Citing Dynamic Datasets

- Andreas Rauber (Vienna)
  - rauber@ifs.tuwien.at
- Webinar 01/10/2014

Problem statement

- many datasets are large and subject to change either through growth or correction
  - sensor data
  - environmental data
  - longitudinal or observational studies
  - econometric and sociological datasets
  (can also apply to software development, computational models etc)
- most researchers actually operate on a subset or extract of the data
- persistent identifiers for the datasets not sufficient, need to also identify accurately subset
- most current solutions suggest redepositing extracts
  - does not scale for big data or where there are many different extracts
- providing description of extract can lead to ambiguity
  - query may give different results

Approach

- any solution needs to be stable across technology changes
- scalable and capable of handling big data
- needs to be machine actionable for automated queries

Proposed solution

- solution is targeted at data only
  - does not reflect any changes in dataset metadata
  - does not provide bibliometric data on dataset use and reuse
- assumes data is stored in a repository or similar data management solution
  - add persistent query store to data management solution to
    - timestamp, track and journal all updates to database
    - timestamp, track and store all database queries
- aim is to allow a replay of the query at a known point in time
  - apply PID's to individual queries to make queries citable
  - normalise queries to ensure consistent replay especially if data later migrated to different platform
- idea is that when someone types in a query PID they get taken to a landing page
  - can choose to rerun query on dataset as was
  - or rerun query on current dataset
    - opportunity to compare results

CSV Data

- solution originally intended for SQL like solutions
- large amount of data exists in CSV format (often derived from spreadsheets)
- CSV often used extensively as a data exchange and distribution format
can leave data in CSV format
  - use existing version control system (eg git, svn, bzr) to handle and track changes
  - data is always as is
  - could become unwieldy where many changes and additions
• alternatively could load csv data into RDMS
  - CSV not one format and need to ensure ingest rules in place
• after loading into RDBMS proceed with SQL approach
  - allow autoextraction of data extracts as CSV
• have also looked at applying to data expressed as XML

**Closing remarks**

• solution is still under test/developement
• iterative approach taking real world datasets and attempting implementation
  - can therefore benchmark effort to implement in real world
  - can assess practicality and usability of solution