Level 1b

Documentation

https://github.com/UMC-Utrecht-RWE/ConcePTION-Level1b

Primary aim

- To asses how (syntactic) study variables are in the CDM of a database (DAP)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	person_id	mo_source_value	mo_meaning	mo_origin	mo_unit	mo_code	mo_recor	sex_at_in	mo_sourc	mo_sourc	visit_occu	mo_date	
2	#ID-00000001#			simulated	RTI-HS	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	19680809	
3	#ID-00000001#	179	height	simulated	cm	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	19680809	
4	#ID-00000001#	55.59	weight	simulated	kg	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	19680809	
5	#ID-00000001#			simulated	RTI-HS	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	19770715	
6	#ID-00000001#			simulated	RTI-HS	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	19930620	
7	#ID-00000001#	23.89	bmi	simulated	kg/m2	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	19930620	
8	#ID-00000001#	21.49	bmi	simulated	kg/m2	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	19980529	
9	#ID-00000001#			simulated	RTI-HS	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	20101004	
10	#ID-00000001#	169	height	simulated	cm	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	20101004	
11	#ID-00000001#	78.71	weight	simulated	kg	RTI-HS	RTI-HS		RTI-HS	RTI-HS	RTI-HS	20101004	

(if weight is a relevant study variable we need to know in which tables, and which columns and which cell value's)

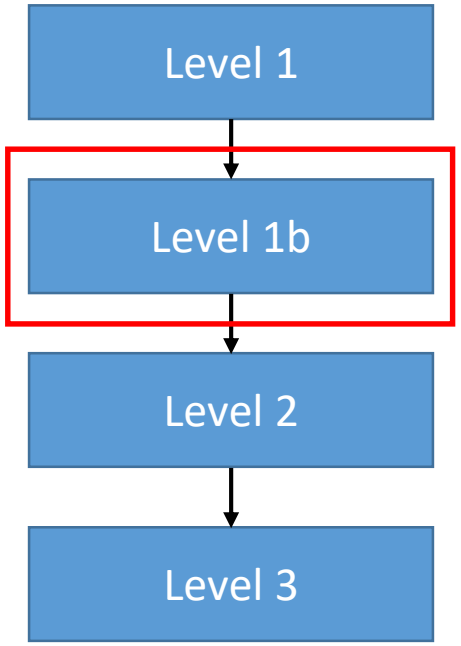
- Previous attempts to let the databases document this themselves failed. The delivered information did not often match reality.
- Mainly meant for programmers and PI's not for DAP's. Programmers need this information to develop there code and sampling data. PI's can check if all needed study information is available.
- Outputs are simple, no HTML reporting or fancy graphs

Secondary aim

- To asses which coding systems and codes are found in the data and relate this to the code lists

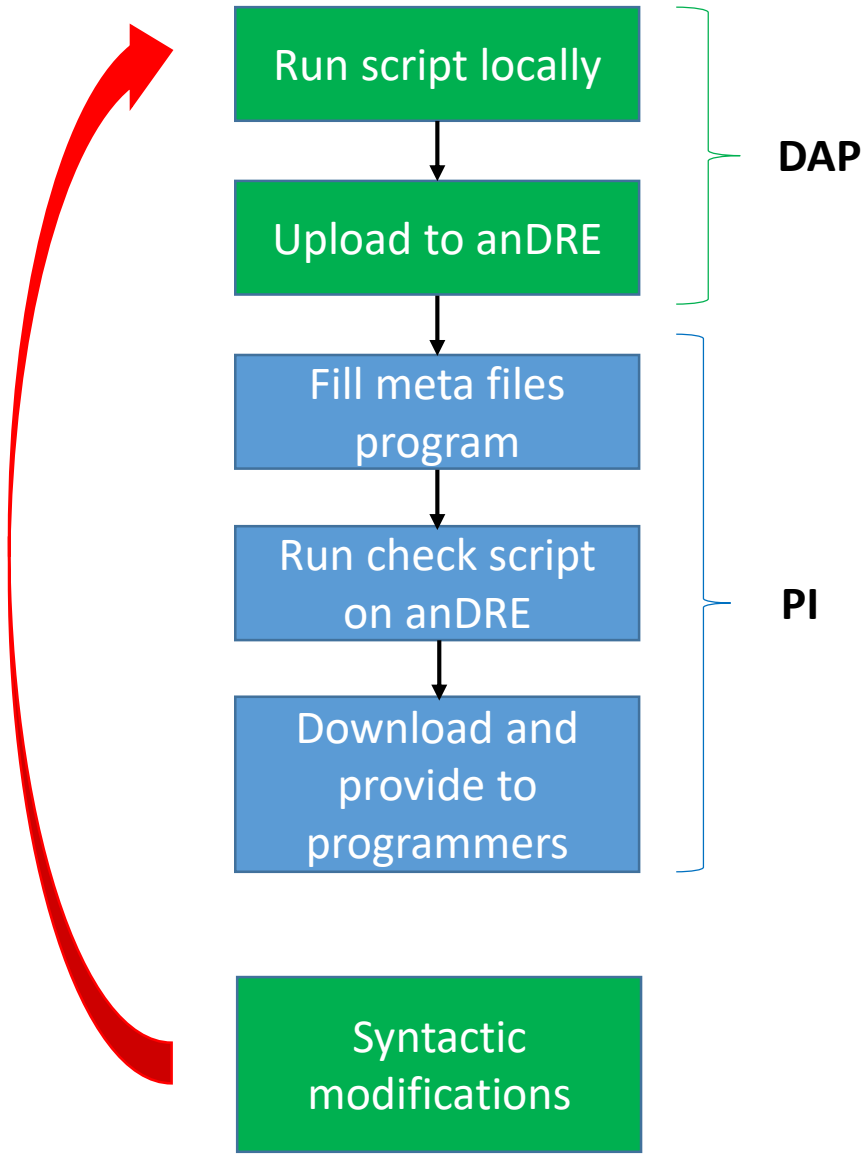
Place in the pipeline

- A part of the quality assessment
- It is independed because some basic checks are done at the beginning. So in theory it can run in all stages.
- The most logical place is to run after level 1 because simple assumptions (semantics) of the CDM need to be met



Execution procedure

- DAP's run the scrip locally
- The output is uploaded to DRE
- On DRE the meta files can be filled and it can be verified if all needed information is in the CDM
- If the meta files are filled, a test script need to be applied to test if the filled meta files are valid
- If valid, download the meta files so they can be used for the analytical script.
- If there any changes in the CDM along the way the procedure starts again from the beginning.



Needed parameters to set in to_run.R

- A reference to the CDM .csv files
  - Via a path:

    StudyName <- NULL

    path_to_fill <- "C:/CDMfolder"
  - Via a studyname/folder name in CDMinstances

    StudyName <- "studyname"

    path_to_fill <- NULL

- The CDM tables you want to analyze. By default all tables are analyzed.

  t.interest <- c("SURVEY_OBSERVATIONS", "MEDICAL_OBSERVATIONS")

  OR

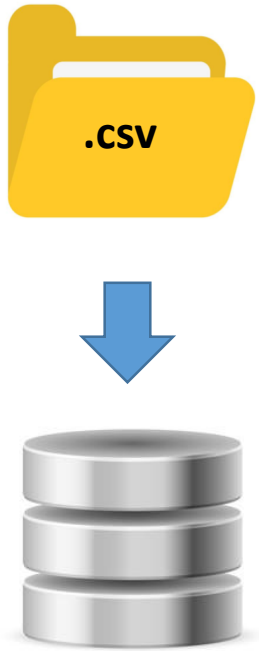
  t.interest <-NULL

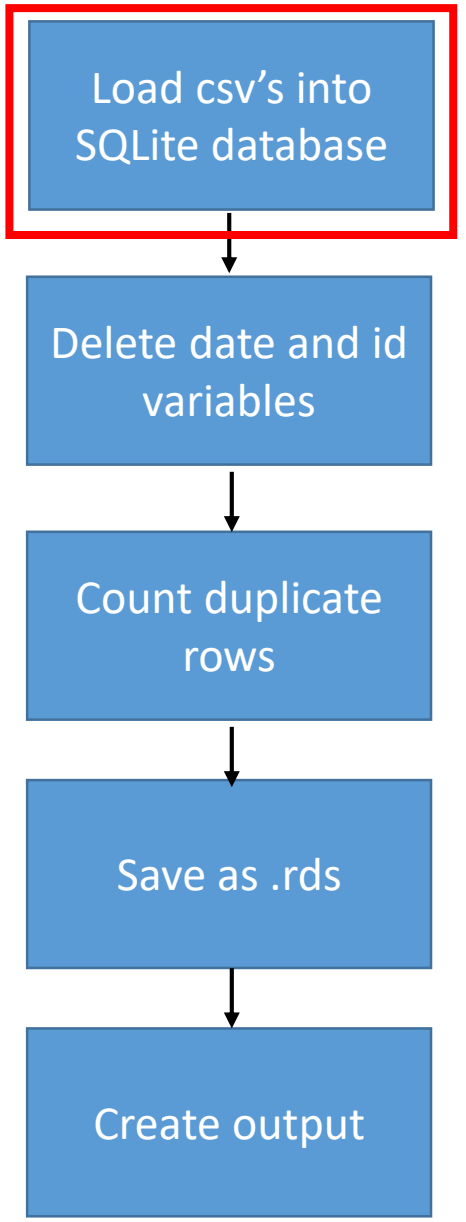
Main principle of the script (1)

person_id	mo_meaning	mo_value	date
p001	covid_test	pos	20000101
p004	covid_test	pos	20000102
p002	covid_test	pos	20000103
p003	covid_test	neg	20000104

person_id	mo_source	mo_value	date
p002	bmi	25	20200105
p003	bmi	23	20200106

person_id	mo_meaning	mo_value	date	mo_source
p001	covid_test	pos	20000101	
p004	covid_test	pos	20000102	
p002	covid_test	pos	20000103	
p003	covid_test	neg	20000104	
p002		25	20200105	bmi
p003		23	20200106	bmi





Advantage to use database

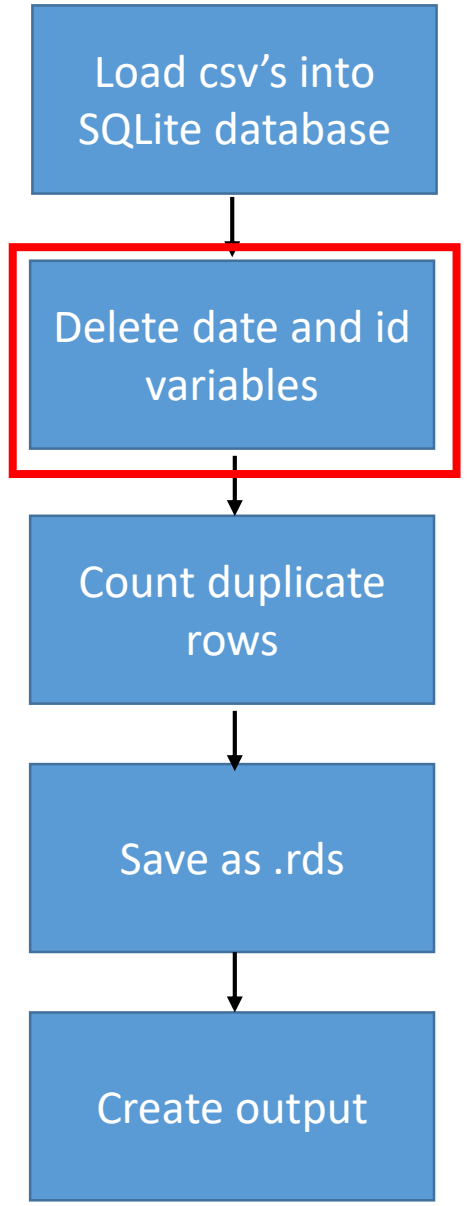
1. 1 table per CDM table instead of multiple
2. Possibility to add indexes
3. You do not have to read the whole table if you need only a part of it
4. Less change of memory issues and no need for loops to prevent this giving unreadable code

For appending column names and order need to be alike

Main principle of the script (2)

person_id	mo_meaning	mo_value	date	mo_source
p001	covid_test	pos	20000101	
p004	covid_test	pos	20000102	
p002	covid_test	pos	20000103	
p003	covid_test	neg	20000104	
p002		25	20200105	bmi
p003		23	20200106	bmi

mo_meaning	mo_value	mo_source
covid_test	pos	
covid_test	pos	
covid_test	pos	
covid_test	neg	
	25	bmi
	23	bmi

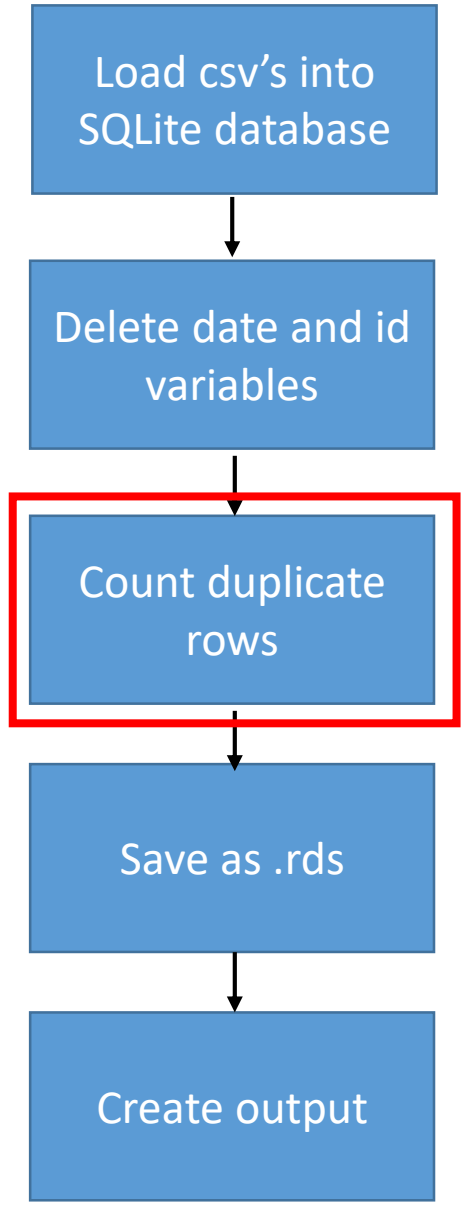


Aim: id's and dates are not relevant to asses the study variables

Main principle of the script (3)

mo_meaning	mo_value	mo_source
covid_test	pos	
covid_test	pos	
covid_test	pos	
covid_test	neg	
	25	bmi
	23	bmi

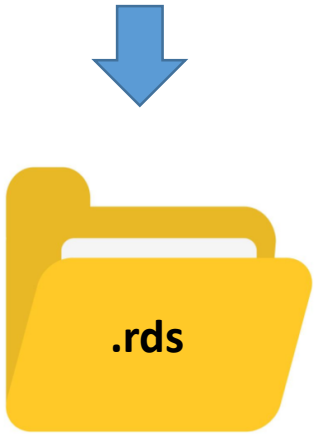
mo_meaning	mo_value	mo_source	N
covid_test	pos		3
covid_test	neg		1
	25	bmi	1
	23	bmi	1

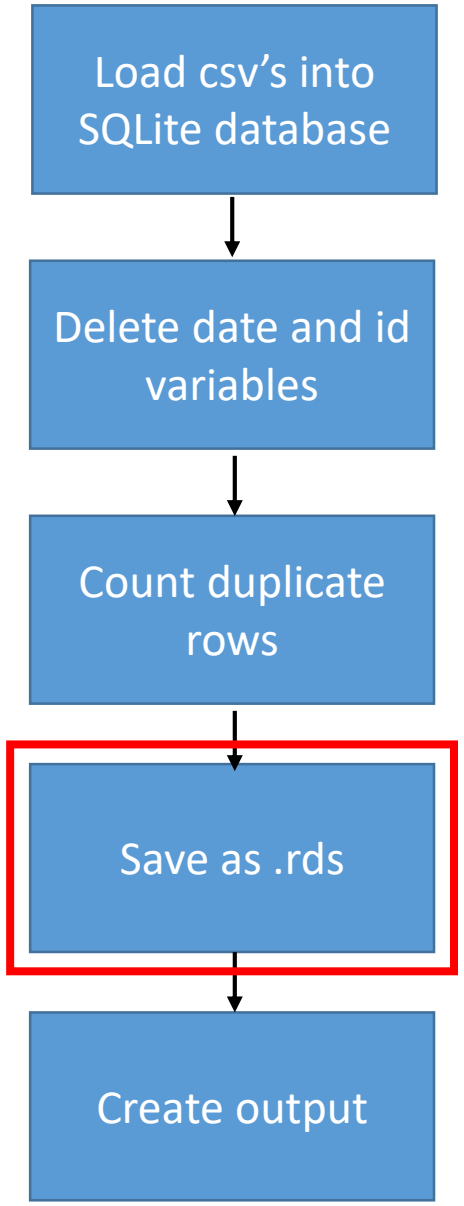


Aim: by aggregation the amount of information reduces drastically
1. Better overview of what is in the data
2. Further coding is less cpu and memory demanding.

Main principle of the script (4)

mo_meaning	mo_value	mo_source	N
covid_test	pos		3
covid_test	neg		1
	25	bmi	1
	23	bmi	1



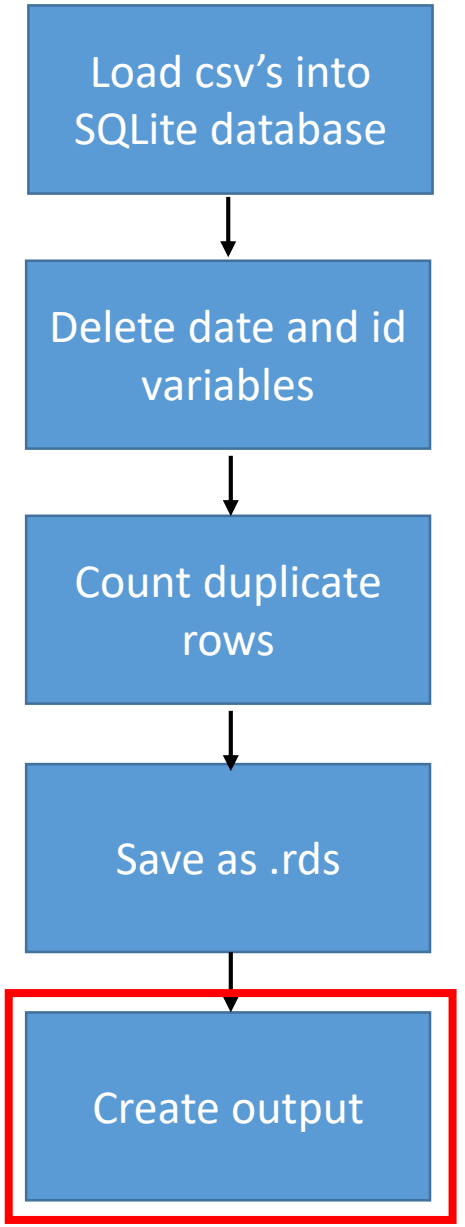


Because the size of the files are reduced .rds files may be considered workable in terms of performance and the need of looping to prevent memory issues will not be expected. Therefore, it was considered that a sqlite database was less needed in this intermediate step.

The advantages of using rds files (which is also a matter of personal opinion)

1. Files are easier to access for an R user
2. Further coding is more flexible in R

Main principle of the script (5)



Minor modifications are made to come from the intermediate .rds to the output files. Post processing will be done on DRE.

Summary of the main steps of the script (1)

1. Set paths needed for the program (99_path.R)
2. Delete all output files in g_output folder
3. Load needed functions (IMPORT_PATTERN.R, GetColumnNamesCDM.R)
4. Load needed packages (packages.R)
5. Retrieve DAP name and instance date from CDM_SOURCE.csv for naming output files (GetInfoOutputFiles.R)
6. Get meta information from CDM. The excel from this location https://drive.google.com/file/d/1hc-TBOfEzRBthGP78ZWIa13C0RdhU7bK/view should is copied to p_meta and need to be refreshed if it is changed in the cloud. The table names and column names that are part of the common data model are retrieved from this file using the function GetColumnNamesCDM.R. This is done in the to_run.R row 62-81.
7. Determine which tables to analyze based on the variable t.interest and the retrieved information from step 6
8. Per CDM table that needs to be analyzed, it imports the csv's, checks if column names are available and correct, and appends the table in a SQL database. (GetCounts.R). If column names are invalid, they are removed. If column names are missing then an empty column is added. This because appending requires equal table semantics.
9. Per CDM table, the data columns and id columns are cut of and 2 file types are created and saved as .rds file in g_intermediate (GetCounts.R). See next slide for examples
    1. WHERECLAUSE: this is a file with distinct rows and a column (N) with the number of rows counted for that distinct row.
    2. AWNSERS: here all the columns all the distinct values are stored and counted. This file can be prevented by setting *GetCountsColumns <- F*
10. Create outputs in .csv and .rds format (Step_002_CreateOutput). If PI's want to access the files csv files are outputted. If only needed for further processing .rds files are outputted.
    1. WHERECLAUSE (.csv): only action that is needed is setting counts less than 5 to "<5" because of privacy concerns
    2. AWSERS (.csv): only action that is needed is setting counts less than 5 to "<5" because of privacy concerns
    3. CODELIST: the code list files are focusing on the combination n of a coding system with a code and are analyzed on DRE to see if all expected codes are in the data.

Step 9 main outputs

WHERECLAUSE

	N	mo_source_value	mo_meaning	mo_origin	mo_unit	mo_record_vocabulary	mo_code	mo_source_table	mo_source_column
	All	All	covi	All	All	All	All	All	All
11551	387	simulated	covid_lab_test	simulated	RTI-HS	RTI-HS	RTI-HS	RTI-HS	RTI-HS
11552	1210	simulated	covid_lab_test	simulated	negtive	RTI-HS	RTI-HS	RTI-HS	RTI-HS
11553	1259	simulated	covid_lab_test	simulated	positive	RTI-HS	RTI-HS	RTI-HS	RTI-HS
11554	1167	simulated	covid_lab_test	simulated	undetermined	RTI-HS	RTI-HS	RTI-HS	RTI-HS
11547	1612	negative	covid19_antigen_test	simulated	RTI-HS	RTI-HS	RTI-HS	RTI-HS	RTI-HS

AWSERS

	N	Result	Column
1	1	1.1	mo_source_value
2	1	1.11	mo_source_value
3	1	1.13	mo_source_value
4	2	1.16	mo_source_value
5	1	1.18	mo_source_value
6	3	1.2	mo_source_value
7	2	1.21	mo_source_value
8	5	1.23	mo_source_value
9	1	1.24	mo_source_value

Further development opportunities

- Since it is working on a distinct interface of the data some simple quality checks can be done via this set of tables with higher performance. This checks only report if an assumption is met or not and it is obliged to fix this.
- If a simple set of assumptions is checked the level 1b may better to run before level 1 instead of after.
- Replace the google drive file with the CDM description to GitHub
- Split the script GetCounts.R in sub steps. First load sqlite database, then do the counts
- Add indexes on sql database and see if it goes faster.
- Detect continues values and delete them like id’s and dates OR standardize the CDM further
- Save .rds file without date and dap name. these files are intermediate