Introduction to `drugprepr`

Belay B. Yimer David A. Selby Meghna Jani Goran Nenadic
Mark Lunt William G. Dixon

June 2021

Motivation

The Clinical Practice Research Datalink (CPRD) is a UK database of anonymised primary care electronic health records. It is widely used to study the effectiveness and safety of medications. However, prescription data from CPRD are often messy, with systemic issues such as missing information on prescription end dates and medication quantities, and prescribing instructions are typically provided as unstructured free text. In pharmaco-epidemiology studies, the data preparation steps used to determine drug exposure are rarely fully reported, yet assumptions made during this stage can have considerable implications for risk attribution of possible adverse events.

We have previously developed a framework[1] for dealing with missing information on prescription end dates and other issues such as overlapping prescription periods. The framework was implemented using Stata software. Besides being only available for Stata users, the earlier algorithm did not deal with the problem of free-text prescriptions. The current R package, `drugprepr` (formerly `drugprepCPRD`), builds upon the earlier algorithm and aims to make the framework available to a wider audience through its implementation as free, open-source software.

This vignette describes how to use the `drugprepr` package to transform CPRD drug data contained in a typical `therapy.txt` file into information on individuals' drug use over time. We will walk through all the steps needed to perform the transformation. We assume the user is already familiar with CPRD data and has basic knowledge of R software.

CPRD data

The CPRD Gold data follows the CPRD Gold specification. It is made up of several tables containing information related to the patients. One of the CPRD tables, called 'Therapy', contains the prescription information for a patient. This table can be linked to the 'common dosages' lookup table to get the text instructions for the prescribed product (i.e. a drug). Table 1, based on the fictional dataset `dataset1` that is bundled with this package, presents hypothetical prescription data for two individuals.

The `event_date` in the table is often used as the start date of exposure, but the stop date of the exposure is not available *per se* and we have to infer it from the available information. CPRD provides two options, namely `numdays` (number of days) and `dose_duration`, which can be used to determine the prescription stop date. However, these values are often missing, and even when present do not offer flexibility for the researcher in computing the stop dates in the case of prescriptions with a variable dose frequency (number of times the prescription is taken per day) and dose number (e.g. the number of tablets to take at a time).

As described below, the `drugprepr` package extracts the dose frequency and dose number from the text instructions and provide a series of data processing steps with multiple options to define start and stop dates of a given prescription. The package works at the `prodcode` level to give granularity for each medication.

[1] is a citation needed here?

Table 1: Prescription data for two fictional individuals. Stored in `dataset1`

patid	pracid	start_date	prodcode	dossageid	text	qty	numdays	dose_duration
2156	156	2011-06-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6
2156	156	2011-06-24	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6
2156	156	2011-07-10	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5
2156	156	2011-07-17	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5
2156	156	2011-07-17	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5
2156	156	2011-08-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	NA	7	6
2156	156	2011-08-24	1	NA		40	7	6
2156	156	2011-09-10	2	2	TAKE 1-2 THREE TIMES A DAY	NA	6	5
2156	156	2011-09-17	2	NA		24	6	5
2256	160	2011-06-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6
2256	160	2011-06-24	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6
2256	160	2011-07-10	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5
2256	160	2011-07-17	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5
2256	160	2011-07-17	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5
2256	160	2011-08-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	NA	7	6
2256	160	2011-08-24	1	NA		40	7	6
2256	160	2011-09-10	2	2	TAKE 1-2 THREE TIMES A DAY	NA	6	5
2256	160	2011-09-17	2	NA		24	6	5

The algorithm

The `drugprepr` is made up of the following functions that must be executed sequentially.

Compute numerical daily dose (ndd) Extract dose frequency (number of times the prescription is to be taken per day) and dose number (number of units of drug to take at a time) from the free text, in order to compute the numerical daily dose.

Define implausible values Given "plausible" values (based on prescribing guidelines and clinical experience), this stage identifies those values outside the plausible range, which may a result of data input or processing errors.

Decision 1: Handle implausible quantities Values outside the "plausible" range may be [1a] ignored, [1b] set to missing, or imputed. Imputation options include: [1c1] set to the mean value for that patient for that product code, [1c2] set to the mean value for that practice for that product code, [1c3] set to the mean value for the whole cohort for that product code, etc. See the package manual for all possible options.

Decision 2: Handle missing quantities Options for missing qty are: [2a] leave as missing, [2b1] set to the mean value for that patient for that product code, [2b2] set to the mean value for that practice for that product code, [2c] set to the mean value for the whole cohort for that product code, etc,.

Decision 3: Handle implausible numerical daily doses Options for implausible numerical daily doses are the same as for decision 1.

Decision 4: Handle missing numerical daily doses Options for missing numerical daily doses are the same as for decision 2.

Decision 5: Clean duration cleans implausibly high values for each of the three available duration variables (numdays, dose_duration, and qty/ndd). Options for cleaning each duration variable are: [5a] make no changes, [5b(X)] set to missing if duration is greater than X months, or [5c(X)] set to X if duration is greater than X months. X is 6, 12, or 24.

Decision 6: Select stop date defines a stop date for each prescription. calculate stop date as prescription start date + one of the following duration definitions: [6a] numdays, [6b] dose_duration, [6c] qty/ndd, or [6d(X)]

Decision 7: Handle missing stop date if stop date is missing: [7a] keep as missing, [7b] set to the mean value for that product code for that patient, [7c] set to the mean value for that product code for the whole cohort, [7d] set to the mean value for that product code for that patient, otherwise set to the mean value for that product code for the whole cohort.

Decision 8: Handle multiple prescriptions for multiple prescriptions for the same product code on the same day, but with different stop dates, options are: [8a] do nothing, ; [8b] calculate the mean duration of prescriptions and drop redundant records; [8c] keep the record with the shortest duration; [8d] keep the record with with the longest duration; [8e] sum the durations and drop

redundant records.

Decision 9: Handle overlapping prescriptions for consecutive records with overlapping start and stop dates, options are [9a] to ignore the overlap but sum ndds; [9b] move the overlapping time.

Decision 10: Handle short gaps between prescriptions handles small gaps between consecutive prescriptions by either allowing these gaps to remain classified as unexposed or reclassifying the gaps as exposed when the gap is less than a specified number of days. options include [10a] do nothing – the gap remains classified as unexposed; [10b(X)] move the stop date of the preceding prescription to "fill in" the gap, reclassifying the time as exposed, if the gap between consecutive prescriptions is less than X days. X is 15, 30, or 60

R package drugprepr

You can install the latest development version from `GitHub` with the following R commands:

```
install.packages('remotes')
remotes::install_github("belayb/drugprepr")
```

Once the package is installed, we load the `drugprepr` package into the working environment as follows.

```
library(drugprepr)
```

Computation of ndd from free text

This step uses another package developed for extraction of dose frequency and dose number—R package `doseminer`—and computes the numerical daily dose using the formula

$$\text{ndd} = \frac{\text{DF} \times \text{DN}}{\text{DI}},$$

where DF is dose frequency (number of doses per day), DN is dose number (number of units of medication taken per dose), and DI is dose interval (number of days between doses, where an interval of 1 means every day). The user must define here what DF or DN values to use, in case the freetext prescribing instructions indicate a range of possible values. Possible values are `min`, `max`, and `mean`. In the case of regular prescriptions (prescriptions with fixed instructions such as 'take 2 tablets 4 times a day'), the `min`, `max`, and `mean` value will be the same. Computation of numerical daily dose can be done using the `compute_ndd` function as follows.

```
data_ndd <- compute_ndd(dataset1, min, min)
```

Table 2: Sample output from the `compute_ndd` function, selecting minimum frequency and minimum dosage number

patid	pracid	prodcode	text	ndd
2156	156	1	TAKE 1 OR 2 4 TIMES/DAY	4
2156	156	2	TAKE 1-2 THREE TIMES A DAY	3
2156	156	1		NA

Here, we specified to use the minimum values for both the DF and DN in the computation of `ndd`. Running `compute_ndd()` creates an additional column names `ndd`, as illustrated in Table 2.

Defining implausible values

The next stage is to define the cut-off for plausible values of prescription quantity and numeric daily dose for each product. The information has to be provided by the users in a table format. The table should have column names: `prodcode`, `max_qty`, `min_qty`, `max_rec_ndd`, `min_rec_ndd`. Such information might be obtained from British National Formulary (BNF). For our hypothetical case, we defined the `min_max` data as in Table 3.

Table 3: Example data frame to supply to `min_max_dat`

prodcode	max_qty	min_qty	max_ndd	min_ndd
1	100	1	10	1
2	50	1	10	1

Once we prepare our `min_max` data in a table format, we can pass it to the second argument of `drug_prep()`. Internally, the function will join the plausible ranges to the different drugs and mark respective quantities and doses as plausible or implausible.

Processing the CPRD prescription data

Once these preliminary pre-processing steps are completed, we can use the main function of `drugprepr`, `drug_prep()`, to evaluate the 10 decision nodes described above. This can be done by executing an R command of the following form:

```
result <- drug_prep(data_ndd,
                    min_max_dat,
                    decisions = c('b', 'b1', 'b', 'b1', 'b_6',
                                  'c', 'a', 'd', 'a', 'b_15'))
```

Table 4: Result of `drug_prep()`

pracid	start_date	prodcode	dossageid	text	qty	numdays	dose_duration	ndd	optional	duration	stop_date
156	2011-06-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6	4	0	10	2011-06-26
156	2011-06-24	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6	4	0	10	2011-07-04
156	2011-08-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6	4	0	10	2011-08-26
156	2011-08-24	1	NA		40	7	6	4	0	10	2011-09-03
160	2011-06-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6	4	0	10	2011-06-26
160	2011-06-24	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6	4	0	10	2011-07-04
160	2011-08-16	1	1	TAKE 1 OR 2 4 TIMES/DAY	40	7	6	4	0	10	2011-08-26
160	2011-08-24	1	NA		40	7	6	4	0	10	2011-09-03
156	2011-07-10	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5	3	0	8	2011-07-18
156	2011-07-17	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5	3	0	8	2011-07-25
156	2011-09-10	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5	3	0	8	2011-09-18
156	2011-09-17	2	NA		24	6	5	3	0	8	2011-09-25
160	2011-07-10	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5	3	0	8	2011-07-18
160	2011-07-17	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5	3	0	8	2011-07-25
160	2011-09-10	2	2	TAKE 1-2 THREE TIMES A DAY	24	6	5	3	0	8	2011-09-18
160	2011-09-17	2	NA		24	6	5	3	0	8	2011-09-25

The result of running the function `drug_prep()` provides the start and stop date (`stop_date`) for each prescription.

To save remembering lots of alphanumeric codes, a helper function can generate them for you from more human-readable parameters:

```
decisions <- make_decisions('ignore',
                            'mean population',
                            'missing',
                            'mean practice',
                            'truncate 6',
                            'qty / ndd',
                            'mean individual',
                            'mean',
                            'allow',
                            'close 15')
decisions
#>      implausible_qty          missing_qty      implausible_ndd
#>                  "a"                 "b3"                  "b"
#>          missing_ndd implausible_duration   calculate_duration
#>                 "b2"                "c_6"                  "c"
```

```
#>     missing_duration           clash_start           overlapping
#>                  "b"                   "b"                   "a"
#>           small_gaps
#>               "b_15"
# drug_prep(example_therapy, plausible_values, decisions)
```