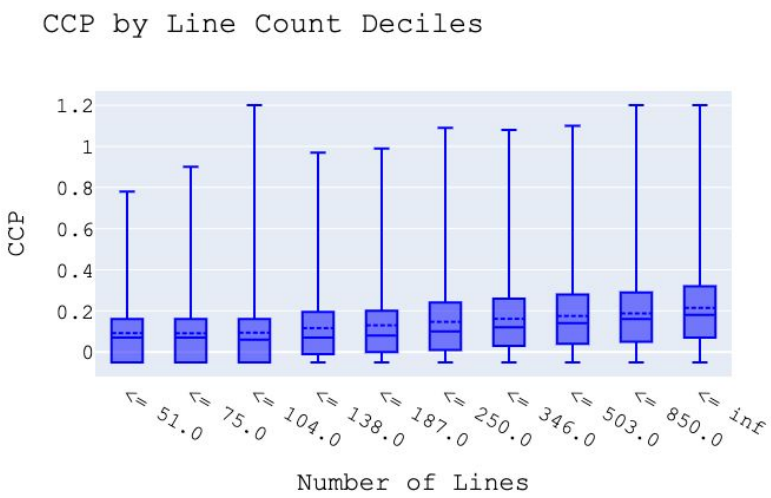
Databases for data analytics

https://github.com/evidencebp/databases-course/

SubQueries and Software Engineering

The need for subqueries

- SQL is a programming language (though declarative and different than the usual).
- The longer code is the higher tendency to bugs, difficulty to understand and modify



Subqueries can be used by encapsulating a part

- Example that we are already familiar with - “Movies per actor distribution”

```
select movies, count(*) as actors
from (select
actor_id, count(distinct movie_id) as movies
from
imdb_ijs.roles
group by actor_id ) as inSql
group by movies order by movies
```

Subqueries can be used in conditions

```
Select *
From imdb_ijs.actors
Where id not in
(select actor_id from imdb_ijs.roles as r
Join imdb_ijs.movies_genres as mg
on r.movie_id = mg.movie_id
Where genre = 'Comedy'
) #comedy_actors
```

NEVER use the previous pattern (0.094 vs 3.578 sec)

PLEASE use

```
Select *
From imdb_ijs.actors as a left join
(select actor_id
from imdb_ijs.roles as r
Join imdb_ijs.movies_genres as mg
on r.movie_id = mg.movie_id
Where genre = 'Comedy') as comedy_actors
on a.id = comedy_actors.actor_id
Where actor_id is null
;
```

View

- SQL equivalent mechanism of functions - break code into parts
- Create view XXX as query;
- Used once created as any table
  - Select * from XXX
- However, the view stores just the query, not the results.
  - You pay for each execution
  - Each execution returns up to date results

Stateful processes

- State should be kept
- It can be kept in tables, columns, are variables

Using databases from Python/R/Whatever

- Common scenario
  - Connect
  - Run a query
  - Get the results
  - Use them (not related to the DB)
  - Repeat as needed
  - Disconnect

Connect

Taken from https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html

```
import mysql.connector


cnx = mysql.connector.connect(user='scott', password='password',
                              host='127.0.0.1',
                              database='employees')
cnx.close()
```

Use DB (see link for original)

```
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 )
cursor.execute(query)
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()
```

In class exercises

- A view of pairs of movies of the same director
- Westerns without cowboys
- Actors per movie with probabilities

Exercises

- Implement “Movies per actor distribution with probabilities” with R/Python