本周五（18日）有【上机考试】

- 18:00-20:15
- 理工配楼二层1、2机房

第5讲：探索数据分析 – 数据获取与准备

“数据科学导论”专业核心课

范举 教授 fanj@ruc.edu.cn

探索数据分析：迭代工作流

技术挑战

- 数据的多样性
- 数据的海量性
- 结果的及时性
- 数据的隐私保护
- ……

提出问题：新冠肺炎求助患者数据画像

天边的一朵小马驹_ +关注

2月11日 00:04 来自 肺炎患者求助超话

肺炎患者求助超话 【姓名】 XYZ

【年龄】56

【所在城市】 武汉

【所在小区、社区】汉阳区麒麟路麒麟社区 X栋Y单元Z楼

【患病时间】1月26日

【联系方式】 1397xxx6027

【其他紧急联系电话】 1341xxx1935

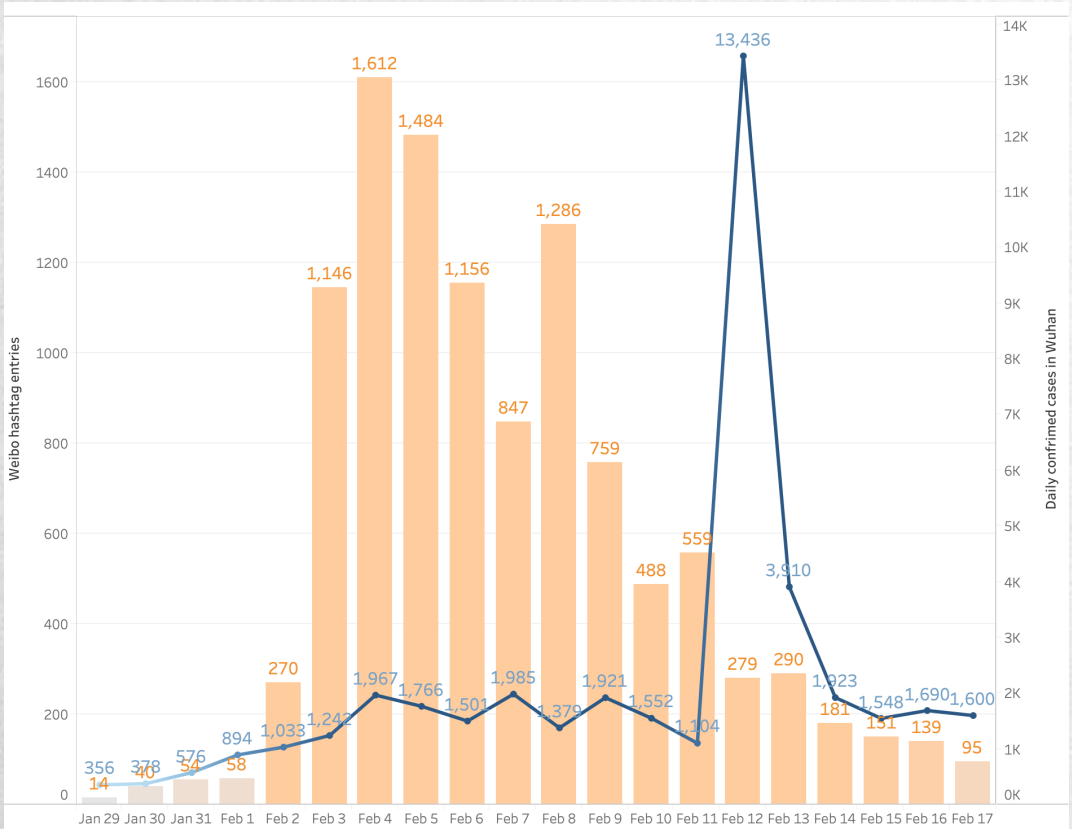
【病情详细描述】病人有高血压，甲状腺癌，发烧十天，高烧39度，双肺感染，咳嗽，无力，2月8日做了试剂盒，结果还没出来，现在病人病危，靠氧气，血氧70。医院不收入院。

【周边家人及隔离情况】家里还有一个姐姐同住，姐姐有糖尿病重症，也昏睡了十几天，咳嗽，乏力，至今也没隔离，连去医院的劲都没有。家里还有一位父亲抗美援朝的离休老干部90岁，于2月5日在家离世（是不是这个病离世未知）我已被隔离，妈妈目前情况非常严峻，只能躺床上靠吸氧根本不能动，连CT片都不能拍给我，我们报给社区说只能在家等，根本无力解决，现在情况紧急，生命危在旦夕，求求哪家医院能收治我的家人

新浪微博 weibo.com

肺炎患者求助超话

提出问题：肺炎患者求助超话的整体分析



- 黄色柱状图
  - 肺炎患者求助超话每天的发帖量
- 蓝色线条图
  - 武汉市卫健委汇报的每日新增新冠确诊数
- 时间区间
  - 1月29日 – 2月17日
- 你能发现什么？

提出问题：新冠肺炎求助患者数据画像

- 我们试图用数据解码数字背后的个体，看清那些宏大叙事之下的具体的人，尤其是那些在社交网络上以个体的身份发出求助的人——

- 他们是谁？
- 他们为什么而发声？
- 他们的处境有何共同点？

提纲

1. **数据获取**
2. **数据准备 – 数据抽取**
3. **数据准备 – 数据清洗**
4. **案例分析：wuhan2020-sos**

探索数据分析

数据获取与准备

从CSV文件读取数据

- CSV的全称是Comma-separated values，是一种用逗号分隔的方式来表示与存储表格数据的文件格式

rankings.csv

```
Team,Win,Loss,Win%
Houston Rockets,20,4,0.833
Golden State Warriors,21,6,0.778
San Antonio Spurs,19,8,0.704
Minnesota Timberwolves,16,11,0.593
Denver Nuggets,14,12,0.538
Portland Trail Blazers,13,12,0.52
New Orleans Pelicans,14,13,0.519
Utah Jazz,13,14,0.481
```

- 技能包
  - 使用Python Pandas读取CSV文件

```python
import pandas as pd

df = pd.read_csv('./samples/nba.csv')
print (df['Team'])
print (df['Win'].max())
```

扩展阅读：https://pythonspot.com/pandas-read-csv/

从JSON文件读取数据

- JSON是一种存储嵌套数据的文件格式（类似Python中的List, Dict）

players.json

```
{
    "Kobe Bryant" :{
        "Born": "08/23/1978",
        "Number": ["8", "24"],
        "Team": ["Los Angeles Lakers"]
    },
    "Michael Jordan":{
        "Born": "02/17/1963",
        "Number": ["23"],
        "Team": ["Chicago Bulls", "Washington Wizards"]
    }
}
```

- 技能包
  - 使用Python Pandas读取JSON文件

```
import pandas as pd

df = pd.read_json('./samples/nba.json')
```

阅读：http://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_json.html

从网页获取数据：“爬虫”

- **爬虫**：一段自动抓取互联网信息的程序，从互联网上抓取对于我们有价值的信息
  - 每个搜索引擎都拥有自己的爬虫程序，比如 360 浏览器的爬虫称作 360Spider，搜狗的爬虫叫做 Sogouspider
  - 爬虫可以用Java、PHP、Python等语言写，Python 更为简单和实用
  - 常用的爬虫框架：Selenium、Scrapy

https://scrapy.org/

https://www.selenium.dev/

基于爬虫的数据分析与决策

- 金融领域需求：获取各大政府、行业网站发布的关键数据、公告
  - 实现方法：使用爬虫提高获取数据的效率
  - 举例：万得软件是我国领先的金融数据服务商之一

万得信息技术股份有限公司

播报 讨论 上传视频

公司

上海万得信息技术股份有限公司是中国大陆领先的金融信息服务企业，金融行业的人习惯称之为“wind资讯”或“万得”，wind资讯因此成为公司的品牌。在国内市场，wind资讯的客户包括超过87%的中国证券公司、基金管理公司、保险公司、银行和投资公司等金融企业；在国际市场，已经被中国证监会批准的合格境外机构投资者（qfii）中 60的机构是wind资讯的客户。

2020年8月4日，《苏州高新区·2020胡润全球独角兽榜》发布，万得排名第70位。

公司名称	上海万得信息技术股份有限公司	成立时间	1996年
所属行业	金融行业	总部地点	上海浦东陆家嘴金融区

https://baike.baidu.com/item/万得信息技术股份有限公司/20101094

万得信息技术股份有限公司 存续

下载报告 笔记 风险监控 关注

曾用名 战略融资 高新技术企业 企业技术中心 投资机构 自身风险 353条 司法案件 999+个 #更多标签 26万+ 3小时37分钟前更新

统一社会信用代码：91310000773282177G
法定代表人：曹剑 关联24家企业>
注册资本：67557.6万元人民币
成立日期：2005-04-04

电话：021-2070**** 更多 1
邮箱：G****M@wind.com.cn
官网：http://www.wind.com.cn
地址：中国（上海）自由贸易试验区浦明路15… 附近企业 通信地址

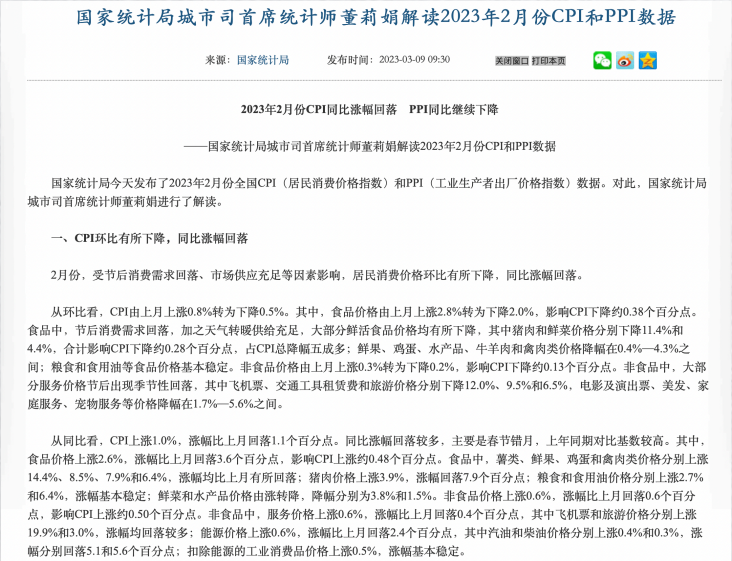
企查查行业：
企业规模：
员工人数：

简介：万得信息技术股份有限公司（下称：Wind）是中国大陆领先的金融数据、信息和软件服务企业，总部位于上海陆家嘴金融中心。 在国内市场，Wind的客户包括中国绝大多数… 展开

财产线索 线索 999+ 预估价值
所属集团：Wind 成员 36 风险 999+
产品信息：万得Wind 最新融资轮次 战略融资
投资机构：万得投资 投资项目 15 投资事件 15
股权穿透图 挖掘深层股权结构

https://www.qcc.com/firm/d1b84f508d958bdd374cba6f903686b3.html

基于爬虫的数据分析与决策



国家统计局城市司首席统计师董莉娟解读2023年2月份CPI和PPI数据

来源：国家统计局　　发布时间：2023-03-09 09:30　　关闭窗口 打印本页

2023年2月份CPI同比涨幅回落　PPI同比继续下降

——国家统计局城市司首席统计师董莉娟解读2023年2月份CPI和PPI数据

国家统计局今天发布了2023年2月份全国CPI（居民消费价格指数）和PPI（工业生产者出厂价格指数）数据。对此，国家统计局城市司首席统计师董莉娟进行了解读。

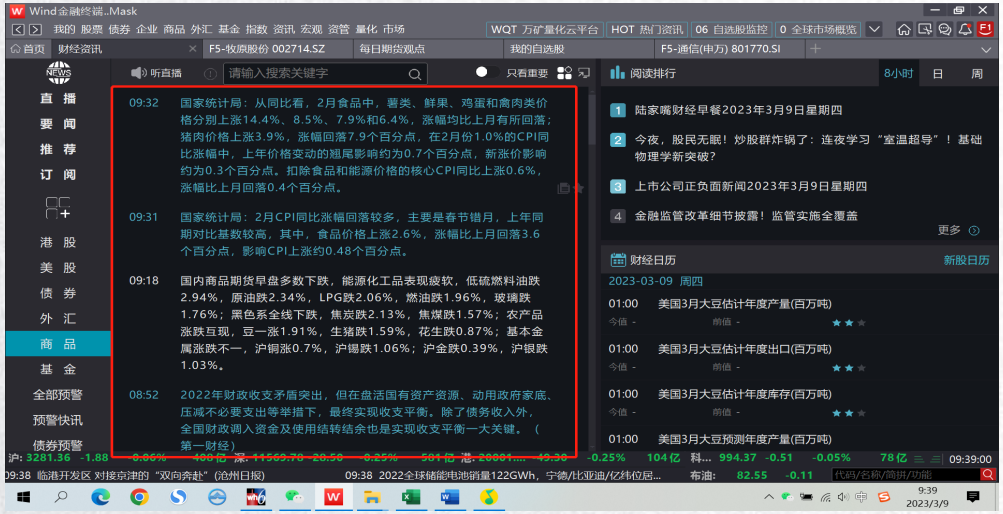
一、CPI环比有所下降，同比涨幅回落

2月份，受节后消费需求回落、市场供应充足等因素影响，居民消费价格环比有所下降，同比涨幅回落。

从环比看，CPI由上月上涨0.8%转为下降0.5%。其中，食品价格由上月上涨2.8%转为下降2.0%，影响CPI下降约0.38个百分点。食品中，节后消费需求回落，加之天气转暖供给充足，大部分鲜活食品价格均有所下降，其中猪肉和鲜菜价格分别下降11.4%和4.4%，合计影响CPI下降约0.28个百分点，占CPI总降幅五成多；鲜果、鸡蛋、水产品、牛羊肉和禽肉类价格降幅在0.4%—4.3%之间；粮食和食用油等食品价格基本稳定。非食品价格由上月上涨0.3%转为下降0.2%，影响CPI下降约0.13个百分点。非食品中，大部分服务价格节后出现季节性回落，其中飞机票、交通工具租赁费和旅游价格分别下降12.0%、9.5%和6.5%，电影及演出票、美发、家庭服务、宠物服务等价格降幅在1.7%—5.6%之间。

从同比看，CPI上涨1.0%，涨幅比上月回落1.1个百分点。同比涨幅回落较多，主要是春节错月，上年同期对比基数较高。其中，食品价格上涨2.6%，涨幅比上月回落3.6个百分点，影响CPI上涨约0.48个百分点。食品中，薯类、鲜果、鸡蛋和禽肉类价格分别上涨14.4%、8.5%、7.9%和6.4%，涨幅均比上月有所回落；猪肉价格上涨3.9%，涨幅回落7.9个百分点；粮食和食用油价格分别上涨2.7%和6.4%，涨幅基本稳定；鲜菜和水产品价格由涨转降，降幅分别为3.8%和1.5%。非食品价格上涨0.6%，涨幅比上月回落0.6个百分点，影响CPI上涨约0.50个百分点。非食品中，服务价格上涨0.6%，涨幅比上月回落0.4个百分点，其中飞机票和旅游价格分别上涨19.9%和3.0%，涨幅均回落较多；能源价格上涨0.6%，涨幅比上月回落2.4个百分点，其中汽油和柴油价格分别上涨0.4%和0.3%，涨幅分别回落5.1和5.6个百分点；扣除能源的工业消费品价格上涨0.5%，涨幅基本稳定。

http://www.stats.gov.cn/tjsj/sjjd/202303/t20230309_1936941.html



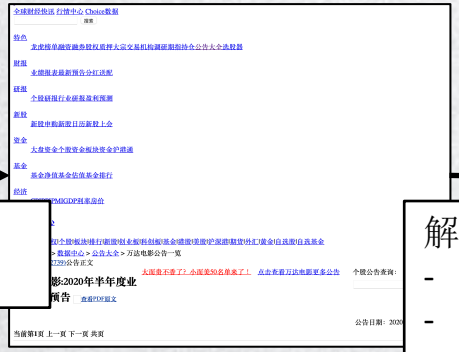
- 万得实时爬取国家统计局、证券交易所等网站数据、公告，并做初步的处理发送给投资者
- 金融领域对数据时效性要求极高，数据获取效率对证券市场的决策意义重大，因此以**更高效的爬虫算法**获取数据就成为了数据服务商战胜行业竞争对手的关键竞争优势

爬虫

- Step 1: 获取目标html文件
- Step 2: 解析目标html文件得到结构化数据
- Step 3: 将结构化数据存储成制定格式

原网页

request
response



获取html文件

解析html
- 正则表达式
- **Beautiful Soup**
- Xpath
- HTMLParser

标题：万达电影2020年半年度业绩预告
日期：2020年7月15日
正文：…

解析得到结构化数据

从开放API获取数据

- API接口开放平台是指一种提供给开发者使用的平台，开发者可以通过该平台获取到特定服务或功能的API接口。

扩展阅读：

使用微博**API**接口获取数据
https://www.jianshu.com/p/2d7fd22462b3

从关系数据库获取数据

- 以MySQL数据库为例
  - 创建连接
  - 写SQL语句
  - 执行SQL语句
  - 解析结果

```python
import datetime
import mysql.connector

cnx = mysql.connector.connect(user='scott', database='employees')
cursor = cnx.cursor()

query = ("SELECT first_name, last_name, hire_date FROM employees "
         "WHERE hire_date BETWEEN %s AND %s")

hire_start = datetime.date(1999, 1, 1)
hire_end = datetime.date(1999, 12, 31)

cursor.execute(query, (hire_start, hire_end))

for (first_name, last_name, hire_date) in cursor:
  print("{}, {} was hired on {:%d %b %Y}".format(
    last_name, first_name, hire_date))

cursor.close()
cnx.close()
```

https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html

提纲

1. 数据获取
2. 数据准备 – 数据抽取
3. 数据准备 – 数据清洗
4. 案例分析：wuhan2020-sos

探索数据分析

数据获取与准备

从文本数据中抽取结构化信息

新浪微博 weibo.com

天边的一朵小马驹_ +关注

2月11日 00:04 来自 肺炎患者求助超话

肺炎患者求助超话 【姓名】 XYZ

【年龄】56

【所在城市】 武汉

【所在小区、社区】汉阳区麒麟路麒麟社区 X栋Y单元Z楼

【患病时间】1月26日

【联系方式】 1397xxx6027

【其他紧急联系电话】 1341xxx1935

【病情详细描述】病人有高血压，甲状腺癌，发烧十天，高烧39度，双肺感染，咳嗽，无力，2月8日做了试剂盒，结果还没出来，现在病人病危，靠氧气，血氧70。医院不收入院。

【周边家人及隔离情况】家里还有一个姐姐同住，姐姐有糖尿病重症，也昏睡了十几天，咳嗽，乏力，至今也没隔离，连去医院的劲都没有。家里还有一位父亲抗美援朝的离休老干部90岁，于2月5日在家离世（是不是这个病离世未知）我已被隔离，妈妈目前情况非常严峻，只能躺床上靠吸氧根本不能动，连CT片都不能拍给我，我们报给社区说只能在家等，根本无力解决，现在情况紧急，生命危在旦夕，求求哪家医院能收治我的家人

- 示例：肺炎患者求助超话

转发！！！【所在小区、社区】汉阳区麒麟路麒麟社区\n 求助！

请使用Python的re库抽取属性名和取值，粘贴到讨论对话框中

从文本中抽取信息

新浪微博 weibo.com

天边的一朵小马驹_ +关注

2月11日 00:04 来自 肺炎患者求助超话

肺炎患者求助超话 【姓名】 XYZ

【年龄】56

【所在城市】 武汉

【所在小区、社区】汉阳区麒麟路麒麟社区 X栋Y单元Z楼

【患病时间】1月26日

【联系方式】 1397xxx6027

【其他紧急联系电话】 1341xxx1935

【病情详细描述】病人有高血压，甲状腺癌，发烧十天，高烧39度，双肺感染，咳嗽，无力，2月8日做了试剂盒，结果还没出来，现在病人病危，靠氧气，血氧70。医院不收入院。

【周边家人及隔离情况】家里还有一个姐姐同住，姐姐有糖尿病重症，也昏睡了十几天，咳嗽，乏力，至今也没隔离，连去医院的劲都没有。家里还有一位父亲抗美援朝的离休老干部90岁，于2月5日在家离世（是不是这个病离世未知）

我已被隔离，妈妈目前情况非常严峻，只能躺床上靠吸氧根本不能动，连CT片都不能拍给我，我们报给社区说只能在家等，根本无力解决，现在情况紧急，生命危在旦夕，求求哪家医院能收治我的家人

- 示例：肺炎患者求助超话

```
import re
blog = """转发！！！【所在小区、社区】汉阳区麒麟路麒麟社区\n
          求助！"""
m = re.search('【(.+)】(.+)',blog)
if m is not None:
    print (m.group(1),m.group(2))
```

所在小区、社区 汉阳区麒麟路麒麟社区

什么是正则表达式

- 正则表达式(Regular Expression)是一种文本模式，包括普通字符（例如，a 到 z 之间的字母）和特殊字符（称为"元字符"）。
- 正则表达式使用单个字符串来描述、匹配一系列匹配某个句法规则的字符串。
- 举例：
  - 11位手机号 ： 1[0-9]{10}
  - moon, moooon, mooooon… : m(oo)+n
  - moon, muun, moooon, muuuun: m((oo)+|(uu)+)n

为什么要用正则表达式

- replace(old, new[, max])

字符串中的 old（旧字符串） 替换成 new(新字符串)，如果指定第三个参数max，则替换不超过 max 次

- split(str,num)

split() 通过指定分隔符对字符串进行切片，如果参数 num 有指定值，则分隔 num+1 个子字符串

- strip([chars])

用于移除字符串头尾指定的字符（默认为空格或换行符）或字符序列。

```
169.237.46.168 - - [26/Jan/2014:10:47:58
-0800] "GET /stat141/Winter04/ HTTP/1.1" 200
2585 "http://anson.ucdavis.edu/courses/"
```

要求从这个日志数据中提取出年、月、日、时、分、秒，该如何操作

One possible solution:

```
day, month, rest = line.split(' [')[1].split(']')[0].split('/')
```

```
year, hour, minute, seconds = rest.split(' ')[0].split(':')
time_zone = rest.split(' ')[1]
```

为什么要用正则表达式

34sdg4gfwr4@ruc.edu.cn
653564309@qq.com
hehehe@gmail.com
chigua@sina.cn
34gdggfdgrtyhyr@163.com
... ...

如果验证一个字符串是否是电子邮箱呢？

依然自己写规则判断吗？

思考题

- 1：判断是否是合法邮箱

  邮箱的规则为，输入的只能是字母、数字、@以及.五种， @和.前后只能是字母或者数字

liyiyan@ruc.edu.cn √

pavlo@cs.cmu.edu √

datascience@@ruc.edu.com ×

hehehe@.gmail.com ×

- 2：使用正则表达式，提取下列字符串中的年/月/日/时/分/秒

```
169.237.46.168 - - [26/Jan/2014:10:47:58
-0800] "GET /stat141/Winter04/ HTTP/1.1" 200
2585 "http://anson.ucdavis.edu/courses/"
```

正则表达式写法

模式	描述
^	匹配字符串的开头。比如："^hhh"可以匹配"hhhhhehehe", 但不可以匹配"ehehehhh"。
$	匹配字符串的末尾。用法同上，不再赘述
. (这里有个dot)	匹配任意字符，除了换行符。如果要加换行符，应该用[.\n]
[...]	用来表示一组字符,单独列出：[amk] 匹配 'a'，'m'或'k'
[^...]	不在[]中的字符：[^abc] 匹配除了a,b,c之外的字符。
re*	匹配0个或多个的表达式。例如："a*"表示匹配0或多次a
re+	匹配1个或多个的表达式。例如："a+"表示匹配1或多次a
re?	匹配0个或1个由前面的正则表达式定义的片段，非贪婪方式。非贪婪方式的意思是匹配范围尽可能小，例如字符串为"<hhh>hehehe>"，正则表达式为"<.*>"的话，则匹配到"<hhh>hehehe>"整个，如果为"<.*?>"的话，则匹配到<hhh>即停止。

正则表达式写法

模式	描述
re{ n}	精确匹配 n 个前面表达式。例如， `o{2}` 不能匹配 "Bob" 中的 "o"，但是能匹配 "food" 中的两个 o。
re{ n,}	匹配 n 个前面表达式。例如， o{2,} 不能匹配"Bob"中的"o"，但能匹配 "foooood"中的所有 o。"o{1,}" 等价于 "o+"。"o{0,}" 则等价于 "o*"。
re{ n, m}	匹配 n 到 m 次由前面的正则表达式定义的片段，贪婪方式
a| b	匹配a或b
(re)	对正则表达式分组并记住匹配的文本。分组之后看作一个整体，比如"abc+"可以匹配"abcccccccc", 而"(abc)+"可以匹配"abcabcabcabc"
转义字符\	涉及到匹配规则用到的特殊字符时，前面加\表示其本身。例如要找到一个字符串"mit(eth)cmu"括号内的内容, 需要用"\(.*?\)"

正则表达式写法

特殊字符类实例	描述
\d	匹配一个数字字符。等价于 [0-9]。
\D	匹配一个非数字字符。等价于 [^0-9]。
\s	匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [\f\n\r\t\v]。
\S	匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。
\w	匹配包括下划线的任何单词字符。等价于'[A-Za-z0-9_]'。
\W	匹配任何非单词字符。等价于 '[^A-Za-z0-9_]'。

正则表达式写法

一些常用的简单实例	描述
[Pp]ython	匹配 "Python" 或 "python"
rub[ye]	匹配 "ruby" 或 "rube"
[aeiou]	匹配中括号内的任意一个字母
[0-9]	匹配任何数字。类似于 [0123456789]
[a-z]	匹配任何小写字母
[A-Z]	匹配任何大写字母
[a-zA-Z0-9]	匹配任何字母及数字
[^aeiou]	除了aeiou字母以外的所有字符
[^0-9]	匹配除了数字外的字符

example	matches	does not match
AABAAB	AABAAB	every other string
AA|BAAB	AA BAAB	every other string
AB*A	AA ABBBBBBA	AB ABABA
A(A|B)AAB	AAAAB ABAAB	every other string
(AB)*A	A ABABABABA	AA ABBA

正则表达式在python里的用法

此处只讲解最常用的函数、参数和功能

match	sub	search	findall
检查格式	替换子串	查找位置	找到全部

用法：

- re.match(pattern, string)，从字符串开头进行匹配
- re.sub(pattern, repl, string)，repl为替换的字符串，也可为一个函数
- re.search(pattern, string)，可以在字符串任何位置匹配
- re.findall(pattern, string)，找到所有满足匹配条件的子串

正则表达式在python里的用法

匹配对象方法：

- **span()**：返回匹配的范围(start, end)
- **group(num)**：在正则表达式中用()可以分组，group() 可以一次输入多个组号，在这种情况下它将返回一个包含那些组所对应值的元组。
- **groups()**：返回一个包含所有小组字符串的元组，从 1 到所含的小组号

```python
import re

line = "Cats are smarter than dogs"

matchObj = re.match( r'(.*) are (.*?) than (.*)', line)

if matchObj:
   print ("matchObj.group() : ", matchObj.group())
   print ("matchObj.span() : ", matchObj.span())
   print ("matchObj.group(1) : ", matchObj.group(1))
   print ("matchObj.group(2) : ", matchObj.group(2))
   print ("matchObj.group(2,3) : ", matchObj.group(2,3))
   print ("matchObj.groups() : ", matchObj.groups())
else:
   print ("No match!!")
```

```
matchObj.group() :  Cats are smarter than dogs
matchObj.span() :  (0, 26)
matchObj.group(1) :  Cats
matchObj.group(2) :  smarter
matchObj.group(2,3) :  ('smarter', 'dogs')
matchObj.groups() :  ('Cats', 'smarter', 'dogs')
```

课堂练习

- checkpoint1：判断是否是合法邮箱

邮箱的规则为，输入的只能是字母、数字、@以及.五种，@和.前后只能是字母或者数字

liyiyan@ruc.edu.cn √

pavlo@cs.cmu.edu √

datascience@@ruc.edu.com ×

hehehe@.gmail.com ×

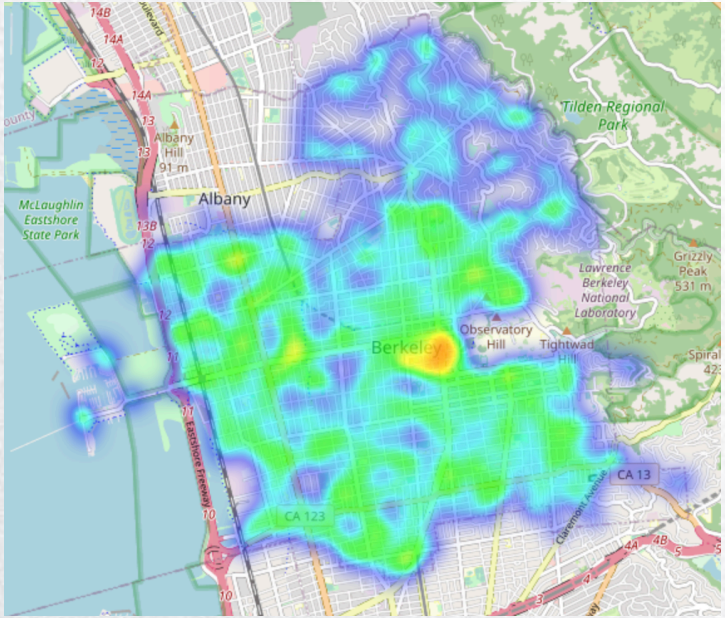
- checkpoint2：使用正则表达式提取，下列字符串中的年/月/日/时/分/秒

```
169.237.46.168 - - [26/Jan/2014:10:47:58
-0800] "GET /stat141/Winter04/ HTTP/1.1" 200
2585 "http://anson.ucdavis.edu/courses/"
```

从文本属性中抽取结构化信息

- 从文本数据中抽取结构化信息，进行丰富的可视化
  - 举例：从文本中提取位置信息

	CASENO	OFFENSE	EVENTDT	EVENTTM	CVLEGEND	CVDOW	InDbDate	Block_Location
0	18092358	THEFT MISD. (UNDER $950)	12/19/2018 12:00:00 AM	09:30	LARCENY	3	06/18/2019 10:28:09 AM	2000 FOURTH ST\nBerkeley, CA\n(37.867613, -122...
1	19000172	BURGLARY RESIDENTIAL	12/19/2018 12:00:00 AM	00:01	BURGLARY - RESIDENTIAL	3	06/18/2019 10:28:09 AM	1200 EVELYN AVE\nBerkeley, CA\n(37.883153, -12...
2	18069978	BURGLARY AUTO	12/19/2018 12:00:00 AM	08:07	BURGLARY - VEHICLE	3	06/18/2019 10:28:09 AM	2900 SAN PABLO AVE\nBerkeley, CA\n(37.853576, ...
3	18070016	THEFT MISD. (UNDER $950)	12/19/2018 12:00:00 AM	00:00	LARCENY	3	06/18/2019 10:28:09 AM	GRAYSON STREET & 7TH ST\nBerkeley, CA\n(37.855...
4	18070047	THEFT MISD. (UNDER $950)	12/19/2018 12:00:00 AM	16:08	LARCENY	3	06/18/2019 10:28:09 AM	2200 SHATTUCK AVE\nBerkeley, CA\n(37.869363, -...



练习

- 请使用三种方法抽取以下日志文件中的日期时间信息

log.txt

```
169.237.46.168 - - [26/Jan/2014:10:47:58 -0800] "GET /stat141/Winter04/ HTTP/1.1" 200 2585 "http://anson.ucdavis.edu/courses/"
193.205.203.3 - - [2/Feb/2005:17:23:6 -0800] "GET /stat141/Notes/dim.html HTTP/1.0" 404 302 "http://eeyore.ucdavis.edu/stat141/Notes/session.html"
169.237.46.240 - "" [3/Feb/2006:10:18:37 -0800] "GET /stat141/homework/Solutions/hw1Sol.pdf HTTP/1.1"
```

	0	1	0	1	2	3
0	26	Jan	2014	10	47	58
1	2	Feb	2005	17	23	6
2	3	Feb	2006	10	18	37

- 完成后示意老师
  - 提示：可使用字符串操作、正则表达式、datatime模块

提纲

1. 数据获取
2. 数据准备 – 数据抽取
3. 数据准备 – 数据清洗
4. 案例分析：wuhan2020-sos

探索数据分析

数据获取与准备

数据清洗有哪些挑战

- 现实世界中，数据通常是脏的 → Garbage In, Garbage Out
  - 数据存在错误和不一致

↓	C1	C2
	Total Cholesterol_1	Total Cholesterol_2
682	214.4	214.4
683	184.4	184.4
684	183.5	183.5
685	240.7	240.7
686	215.1	215.1
687	198.6	198.6
688	2800.0	280.0
689	210.8	210.8
690	182.5	182.5
691	192.6	192.6

数据清洗有哪些挑战

- 现实世界中，数据通常是脏的 → Garbage In, Garbage Out
  - 数据存在错误和不一致
  - 数据存在缺失（Missing）

Exhibit 2: Examples of variables that are set to unknown values

Administrative dates:	set to 0101YY, 010199, 999999
Date of Birth	0101YY, 1506YY, 3006YY, 0107YY, 1507YY, 0101YEAR
Names:	set to spaces, NK, UNKNOWN, or ZZZZ BABY, MALE, FEMALE, TWIN, TRIPLET, INFANT
Other variables:	set to 9, 99, 9999, -1 NK (Not Known) NA (Not applicable) NC (Not coded) U (Unknown)

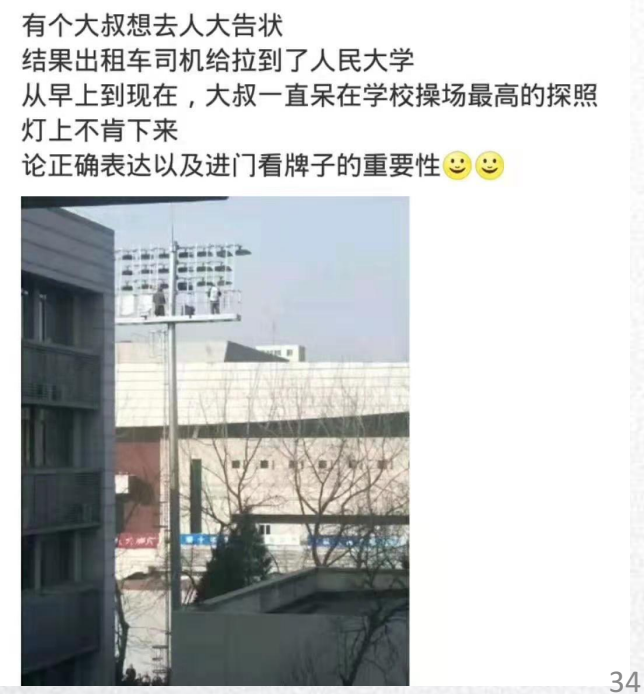
[Gill et al; Univ of Oxford 2003]

数据清洗有哪些挑战

- 现实世界中，数据通常是脏的 → Garbage In, Garbage Out
  - 数据存在错误和不一致
  - 数据存在缺失（Missing）
  - 名称/属性的二义性

Michael Jordan

人大



数据清洗的基本任务

- 将文本拆分成不同的属性（Fields） → 解决分隔符问题
  - 例：教师列表 Ju Fan: Associate Prof., Computer Science | 35
- 补充缺失的数据
  - 例：如果Ju Fan的年龄信息缺失，应该如何填充呢？
    - 平均值填充、用最近似教师年龄、贝叶斯估计

请写Python代码帮助范老师进行文本拆分，输出一个字符串数组

- 格式转换问题
  - 日期的表示：
- 异常值检测
  - 例： Salary =

```
info = 'Ju Fan: Associate Prof., Computer Science | 35'
fields = info.split(':')
print(fields)
fields = re.split('[:,|]', info)
print(fields)
```

```
['Ju Fan', ' Associate Prof., Computer Science | 35']
['Ju Fan', ' Associate Prof.', ' Computer Science ', ' 35']
```

如何做数据清洗

- 查看数据描述
  - 检查数据量和数据属性
  - 属性的类型、值域区间、关联，数据的可获得性
  - 从业务角度理解数据属性和属性值的含义
  - 计算每个属性的统计信息（如最大值、最小值、均值、方差等）

- 检测数据质量问题并修复
  - 检查数据值是否有异常值和错误
  - 值和属性本身的含义是否符合
  - 有无缺失值
  - 有无重复属性

使用pandas描述数据

- 以Titanic数据集为例

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
train_df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
train_df.describe(include=['O'])
```

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Dahlberg, Miss. Gerda Ulrika	male	CA. 2343	G6	S
freq	1	577	7	4	644

数据值的类型

- Nominal 标记值
  - ID、姓名、电话等，和数据项个数相当
- Categorical 离散值
  - 邮编、性别、型号等
- Ordinal 排序型
  - 分数分级A、B、C等
- Numerical 数值型
  - 度量值，薪水、高度等
- Interval 区间型
  - 数据值范围
- Ratio 比例

定性

定量

信息量增加

数据缺失

- 有数据的地方，通常也就有缺失的数据
  - 我们没有上帝视角
- 数据科学，必须面对数据缺失的问题
  - 仪器故障
  - 因为与以它记录冲突被删除
  - 人为因数没有输入
  - 当初录入数据时数据项没有被重视
  - 数据变化没有被记录
- 缺失数据对分析建模有很大影响
  - 忽略还是弥补，这是个问题

缺失值处理

- 忽略记录：保留全的记录
- 忽略属性：保留全的属性
- 手动填充：可行吗？
- 用一个全局值代替，例如-1， unknown
- 使用对应属性的平均值
- 使用相同类标签样本对应属性上的平均值
- 用最大可能的值推理
  - 如找最相似的点推理缺失值
  - 使用贝叶斯或决策树推理
- 缺失值填充：猜的越多，离真实数据越远

使用pandas处理缺失值

```
test_df.isnull().any()
```

```
PassengerId    False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare            True
Cabin           True
Embarked       False
dtype: bool
```

请问你会如何处理Age缺失的记录？

A. 将这些记录删除
B. 使用平均年龄填充
C. 使用年龄的中位数填充

使用pandas处理缺失值

```
test_df.isnull().any()
```

```
PassengerId    False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare            True
Cabin           True
Embarked       False
dtype: bool
```

```
test_df['Fare'].fillna(test_df['Fare'].median(), inplace=True)
test_df['Age'].fillna(test_df['Age'].median(),inplace=True)
```

```
test_df.drop(['PassengerId','Name','Ticket', 'Cabin'], axis=1,inplace=True)
test_df.head()
```

数值间的转换

- 排序型转换成数值型
  - A -> 4.0　A- ->3.7
  - B -> 3.3　B- ->3.0
- 多值属性处理
  - 颜色、国家、省/州等有限数值，可以用数值编号字典化处理数据
- ID属性
  - 一般保持不变。过长且带冗余信息的属性值，可以做压缩，注意在不同数据集上压缩ID解释的一致性
- 属性值分组
  - 省份按区域换分成5-6组

使用pandas处理数值转换

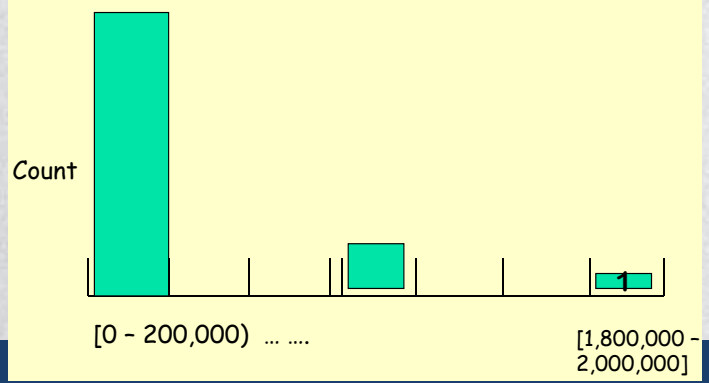
```
train_df['Sex'].replace(['male','female'],[0,1],inplace=True)
test_df['Sex'].replace(['male','female'],[0,1],inplace=True)

train_df['Embarked'].replace(['S','C','Q'],[0,1,2],inplace=True)
test_df['Embarked'].replace(['S','C','Q'],[0,1,2],inplace=True)
```

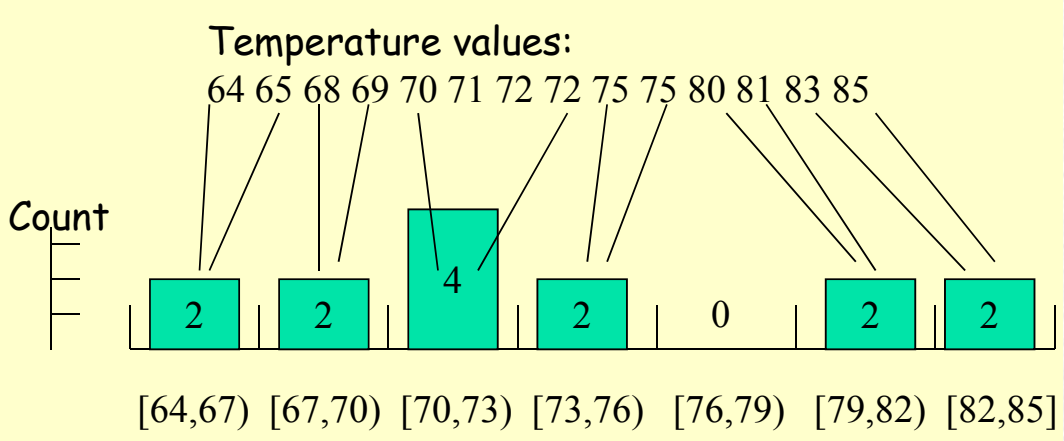
```
train_df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	target_name
0	0	3	0	22.0	1	0	7.2500	0.0	Not Survived
1	1	1	1	38.0	1	0	71.2833	1.0	Survived
2	1	3	1	26.0	0	0	7.9250	0.0	Survived
3	1	1	1	35.0	1	0	53.1000	0.0	Survived
4	0	3	0	35.0	0	0	8.0500	0.0	Not Survived

数据离散化处理

- 把连续型的数值数据分成区间型数据Binning
  - 如4分制到A/B/C转换
  - 可以减少数据量、压缩数据
- 等宽度划分离散化Equal-Width Binning
  - 优点：简单、易懂
  - 缺点：分成多少个桶？
  - 受噪音影响大





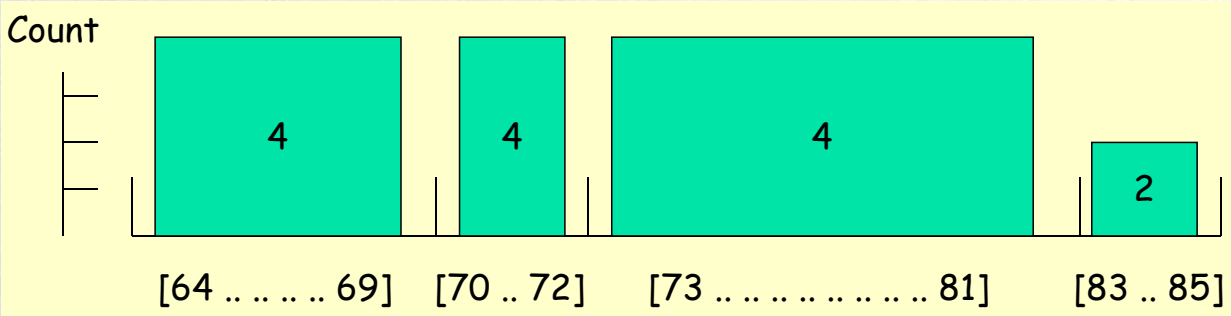
使用pandas进行数据离散化

```
bins = [18,25,35,60,100]
group_names = ['Youth','YounthAdult','MiddleAged','Senior']
cats = pd.cut(train_df['Age'],bins, labels = group_names)
cats
```

```
0            Youth
1       MiddleAged
2      YounthAdult
3      YounthAdult
4      YounthAdult
5              NaN
6       MiddleAged
7              NaN
8      YounthAdult
9              NaN
10             NaN
```

数据离散化处理

- 等高划分离散化 Equal-Depth Binning
  - 对比一下与EW的优缺点？

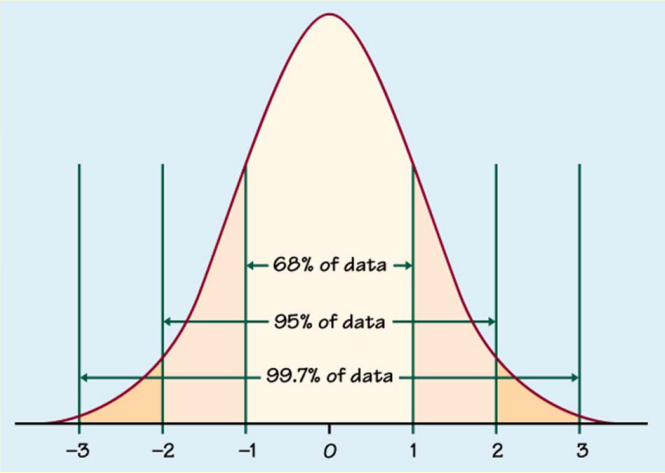


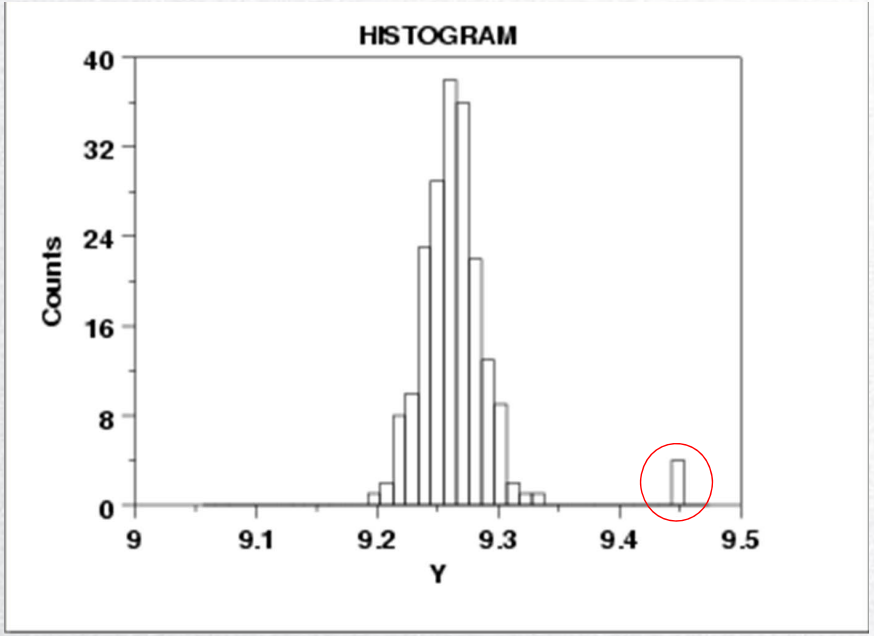
- 请将下面数据用EW和ED离散化处理
  - 13,15,16,16,19,20,21,22,22,25,30,33,35,35,36,40,45

发现异常值（outlier）

- 单变量
  - 统计方法，k=2 or 3

$$(\bar{x} - ks, \bar{x} + ks)$$





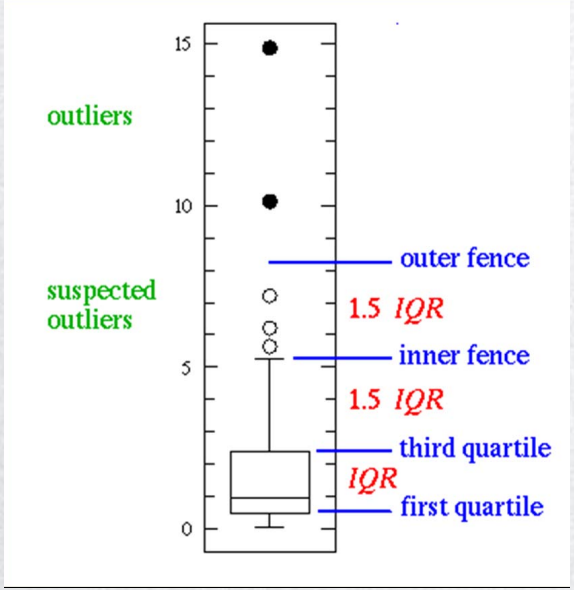
发现异常值（outlier）

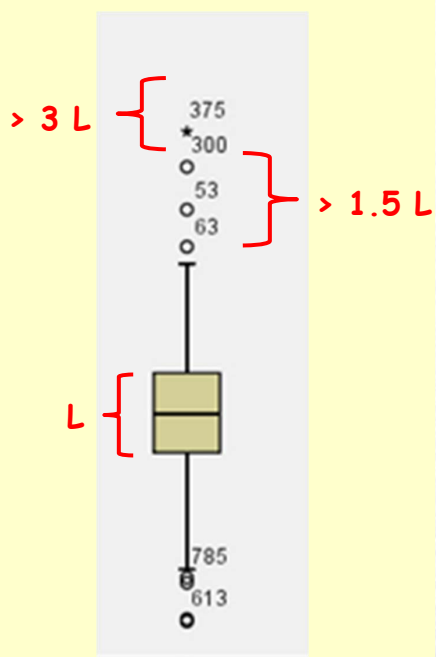
- 单变量
  - 超级异常值：

(Q1-3×IQR, Q3+3×IQR)， where IQR=Q3-Q1

  - 普通异常值

(Q1-1.5×IQR, Q3+1.5×IQR)





发现异常值（outlier）

- 单变量
  - 统计方法：k倍标准差之外（k = 2, 3）
  - 思考：这种方法存在什么问题？
  - 考虑一组年龄数据

1	20	21	21	22	26	33	35	36	37	39	42	45	47	54	57	61	62	400

  - 计算出：均值为56；标准差为83 → 两倍标准差区间[-109, 221]
  - 为什么会出现这种情况？
  - 均值Mean对异常值很敏感！
  - 怎么解决？中位数Median

数据归一化(normalization)处理

- Min-max归一化

$$v' = \frac{v - \min_v}{\max_v - \min_v}(new_max_v - new_min_v) + new_min_v$$

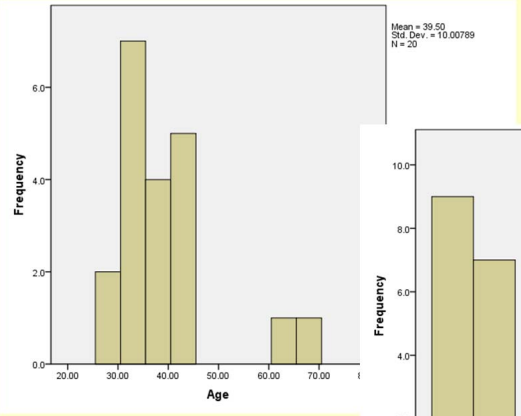
- Z-score归一化
  - 基于统计

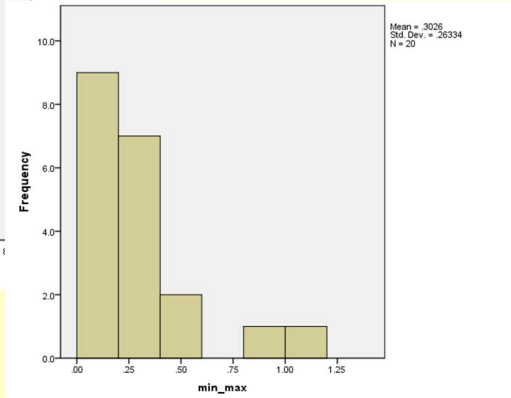
$$v' = \frac{v - \bar{v}}{\sigma_v}$$

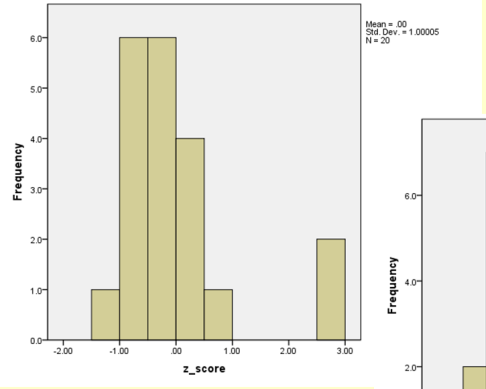
- 十位数归一化
  - j取值刚好让最大的数小于等于1

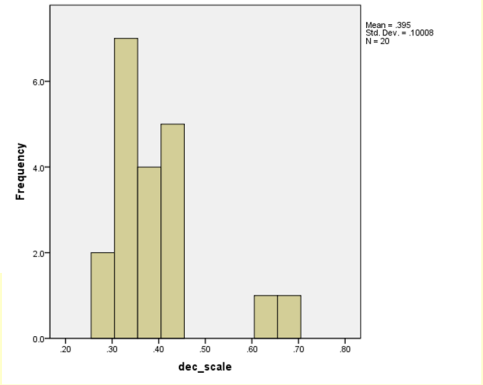
$$v' = \frac{v}{10^j}$$

Age	min-max (0-1)	z-score	dec. scaling
44	0.421	0.450	0.44
35	0.184	-0.450	0.35
34	0.158	-0.550	0.34
34	0.158	-0.550	0.34
39	0.289	-0.050	0.39
41	0.342	0.150	0.41
42	0.368	0.250	0.42
31	0.079	-0.849	0.31
28	0.000	-1.149	0.28
30	0.053	-0.949	0.3
38	0.263	-0.150	0.38
36	0.211	-0.350	0.36
42	0.368	0.250	0.42
35	0.184	-0.450	0.35
33	0.132	-0.649	0.33
45	0.447	0.550	0.45
34	0.158	-0.550	0.34
65	0.974	2.548	0.65
66	1.000	2.648	0.66
38	0.263	-0.150	0.38
28	minimun		
66	maximum		
39.50	avgerage		
10.01	standard deviation		









提纲

1. 数据获取
2. 数据准备 – 数据抽取
3. 数据准备 – 数据清洗
4. 案例分析：wuhan2020-sos

探索数据分析

数据获取与准备

数据准备

- 信息抽取：从文本数据中抽取出结构化信息

天边的一朵小马驹_ +关注

2月11日 00:04 来自 肺炎患者求助超话

肺炎患者求助超话 【姓名】 XYZ

【年龄】56

【所在城市】 武汉

【所在小区、社区】汉阳区麒麟路麒麟社区 X栋Y单元Z楼

【患病时间】1月26日

【联系方式】 1397xxx6027

【其他紧急联系电话】 1341xxx1935

【病情详细描述】病人有高血压，甲状腺癌，发烧十天，高烧39度，双肺感染，咳嗽，无力，2月8日做了试剂盒，结果还没出来，现在病人病危，靠氧气，血氧70。医院不收入院。

【周边家人及隔离情况】家里还有一个姐姐同住，姐姐有糖尿病重症，也昏睡了十几天，咳嗽，乏力，至今也没隔离，连去医院的劲都没有。家里还有一位父亲抗美援朝的离休老干部90岁，于2月5日在家离世（是不是这个病离世未知）

我已被隔离，妈妈目前情况非常严峻，只能躺床上靠吸氧根本不能动，连CT片都不能拍给我，我们报给社区说只能在家等，根本无力解决，现在情况紧急，生命危在旦夕，求求哪家医院能收治我的家人

危在旦夕，求求哪家医院能收治我的家人

id	name	age	city	addr
10815	ZBX	52	湖北武汉	长山社区万科嘉园小区
10842	ZBX	52	湖北武汉	长山社区万科嘉园小区
10473	CTC	75	武汉市洪山区	东湖御院 红星社区
84	DSX	51	武汉	云顶居，清水源社区
248	DSX		武汉	云顶居，清水源社区
565	DSX	51	武汉	云顶居，清水源社区
1205	RC	36	武汉	洪山区
1204	RC	36	武汉	洪山区
725	RC	36	武汉市	洪山区
729	RC	36	武汉市	洪山区
855	RC		湖北省武汉市	洪山区关山大道新竹路保利时代
854	RC	35	湖北省武汉市	保利时代
1826	RC	36	武汉	洪山区

你能观察到真实数据有多“脏”吗？

数据清洗

name	age	city	addr
ZBX	52	湖北武汉	长山社区万科嘉园小区
ZBX	52	湖北武汉	长山社区万科嘉园小区
CTC	75	武汉市洪山区	东湖御院 红星社区
DSX	51	武汉	云顶居，清水源社区
DSX		武汉	云顶居，清水源社区
DSX	51	武汉	云顶居，清水源社区
RC	36	武汉	洪山区
RC	36	武汉	洪山区
RC	36	武汉市	洪山区
RC	36	武汉市	洪山区
RC		湖北省武汉市	洪山区关山大道新竹路保利时代
RC	35	湖北省武汉市	保利时代
RC	36	武汉	洪山区

- 数据缺失
  - 属性age
- 数据重复
  - 同一个名字的人出现在多条微博中，是一个人吗？
- 属性值冲突
  - RC的年龄是35还是36？
- 属性值粒度粗细不一
  - 洪山区
  - 保利时代
  - 洪山区关山大道新竹路保利时代

数据扩充（Data Enrichment）

- 能否在地图上定位每位求助者所在的小区
- 能否找到每位求助者所在小区的平均房价和建造年份
- 数据扩充
  - 将求助者自己填写的小区名称对齐到链家小区名录中

name	age	city	addr
ZBX	52	湖北武汉	长山社区万科嘉园小区
ZBX	52	湖北武汉	长山社区万科嘉园小区
CTC	75	武汉市洪山区	东湖御院 红星社区
DSX	51	武汉	云顶居，清水源社区
DSX		武汉	云顶居，清水源社区
DSX	51	武汉	云顶居，清水源社区
RC	36	武汉	洪山区
RC	36	武汉	洪山区
RC	36	武汉市	洪山区
RC	36	武汉市	洪山区
RC		湖北省武汉市	洪山区关山大道新竹路保利时代
RC	35	湖北省武汉市	保利时代
RC	36	武汉	洪山区

实体链接

小区名	区县	街道	年份	二手房价
洪山区 027社区	洪山	老南湖	2008	18869
黄陂区 08经典	黄陂	盘龙城	2010	12231
硚口区 2008城市花	硚口	长丰常码头	2007	13682
武昌区 2008新长江	武昌	积玉桥	2005	22585
青山区 20街坊	青山	青山	1987	17574
青山区 22街坊	青山	青山	1985	16752

链家小区名录

数据扩充（Data Enrichment）

- 能否在地图上定位每位求助者所在的小区
- 能否找到每位求助者所在小区的平均房价和建造年份
- 数据扩充
  - 通过百度地图获取求助者对应到链家小区的经纬度坐标

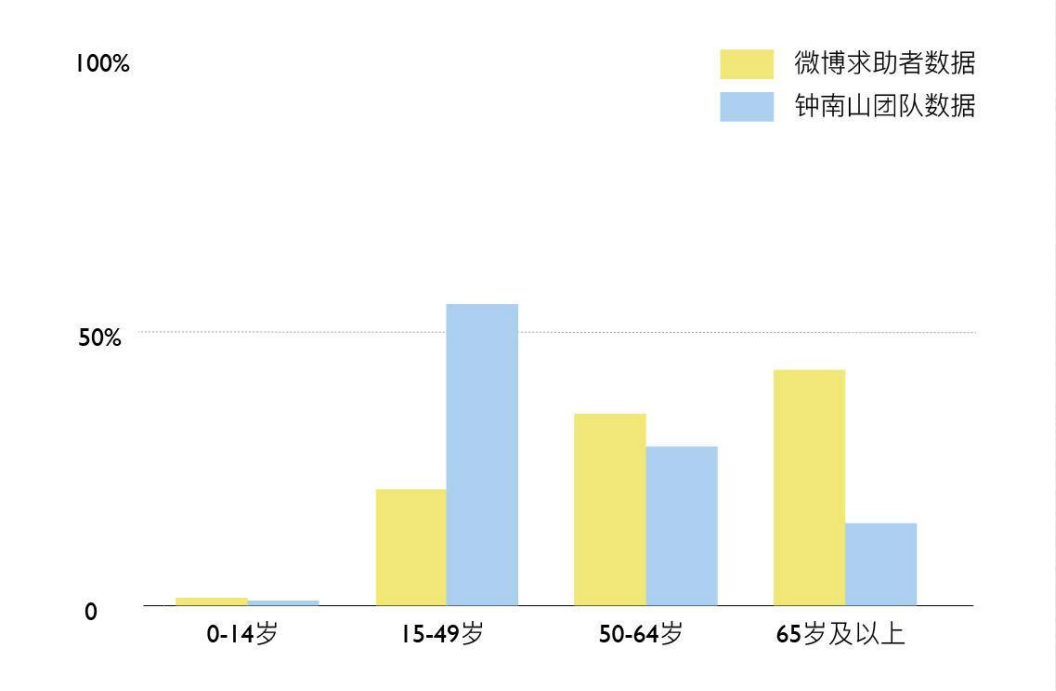
小区名	区县	街道	年份	二手房价
洪山区 027社区	洪山	老南湖	2008	18869
黄陂区 08经典	黄陂	盘龙城	2010	12231
硚口区 2008城市花	硚口	长丰常码头	2007	13682
武昌区 2008新长江	武昌	积玉桥	2005	22585
青山区 20街坊	青山	青山	1987	17574
青山区 22街坊	青山	青山	1985	16752

链家小区名录

调用百度API

数据分析

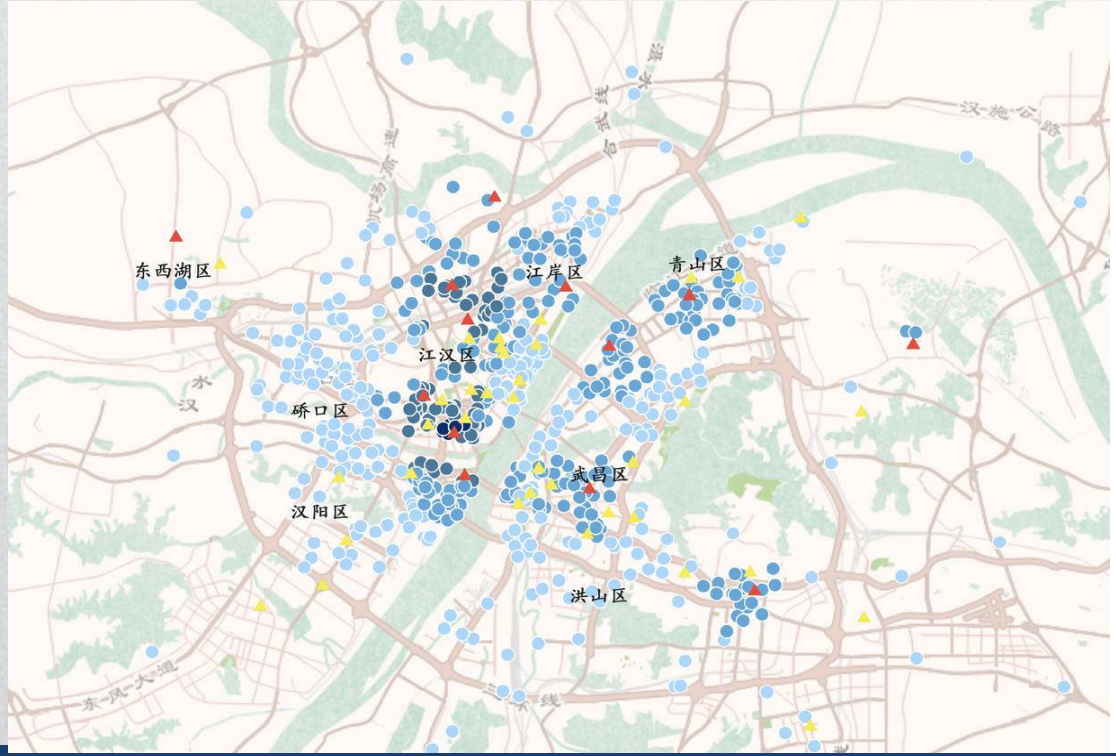
- **围城之中，越老越无助**



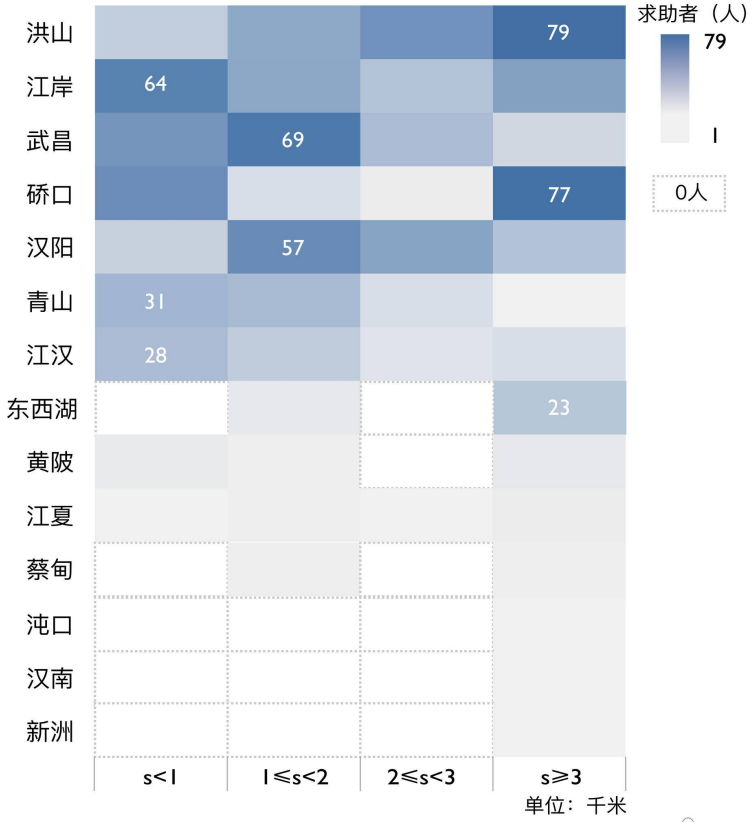
1183条求助微博的词频统计

数据分析

- **求助者离医院有多远？**

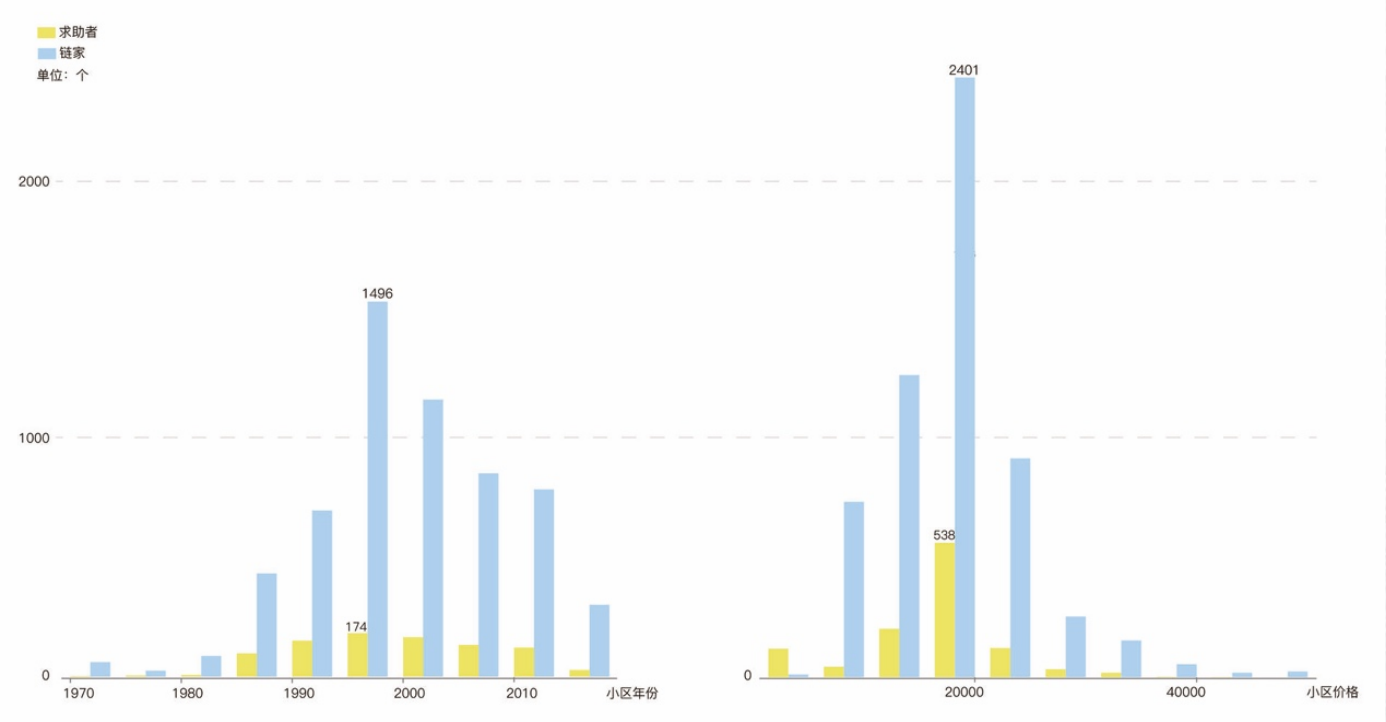


各区县求助者距最近医院的距离分布



数据分析

- **求助者所居住的房屋，无论是价格还是年份，都与武汉住宅的整体分布相当**



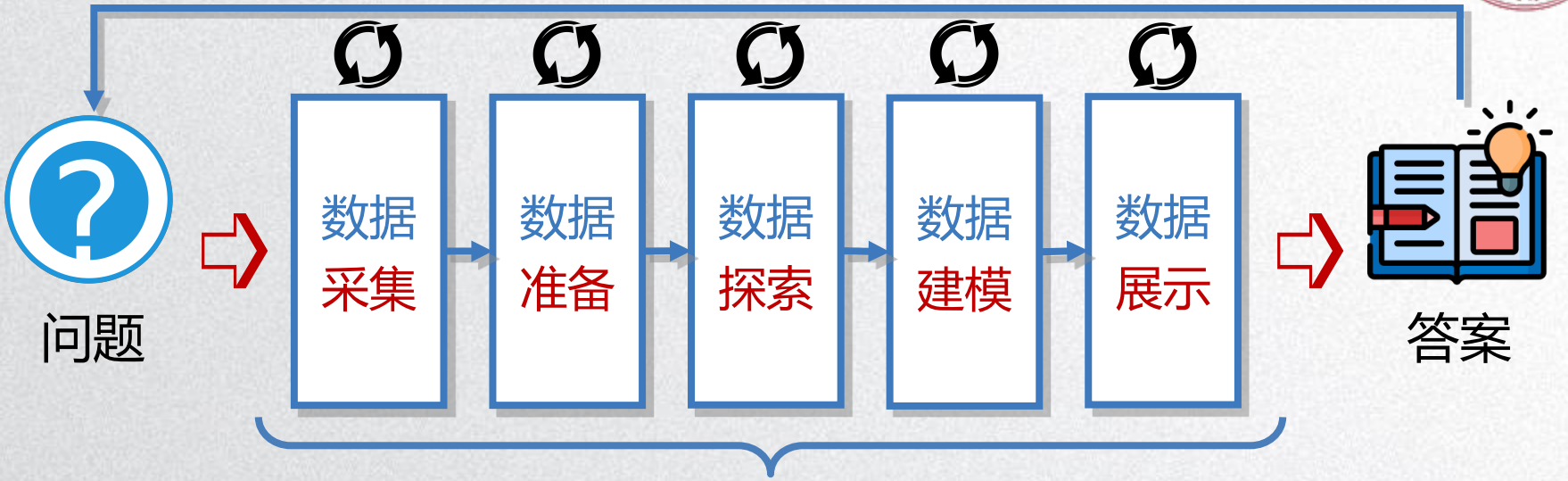
数据分析结论

- 时代的雨滴打在每一个人头上，数字背后甚至被排除在数字之外的，是一个个鲜活的人，用文字、图片、视频、转发、点赞……拼凑聚合成历史本身，而不甘愿沦为其注脚。

- 正如作家李静睿所言，“灾难面前，我们不是旁观，只是幸存。”这些陷入窘境的人、苦苦挣扎的人、奋力求生的人，不是弱者也不是他者，正是另一种处境下的你我。

- 照见他们，也是看到我们自己。

总结：数据探索分析与可视化



技术挑战

- 数据的多样性
- 数据的海量性
- 结果的及时性
- 数据的隐私保护
- ……