Construction of the UKB Synthetic Dataset

Naomi Allen, Howard Callen, Edward Horn, Shelly Lachish, Alan Young

1. Background

In late 2019, in anticipation of UK Biobank's (UKB) transition to utilising a cloud-based research analysis platform, the UKB data analyst (DA) team and the NDPH core programming team constructed a synthetic version of the UKB dataset. The purpose of this synthetic dataset was to provide the organisations that had applied to the cloud platform tender with a simulated dataset comparable in size, structure and complexity to that of the real UKB dataset, which they could use to develop, test and demonstrate their products and systems.

The synthetic UKB dataset contains 24,254 columns of simulated data, corresponding to 7,324 data-fields contained in the 'main' UKB dataset, as well as a set of tables containing simulated primary care record-level data that are not included in the 'main' dataset (record-level hospital data was not simulated). Additional files simulating genetics and bulk data were also constructed. This document outlines the construction of the main dataset and the primary care records by the UKB DA team.

2. Caveats

The remit for this task was to construct a synthetic version of the UKB dataset where simulated data fields were labelled as per those that are available to researchers on the UKB Showcase website and that was no smaller, and ideally larger, in size to the existing UKB dataset (to ensure any cloud-based infrastructure developed would be able to deal with the anticipated future growth in size of the UKB dataset).

In order to maintain some consistency with the real data, and to provide enough deviation from complete randomness so that analysis questions could be asked during the tendering process, simulated dates of events for participants were allocated within the real date range of events where possible (e.g. for baseline assessment centre visit and imaging follow-up visits), and certain other correspondences that exist within the real data were reproduced to a certain extent. However, there was no general requirement to ensure any internal coherence, clinical logic, or medical diagnostic 'sense' in the simulated data.

Hence, for example it is entirely possible for a simulated 'female' participant to have a simulated cancer record for prostate cancer or conversely for a simulated 'male' participant to have a simulated GP visit for a pregnancy ultrasound. In a similar way, dependencies between fields, such as one touchscreen question only being asked if an earlier question was answered a certain way, will not be respected in the synthetic dataset. Many other such clinical or logical discordances will occur within this dataset. Researchers should bear this in mind when using this data.

3. Simulating participants and instances

To ensure the simulated dataset size was larger than the real current UKB dataset, the DA team simulated 600,000 participant id numbers (pids: 1 through 600,000) at baseline (instance 0 – from 2006 to 2010). Odd pids were designated 'female'; even pids were designated 'male'. The first 100,000 pids numbers were assigned to the imaging visit (instance 2 – from 2014), while the first 70,000 pids were assigned to the repeat imaging visit (instance 3 – from 2019). Finally, pids in the range 65,001 to 75,000 and 590,001 to 600,000 were assigned to the repeat assessment visit (instance 1 – 2012). Once these selections were

made (and after these pids had been used to simulate data; see below), a random six digit encoded id (eid) was assigned to each participant id and the data sorted by the new eid column for release.

4. Simulating participation in activities

The selection as to which participants had data on which activities (e.g., questionnaires, physical activity monitors, imaging assessments) was made in such a way so as to reflect the complexity, structure and nuances of the real data and introduce some "light & shade" into the synthetic data.

An algorithm was devised that assigned participants to these activities with varying probabilities. As such, participants that were selected to have imaging data (i.e. to have data at instance 2, corresponding to the initial imaging visit) were also more likely to have data from online follow-up questionnaires and to have data from physical activity monitors, and those participants selected for both the initial and repeat imaging visits (instances 2 and 3) were even more likely to also have these data. From the remaining groups, ranges of pids were selected to have higher or lower likelihood of involvement in these activities. A random subset of pids was intentionally given a very low probability of questionnaire/accelerometer participation, to mimic those participants who rarely participate in additional activities (~40% of pids).

5. Simulating deaths and cancer diagnoses

A random sample of ~40,000 participant ids were generated and these participants were deemed to have died (with the date and cause of death then simulated separately). To ensure deaths occurred in participants with data across all instances, deaths were not randomly sampled across all participants but were done so in batches; i.e. 604 deaths were randomly drawn from the first 70,000 pids, 1277 deaths were drawn from pids in range of 70001 to 100,000 and 37,683 deaths were drawn from pids in the range 100,001 to 600,000. Dates of death of participants were chosen to occur after their last attendance at an assessment centre.

Participants with cancer diagnoses were 'selected' in a similar fashion; 27,418 eids were randomly selected to have ICD-9 cancers that occurred prior to October 2011 and 96,892 eids were randomly selected to have ICD-10 cancers that occurred thereafter.

6. Simulating data fields by data type

To simplify the simulation process, the data fields were simulated in batches according to their data type: real number fields, integer number fields, date fields, time fields, string fields, and bulk fields.

a. Simulating real number fields

- [real_fields1.tsv, real_fields2.tsv]

Simulation of real number fields was performed assuming that all fields followed a normal distribution. Data for each real number field were extracted and any extreme values, or values with a special coded meaning, were removed. The mean and variance of each combination of field, instance and array, and the covariance between each pair of combinations were calculated for use as input parameters to the simulation algorithm (*simnormal* function in SAS 9.4). This approach was used to capture the correlation between fields in the simulated dataset. In cases where the covariance was inestimable (e.g. where values for different field, instance and array combinations did not exist for more than one participant), the simulated data were generated independently using a random number generator. The mean and standard deviation of the corresponding field were used as input parameters to the algorithm in order to constrain the distribution of the simulated data.

b. *Simulating integer number fields*

- [integer_no_arrays.tsv , integer_arrays_part1.tsv, integer_arrays_part2.tsv, integer_diet_quest_fields.tsv, integer_other_quest_fields.tsv]

Simulation of integer number fields was achieved by randomly sampling (with replacement) from the range of true integer values possible for a particular field with sampling probabilities derived from a random uniform distribution (*runif* function in R). 600,000 simulated integers were generated for each integer number field at each instance. The simulated integer values corresponding to the pids present at a particular instance were then assigned accordingly (e.g. at baseline all pids would have been assigned a simulated integer number for a particular integer field, whereas at instance 3 for a particular field only the first 70,000 pids would have been assigned a simulated integer number). This approach was used for integer fields with and without arrays (arrays = multiple integer values are possible per pid per instance), and for integer fields where values are true numbers and those where values are encodings.

c. *Simulating date/time fields*

- [datetime_fields.tsv , datetime_fields_2.tsv, dates_death.tsv, *_HES_SimDates.tsv]

The approach here was to create a set of rules for how each combination of field, instance, and array should be simulated for each participant. For each combination, these rules would depend on the nature of the field, and included the following types:

- A start date/time was manually specified along with a minimum (often 0) and maximum number of days/seconds. A random number of days/seconds between the min and max was then generated and added to the start date/time to derive the value for that field.

- A particular field/instance/array combination could be specified to have the same value as another field for which a value had already been simulated for that participant (and having the same instance and array index), or to have a random number of days/seconds added to that previous value (between specified limits).

The aim of this approach was to make the dates have a certain amount of logic to them. For example, the date of the baseline assessment visit for all participants was simulated to occur during the true date range for this event (i.e. 2006-2010), and participants selected for the repeat imaging visit were assigned simulated dates for that visit that were approximately two years after their first imaging visit date (as per the real dataset). In addition, the time values generated for measurements obtained at an assessment centre were simulated to occur on the correct day and in a particular order.

Researchers should note however that many date/time values will nonetheless still be illogical since not every interdependency will have been faithfully represented by the rules used and many randomly chosen increments of days/seconds will be too short or long to be genuinely realistic.

d. *Simulating first occurrence and algorithmically defined outcome fields*

- [fo_fields_trimmed.tsv, oaa_fields.tsv]

Simulation of the date and source fields corresponding to the first occurrence (FO) and algorithmically-defined outcomes (AO) fields were performed separately. This was done to:

(a) ensure that each FO or AO source field (which was simulated as per the integer fields) would always have an accompanying date field, and

(b) mimic the frequency of the outcome events in the real dataset, in particular simulating the extreme sparseness of these fields within a main dataset.

For each source value field/date field combination the number of participants having data for that pair of fields was taken from the real data and then adjusted by up to a few hundred either way.

In simulating FO and AO date fields no attempt was otherwise made to get the dates to be 'sensible' or to coordinate with any other data for that participant.

e. Simulating string fields

- [string_fields1.tsv, string_fields2.tsv]

The method used for simulating string data was determined based on whether the values for the field were encoded. For fields made up of encoded values, data were simulated by drawing random samples with replacement from the set of encoded values corresponding to the simulated field (performed using the 'surveyselect' procedure in SAS 9.4). For the remaining fields, data for each combination of field, instance and array were extracted, and the minimum and maximum number of characters across all values was calculated for each combination. A sequence of integer values between 65 and 90, with random length between the calculated minimum and maximum number of characters for the corresponding combination was generated using a random number generator. Each integer within the sequence was then mapped to an alphabetic character based on the ASCII collating sequence.

f. Simulating summary bulk data fields and bulk data

- [bulk_strings.tsv]

Bulk data types in the UKB resource are obtained separately to a researchers' main dataset. However, for every bulk data field available there is a field in the main dataset indicating whether that participant has data for that particular bulk data field. This model was followed to simulate bulk data files in this synthetic dataset, by generating synthetic summary bulk data fields (string fields) by inserting the name of the particular bulk field against each participant 'selected' to have that bulk field. Subsequently, simulated bulk data files were created wherever there was simulated content for a participant in a summary bulk data field (using a Linux script that generated a file type and size typical for the particular bulk data field).

g. Simulating synthetic GP data

The GP data is presented in 6 files (set3a1, set3a2a, set3a3, set3a4, set3b and set3c), together containing 400,000,000 rows of data. Files with '3a' in the name contain Read2 codes, those with a '3b' or '3c' contain Read3 (to reflect the fact that TPP GP data is coded in this way) though obviously both files have both columns. The file "set3a2a" contains some long strings to replicate the fact that the Scottish values contain non-numerical free-text - up to 723 characters for value1, 566 for value2 and 237 for value3.

A large dataset of synthetic Read2 clinical codes was obtained from the list of existing (real) Read2 clinical codes (see Showcase Resource 592). The selection procedure was run four times, each time obtaining 40,000,000 clinical codes (using an unrestricted random sampling method, with equal probability and replacement, in SAS). The four datasets generated were combined to create one dataset of 160,000,000 Read2 codes, with codes indicating 'data provider' assigned to each record based on the value of a random number (between 0,1) assigned to that record (<0.35 = provider 1, 0.35 – <0.65 = provider 2, >=0.65 = provider 4). A similar process was used to construct the 240,000,000 row dataset of synthetic Read3 clinical codes (though with all records assigned to data provider 3).

Synthetic dates were generated for each of the clinical codes by randomly sampling from the range of true dates. Each clinical code/date combination in this large dataset was then assigned to an eid according to a table of visits per participant (itself generated by drawing randomly from the distribution of visits per participant in the real data), until all records had been assigned.

7. Post-processing

Some post-processing of the various synthetic data files produced was undertaken in Linux to align the column name format across all files, to check which columns were included in each file, to remove duplicate columns where they occurred, and combine different files.