1. Instructions to execute the standardize FAERS data and generate safety signals process

LTS Computing LLC

<http://www.ltscomputingllc.com/>

The source code (shell scripts and SQL scripts) developed by LTS Computing LLC to execute this process is available as open source under the Apache 2.0 license on [github](https://github.com/ltscomputingllc/faersdbstats)

* 1. System Prerequisites

1. Linux. This process was developed and executed on Ubuntu LTS 14.04.
2. OHDSI Usagi mapping tool
3. PostgreSQL database. This process was developed and executed on PostgreSQL 9.3.
4. Pgadmin III PostgreSQL SQL client. Use this client to run the SQL scripts.
   1. Reference Data Prerequisites

* Load the FDA Orange book file. NDA reference data file into a PostgreSQL table called “nda” to look-up ingredients by NDA number
* Download the data file from the [FDA Orange book website](http://www.fda.gov/Drugs/InformationOnDrugs/ucm129689.htm)
  + Run the following SQL script to load the data file:
    - load\_nda\_table.sql
* Load the latest available OHDSI CDMV5 Vocabulary tables into a separate PostgreSQL schema (e.g. cdmv5). The data files and the PostgreSQL table load scripts are available from the [OHDSI Athena website](http://www.ohdsi.org/web/athena/)
  1. Process FAERS current data
     1. Download current data files from the FDA website

Run the current FAERS data download shell script:

* download\_current\_files\_from\_faers.sh
  + 1. Combine the current data individual files into combined files to load

Run the following current data shell scripts:

* create\_current\_all\_demo\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_drug\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_indi\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_outc\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_reac\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_rpsr\_data\_files\_with\_filename\_column.sh
* create\_current\_all\_ther\_data\_files\_with\_filename\_column.sh
  + 1. Create current data staging tables and Load data from each file

Run the following SQL scripts:

* load\_current\_demo\_table.sql
* load\_current\_drug\_table.sql
* load\_current\_indi\_table.sql
* load\_current\_outc\_table.sql
* load\_current\_reac\_table.sql
* load\_current\_rpsr\_table.sql
* load\_current\_ther\_table.sql
  1. De-duplicate current FAERS data

The aim of this section of the process is to remove duplicate cases from the current FAERS data

* + 1. Single value imputation of demographic keys then remove from further processing any rows with missing demographic keys
* Run the derive\_demo\_with\_imputed\_keys.sql SQL script
* Run the derive\_unique\_case.sql script SQL script
  1. Map current FAERS data drug names to RxNorm

The aim of this section of the process is to create a combined\_drug\_mapping table with the original drug name and the associated RxNorm concept\_id.

* + 1. Map current data drug name to RxNorm - Regular Expression
* Run the regex mapping section of the following SQL script to use regular expression mapping of current data drug table drugname to the cdmv5.concept\_table concept\_name and generate the drug\_regex\_mapping table:
  + map\_current\_drugname\_to\_rxnorm.sql
* Note. This will also create the combined\_drug\_mapping table with all unique current data drugnames, setting the update\_type column to 'regex - <regex type>' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_regex\_mapping table.
  + 1. Map current data drug name to RxNorm – Active Ingredient
* Run the Active Ingredient number mapping section of the following SQL script to generate the drug\_ai\_mapping table
  + map\_current\_drugname\_to\_rxnorm.sql
* Note. This will also set the combined\_drug\_mapping table update\_type column to 'drug active ingredients' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_regex\_mapping table.
  + 1. Map current data drug name to RxNorm – NDA Number
* Run the NDA number mapping section of the following SQL script to generate the drug\_nda\_mapping table
  + map\_current\_drugname\_to\_rxnorm.sql
* Note. This will also set the combined\_drug\_mapping table update\_type column to 'drug nda\_num ingredients' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_ai\_mapping table.
  + 1. Map current data drug name to RxNorm - USAGI
* Manually Export any unmapped drug names in the combined\_drug\_mapping table into a file to use as input into the USAGI tool using the following sql statement:
  + SELECT row\_number(), DRUG\_NAME\_ORIGINAL, count(\*) as number\_of\_cases FROM COMBINED\_DRUG\_MAPPING WHERE CONCEPT\_ID IS NULL;
* Open the USAGI tool. Follow the instructions on the [USAGI OHDSI wiki page](http://www.ohdsi.org/web/wiki/doku.php?id=documentation:software:usagi) to initialize and load the data file into USAGI. Note. As part of USAGI setup you will need the CDMV5 vocabulary data files from the Athena website.
* Manually map the drug names to RxNorm concept\_ids in USAGI and export the results to a file.
* Import the USAGI results data file back into a PostgreSQL table called usagi.
* Run the USAGI mapping section of the following SQL script to generate the drug\_usagi\_mapping table.
  + map\_current\_drugname\_to\_rxnorm.sql
* Note. This will also set the combined\_drug\_mapping table update\_type column to 'usagi' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_usagi\_mapping table.
  1. Standardize current FAERS RxNorm concepts to standard RxNorm concepts of Ingredient or Clinical Drug Form

The aim of this section of the process is to create a standard\_drug table with the original drug name and the associated standard RxNorm concept\_id.

* Run the standardize\_combined\_drug\_mapping.sql script (run-time is approximately 12 minutes)
  1. Derive current FAERS standard outcome category (FAERS outcome)

Derive the high level FAERS outcome categories (death, disability etc) for the current data standard\_drug table

* Run the derive\_standard\_case\_outcome\_category.sql script
  1. Derive current FAERS standard case outcome (FAERS reaction) from MedDRA preferred term to MedDRA concept\_id

Convert the FAERS reactions (adverse event outcomes) MedDRA preferred terms into MedDRA concept ids (limited to the cases in the unique\_case table) in a new table called standard\_case\_outcome.

* Run the derive\_standard\_case\_outcome.sql script
  1. Derive current FAERS standard drug outcome (FAERS reaction) counts

Create drug/outcome combination counts (counts for pairs of drug RxNorm concept\_id, outcome (reaction) Meddra concept\_id) and store the counts in a new table called standard\_drug\_outcome\_count

* Run the derive\_standard\_drug\_outcome\_count.sql script
  1. Derive current FAERS standard drug outcome (FAERS reaction) 2x2 contingency tables

Compute the 2x2 contingency table for each drug/outcome pari in a new table called standard\_drug\_outcome\_contingency\_table

* Run the derive\_standard\_drug\_outcome\_statistics.sql script (run-time is around 45 minutes)
  1. Derive current FAERS standard drug outcome (FAERS reaction) statistics

Compute the PRR, ROR and associated 95% CI limit statistics in a new table called standard\_drug\_outcome\_statistics

* Run the derive\_standard\_drug\_outcome\_statistics.sql script (run-time is around 45 minutes)
  1. Process FAERS legacy data
     1. Download legacy data files from the FDA website

Run the legacy FAERS data download shell script:

* download\_legacy\_files\_from\_faers.sh
  + 1. Combine the legacy data individual files into combined files to load

Run the following legacy data shell scripts:

* create\_legacy\_all\_demo\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_drug\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_indi\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_outc\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_reac\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_rpsr\_data\_files\_with\_filename\_column.sh
* create\_legacy\_all\_ther\_data\_files\_with\_filename\_column.sh
  + 1. Create legacy data staging tables and Load data from each file

Run the following SQL scripts:

* load\_legacy\_demo\_table.sql
* load\_legacy\_drug\_table.sql
* load\_legacy\_indi\_table.sql
* load\_legacy\_outc\_table.sql
* load\_legacy\_reac\_table.sql
* load\_legacy\_rpsr\_table.sql
* load\_legacy\_ther\_table.sql
  1. De-duplicate legacy FAERS data

The aim of this section of the process is to remove duplicate cases from the legacy FAERS data

* + 1. Single value imputation of demographic keys then remove from further processing any rows with missing demographic keys
* Run the derive\_demo\_legacy\_with\_imputed\_keys.sql SQL script
* Run the derive\_unique\_legacy\_case.sql script SQL script
  1. Map legacy FAERS data drug names to RxNorm

The aim of this section of the process is to create a combined\_drug\_legacy\_mapping table with the original drug name and the associated RxNorm concept\_id. Note:

* It is sufficient to re-use the drug\_usagi\_mapping tables that we created previously to map the current FAERS data.
* There is no active ingredient in the legacy\_drug FAERS data table
  + 1. Map legacy data drug name to RxNorm - Regular Expression
* Run the regex mapping section of the following SQL script to use regular expression mapping of legacy data drug table drugname to the cdmv5.concept\_table concept\_name and generate the drug\_legacy\_regex\_mapping table:
  + map\_legacy\_drugname\_to\_rxnorm.sql
* Note. This will also create the combined\_drug\_legacy\_mapping table with all unique legacy data drugnames, setting the update\_type column to 'regex legacy - <regex type>' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_regex\_mapping table.
  + 1. Map legacy data drug name to RxNorm – NDA Number
* Run the NDA number mapping section of the following SQL script to generate the drug\_legacy\_nda\_mapping table
  + map\_legacy\_drugname\_to\_rxnorm.sql
* Note. This will also set the combined\_drug\_legacy\_mapping table update\_type column to 'drug legacy nda\_num ingredients' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_ai\_mapping table.
  + 1. Map legacy data drug name to RxNorm - USAGI
* It is sufficient to re-use the drug\_usagi\_mapping table that we created previously to map the current FAERS data.
* Run the USAGI mapping section of the following SQL script to generate the drug\_legacy\_usagi\_mapping table.
  + map\_legacy\_drugname\_to\_rxnorm.sql
* Note. This will also set the combined\_drug\_elgacy\_mapping table update\_type column to 'usagi legacy' and the concept\_id to the matching (RxNorm) concept\_id in the drug\_legacy\_usagi\_mapping table.
  1. Standardize legacy FAERS RxNorm concepts to standard RxNorm concepts of Ingredient or Clinical Drug Form

The aim of this section of the process is to create a standard\_drug\_legacy table with the original drug name and the associated standard RxNorm concept\_id.

* Run the standardize\_combined\_drug\_legacy\_mapping.sql script
  1. Derive legacy FAERS standard case outcome category (FAERS outcome)

Derive the high level FAERS outcome categories (death, disability etc) for the legacy data standard\_drug\_legacy table in a new table called standard\_legacy\_case\_outcome\_category.

* Run the derive\_standard\_legacy\_case\_outcome\_category.sql script
  1. Derive legacy FAERS standard drug outcome (FAERS reaction) from MedDRA preferred term to MedDRA concept\_id

Convert the legacy FAERS reactions (adverse event outcomes) MedDRA preferred terms into MedDRA concept ids (limited to the cases in the unique\_case table) in a new table called standard\_legacy\_case\_outcome.

* Run the derive\_standard\_legacy\_case\_outcome.sql script
  1. Derive legacy FAERS standard drug outcome (FAERS reaction) counts

Create legacy drug/outcome combination counts (counts for pairs of drug RxNorm concept\_id, outcome (reaction) Meddra concept\_id) and store the counts in a new table called standard\_legacy\_drug\_outcome\_count

* Run the derive\_standard\_legacy\_drug\_outcome\_count.sql script
  1. Derive legacy FAERS standard drug outcome (FAERS reaction) 2x2 contingency table

Compute the 2x2 contingency tables for each drug/outcome pair in a new table called standard\_legacy\_drug\_outcome\_contingency\_table

* Run the derive\_standard\_legacy\_drug\_outcome\_contingency\_table.sql script
  1. Derive legacy FAERS standard drug outcome (FAERS reaction) statistics

Compute the PRR, ROR and associated 95% CI limit statistics in a new table called standard\_legacy\_drug\_outcome\_statistics

* Run the derive\_standard\_legacy\_drug\_outcome\_statistics.sql script (run-time is around 1 hour and 45 minutes)
  1. Derive combined legacy FAERS and current FAERS database 2x2 contingency table

Sum the combined counts from the standard\_legacy\_drug\_outcome\_count and standard\_drug\_outcome\_count tables into a new table called standard\_all\_drug\_outcome\_count and compute the 2x2 contingency table for each drug/outcome pair

* Run the derive\_standard\_all\_drug\_outcome\_contingency\_table.sql script
  1. Derive combined legacy FAERS and current FAERS database statistics

Compute the PRR, ROR and associated 95% CI limit statistics in a new table called standard\_all\_drug\_outcome\_statistics

* Run the derive\_standard\_all\_drug\_outcome\_statistics.sql script