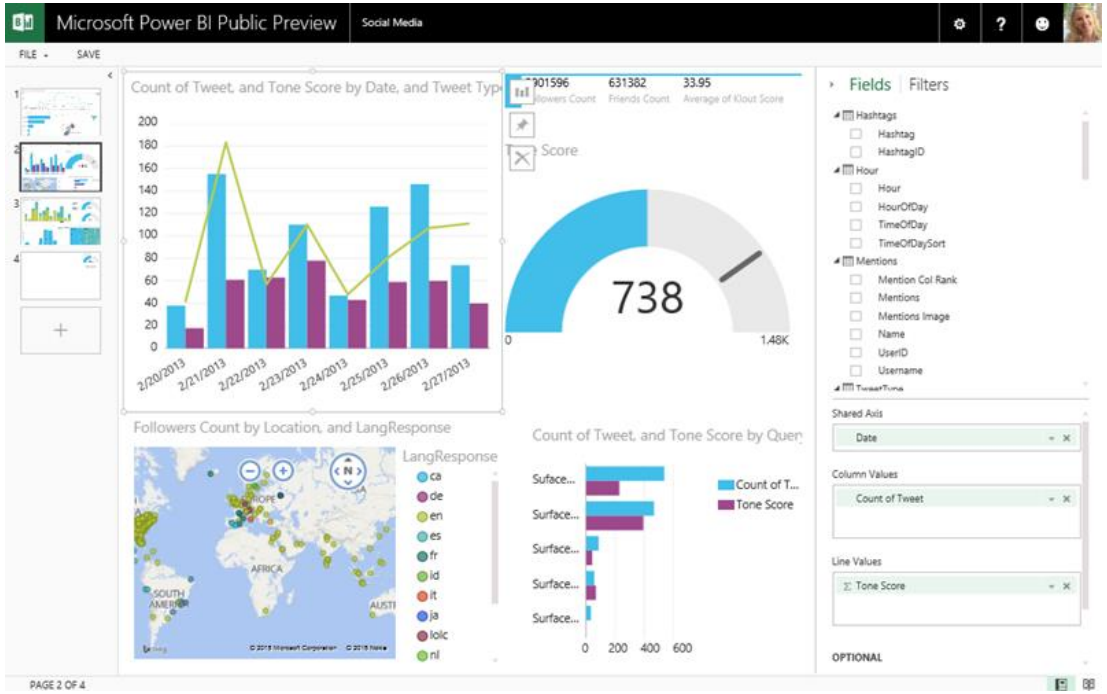
大作业讲解：近似查询处理

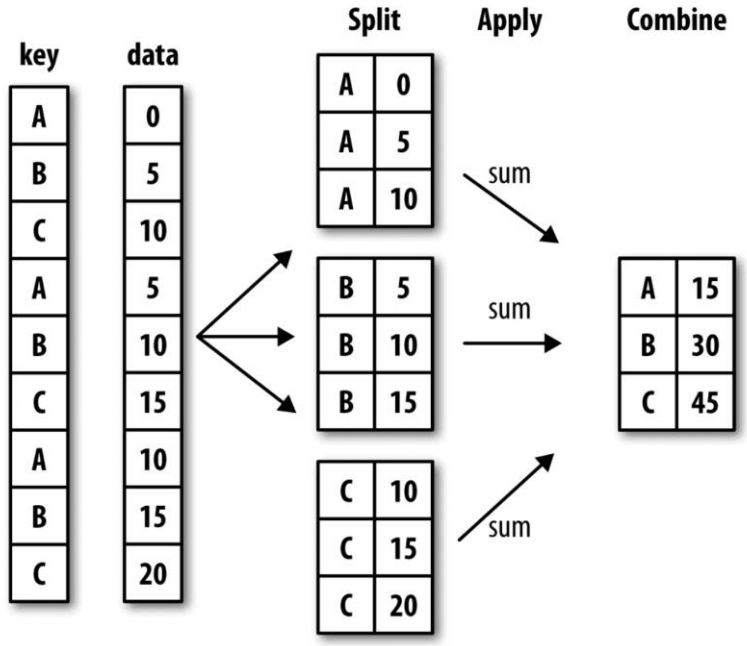
背景：交互式数据探索

- 交互式数据探索有着十分广泛的应用
  - 数据可视化
  - 商务智能（BI）分析
  - 领导驾驶舱



聚合查询：数据的聚合与分组运算

- 聚合查询（Aggregate Queries）
  - 使用一个或多个键（如DataFrame列名)将表格数据进行分组
  - 计算分组的统计信息，比如数量、平均值或求和
- 聚合函数（Aggregate Function）
  - count, sum, avg
- 分组机制（GroupBy）
  - 一个或多个属性
- 选择条件（Where）
  - 对数据进行筛选



聚合查询的SQL形式

- SQL是一种声明性的数据库查询语言，用户只需要告诉数据库系统查询目的，并不需要告诉系统怎么样去获取数据

```
SELECT AVG(jobtime)
FROM very_big_log
WHERE src = 'hadoop'
LEFT OUTER JOIN logs2
ON very_big_log.id = logs.id
```

代码参考：data-aggregation-group.ipynb

近似查询处理

- 近似查询处理（Approximate Query Processing，简称AQP）的目标
  - 在大规模的数据上支持交互式的聚合查询
- 举例
  - 数据规模：100 TB on 1000 machines

½ - 1 Hour

Hard Disks

1 - 5 Minutes

Memory

1 second

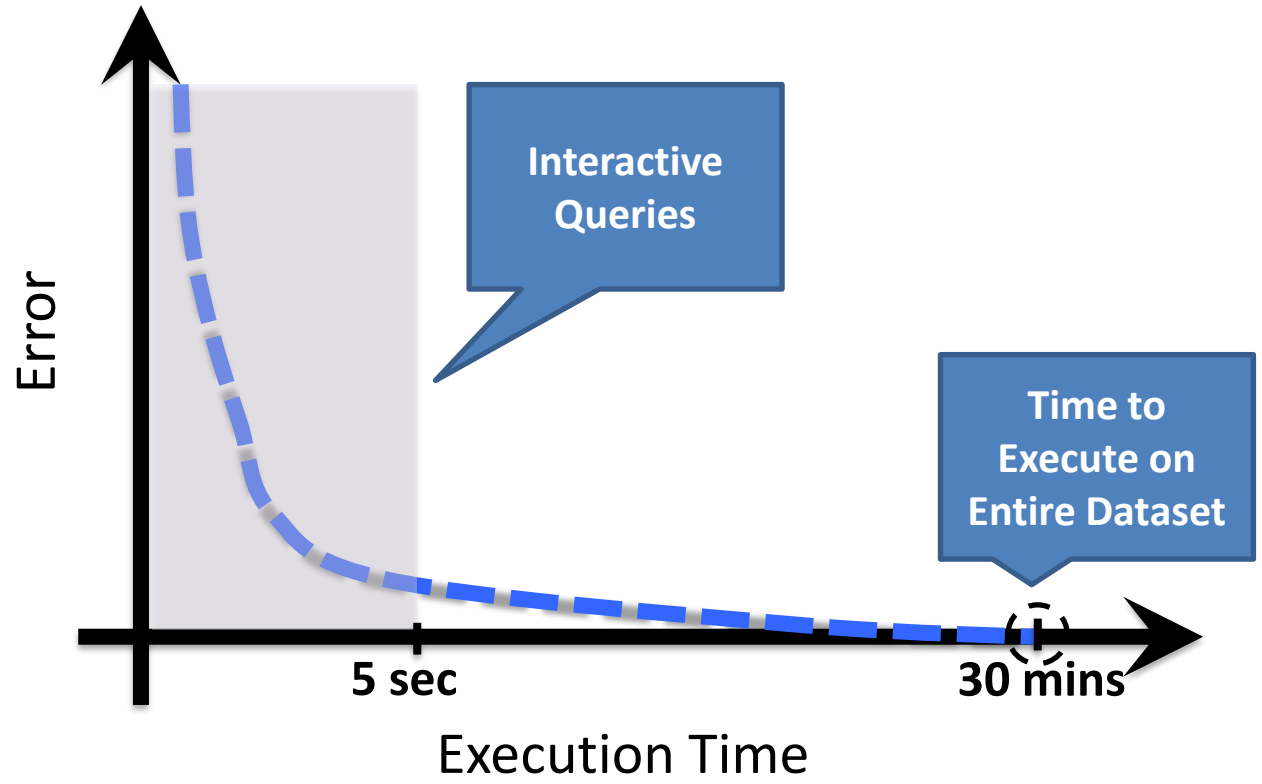
?

近似查询处理

- 基本思路
  - 降低数据规模通过引入近似/允许误差，将“大数据”变为“小数据”
  - 近似的结果往往能满足用户的一定需求。此时可以使用采样、索引、机器学习等算法（而不是筛选和遍历）快速给出用户查询的近似结果
  - 需要对误差的范围有一定的保证

- Online
  - Sampling
- Offline
  - Histogram
  - Wavelet
  - Sketch

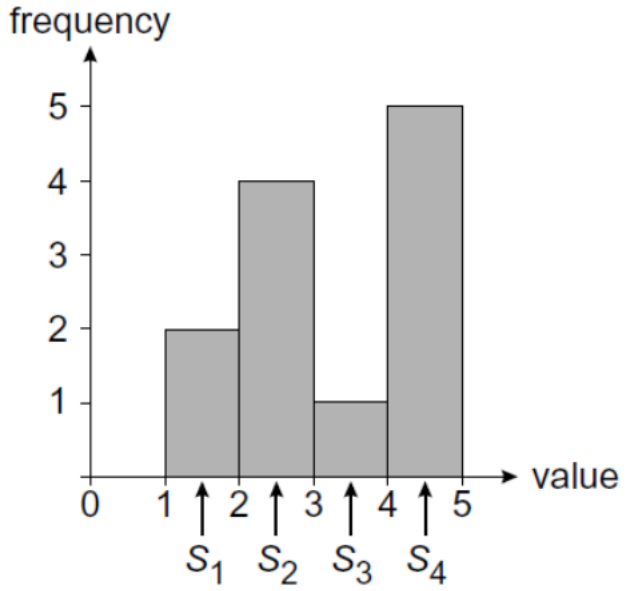
Speed/Accuracy Trade-off



Histogram

设计思路

- 把整体数据分块（把相邻的具体数据整合）
- 预先处理数据的统计信息，组合成结果的估计



$$D = \{1.61, 1.72, 2.23, 2.33, 2.71, 2.90, 3.41, 4.21, 4.70, 4.82, 4.85, 4.91\}$$

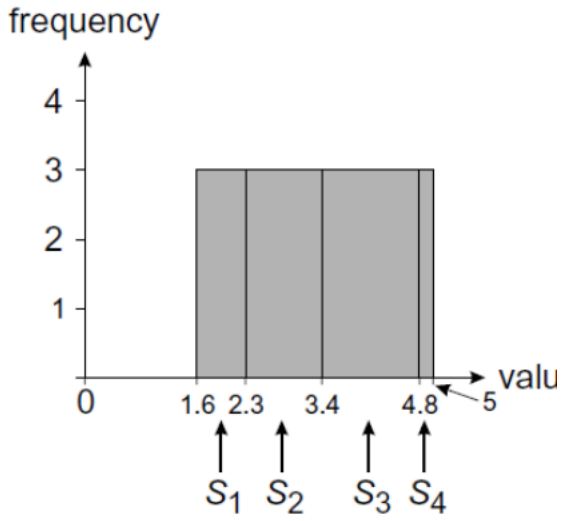
Histogram Design

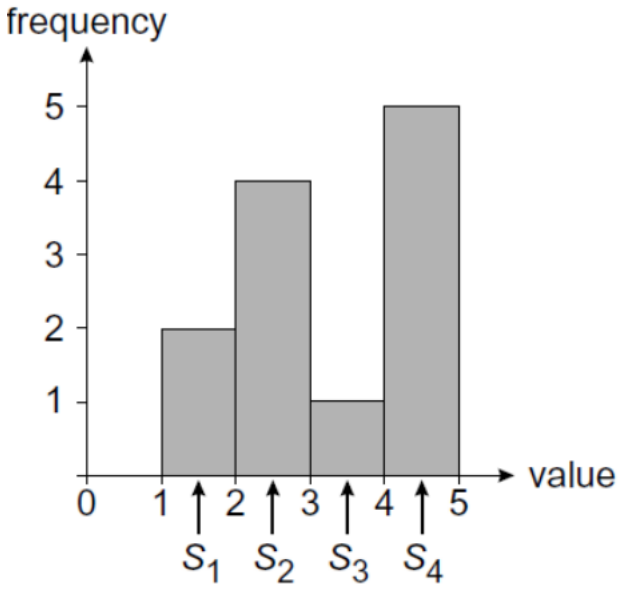
- 分块策略
  - 怎么分块，每个块相不相交
- 估计策略
  - 每个块存什么统计信息
  - 怎么存
- 效率
- 精确度

1D-Histogram

$$D = \{1.61, 1.72, 2.23, 2.33, 2.71, 2.90, 3.41, 4.21, 4.70, 4.82, 4.85, 4.91\}$$

- 分块
  - 等宽VS等高
- 估计
  - 维护频率





Example：

SELECT COUNT(*) FROM D WHERE D.val >= 1.1 AND D.val <= 4.5

- 等宽： (2.0 – 1.1)/(2.0 – 1.0) = 0.9，(5.0 – 4.5)/(5.0 – 4.0) = 0.5，ans = 0.9*2 + 4 + 1 + 0.5*5 = 9.3
- 等高： ans =3 + 3 + ((4.5 – 3.4)/(4.8 – 3.4))*3 = 8.4

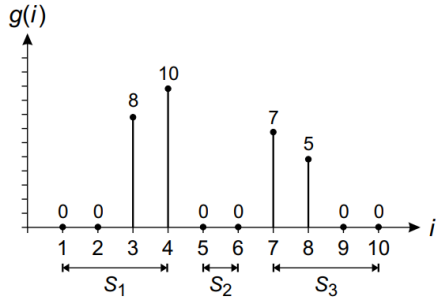
分块策略plus

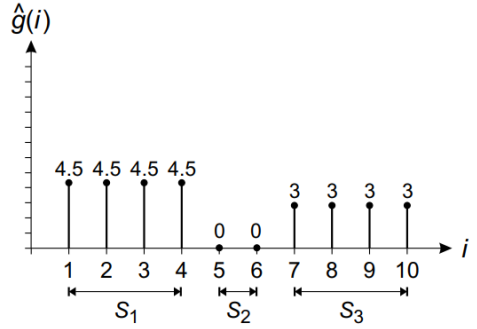
- 启发式方法：
  - 等宽、等高
  - Max-diff
    - 设分B块，相邻项作差排序后取间隔最大的作为分割点
  - Singleton-Bucket Histograms
    - 类似分层采样，对某些数据单独拎出来成块（e.g. 频率奇高奇低）
  - 分到接受的精度就停止
  - 混合不同策略

- 最优化方法
  - 确定目标函数后动态规划

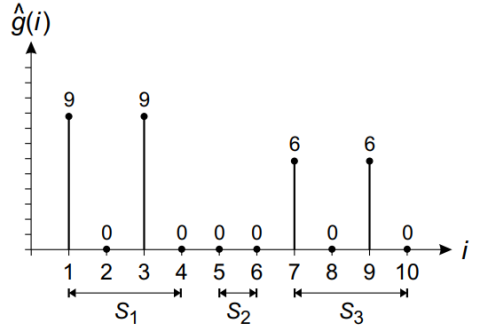
估计策略plus

$$V = (0, 0, 8, 10, 0, 0, 7, 5, 0, 0)$$





(a) Continuous-value assumption

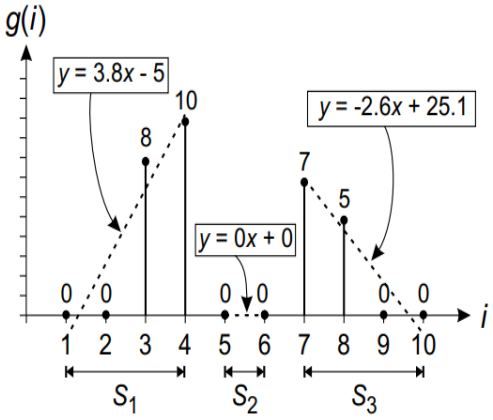


(b) Uniform-spread assumption

- uniform-spread assumption
  - 第j个块中的数据分布在如下集合上,其中d为块中不同值的数量l为区间起始

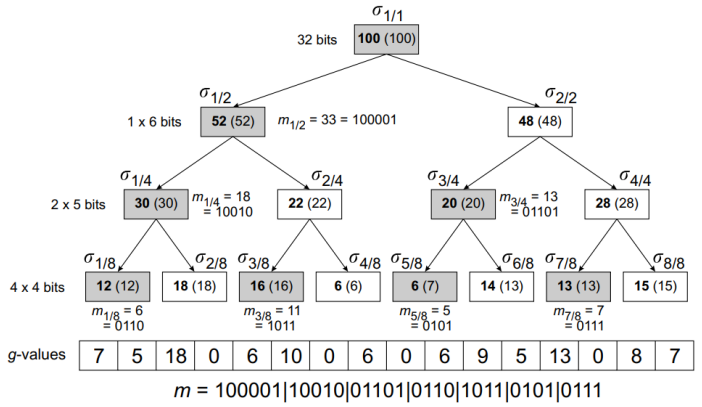
$$\hat{P}_j = \{l_j, l_j + k_j, l_j + 2k_j, \ldots, l_j + (d_j - 1)k_j\}$$

- Spline-Based Schemes
  - $\gamma\gamma$斜率$\beta\beta$截距
  - 添加约束更加精准



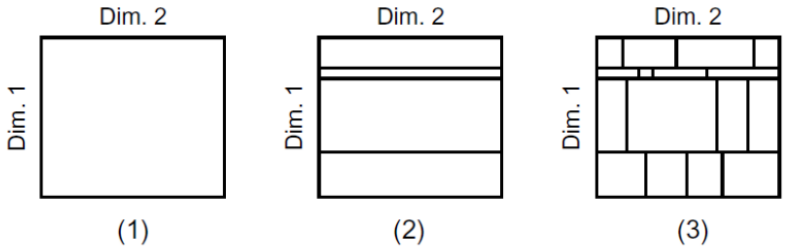
$$\sum_{i \in S_j} (\gamma_j i + \beta_j) = \sum_{i \in S_j} g(i) \quad \&\& \quad \sum_{i \in S_j} i(\gamma_j i + \beta_j) = \sum_{i \in S_j} i g(i)$$

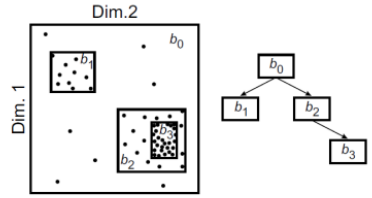
- Four-Level Trees

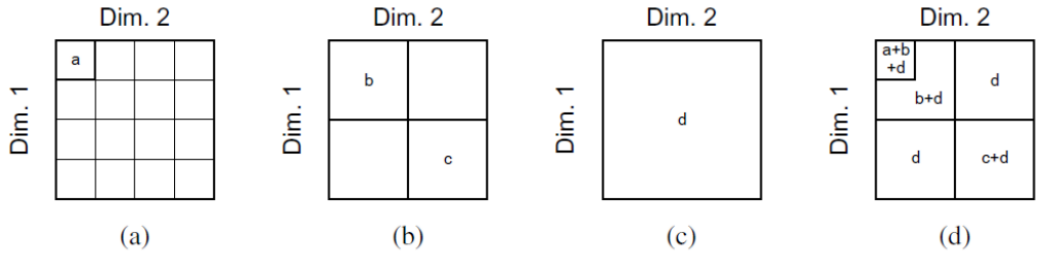


Multi-Dimensional Histogram

- Approach1: 自顶向下
  - 切割高维矩形
    - 每一维用1D的各种切法
- Approach2:自底向上
  - 合并属性相近数据
    - 定义密度 $f_S = \sum_{i \in S} g(i)/g^+$
    - 不均匀就剃掉突出的部分
    - 比较均匀就合成一块







Checkpoint

- 代码实现AQP（任意策略）
  - SELECT COUNT(*) FROM D WHERE D.val >= 1.1 AND D.val <= 4.5

$$D = \{1.61, 1.72, 2.23, 2.33, 2.71, 2.90, 3.41, 4.21, 4.70, 4.82, 4.85, 4.91\}$$

- 做完找助教检查并签到（**记考勤**）