

数据思维与实践

王伟

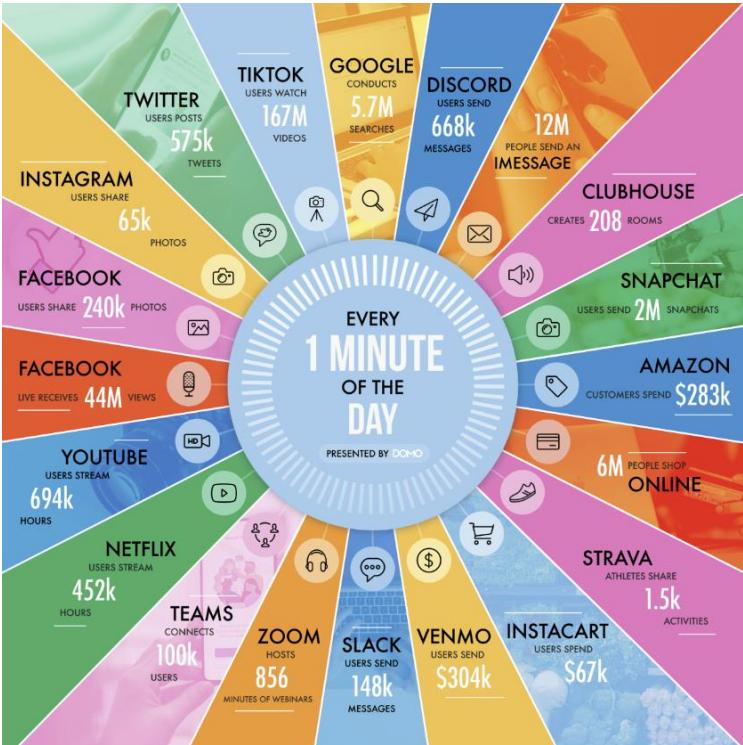
华东师范大学

数据科学与工程学院

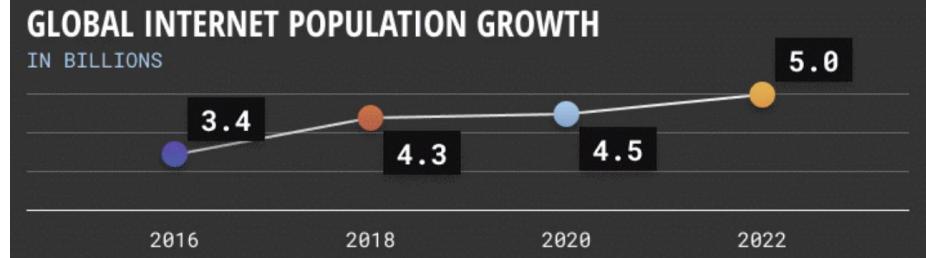
全民数字素养与技能培训基地



开篇实例：数据永不眠

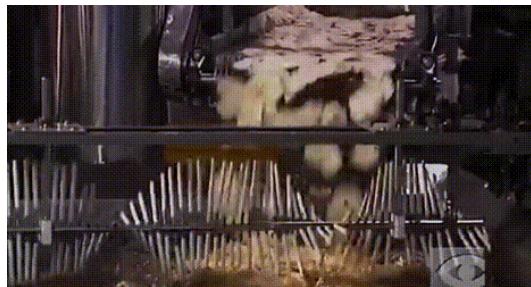
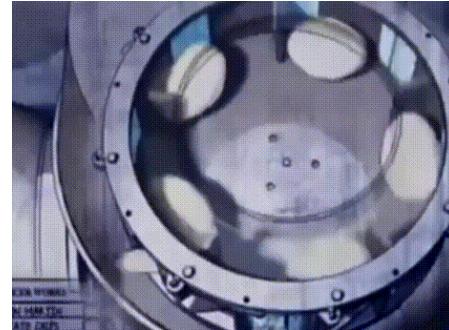


在过去的十年中，通过社交媒体、流媒体内容、在线购物、点对点支付以及其他活动进行数字互动的比例已经增长了数百甚至数千个百分点。尽管世界面临着大流行病、经济起伏和全球动荡，但社会中有一个恒定的事物：我们为了支持个人和商业需求——从连接沟通到进行交易和业务——对新数字工具的使用不断增加。



Data Never Sleeps 10.0: <https://www.domo.com/data-never-sleeps>

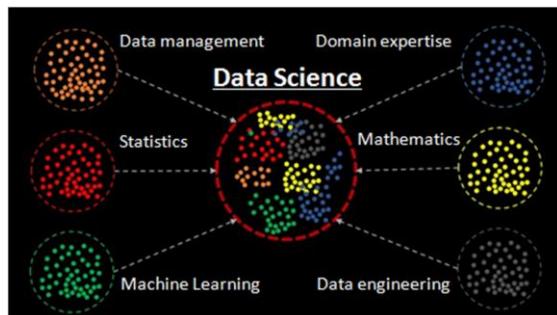
开篇实例：薯片流水线



1	2	3
4	5	6

数据思维与实践

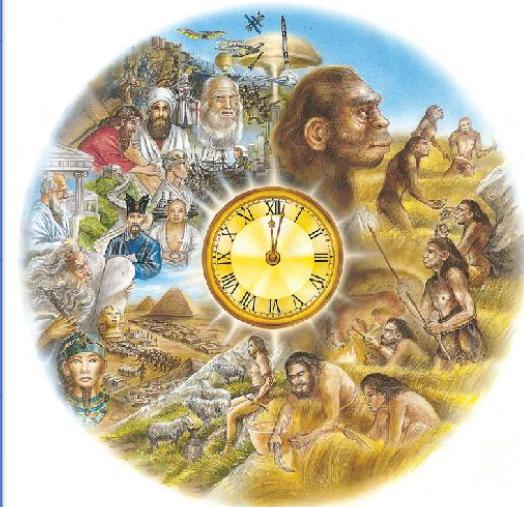
第03讲 数据收集与管理



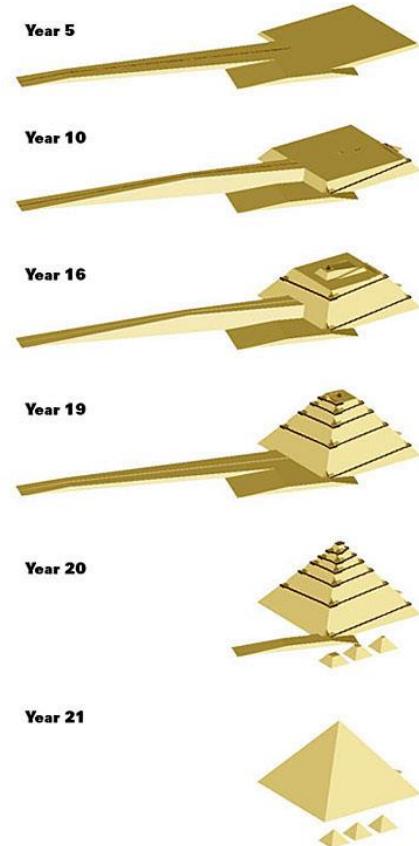
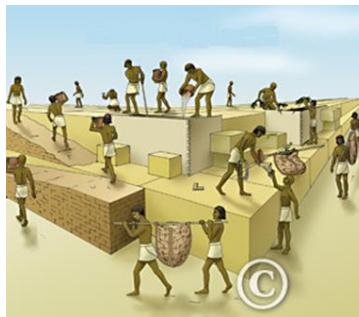
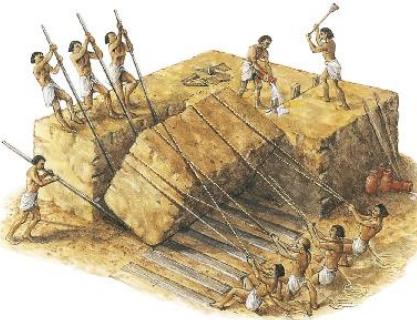
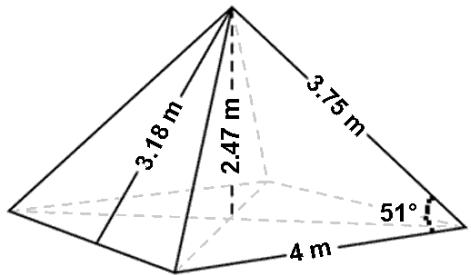
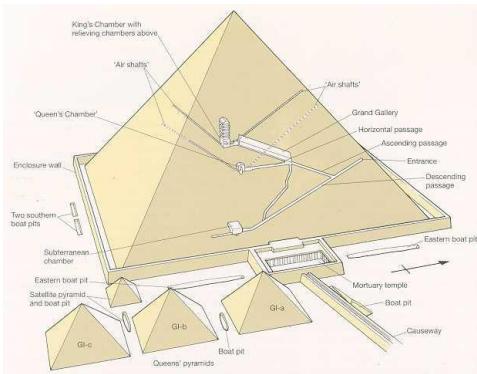
- 大数据时代
- 数据的全生命周期
- Python 数据收集方法
- 开源数字王国中的数据生态

人类文明的发展

原始文明	农业文明	工业文明	信息文明	人类未来
农业革命	工业革命	信息革命	智能革命 (解放体力) (解放脑力) (超越脑力)	
采集时代	农耕时代	机械时代	数字时代	智慧时代
人之力	物之力	能之力	算之力	智之力

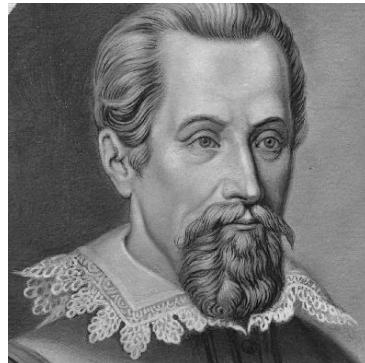
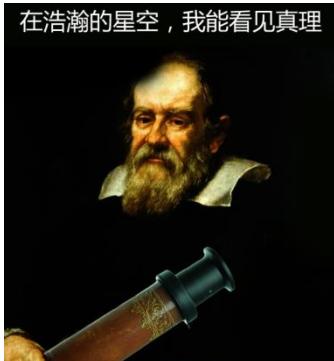
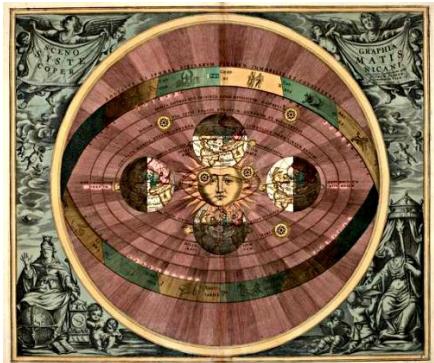


人类对数据的利用



通过古埃及人建造胡夫大金字塔的尺寸数据分析出
4600年前的古埃及人已知道勾股定理。

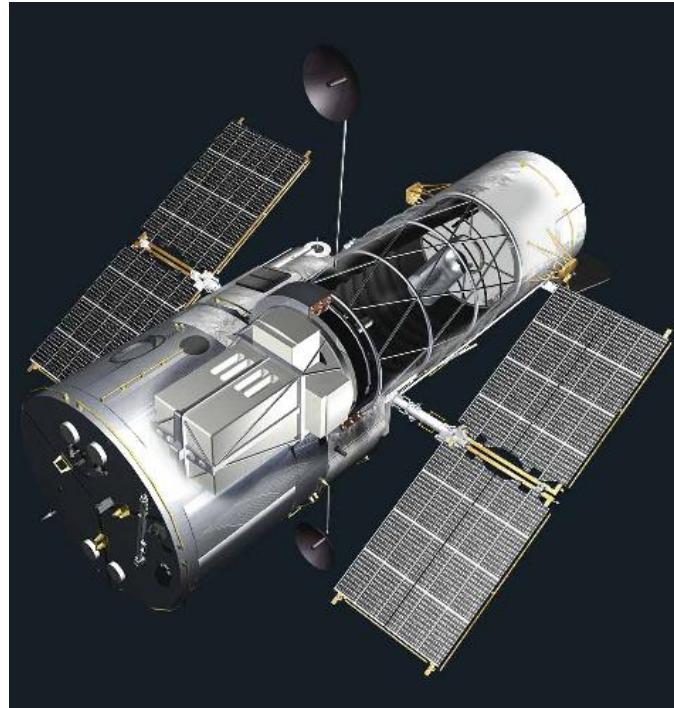
获得和利用数据的水平反映文明的水平



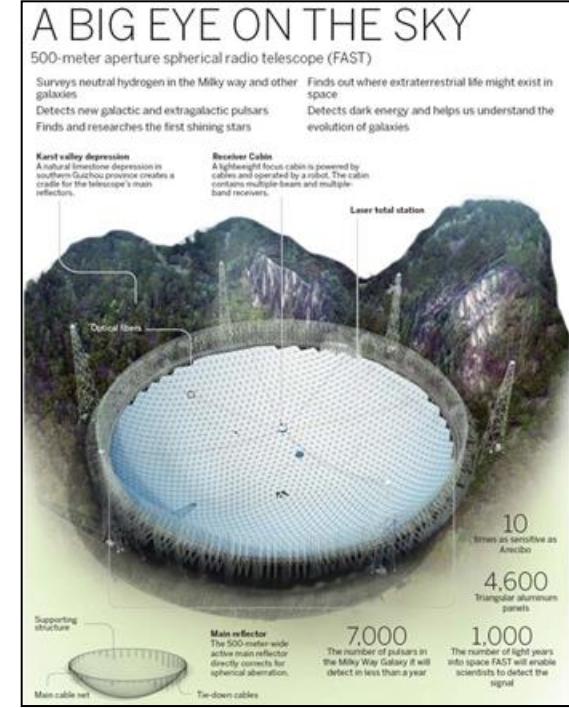
观测手段的飞跃



伽利略的望远镜/1609



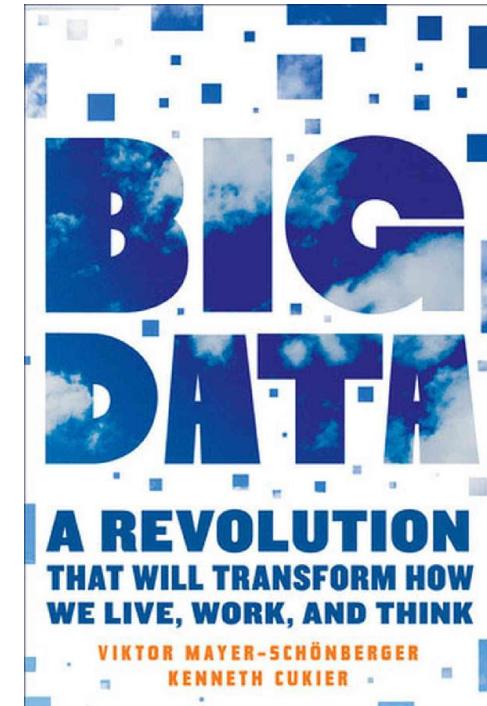
哈勃太空望远镜/1990



“天眼” FAST射电望远镜/2016

六次信息革命

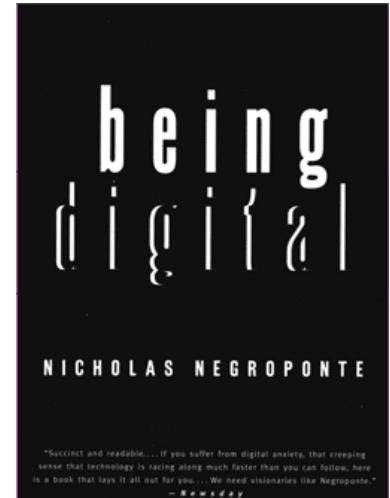
信息革命	技术
第一次信息革命	语言
第二次信息革命	文字
第三次信息革命	造纸术及印刷术
第四次信息革命	电报、电话和电视
第五次信息革命	计算机、互联网和物联网
第六次信息革命	大数据和人工智能



大数据是信息革命的必然趋势

- 大数据是我们数字化到一定阶段之后，必然出现的一个自然现象，这种史无前例的变化有几个主要的驱动力：

- 摩尔定律驱动的指数增长模式（**比特化**）
- 互联网驱动的人机物大规模互联（**网络化**）
- 技术创新驱动的万物数字化（**数字化**）

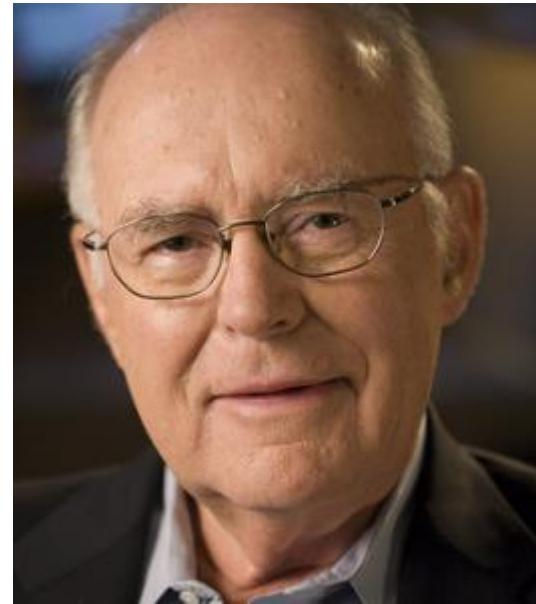
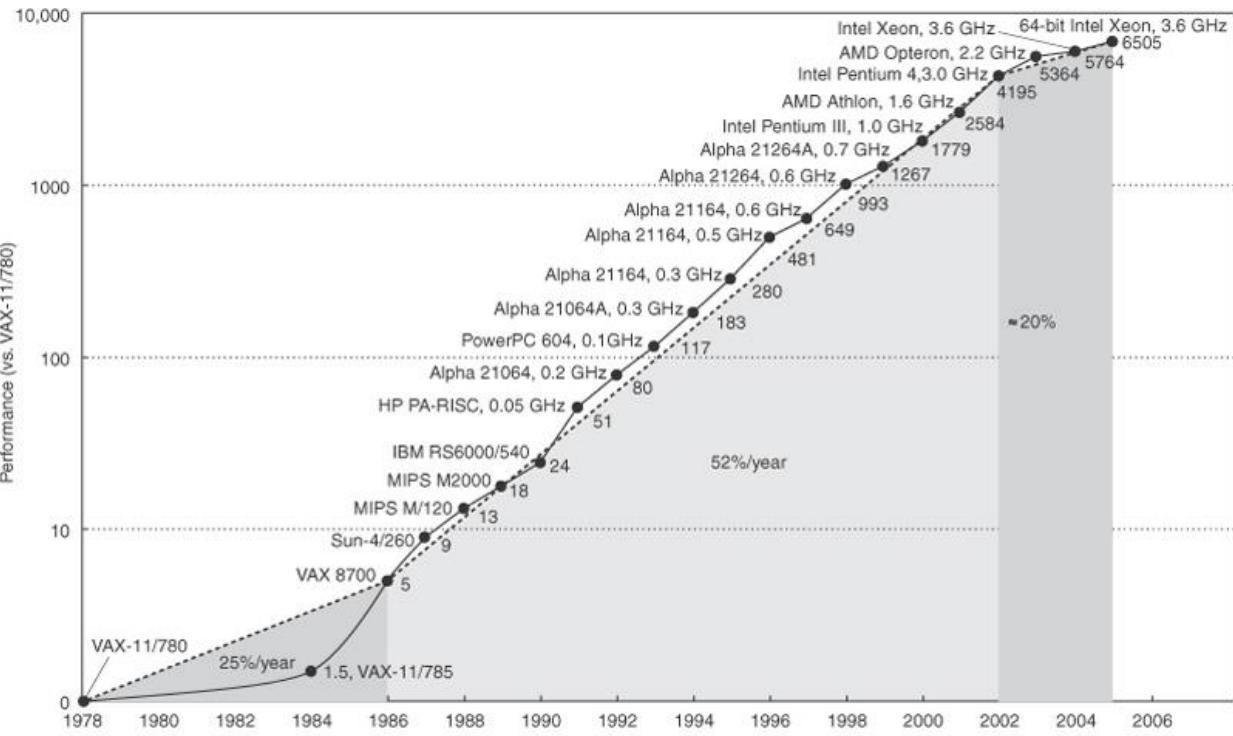


尼葛洛庞帝，《数字化生存》，1995

比特化

- 1965 年微芯片上的元件数增加了 1 倍, **Gordon Moore** 于是预言这一趋势近期内将继续。1975 年他修改为每两年翻一番, 后来又说是 18 个月, 或者说按指数律增长 (每年 46%)。这就是著名的**摩尔定律**。
 - **美国的主粮玉米**, 从 1950 年以后平均产量每年增长 2%;
 - **蒸汽涡轮发电机**, 热能转换为电能效率, 在 20 世纪年增长率为 1.5%;
 - **室内灯光有效性**, 1881 - 2014 年平均增长 2.6%, 而室外为 3.1%;
 - **洲际旅行远洋客轮**, 效率平均每年提高 5.6%;
 - **汽车的燃油**, 1973 - 2014 的效能转换率, 年平均提高 2.5%。

摩尔定律



摩尔定律的结果

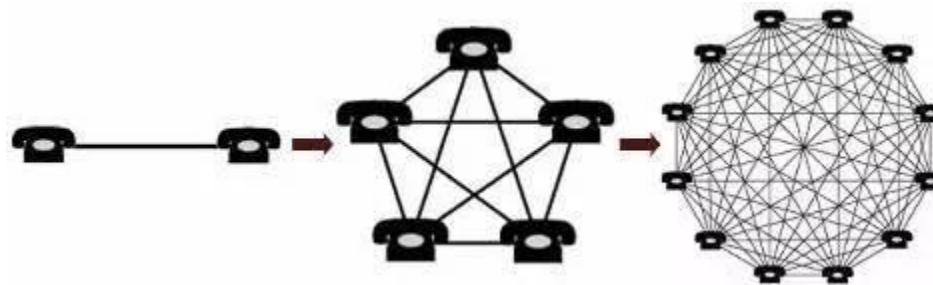
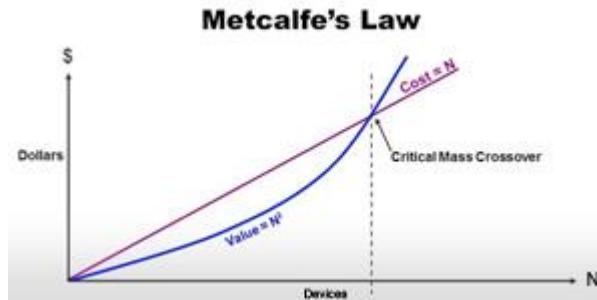
- 元件越来越小、越来越密、越来越快、越来越便宜
- 数据/信息的采集、存储、分析、展示越来越方便



网络化

- 梅特卡夫定律

- 网络价值随着用户数量的平方数增加而增加
- 联网的用户越多，网络的价值越大，联网的需求也就越大



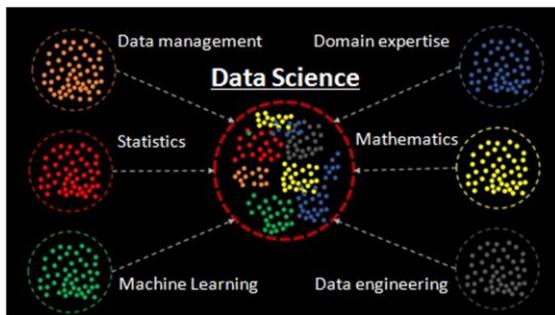
数字化产生新物种

- 数字企业、政府信息化、数字个体
- 智慧城市、虚拟现实、人工智能
- 交互界面、数据、算法、软件定义世界、X-GPT



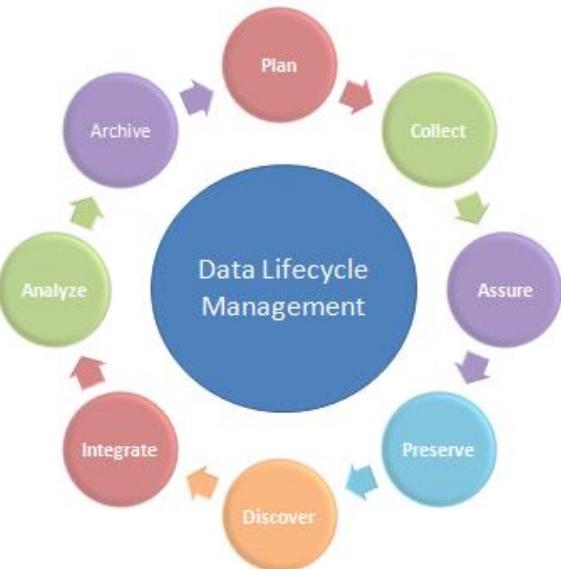
数据思维与实践

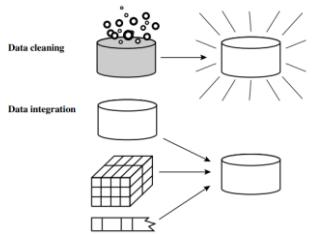
第03讲 数据收集与管理



- 大数据时代
- **数据的全生命周期**
- Python 数据收集方法
- 开源数字王国中的数据生态

数据的全生命周期





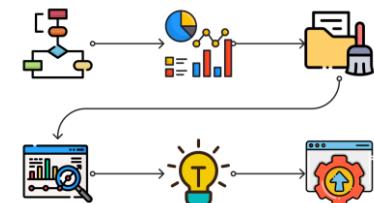
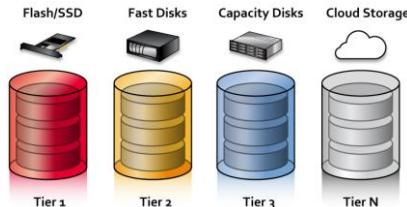
数据采集

数据预处理

数据存储

数据管理

数据分析



数据生成的模式

- **阶段1：20世纪90年代**

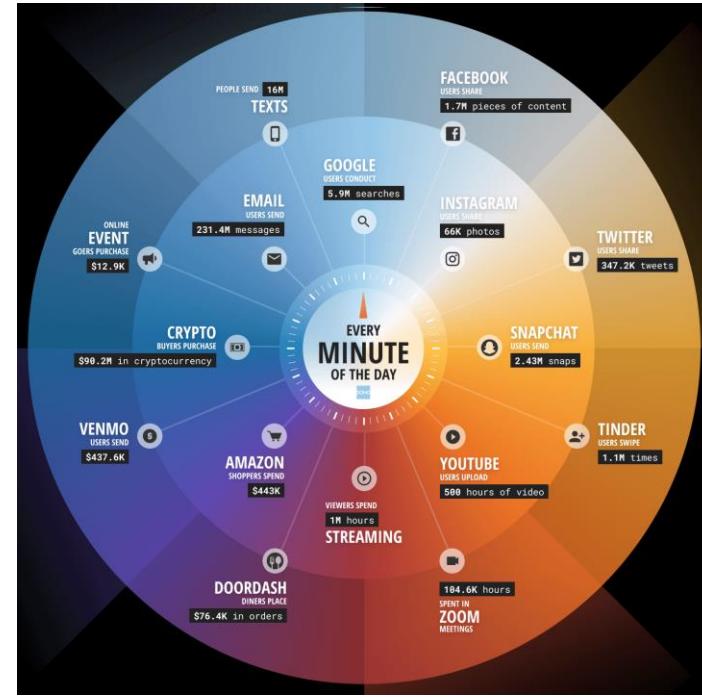
- 数字技术和数据库系统
- 企业信息的管理系统
- 结构化数据集

- **阶段2：Web系统的日益流行**

- 搜索引擎和电子商务
- Web 2.0和社交网络
- 半结构化和无结构的数据

- **阶段3：移动设备的普及**

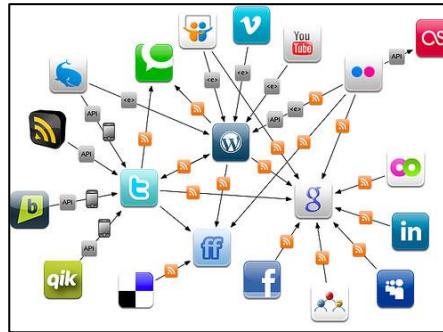
- 智能手机、平板、移动APP
- 物联网
- 结构化、半结构化、无结构化数据



三类典型数据



商业数据



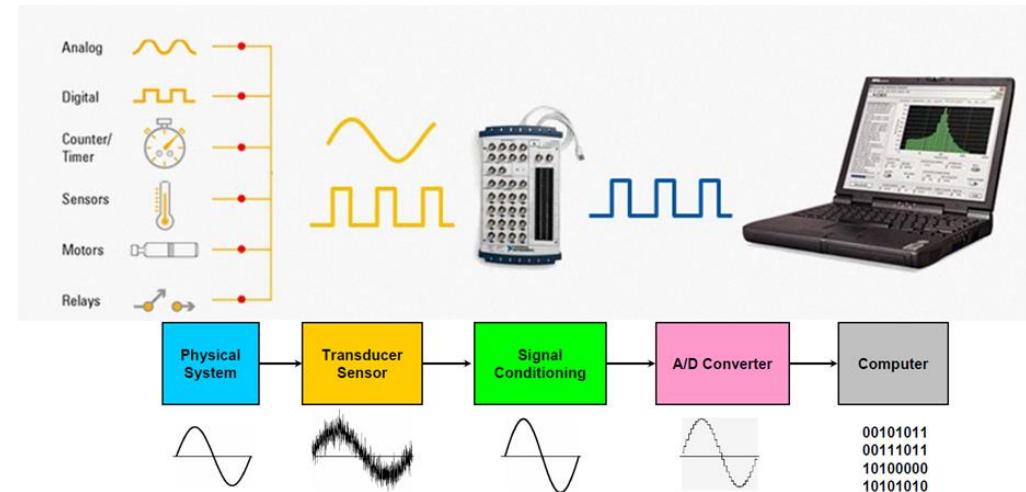
网络数据



科学研究数据

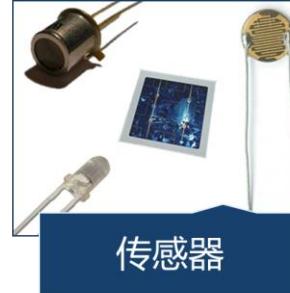
数据获取步骤

- 数据获取阶段的任务是以数字形式将信息聚合, 以待存储和分析处理, 数据获取过程可分为三个步骤:
 - 数据采集
 - 数据传输
 - 数据预处理



1、数据采集

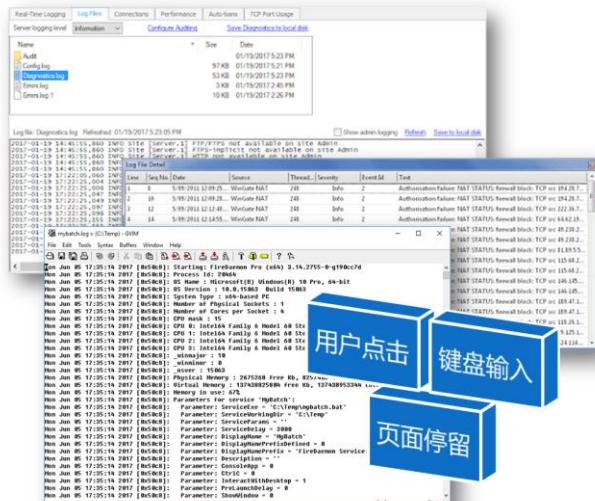
- 数据采集是指从真实世界对象中获得原始数据的过程。采集方法不但要依赖于数据源的物理性质，还要考虑数据分析的目标。
- 三种常用的数据采集方法：
 - 传感器
 - 日志文件
 - Web 爬虫



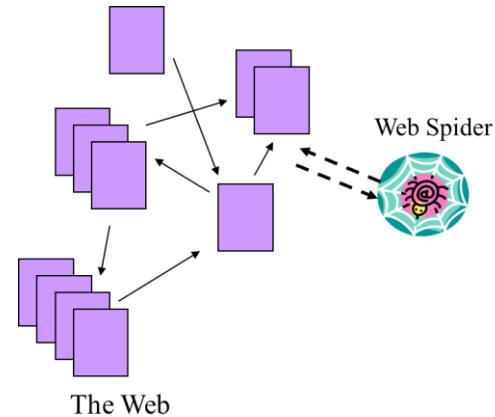
常见数据采集方法



传感器



日志文件

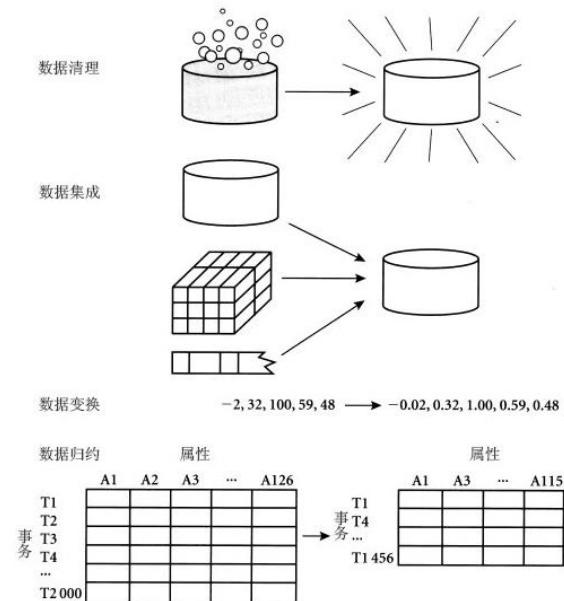


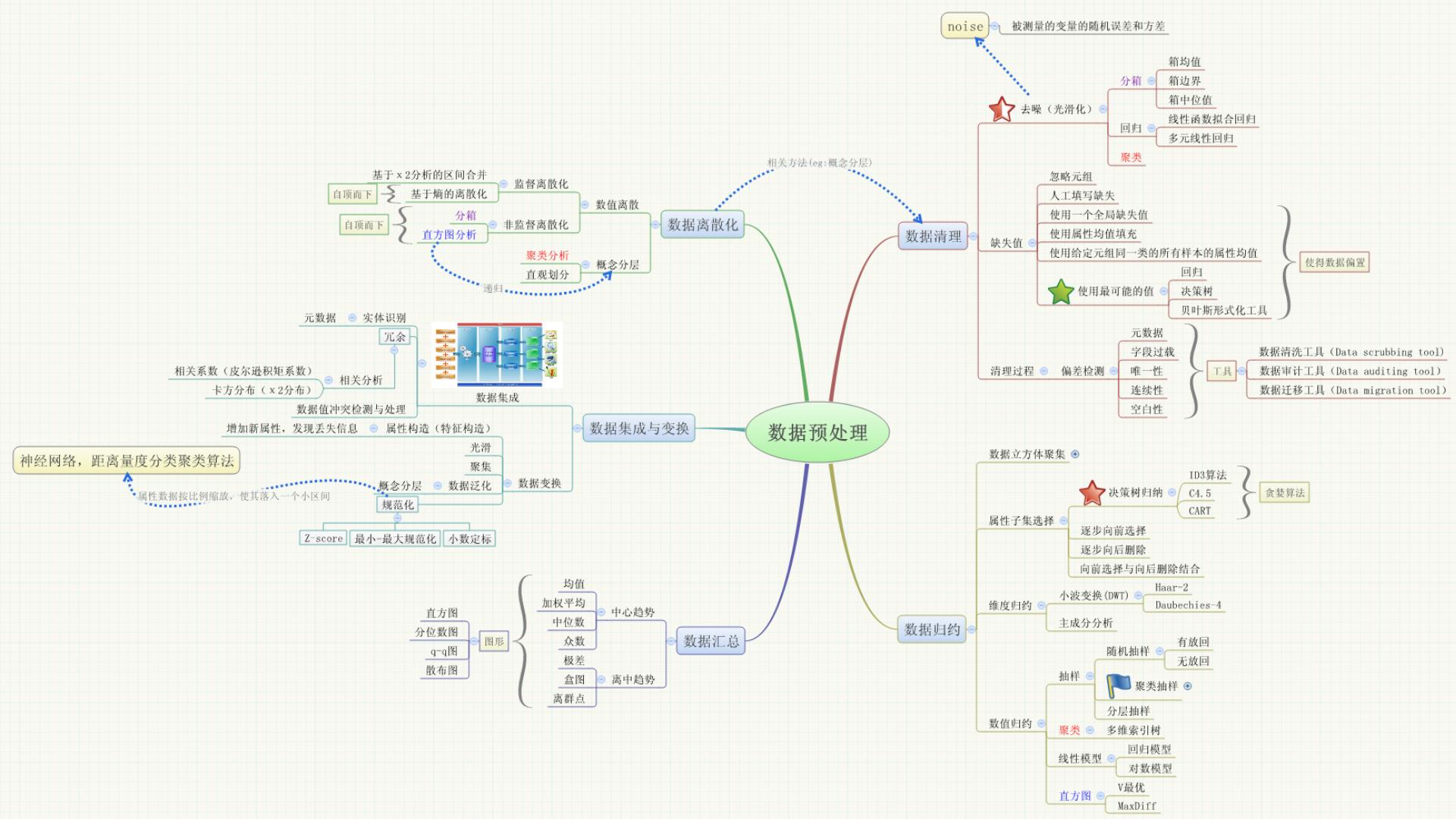
The Web

Web 爬虫

2、数据预处理

- 由于数据源的多样性，数据集由于干扰、冗余和一致性等因素的影响具有不同的质量。
- 一些数据分析工具和应用对数据质量有着严格的要求。因此在大数据系统中需要数据预处理技术提高数据的质量。
 - 数据集成 (Data integration)
 - 数据清洗 (Data cleansing)
 - 冗余消除 (Redundancy elimination)





3、数据存储

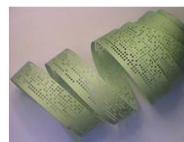
- 数据以某种格式记录在计算机内部或外部存储介质上。

存储结构

- 文件：txt、xls、xml、json
- 数据库：关系模型
- 对象图
- 属性列表

存储方式

- 磁带
- 随机存取存储器（RAM）
- 磁盘和磁盘阵列（RAID）
- 存储级存储器：SSD
- 光盘



4、数据管理

student

姓名	学号	班级	年龄	性别	住址	课号	电话
张三	100	计91	20	男	上海杨浦	上海	89150
李四	200	计92	19	男	上海徐汇	上海	88888
王五	300	计93	18	女	上海浦东	上海	77777
赵六	400	计94	19	女	上海静安	上海	99999
刘七	500	计95	21	男	上海普陀	上海	88666

course

课程名	课号	地点	教师
DB	1	5101	周老师
DB	2	5102	钱老师
DM	3	5103	金老师

关系数据库示例

grade

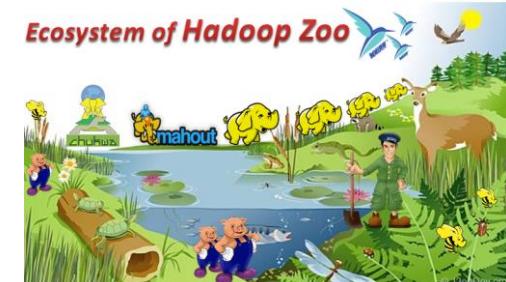
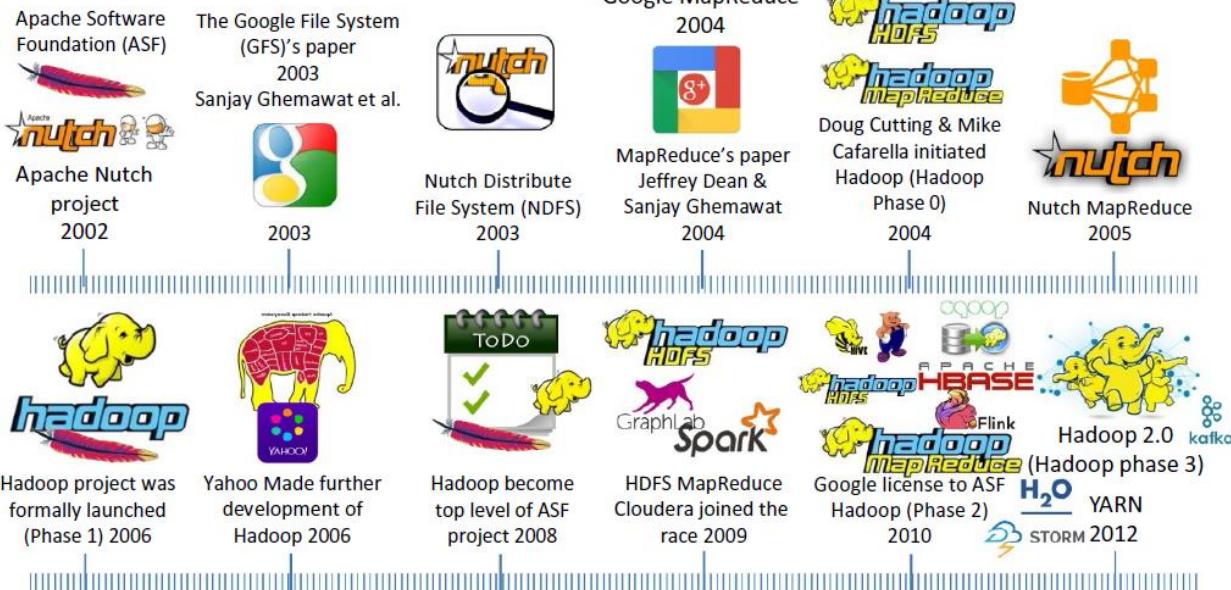
学号	课号	分数
100	1	99
200	1	98
300	2	97

结构化查询语言 (SQL)

```
SELECT name, course, grade  
FROM grade2020  
WHERE grade >= 70  
ORDER BY name DESC;
```



大数据软件生态



5、数据分析

- **描述性分析** (Descriptive analysis) : 发生了什么 (过去与现在)
- **诊断性分析** (Diagnostic analysis) : 发生的原因 (动因与洞察)
- **预测性分析** (Predictive analysis) : 将要发生什么 (趋势与可能)
- **指导性分析** (Prescriptive analysis) : 应该做什么 (决策与优化)

描述性统计
(**Descriptive Statistics**)

探索性统计
(**Exploratory Statistics**)

推断性统计
(**Inferential Statistics**)

统计决策
(**Statistical decision**)

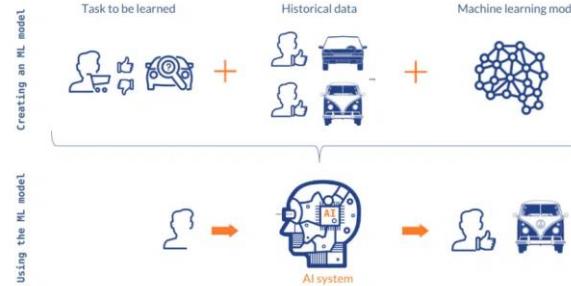
数据分析的方法

常用方法

统计方法
可视化方法
机器学习方法

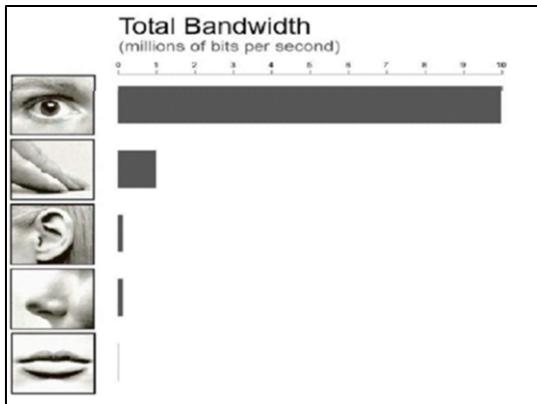


A statistical model

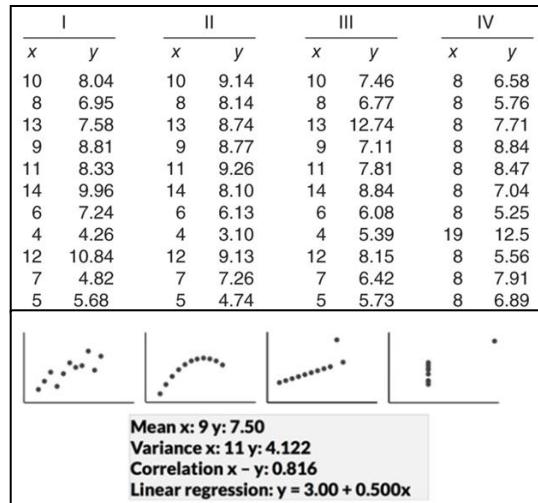


为什么数据可视化?

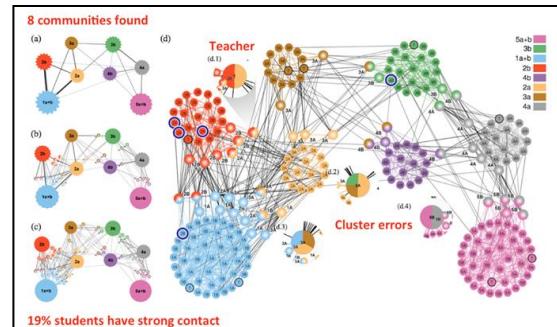
认知效率



认识全面

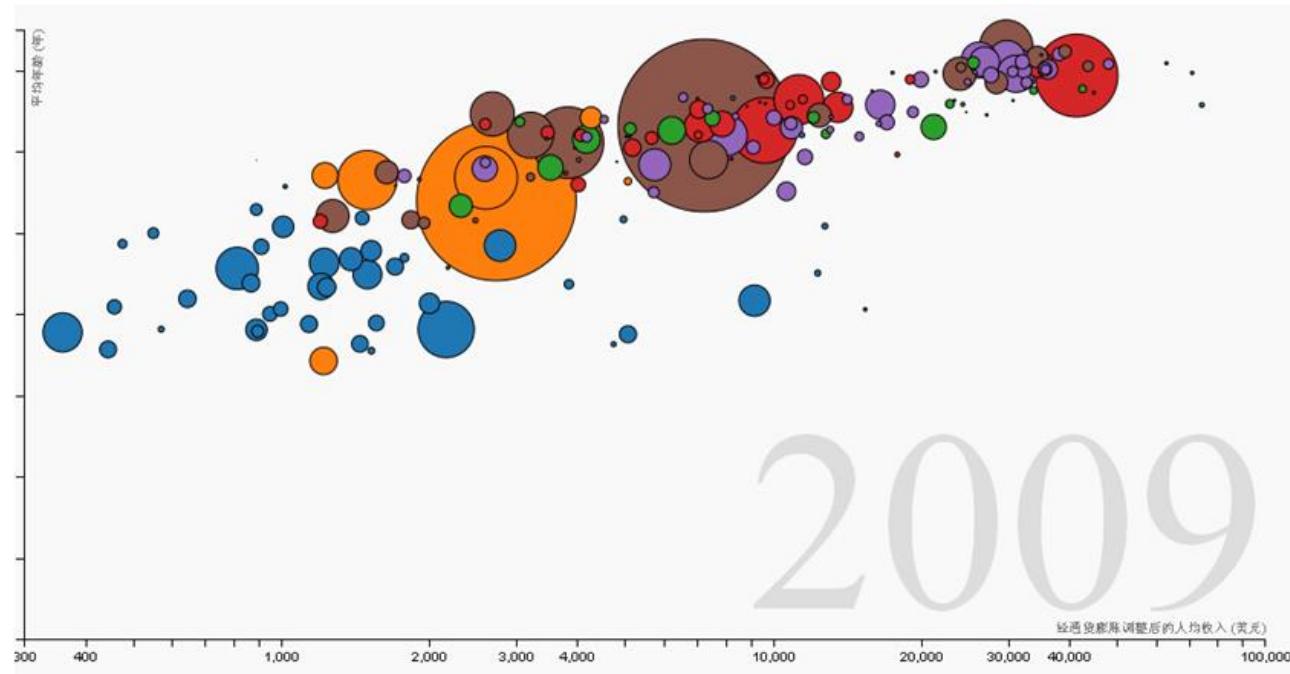


化繁为简



可视化案例：世界国家健康与财富之间的关系

利用可视化技术，把世界上200个国家，从1810年到2010年历时200年其各国国民的健康、财富变化数据制作成三维动画进行了直观展示。





可视化案例：上海地铁系统进站流量图

工作日

公交换乘地铁站点统计图

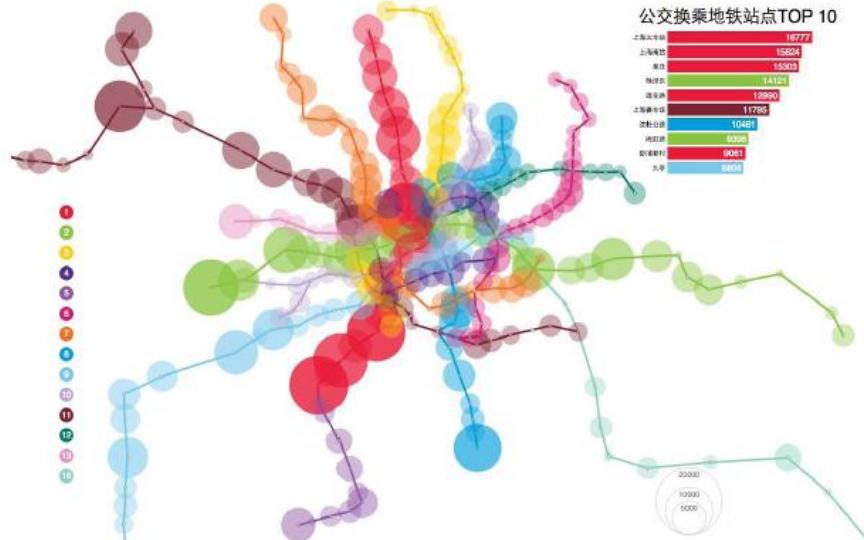
工作日 休息日



休息日

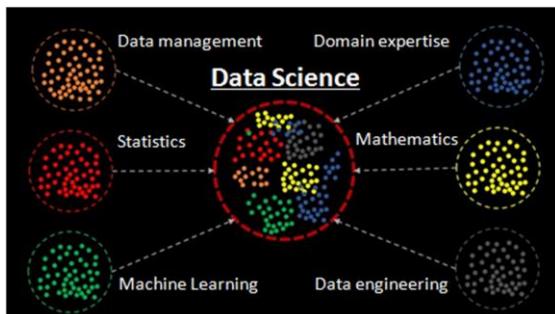
公交换乘地铁站点统计图

工作日 休息日



数据思维与实践

第03讲 数据收集与管理



- 大数据时代
- 数据的全生命周期
- Python 数据收集方法
- 开源数字王国中的数据生态

1、 studin 和 stdput

- studin 和 stdput 以管道的形式传递数据

```
# egrep.py
import sys, re

# sys.argv is the list of command-line arguments
# sys.argv[0] is the name of the program itself
# sys.argv[1] will be the regex specified at the command line
regex = sys.argv[1]

# for every line passed into the script
for line in sys.stdin:
    # if it matches the regex, write it to stdout
    if re.search(regex, line):
        sys.stdout.write(line)
```

2、读取文件

- 显式地用代码读取文件，用 open 获取文件对象

```
# 'r' means read-only, it's assumed if you leave it out
file_for_reading = open('reading_file.txt', 'r')
file_for_reading2 = open('reading_file.txt')

# 'w' is write -- will destroy the file if it already exists!
file_for_writing = open('writing_file.txt', 'w')

# 'a' is append -- for adding to the end of the file
file_for_appending = open('appending_file.txt', 'a')

# don't forget to close your files when you're done
file_for_writing.close()
```

3、网络抓取

- Web 上面的页面使用 HTML 编写的

```
<html>
  <head>
    <title>A web page</title>
  </head>
  <body>
    <p id="author">Joel Grus</p>
    <p id="subject">Data Science</p>
  </body>
</html>
```

- 使用 BeautifulSoup 库

```
from bs4 import BeautifulSoup
import requests

# I put the relevant HTML file on GitHub. In order to fit
# the URL in the book I had to split it across two lines.
# Recall that whitespace-separated strings get concatenated.
url = ("https://raw.githubusercontent.com/"
       "joelgrus/data/master/getting-data.html")
html = requests.get(url).text
soup = BeautifulSoup(html, 'html5lib')
```

4、使用 API

- 许多网站和 Web 服务提供应用程序接口（Application programming interface, API），可以显式的请求结构化格式数据。
- 由于 HTTP 适用于传输文本的协议，因此通过 API 请求的数据需要序列化为字符串格式，并使用 JavaScript 对象符号（JSON），与 Python 的字典非常相似。

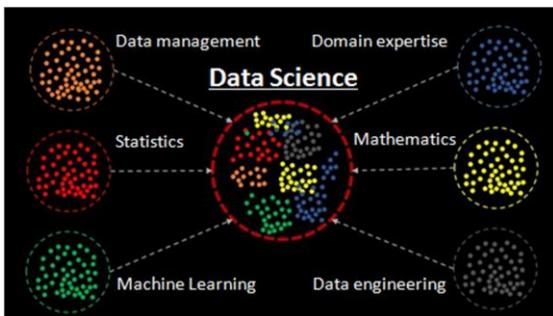
```
{ "title" : "Data Science Book",
  "author" : "Joel Grus",
  "publicationYear" : 2019,
  "topics" : [ "data", "science", "data science" ] }
```

```
import json
serialized = """{ "title" : "Data Science Book",
  "author" : "Joel Grus",
  "publicationYear" : 2019,
  "topics" : [ "data", "science", "data science" ] }"""

# parse the JSON to create a Python dict
deserialized = json.loads(serialized)
assert deserialized["publicationYear"] == 2019
assert "data science" in deserialized["topics"]
```

数据思维与实践

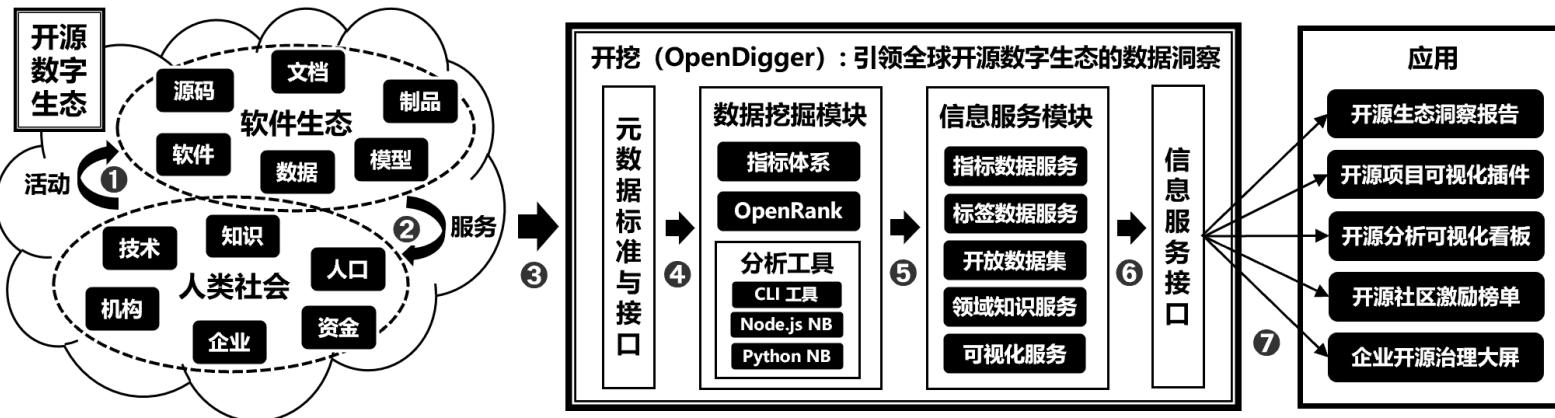
第03讲 数据收集与管理



- 大数据时代
- 数据的全生命周期
- Python 数据收集方法
- **开源数字王国中的数据生态**

开源数据生态

在上次课的任务重，我们初步分析了[开源排行榜 OpenLeaderboard](https://open-leaderboard.x-lab.info) (<https://open-leaderboard.x-lab.info>) 里面的数据。那么这里面的数据从哪里来的呢？答案是从 OpenDigger 中来的。OpenDigger 也是一个开源项目，它收集了包括 GitHub 在内的各种开源生态中的活动数据。



OpenDigger 中的数据指标

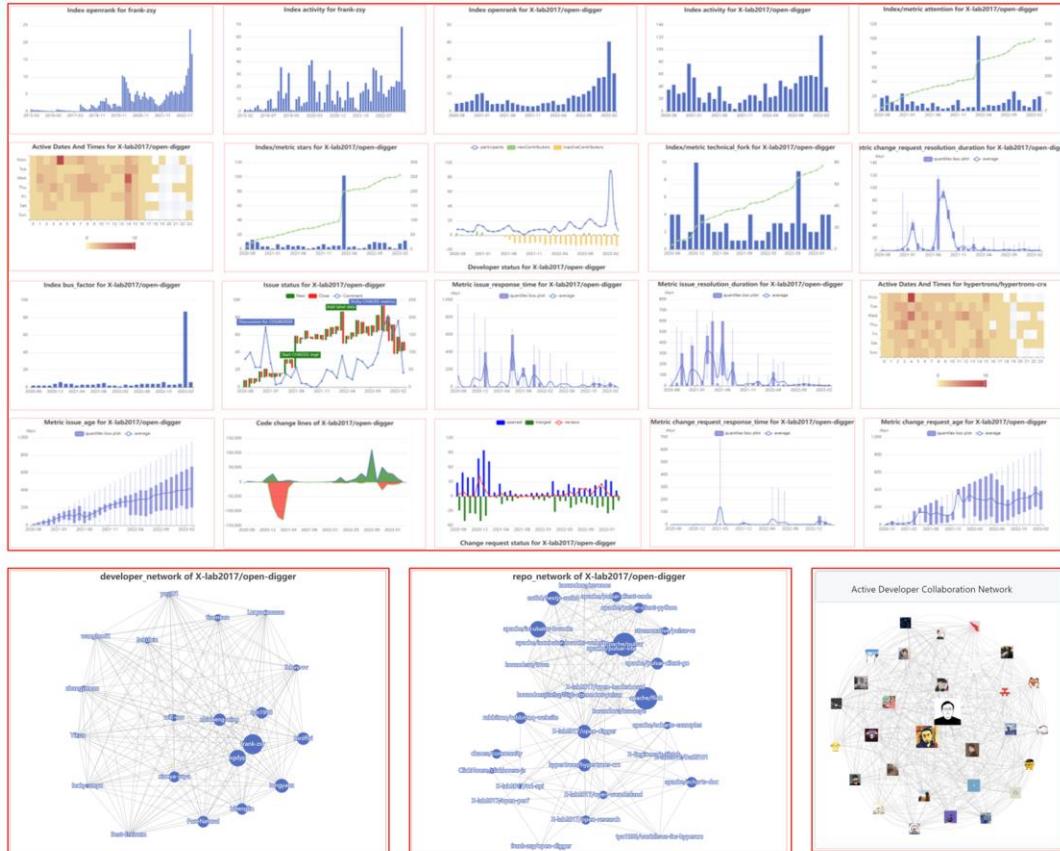
指标

For repos

- [标准院/X-lab] **activity** 💎
- [标准院/X-lab] **openrank** 💎
- [标准院/X-lab] **attention**
- [标准院/X-lab] **stars**
- [标准院/X-lab] **issue_comments**
- [标准院/X-lab] **participants**
- [LF/CHAOS] **technical_fork**
- [LF/CHAOS] **issues_new**
- [LF/CHAOS] **issues_closed**
- [LF/CHAOS] **code_change_lines_add**
- [LF/CHAOS] **code_change_lines_rem**
- [LF/CHAOS] **code_change_lines_sum**
- [LF/CHAOS] **change_requests**
- [LF/CHAOS] **change_requests_accept**
- [LF/CHAOS] **change_requests_reviews**
- [LF/CHAOS] **bus_factor**

For users

- [标准院/X-lab] **activity**
- [标准院/X-lab] **openrank**



开源数据生态

README Apache-2.0 license

OpenDigger

license Apache 2 Data OpenDigger Node.js CI passing

OpenDigger is an open source analysis report project for all open source data initiated by [X-lab](#), this project aims to combine the wisdom of global developers to jointly analyze and insight into everyone better understand and participate in open source.

<https://github.com/X-lab2017/open-digger>

Metrics or Indices Usage

All implemented metrics are open for anyone to use, you can find the data with OpenDigger static data is <https://oss.x-lab.info/open-digger/github/> right now, just replace the `org/repo` or `owner` to get your data.

Feel free to use the data to construct your own data application and you can reach me at zhangxiao@x-lab.org and welcome to use the following badge in your project to show the data source.

Data OpenDigger

For repos

Type	Name	From	Example	Code	CodePen
Index	OpenRank	X-lab	openrank.json	Link	Demo
	Activity	X-lab	activity.json	Link	Demo
	Attention	X-lab	attention.json	Link	Demo
	Active dates and times	CHAOS	active_dates_and_times.json	Link	Demo
	Stars	X-lab	stars.json	Link	Demo
Technical	Technical fork	CHAOS	technical_fork.json	Link	Demo
	Participants	X-lab	participants.json	Link	
	New contributors	CHAOS	new_contributors.json new_contributors_detail.json	Link	Demo

本节任务

从 GitHub 获取相关数据

1. 生成 GitHub 的个人 token
2. 用 GitHub API 获取仓库标星者（starred）的用户信息（不使用 token）
3. 用 GitHub API 查询个人仓库信息（使用个人 token）

课后作业

- 通过学习 PyGithub 文档，利用 GitHub API，首先生成个人的 token，查询自己所有关注者的仓库的数据，将其存到本地。

THANK YOU



本节任务

从 GitHub 获取相关数据

1. 生成GitHub的个人 token
2. 用GitHub API 获取仓库标星者（starred）的用户信息（不使用token）
3. 用GitHub API 查询个人仓库信息（使用个人token）

课后作业

- 通过学习PyGithub文档，利用GitHub API，首先生成个人的token，查询自己所有关注者的仓库的数据，将其存到本地。
- 截止时间：12月22日0点
- 提交地址：51255903058@stu.ecnu.edu.cn
- 注明姓名和学号