10 Data Management I

DMV
suzanne.little@dcu.ie

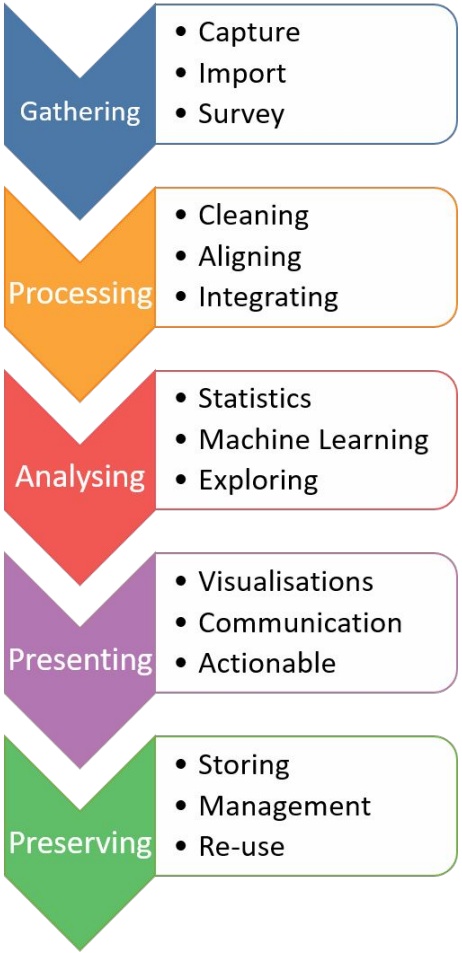
Today

- Data Storage overview
- What storage approach should you use for …?
- Review of Exercises

Reminder - Data Analytics Pipeline

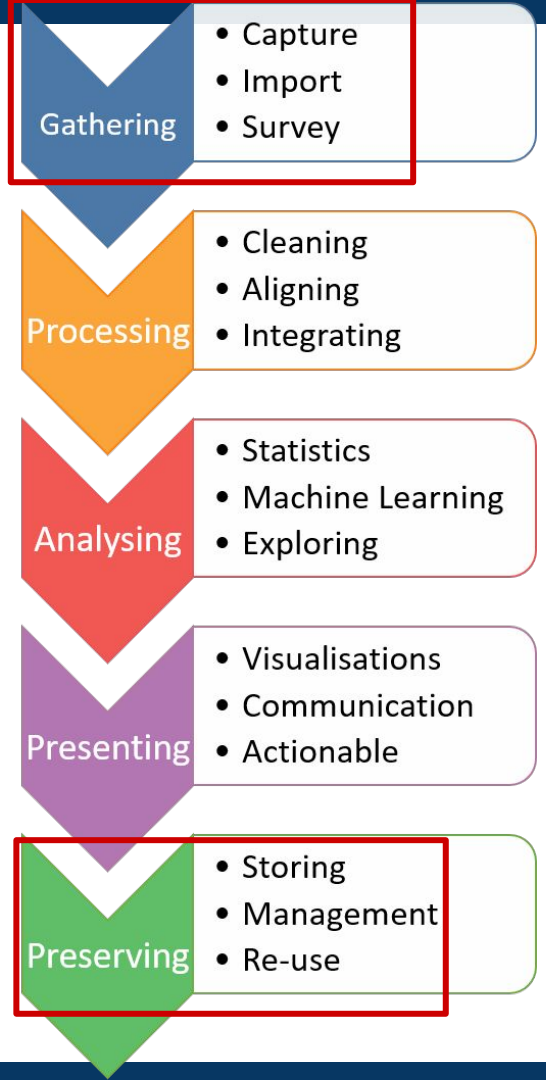
Finished Data Visualisation (“Presenting”)

Next 2.5 weeks are on “Preserving”



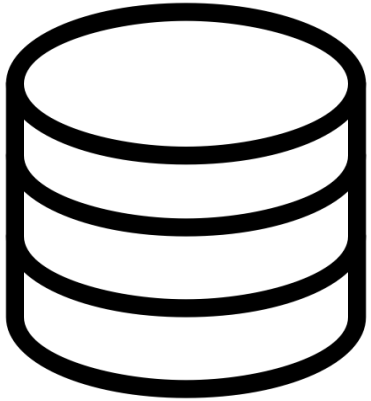
Data (recap)

- Data is collected information
- Where does data come from?
  - Files
  - The Internet
  - Databases



Data storage .... some options

Database management tools



Why? Data persistence & access

Find relevant information quickly

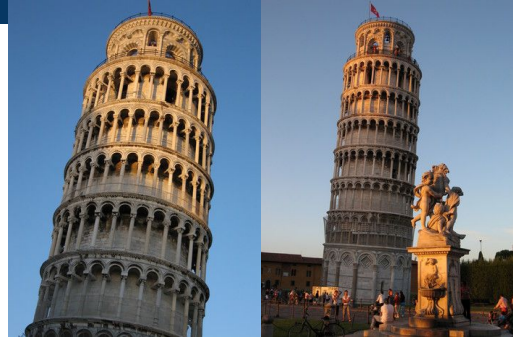
Select required data subset for further analysis

Apply analysis function to (large, distributed?) datasets

Calculate regular reports as data is updated

Ensure **consistency** and **protection** of data

Scenario: my photo collection



Data consists of:

- CSV file with list of photo id, path to EXIF file & path to jpg file
- Folder with JPG & EXIF files organised into subfolders of year & event name (e.g, 2018/Italy_trip_2018, 2006/Pisa2006, 2007/AustChristmas)

1. How do I find the photo of the leaning tower taken Oct 2006?
2. How do I delete the photo of the leaning tower taken Oct 2006?
3. How do I find all photos of the leaning tower of Pisa?

What makes a “database”?

- Structure - tables, documents, “chunks”
- Minimise redundancy - efficient storage, normalisation
- Maintain consistency - updates, transactions, deletions
- Multiple user and concurrent access
- Query options - eg. language like SQL, SPARQL, Cypher
- Security

Data storage approaches

Database: structured set of data that can be accessed, managed and updated (easily)

1. Relational (traditional & modern)
2. Column
3. MPP, Data Warehouse
4. NoSQL
5. Big Data (e.g., MapReduce, Hadoop, PySpark, Kubernetes) → more next week

1 Relational DBs (the all-purpose solution for not-that-big data)

- Structured according to the relationship between data
- Tables, Records and Columns
- Relationship facilitates searching, organisation & reporting

Consider:

- “adequate for all tasks but not excellent at any of them” - ?
- easy to use
- low resource requirements
- well-supported by all software
- familiar
- not suitable for really big data

Eg: Oracle, PostgreSQL, MySQL/MariaDB, sqlite, IBM DB2, Microsoft SQL Server

1 Relational DBs

Visualisation - a genealogy of RDBMS! Only to 2018
https://hpi.de/naumann/projects/rdbms-genealogy.html

Popular ones: Oracle, MySQL/MariaDB, Microsoft SQL Server, PostgreSQL

Why so many? What's the difference?

Legacy databases are a common source of data. Programming languages provide APIs to query and update databases (of most sorts).

2 Columnar stores?

- inversion of a row store: indexes become data & data becomes indexes
- for aggregations and transformations of highly structured data
- good for BI, analytics, some archiving but not data mining
- moderately big data (0.5-100TB) → compression
- slow to add new data / purge data
- Eg: Cassandra, Bigtable, HBase, PostgreSQL (option)

https://database.guide/what-is-a-column-store-database/

3 DW & MPP

Data Warehouse (**DW**)

- a centralized repository
- stores data from multiple information sources
- transformed into a common, multidimensional data model
- efficient querying and analysis

https://www.datawarehouse4u.info/index_en.html

Massively Parallel Processing Database (**MPP**)

- optimized to be processed in parallel
- many operations performed by many processing units at a time.

OLAP, OLTP, DW ??

OLTP: On-Line Transactional Processing

Operations: INSERT, UPDATE, DELETE

OLAP: On-Line Analytical Processing

Information: complex analytics, aggregations, batch

DW: Data Warehouse

OLTP vs Data Warehouse/OLAP

- many single-row writes
- current data
- queries generated by user activity
- < 1s response times
- 1000's of users

- few large batch imports
- years of data
- queries generated by large reports
- queries can run for minutes/hours
- 10's of users

OLTP	vs	Data Warehouse/OLAP
big data for many concurrent requests to small amounts of data each		big data for low concurrency requests to very large amounts of data each

4 NoSQL

- Non-relational or sometimes “not only SQL” - https://en.wikipedia.org/wiki/NoSQL
- Eg: key-value, document, object or graph-based data stores
- Eg: MongoDB, Solr, HBase, Splunk, Neo4j
- Why?
  - Large volumes of structured, semi-structured, and unstructured data
  - Quick iteration
  - Efficient, scale-out architecture instead of expensive, monolithic architecture

https://www.devbridge.com/articles/benefits-of-nosql/

5 Big Data

Next week

Actual databases ...

http://db-engines.com/en/ranking

423 systems in ranking, November 2024

Rank Nov 2024	Rank Oct 2024	Rank Nov 2023	DBMS	Database Model	Score Nov 2024	Score Oct 2024	Score Nov 2023
1.	1.	1.	Oracle	Relational, Multi-model	1317.01	+7.57	+39.98
2.	2.	2.	MySQL	Relational, Multi-model	1017.80	-4.95	-97.44
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	799.81	-2.28	-111.61
4.	4.	4.	PostgreSQL	Relational, Multi-model	654.34	+2.18	+17.48
5.	5.	5.	MongoDB	Document, Multi-model	400.93	-4.28	-27.62
6.	6.	6.	Redis	Key-value, Multi-model	148.64	-0.99	-11.38
7.	7.	↑ 11.	Snowflake	Relational	142.50	+1.90	+21.50
8.	8.	↓ 7.	Elasticsearch	Multi-model	131.64	-0.20	-7.98
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model	121.74	-1.03	-14.26
10.	10.	↓ 9.	SQLite	Relational	99.49	-2.43	-25.09
11.	11.	↑ 12.	Apache Cassandra	Wide column, Multi-model	97.71	+0.10	-11.45
12.	12.	↓ 10.	Microsoft Access	Relational	91.31	-0.84	-33.18

Actual databases ...

http://db-engines.com/en/ranking

416 systems in ranking, November 2023

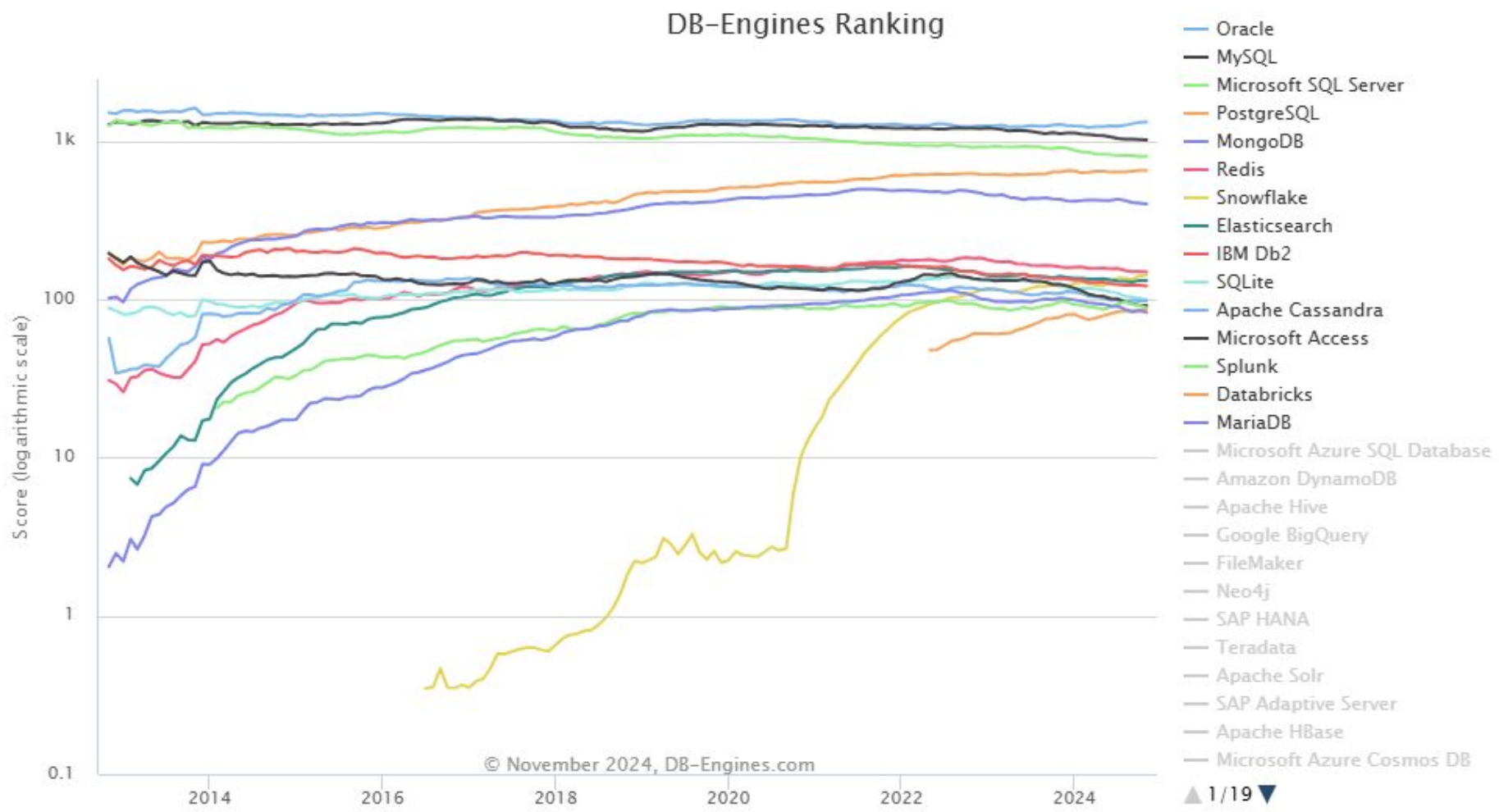
Rank Nov 2023	Rank Oct 2023	Rank Nov 2022	DBMS	Database Model	Score Nov 2023	Score Oct 2023	Score Nov 2022
1.	1.	1.	Oracle	Relational, Multi-model	1277.03	+15.61	+35.34
2.	2.	2.	MySQL	Relational, Multi-model	1115.24	-18.07	-90.30
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	911.42	+14.54	-1.09
4.	4.	4.	PostgreSQL	Relational, Multi-model	636.86	-1.96	+13.70
5.	5.	5.	MongoDB	Document, Multi-model	428.55	-2.87	-49.35
6.	6.	6.	Redis	Key-value, Multi-model	160.02	-2.95	-22.03
7.	7.	7.	Elasticsearch	Search engine, Multi-model	139.62	+2.48	-10.70
8.	8.	8.	IBM Db2	Relational, Multi-model	136.00	+1.13	-13.56
9.	9.	↑ 10.	SQLite	Relational	124.58	-0.56	-10.05
10.	10.	↓ 9.	Microsoft Access	Relational	124.49	+0.18	-10.53
11.	11.	↑ 12.	Snowflake	Relational	121.00	-2.24	+10.84
12.	12.	↓ 11.	Cassandra	Wide column, Multi-model	109.17	+0.34	-8.96

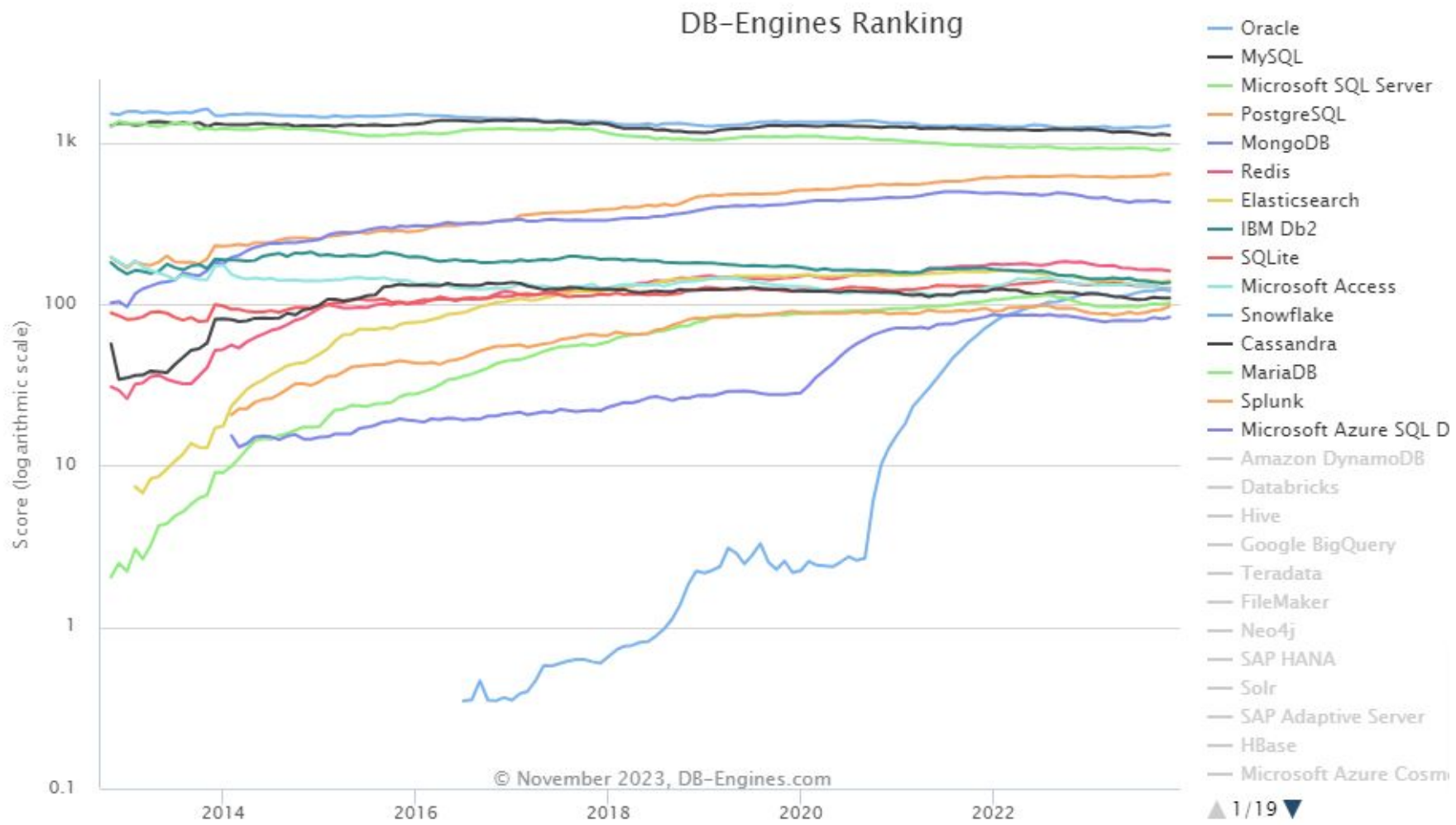
Actual databases ...

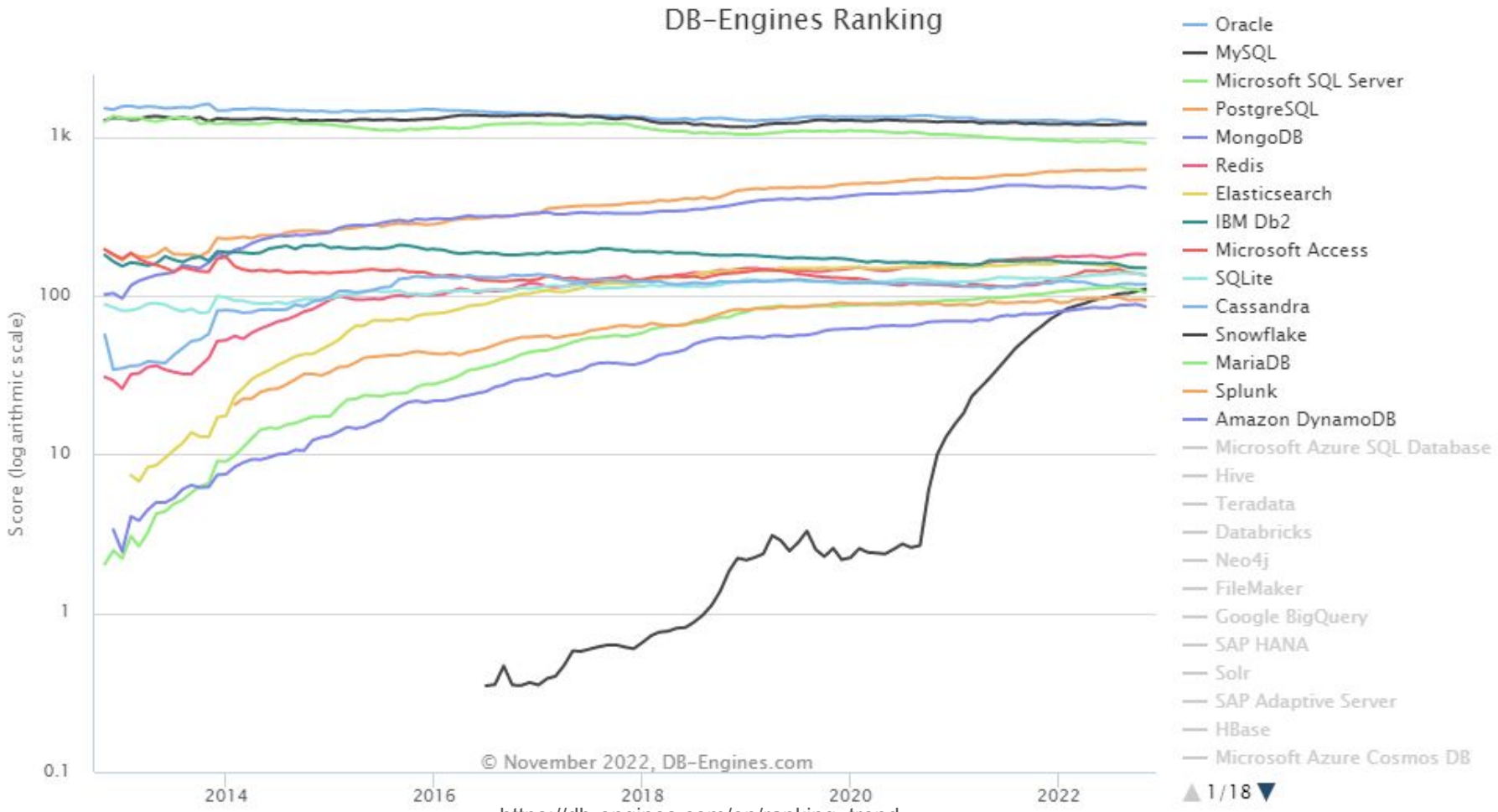
http://db-engines.com/en/ranking

397 systems in ranking, November 2022

Rank Nov 2022	Rank Oct 2022	Rank Nov 2021	DBMS	Database Model	Score Nov 2022	Score Oct 2022	Score Nov 2021
1.	1.	1.	Oracle	Relational, Multi-model	1241.69	+5.32	-31.04
2.	2.	2.	MySQL	Relational, Multi-model	1205.54	+0.17	-5.98
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	912.51	-12.17	-41.78
4.	4.	4.	PostgreSQL	Relational, Multi-model	623.16	+0.44	+25.88
5.	5.	5.	MongoDB	Document, Multi-model	477.90	-8.33	-9.45
6.	6.	6.	Redis	Key-value, Multi-model	182.05	-1.33	+10.55
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	150.32	-0.74	-8.76
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model	149.56	-0.10	-17.96
9.	9.	↑ 11.	Microsoft Access	Relational	135.03	-3.14	+15.79
10.	10.	↓ 9.	SQLite	Relational	134.63	-3.17	+4.83
11.	11.	↓ 10.	Cassandra	Wide column	118.12	+0.18	-2.76
12.	↑ 13.	↑ 18.	Snowflake	Relational	110.15	+3.43	+45.97


DB-Engines Ranking
Score (logarithmic scale)
1k
100
10
1
0.1
2014
2016
2018
2020
2022
2024
© November 2024, DB-Engines.com
Oracle
MySQL
Microsoft SQL Server
PostgreSQL
MongoDB
Redis
Snowflake
Elasticsearch
IBM Db2
SQLite
Apache Cassandra
Microsoft Access
Splunk
Databricks
MariaDB
Microsoft Azure SQL Database
Amazon DynamoDB
Apache Hive
Google BigQuery
FileMaker
Neo4j
SAP HANA
Teradata
Apache Solr
SAP Adaptive Server
Apache HBase
Microsoft Azure Cosmos DB
1/19


DB-Engines Ranking
Score (logarithmic scale)
1k
100
10
1
0.1
2014
2016
2018
2020
2022
© November 2023, DB-Engines.com
Oracle
MySQL
Microsoft SQL Server
PostgreSQL
MongoDB
Redis
Elasticsearch
IBM Db2
SQLite
Microsoft Access
Snowflake
Cassandra
MariaDB
Splunk
Microsoft Azure SQL D
Amazon DynamoDB
Databricks
Hive
Google BigQuery
Teradata
FileMaker
Neo4j
SAP HANA
Solr
SAP Adaptive Server
HBase
Microsoft Azure Cosm
1/19

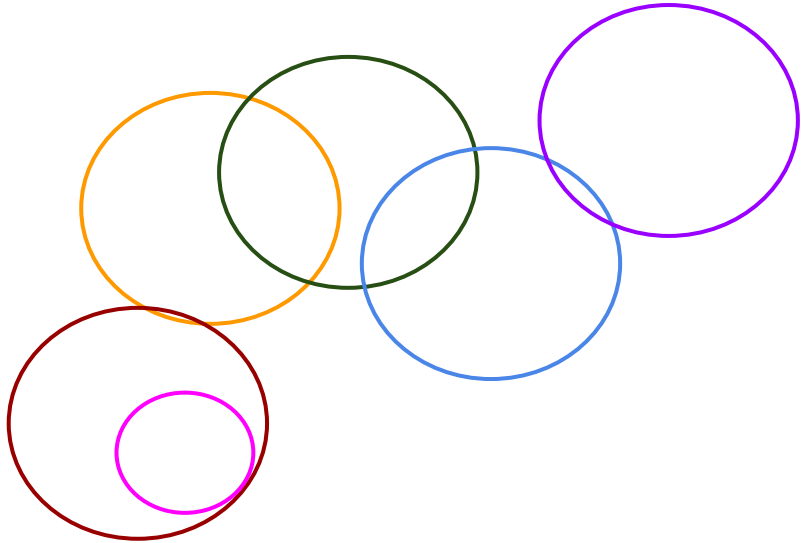


https://db-engines.com/en/ranking_trend

Data storage approaches

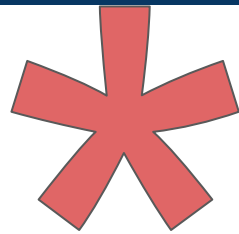
Database: structured set of data that can be accessed, managed and updated (easily)

1. Relational (traditional & modern)
2. Column
3. MPP, Data Warehouse
4. NoSQL
5. Big Data (MapReduce, Hadoop)



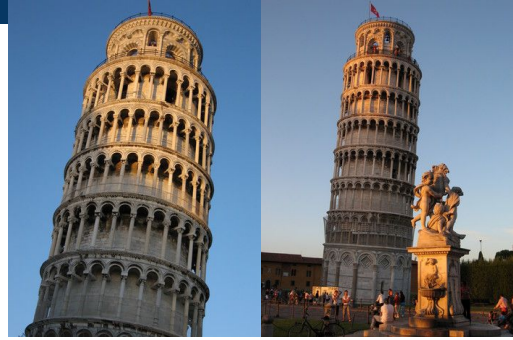
→ In practice, commonly use “polyglot persistence”

Database management - Some questions to ask



1. How much data do I have now? What rate will I get new data?
2. Is the data structured? What format is the data?
3. Does the data need to be processed before loading?
4. How many queries will be run? Will they be concurrent? How many users?
5. What questions will the users be trying to answer? Do I know these questions?
6. Do I need to perform complex calculations on the data?
7. Do I have metadata or catalogue information? Is there a domain standard I can use?

Scenario: my photo collection



Data consists of:

- CSV file with list of photo id, path to EXIF file & path to jpg file
- Folder with JPG & EXIF files organised into subfolders of year & event name (e.g, 2018/Italy_trip_2018, 2006/Pisa2006, 2007/AustChristmas)

1. How do I find the photo of the leaning tower taken Oct 2006?
2. How do I delete the photo of the leaning tower taken Oct 2006?
3. How do I find all photos of the leaning tower of Pisa?

Exercise - Which data storage method?

1. Sales and Customer data for a Small-to-Medium-Enterprise (SME)
2. Website logs, audience profiles, content generation for a media organisation
3. Building Management System - CCTV, power, floor plans, energy usage, work rosters, emergency plans, alarms, other sensors, etc.

Simple relational DW/MPP

NoSQL → Column, Graph, Document?

Map/Reduce ELK / Elasticstack

Hang on! What do I need to know?

The characteristics (pros/cons) of different types of data management methods and some examples

Specific terms and acronyms related to data management

How to approach a data collection and storage task

Eg: What questions to ask; What attributes to look for

Solutions are often multipart (“polyglot persistence”)

References

DCU library ebook: R. Stevens, Beginning Database Design Solutions (Part 1 only) - https://ebookcentral-proquest-com.dcu.idm.oclc.org/lib/dcu/reader.action?ppg=39&docID=427853&tm=1543835305326

Recap of lab exercises

Data Cleaning - What do we think of LLMs?

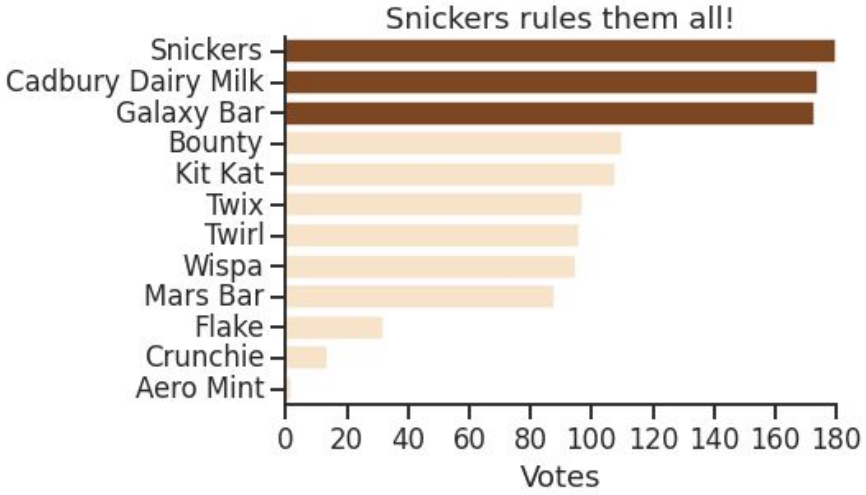
Which LLM gave the "best" results when trying to clean data?
ChatGPT had most votes

“Only Claude used a python script to clean the names”

“The o1 models [ChatGPT] took better care to account for titles. The older GPT4 model didn’t do a great job at this”

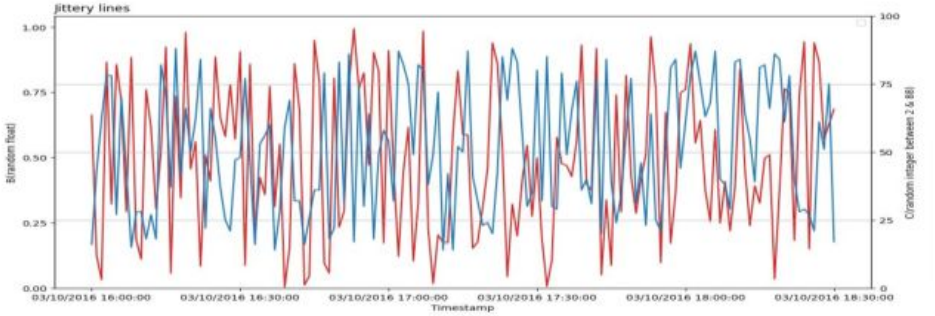
Notebook - create a graph using Python

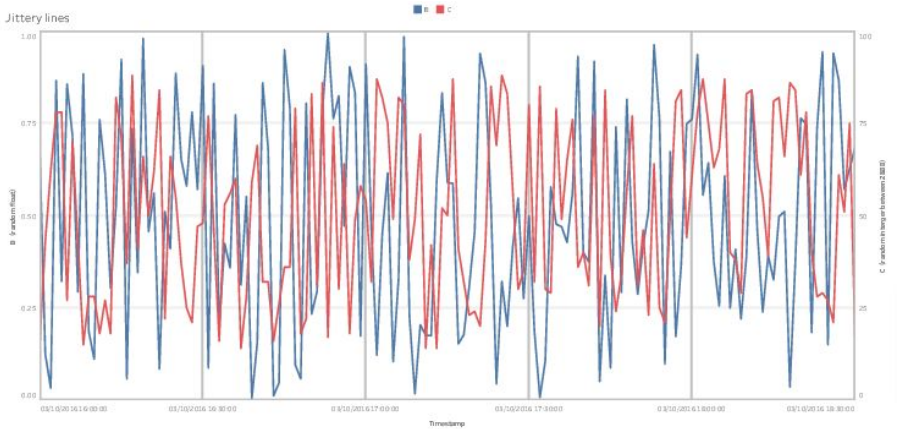
Solution:
https://github.com/suzannelittle/ca682i/blob/master/notebooks/solutions/3_1_11_Data_Visualisation_with_Python-solutions.ipynb



Replicate this graph

Congratulations **Anand** for a very close replication of jitterylines in python/matplotlib

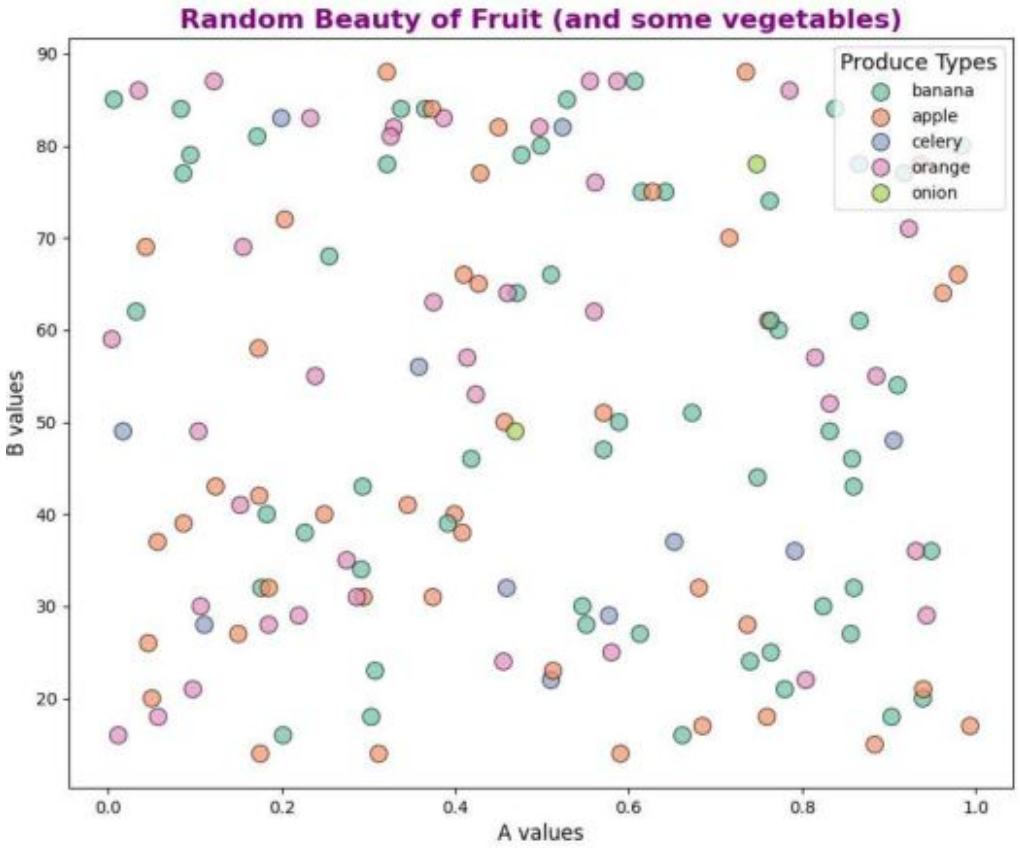




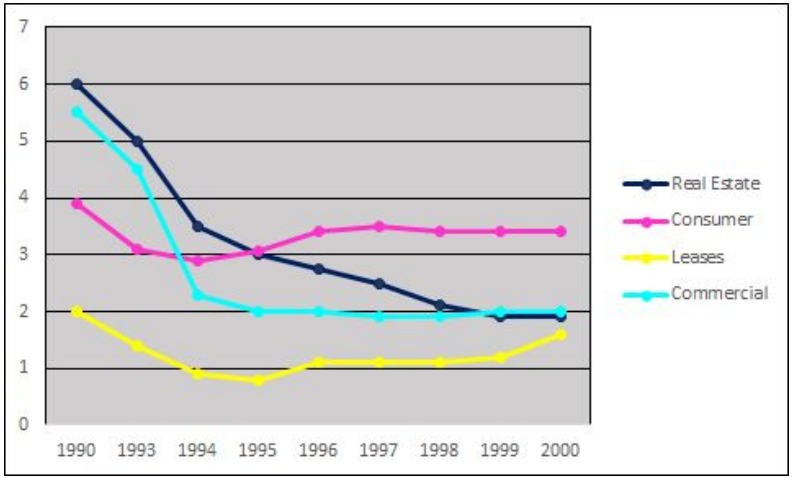
And **Tianrui** in Tableau

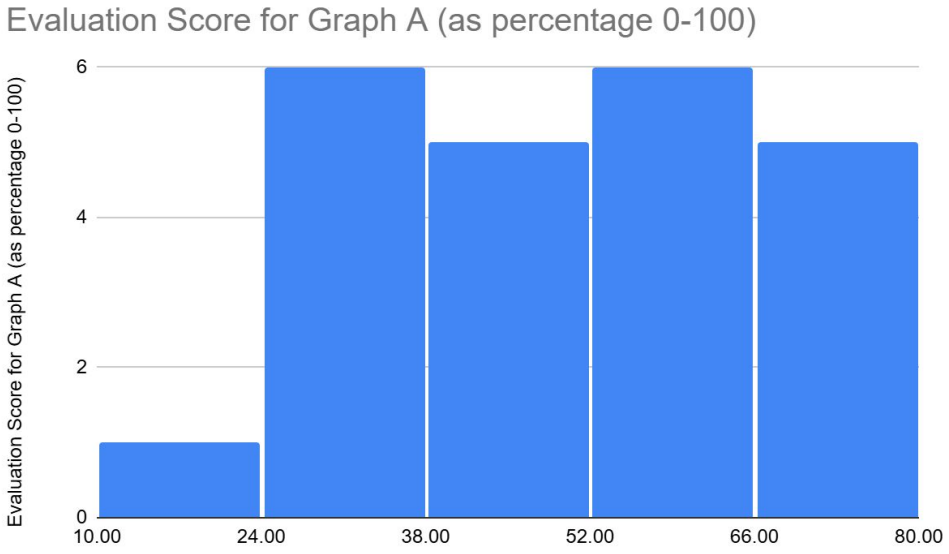
Replicate this graph

Haoquan for a very good effort on the Random Beauty of Fruit in python

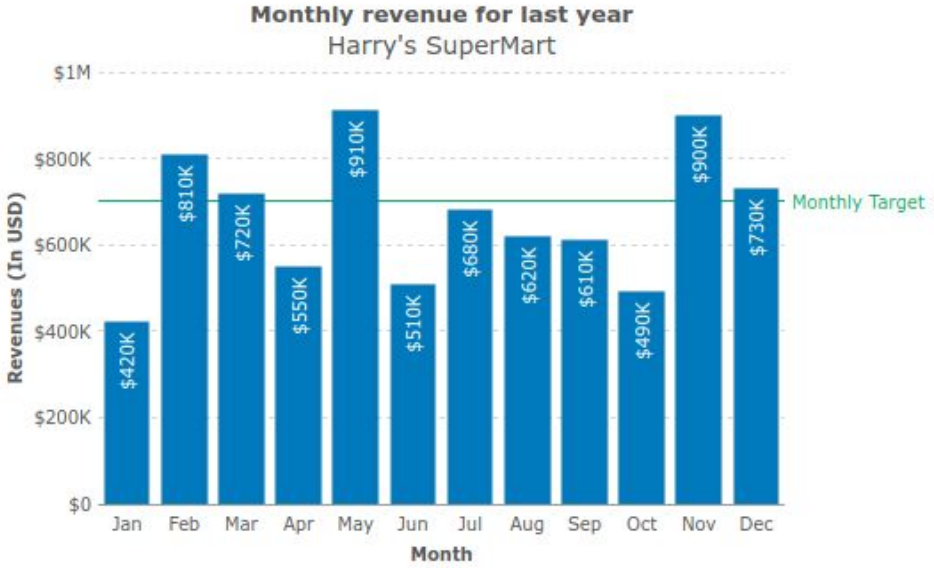


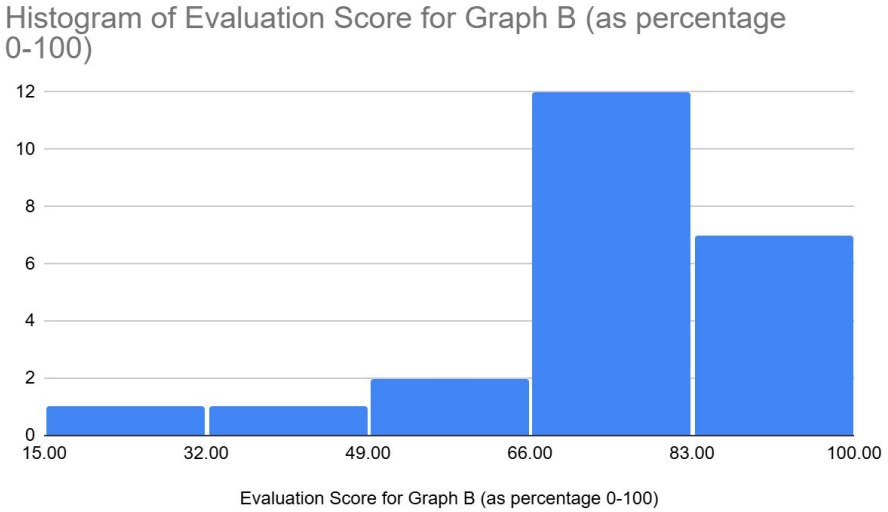
Graph Critique Scores: A



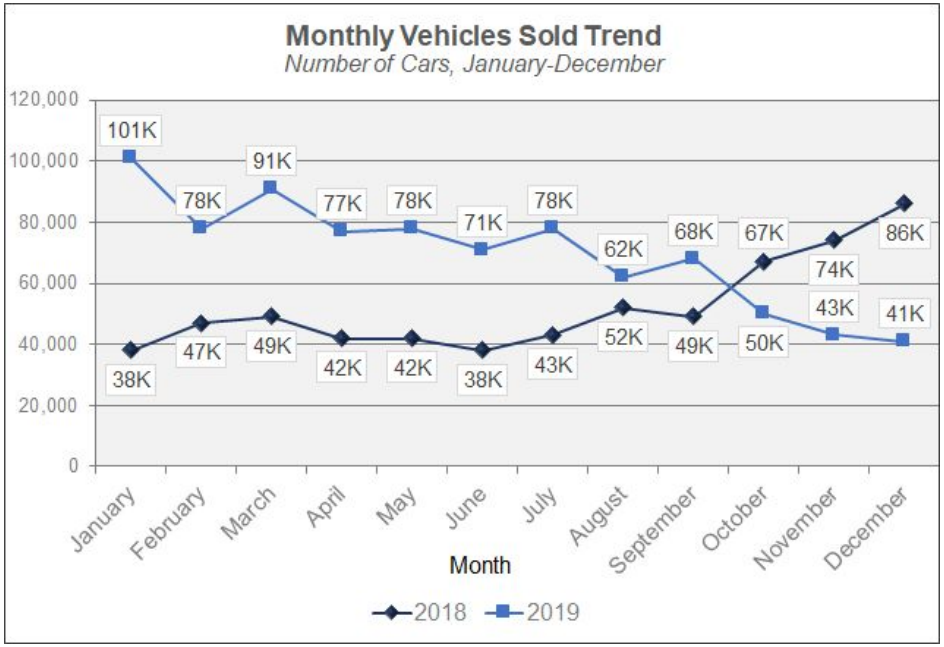


Graph Critique Scores: B





Improvements?



Switch to a Bar Chart: Use a bar chart instead of a line graph to represent data differences clearly.
Simplify Axes and Labels: Shorten month labels, remove the "Month" title, and adjust the Y-axis to display values in thousands (k) for clarity.
Refine Visuals: Remove unnecessary elements like gridlines, background, borders, and data point labels to reduce clutter.
Improve Labeling and Context: Add labels directly on bars or lines to avoid a legend, and ensure the Y-axis has a clear title.
Enhance Color and Contrast: Use contrasting, colorblind-friendly colors and add a distinct color to highlight target achievement where applicable.

Note: I asked ChatGPT to summarise all of your submitted text into 5 suggestions to improve the graph!

Labs today (LG25 & LG26)

Three options

1. Assignment
2. Datacamp
3. Or play with Map/Reduce in advance of next week: https://nbviewer.org/github/phelps-sg/python-bigdata/blob/master/src/main/ipynb/spark-mapreduce.ipynb (example of word counting using python)