HDAT9400 Data Management: S, M, L, XL Data

Malcolm Gillies

https://github.com/mbg-unsw/hdat9400

19 October 2021

main@283df09 2021-10-18

Lecture outline

- Why does data size matter?
- How big are health data sets?
- How fast can we process data?
- Asymptotic algorithmic complexity
- Real world examples

main@283df09 2021-10-18

About me

- When I studied computer science (1990), a PC had
  - 4MB RAM (1000th today's phones)
  - 200MB Disk (1000th today's phones)
  - 33MHz Processor (100th today's phones)

Health data I've worked with

- NPS MedicineWise: GP electronic medical records (MedicineInsight)
- NSW Ministry of Health: hospital, ambulance, births
- CBDRH: Pharmaceutical Benefits Scheme (PBS)
- SAS, R, MS SQL Server, PostgreSQL, SQLite, DuckDB

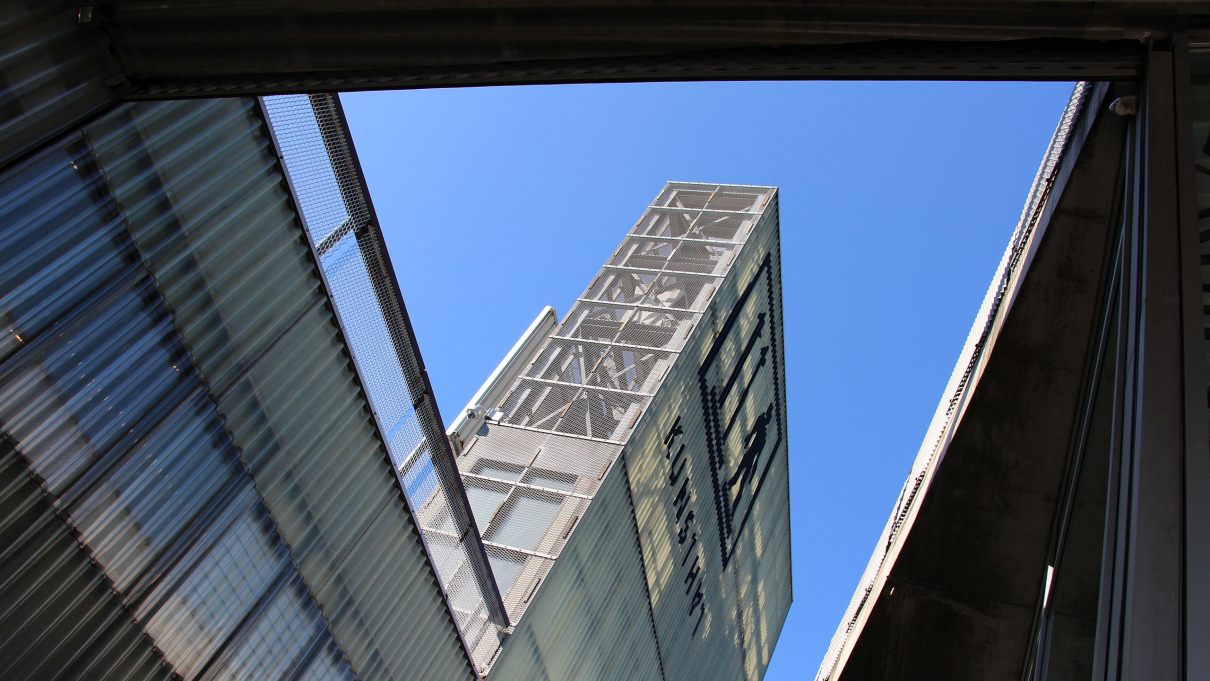
Why does data size matter?

- Time and space are finite
- We have budgets and deadlines
- Two times bigger can take more than twice the time

What can we do about it?

- *Work smarter, not harder*
- Learn from the theory of algorithms
- Use efficient data-processing tools

main@283df09 2021-10-18


KUNSTHAL

How big are health data sets?

Data	Records	Gigabytes
NSW congenital conditions (5 years)	10 000	0.001
NSW perinatal (20 years)	1 000 000	1
NSW Admitted patients (20 years)	100 000 000	15
AU Pharmaceutical benefits (20 years)	1 000 000 000	400
Health “data lake”	1 000 000 000 000 000	1 000 000

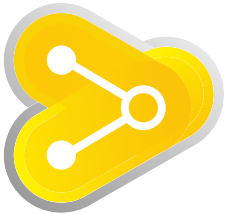
20–200 variables per record

What is “big data”?

> **Big data** is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software.
>
> — Wikipedia

main@283df09 2021-10-18

So...



CENTRE FOR
~~BIG~~ DATA RESEARCH
IN HEALTH

MODESTLY SIZED

main @ 283df09 2021-10-18

Examples of different data processing technologies

Method	Max size	Rec per sec	Notes
In memory [R]	GB	1000M	Simple!
Disk streaming [SAS]	TB	1M	Slower
Relational database [SQLite]	TB	2M	Complicated
Column-store database [DuckDB]	TB	50M	Specialised
NoSQL [Apache Spark]	PB	????	Don't ask

main@283df09 2021-10-18

Starting simple: process all the data

- Sometimes you need to look at every record aka *full table scan*
  - e.g. What is the total length of stay of all NSW admissions?
- All else being equal, twice the data takes twice the time
- Most important distinction: scan in memory (RAM) or disk?

main@283df09 2021-10-18

Making things more complicated

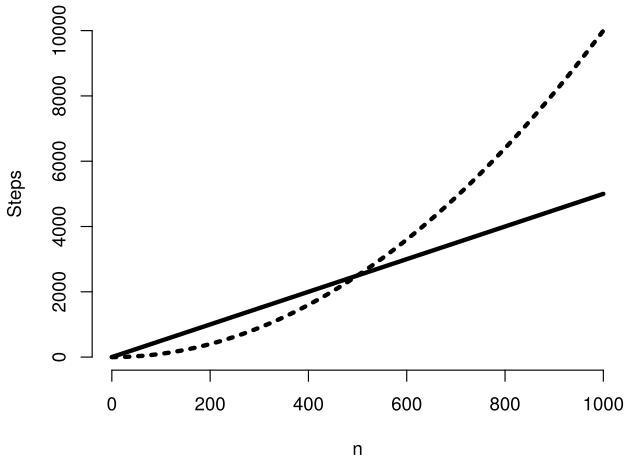
- Sort all prescriptions by date of prescription
- Analyse all data from hospitals in Sydney
- For each antibiotic prescription, find the corresponding doctor visit
- Build a regression model for risk of low birth weight based on maternal characteristics

Time (and space) complexity

- How does an algorithm "scale" with the amount of data?
- Ignore details of the implementation, specific computer etc
- Insight: how does the time/space cost change as $n \to \infty$?
- *Asymptotic complexity*

main@283df09 2021-10-18

$O(n)$ vs $O(n^2)$



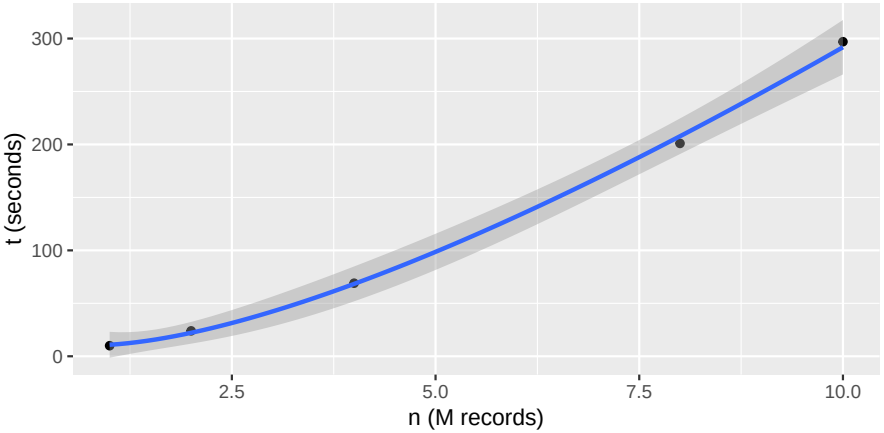
main@283df09 2021-10-18

Optimal speed of common algorithms

Hash lookup	$O(1)$
(Binary) search	$O(\log n)$
Sort	$O(n \log n)$
Matrix inversion	$O(n^2 \log n)$

main@283df09 2021-10-18

Experiment: sorting in SAS



main@283df09 2021-10-18

Speeding up WHERE using an index (1)

```
/* test.data contains 10,000,000 records of 200 variables */
/* each variable contains a random number from 1 to 1000 */

proc sql;
    create index xi200 on test.data(xi200);
quit;

/* data.sas7bdat: 16GB */
/* data.sas7bndx: 80MB */
```

Speeding up WHERE using an index (2)

```
data benchmark1;
    set test.data;
    keep total;
    total+xi1;
    * no index;
    where xi199=10;
run;
```

real time 14.76 seconds

```
data benchmark2;
    set test.data;
    keep total;
    total+xi1;
    * index;
    where xi200=10;
run;
```

real time 2.52 seconds

Real world example: Matching ED and in-patient episodes

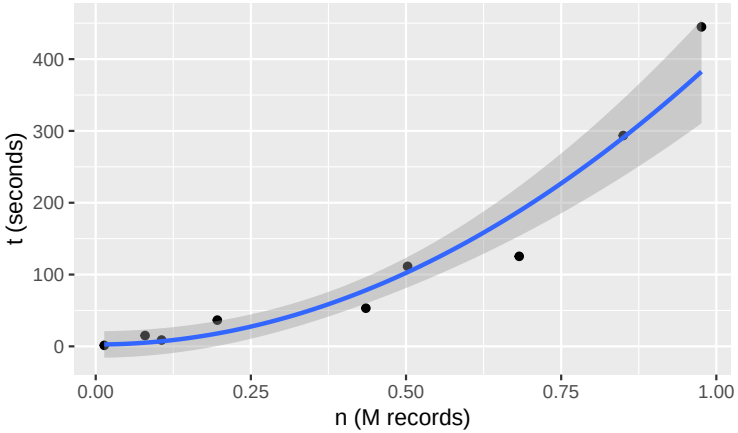
- 50M records in Admitted Patient Data Collection
- 50M records in Emergency Department Data Collection
- SAS merge on PPN (patient ID) and date
- Sort, sort and merge $\longrightarrow$ *hours*
- R: cannot load complete data set into memory
  - 1M records, subset of 16 variables $\longrightarrow$ 2.5s
  - need to process in chunks, similar total run time to SAS

Real world example: Modelling ICU glucose control (1)

- Blood glucose levels in ICU monitored and insulin given
- What are risk factors for glucose >180 mg/dL?
- Covariates: severity score (APACHE), diabetes dx, age, gender, insulin
- 1M observations, 200K patients, 200 ICUs
- Generalised linear mixed model (3 level)

main@283df09 2021-10-18

Real world example: Modelling ICU glucose control (2)



- MixedModels.jl
- *Full model too large for R lme4*

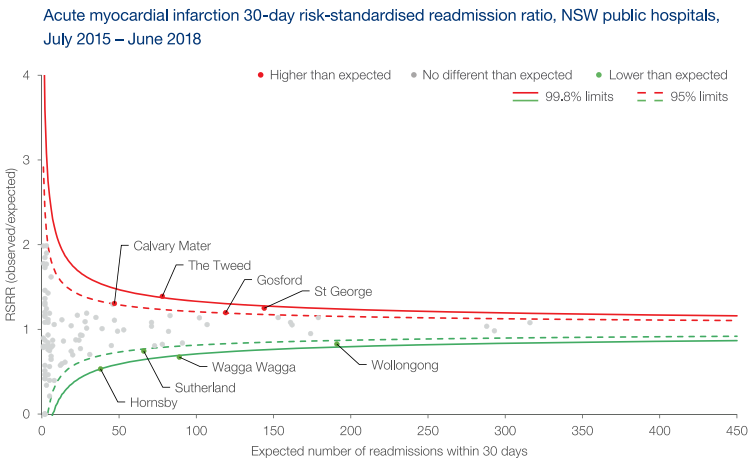
main@283df09 2021-10-18

Real world example: NSW hospital readmission rates (1)

- Bureau of Health Information
- Quarterly report on hospital performance
- Mixed models, SAS
- 32 000 records, 177 hospitals
- Run time for the analysis: 1 minute

main@283df09 2021-10-18

Real world example: NSW hospital readmission rates (2)



Summary

- Computers are fast and most data isn't that big
- We have a bunch of excellent data processing tools
- If you do things the wrong way, time and space limits can bite
- Some tasks will always be "slow" asymptotically
- Think about the size of your data before you start

Thanks

- Sadaf Marashi-Pour (Bureau of Health Information)
- Sandy Sa (NSW Ministry of Health)
- Juan Quiroz Aguilera (CBDRH)
- Oisin Fitzgerald (CBDRH)

main@283df09 2021-10-18

Further reading

- A Gentle Introduction to Algorithm Complexity Analysis by Dionysis Zindros
- Database-like ops benchmark by `data.table` developers
- Mixed Models for Big Data by Michael Clark

Picture credits (Creative Commons)

- Nieuwbouw IJplein © Bart Molendijk / Anefo
- Kunsthal, Rotterdam © Fred Romero
- Seattle Public Library © Moody 75
- De Rotterdam © Fred Romero
- Kunsthal, Rem Koolhaas portrait © Jacco van Giessen

main@283df09 2021-10-18