Processing OMOP data on Apache Spark Cluster by R Script

Masafumi Okada
masafumi.okada@iqvia.com
IQVIA Solutions Japan K.K.

Motivation

To handle *Real World Data* such as administrative claims database, the **first barrier** is often its large data size. Usually this kind of dataset will be stored in a relational database system, but it is sometimes troublesome **for statistical programmers** to operate the unfamiliar database system using SQL.
And the **second barrier** might be its non-standardized column names. Programmers do **not** like much of definition documents written in Word or PDF.

What is needed to resolve the problems

Large Data Manipulation without SQL

With Spark+R, clinical programmer …

- Can load ~50GB CSV/TSV text into Spark cluster.
- Can Extract/Transform data using normal dplyr functions. No SQL required.
- With Spark cluster, can handle ~50GB data from 8GB memory PC.
- Can apply an algorithm for each patient's data in parallel on Spark cluster.

Simple Standard for Column Names/Types

OHDSI OMOP CDM

- Public Domain Standard + Open Source Tools
- Only Tables, No XML.
- Standard defines column name, type, isRequired, Description, that's all.
- No ~400 page PDF documents.

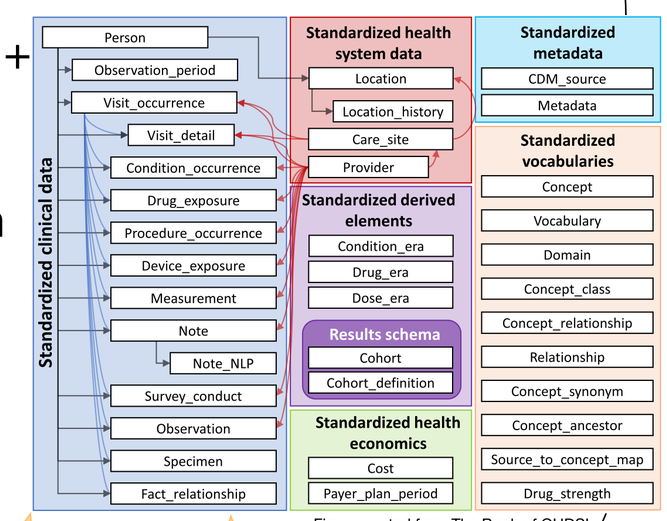


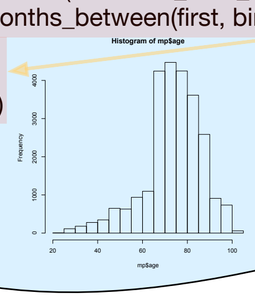
Figure quoted from The Book of OHDSI

Example code

Using Centers for Medicare and Medicaid Services (CMS) Medicare Claims Synthetic Public Use Files (SynPUF) 5% sample converted to OMOP CDM v5, draw histogram of age among CKD patients.

```
> library(sparklyr)
> library(dplyr)
> spark_install()
— Snip —
Installation complete.
> sc <- spark_connect(master="local")
* Using Spark: 2.4.3

> object.size(sc)
7208 bytes
> system.time(condition_occurence <- spark_read_csv(sc,"~/Downloads/synpuf5pct_20180710/
condition_occurrence.csv", delimiter='\t', header=FALSE, memory=FALSE))
   user  system elapsed
  0.469   0.134  82.827
> object.size(condition_occurence)
10808 bytes
> co2 <- condition_occurence %>%
select( `person_id`=V2 ,`condition_concept_id`=V3 ,`condition_start_date`=V4 ,  `condition_end_date`=V6 )
> person <- spark_read_csv(sc, "~/Downloads/synpuf5pct_20180710/person.csv",
delimiter='\t',header=FALSE)
> ps <- person %>% select(person_id=V1, year_of_birth=V3, month_of_birth=V4, day_of_birth=V5) %>%
mutate(birth_datetime=to_date(paste(year_of_birth, month_of_birth, day_of_birth, sep="-")))
> mp <- co2 %>% filter(condition_concept_id == 46271022 ) %>% group_by(person_id) %>%
summarise(first=min(condition_start_date)) %>% left_join(ps, by='person_id') %>%
mutate(age=months_between(first, birth_datetime)/12) %>% select(person_id, age) %>% collect
> nrow(mp)
[1] 25101
> hist(mp$age)
```



Import "sparkly" and "dplyr" package

Download and Install spark for local use automatically

Connect to spark local instance. If you have already set up spark cluster, you can connect it by specifying it as "master" argument.

spark object itself is very small
Load CSV large text file(1,637,181,751bytes) to spark. It took ~83 secs on my Desktop PC.
This file has "Condition_occurence" standardized table.

Data was loaded into Spark, not R. So the size of object is only 10,808 bytes.

DeSynPUF CSV file does not have headers, but the order of columns follow the OMOP CDM v5 standard.So we can name these safely.

Load another file (7,811,593bytes) to spark. This file has "Person" standardized table.

birth_datetime is not mandatory field in CDM v5, so generate it from year_of_birth, month_of_birth, and day_of_birth using R function `paste()`

From Condition_occurence table, first select CKD diagnosis (concept ID: 46271022), then pick up first date of diagnosis: min(condition_start_date). Next join this to Person table, calculate age by SparkSQL function `months_between()`. Finally copy the result in spark into R workspace by `collect()`

Resulted data has 25101 patients. This is normal R dataframe, so we can use `hist()` to draw histogram.

This code uses standardized name of columns only. Thus we can re-use this code for any other data sources which follows OMOP CDM. OHDSI people release some tools for data management and cohort data analysis that assumes data in OMOP CDM tables.

References for further information

- Javier Luraschi, Kevin Kuo, Edgar Ruiz, Mastering Apache Spark with R. `https://therinspark.com`
- Observational Health Data Sciences and Informatics, The Book of OHDSI. `https://ohdsi.github.io/TheBookOfOhdsi/`
- Observational Health Data Sciences and Informatics, A 5% sample (116,352 people) of simulated CMS SynPUF data in CDM Version 5.2.2 format. `https://www.ohdsi.org/data-standardization/`
- Centers for Medicare & Medicaid Services. Medicare Claims Synthetic Public Use Files (SynPUFs). `https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/index.html`