周六	节假日	1.27-2.1	7.15-8.1	
2019-01-02 00:00:00	0:00:00	2.44	星期三					
2019-01-02 00:00:00	1:00:00	2.44	星期三					
2019-01-02 00:00:00	2:00:00	2.44	星期三					
2019-01-02 00:00:00	3:00:00	1.55	星期三					
2019-01-02 00:00:00	4:00:00	1.6	星期三					
2019-01-02 00:00:00	5:00:00	1.6	星期三					
2019-01-02 00:00:00	6:00:00	1.6	星期三					
2019-01-02 00:00:00	7:00:00	4.34	星期三					
2019-01-02 00:00:00	8:00:00	6.64	星期三					
2019-01-02 00:00:00	9:00:00	6.64	星期三					
2019-01-02 00:00:00	10:00:00	6.64	星期三					
2019-01-02 00:00:00	11:00:00	7.3	星期三					
2019-01-02 00:00:00	12:00:00	6.48	星期三					
2019-01-02 00:00:00	13:00:00	6.48	星期三					
2019-01-02 00:00:00	14:00:00	6.48	星期三					
2019-01-02 00:00:00	15:00:00	6.95	星期三					
2019-01-02 00:00:00	16:00:00	5.92	星期三					
2019-01-02 00:00:00	17:00:00	5.52	星期三					
2019-01-02 00:00:00	18:00:00	6.52	星期三					
2019-01-02 00:00:00	19:00:00	6.53	星期三					
2019-01-02 00:00:00	20:00:00	6.2	星期三					
2019-01-02 00:00:00	21:00:00	6.2	星期三					
2019-01-02 00:00:00	22:00:00	6.2	星期三					
2019-01-02 00:00:00	23:00:00	6.79	星期三					
2019-01-02 00:00:00	0:00:00	2.06	星期四					



4、点击 C1 单元格的小倒三角按钮——除空白外选中其余全部数据

	A	B	C	D	E	F	G	H
1	采集时间	时间	教学楼	工作日	周六	节假日	1. 27-2. 1	7. 15-8. 1
26	2019-01-02 00:00:00							
27	2019-01-02 00:00:00							
28	2019-01-02 00:00:00							
29	2019-01-02 00:00:00							
30	2019-01-02 00:00:00							
31	2019-01-02 00:00:00							
32	2019-01-02 00:00:00							
33	2019-01-02 00:00:00							
34	2019-01-02 00:00:00							
35	2019-01-02 00:00:00							
36	2019-01-02 00:00:00							
37	2019-01-02 00:00:00							
38	2019-01-02 00:00:00							
39	2019-01-02 00:00:00							
40	2019-01-02 00:00:00							
41	2019-01-02 00:00:00							
42	2019-01-02 00:00:00							
43	2019-01-02 00:00:00							
44	2019-01-02 00:00:00							
45	2019-01-02 00:00:00							
46	2019-01-02 00:00:00							
47	2019-01-02 00:00:00							
48	2019-01-02 00:00:00							
49	2019-01-02 00:00:00							
50	2019-01-03 00:00:00	0:00:00						

4、选中这两列的所有数据——Ctrl+C

	A	B	C	D	E	F	G	H
1	采集时间	时间	教学楼	工作日	周六	节假日	1. 27-2. 1	7. 15-8. 1
26	2019-01-02 00:00:00	0:00:00	2.44	星期一				
27	2019-01-02 00:00:00	1:00:00	2.44	星期一				
28	2019-01-02 00:00:00	2:00:00	2.44	星期一				
29	2019-01-02 00:00:00	3:00:00	1.55	星期三				
30	2019-01-02 00:00:00	4:00:00	1.6	星期三				
31	2019-01-02 00:00:00	5:00:00	1.6	星期三				
32	2019-01-02 00:00:00	6:00:00	1.6	星期三				
33	2019-01-02 00:00:00	7:00:00	4.34	星期三				
34	2019-01-02 00:00:00	8:00:00	6.64	星期三				
35	2019-01-02 00:00:00	9:00:00	6.64	星期三				
36	2019-01-02 00:00:00	10:00:00	6.64	星期三				
37	2019-01-02 00:00:00	11:00:00	7.3	星期三				
38	2019-01-02 00:00:00	12:00:00	6.48	星期三				
39	2019-01-02 00:00:00	13:00:00	6.48	星期三				
40	2019-01-02 00:00:00	14:00:00	6.48	星期三				
41	2019-01-02 00:00:00	15:00:00	6.95	星期三				
42	2019-01-02 00:00:00	16:00:00	5.92	星期三				
43	2019-01-02 00:00:00	17:00:00	5.52	星期三				
44	2019-01-02 00:00:00	18:00:00	6.59	星期二				

5、把第四步的两列复制到 各类时间工作表中的工作日列



A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1														
A2		B2		C2		D2		E2		F2		G2		H2		I2		J2		K2		L2		M2		N2	
工作日		宿舍				时间		宿舍				节假日		时间		宿舍				1.27-2.17				时间		宿舍	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
1	0:00:00	13.28				0:00:00	14.45			0:00:00	9.72			0:00:00	18.57			0:00:00	15.54								
2	1:00:00	13.23				1:00:00	13.92			1:00:00	10.86			1:00:00	9.62			1:00:00	6.74								
3	2:00:00	12.38				2:00:00	14.12			2:00:00	10.46			2:00:00	8.23			2:00:00	10.59								
4	3:00:00	11.74				3:00:00	15.41			3:00:00	12.67			3:00:00	12.76			3:00:00	9.62								
5	4:00:00	11.8				4:00:00	7.88			4:00:00	12			4:00:00	12.87			4:00:00	13.44								
6	5:00:00	11.8				5:00:00	7.88			5:00:00	12			5:00:00	13.32			5:00:00	10.05								
7	6:00:00	11.8				6:00:00	7.94			6:00:00	12			6:00:00	3.28			6:00:00	8.15								
8	7:00:00	13.1				7:00:00	8.81			7:00:00	7.84			7:00:00	7.36			7:00:00	15.25								
9	8:00:00	8.4				8:00:00	10.7			8:00:00	9.84			8:00:00	8.77			8:00:00	14.02								
10	9:00:00	10.3				9:00:00	10.82			9:00:00	10.04			9:00:00	15.22			9:00:00	13.12								
11	10:00:00	9.3				10:00:00	11.85			10:00:00	9.64			10:00:00	13.2			10:00:00	8.6								
12	11:00:00	9.85				11:00:00	14.08			11:00:00	13.02			11:00:00	14.35			11:00:00	9.52								
13	12:00:00	7.24				12:00:00	8.73			12:00:00	10.32			12:00:00	10.8			12:00:00	9.93								
14	13:00:00	7.24				13:00:00	8.85			13:00:00	10.32			13:00:00	8.49			13:00:00	12.81								
15	14:00:00	7.24				14:00:00	8.85			14:00:00	10.32			14:00:00	9.39			14:00:00	6.7								
16	15:00:00	5.99				15:00:00	7.19			15:00:00	7.35			15:00:00	8.98			15:00:00	4.85								
17	16:00:00	6.28				16:00:00	8.6			16:00:00	8.38			16:00:00	6.41			16:00:00	3.86								
18	17:00:00	6.28				17:00:00	8.84			17:00:00	8.18			17:00:00	7.94			17:00:00	5.84								
19	18:00:00	6.28				18:00:00	9.53			18:00:00	8.18			18:00:00	13.5			18:00:00	8.6								
20	19:00:00	8.79				19:00:00	14.4			19:00:00	10.06			19:00:00	9.44			19:00:00	9.81								
21	20:00:00	18				20:00:00	20.18			20:00:00	18.96			20:00:00	17.61			20:00:00	11.38								
22	21:00:00	18				21:00:00	20.14			21:00:00	18.96			21:00:00	17.35			21:00:00	19.73								
23	22:00:00	18				22:00:00	20.18			22:00:00	18.96			22:00:00	18.85			22:00:00	27.75								
24	23:00:00	24				23:00:00	25.36			23:00:00	22.15			23:00:00	21.1			23:00:00	30.52								
25	0:00:00	16.7				0:00:00	18.66			0:00:00	20.96			0:00:00	14.51			0:00:00	16.15								
26	1:00:00	14.4				1:00:00	16.16			1:00:00	14.72			1:00:00	11.41			1:00:00	5.4								
27	2:00:00	13.7				2:00:00	15.65			2:00:00	10.27			2:00:00	12.27			2:00:00	9.88								
28	3:00:00	11.26				3:00:00	13.3			3:00:00	13.85			3:00:00	12.29			3:00:00	13.92								
29	4:00:00	10.8				4:00:00	7.3			4:00:00	13.87			4:00:00	12.76			4:00:00	14.15								
30	5:00:00	10.8				5:00:00	7.26			5:00:00	3.93			5:00:00	6.74			5:00:00	8.59								

6、在选中 A、B 两列全部数据后——数据透视表——点击确定

The screenshot shows the Microsoft Excel ribbon with the 'Insert' tab selected. In the 'Tables' group, the '数据透视表' (PivotTable) icon is highlighted with a red box. A second red box highlights the '确定' (OK) button in the 'Create PivotTable' dialog box, which is open over the worksheet area.

	A	B
1	工作日	
2	时间	宿舍
3	0:00:00	13.28
4	1:00:00	13.28
5	2:00:00	12.38
6	3:00:00	11.74
7	4:00:00	11.8

7、把 时间、宿舍分别拖动到 行和值 的区域——点击宿舍——选择值字段设置——选择平均值——选择确定



选择要添加到报表的字段:

搜索

时间
 宿舍

在以下区域间拖动字段:

筛选 列

行 值

时间 求和项:宿舍

值字段设置

源名称: 宿舍
自定义名称(C): 平均值项:宿舍

值汇总方式 值显示方式

值字段汇总方式(S)

选择用于汇总所选字段数据的
计算类型

求和
计数
平均值
最大值
最小值
乘积

确定 取消

延迟布局更新 更新

8、选择 B 列 5 到 28 行的数据——Ctrl+C——粘贴到功能区工作表中的 B 列 3 到 26 行处。

A	B	C	A	B	C	D	E	F
1	将报表筛选字段拖至此处	2	宿舍	1	工作日	周六日	节假日	1. 27-2. 17
3	平均值项:宿舍	3	20.40257	19.40605	15.5635	9.14090909	18.363125	
4	时间	4	9.93319	9.634321	9.544	6.17	8.2528125	
5	0:00:00	5	10.09771	10.04358	10.6665	7.17909091	10.01875	
6	1:00:00	6	10.9861	9.711481	10.056	7.97	11.41375	
7	2:00:00	7	10.55581	8.808642	9.142	6.33681818	13.5628125	
8	3:00:00	8	6.695857	5.669012	5.8325	3.98454545	9.43	
9	4:00:00	9	9.335476	6.431481	6.4455	4.02272727	10.0775	
10	5:00:00	10	17.25695	11.00284	9.878	6.97363636	17.65	
11	6:00:00	11	14.48395	13.70123	12.247	6.15454545	15.615625	
12	7:00:00	12	13.3039	15.98728	13.4075	7.50954545	13.8553125	
13	8:00:00	13	11.18143	15.35642	13.0785	7.02363636	11.1528125	
14	9:00:00	14	10.80719	14.62617	12.778	6.25318182	10.4271875	
15	10:00:00	15	10.895	14.0337	11.5235	5.66272727	11.4834375	
16	11:00:00	16	10.72671	13.28778	10.5605	5.79909091	10.9490625	
17	12:00:00	17	9.558429	10.59704	9.306	5.62590909	10.1428125	
18	13:00:00	18	8.590476	9.126296	8.544	4.84045455	8.6746875	
19	14:00:00	19	7.255048	8.597037	8.0665	4.49454545	7.13125	
20	15:00:00	20	8.486571	9.535309	8.457	4.72545455	8.46125	
21	16:00:00	21	9.667238	9.669136	8.861	5.65863636	9.8990625	
22	17:00:00	22	11.27976	11.76074	9.2975	7.25727273	11.893125	
23	18:00:00	23	14.2369	14.44519	11.2055	9.37863636	13.980625	
24	19:00:00	24	22.18005	22.45444	15.4085	12.6545455	23.2946875	
25	20:00:00	25	32.17014	30.57432	22.1355	14.1359091	33.0425	
26	21:00:00	26	34.40043	32.33889	24.4	14.4740909	35.6275	
27								
28								
29	总计		13.5202877					
30								

图表标题

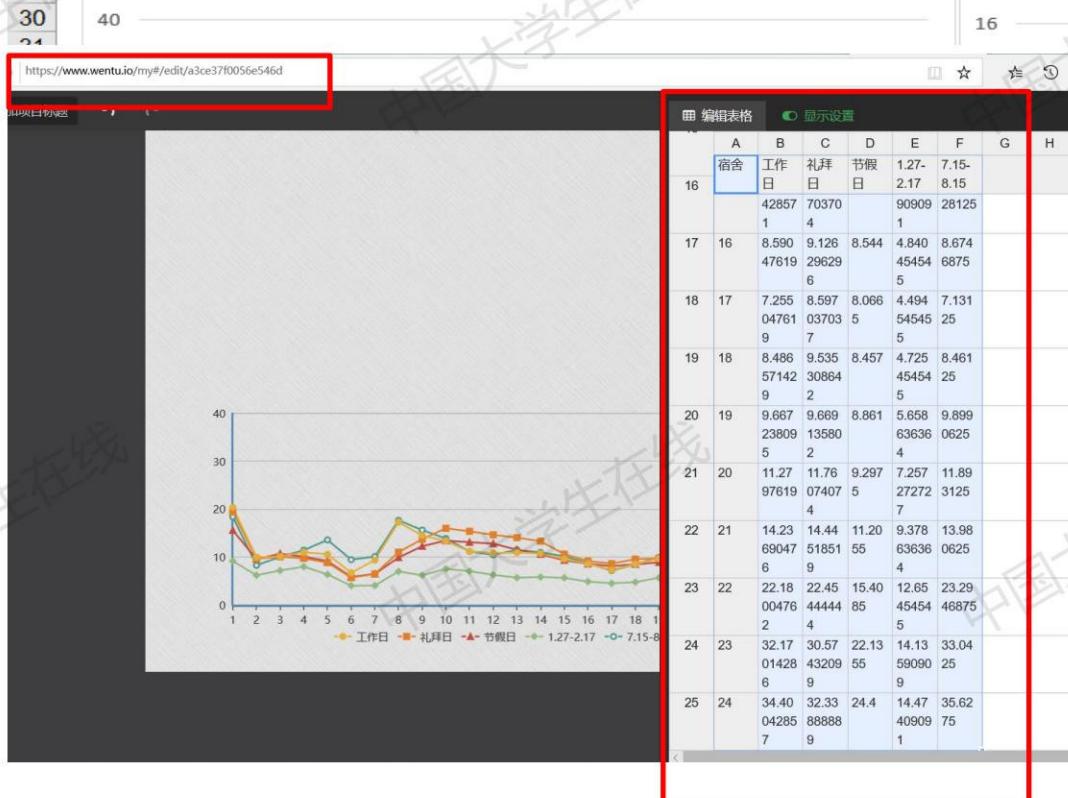
全年按小时 | Sheet8 | 各类别时间 | 功能区 | +

9、重复 4—8 的步骤后会得到以下全部数据——Ctrl+C——粘贴到如下网址中



	A	B	C	D	E	F	G	H
1								
2	宿舍	工作日	礼拜日	节假日	1. 27-2. 17	7. 15-8. 15	教学楼	
3	1	20.40257	19.40605	15.5635	9.14090909	18.363125	1	
4	2	9.93319	9.634321	9.544	6.17	8.2528125	2	
5	3	10.09771	10.04358	10.6665	7.17909091	10.01875	3	
6	4	10.9861	9.711481	10.056	7.97	11.41375	4	
7	5	10.55581	8.808642	9.142	6.33681818	13.5628125	5	
8	6	6.695857	5.669012	5.8325	3.98454545	9.43	6	
9	7	9.335476	6.431481	6.4455	4.02272727	10.0775	7	
10	8	17.25695	11.00284	9.878	6.97363636	17.65	8	
11	9	14.48395	13.70123	12.247	6.15454545	15.615625	9	
12	10	13.3039	15.98728	13.4075	7.50954545	13.8553125	10	
13	11	11.18143	15.35642	13.0785	7.02363636	11.1528125	11	
14	12	10.80719	14.62617	12.778	6.25318182	10.4271875	12	
15	13	10.895	14.0337	11.5235	5.66272727	11.4834375	13	
16	14	10.72671	13.28778	10.5605	5.79909091	10.9490625	14	
17	15	9.558429	10.59704	9.306	5.62590909	10.1428125	15	
18	16	8.590476	9.126296	8.544	4.84045455	8.6746875	16	
19	17	7.255048	8.597037	8.0665	4.49454545	7.13125	17	
20	18	8.486571	9.535309	8.457	4.72545455	8.46125	18	
21	19	9.667238	9.669136	8.861	5.65863636	9.8990625	19	
22	20	11.27976	11.76074	9.2975	7.25727273	11.893125	20	
23	21	14.2369	14.44519	11.2055	9.37863636	13.980625	21	
24	22	22.18005	22.45444	15.4085	12.6545455	23.2946875	22	
25	23	32.17014	30.57432	22.1355	14.1359091	33.0425	23	
26	24	34.40043	32.33889	24.4	14.4740909	35.6275	24	

图表标题



10、最终得到如下结果



