**README FOR CHAPTER 4 CODE**

Table of Contents

[General instructions 1](#_Toc36473844)

[Packages 2](#_Toc36473845)

[Data information 2](#_Toc36473846)

[Program information 3](#_Toc36473847)

**DISCLAIMER:** The data supplied is simulated at random from the real distributions of the respective variables, the code will run and produce results and figures, but the results are meaningless

# General instructions

* All programs run from a root/parent directory, where all the provided files and directories should be stored. At the top of each R program, set the working directory to this, or place an Rproject file in this root directory and then nothing needs to be added to the code.
* The programs are based on two imputed datasets (R data frames) that are stored in the /R\_out\_C4/chapter4\_fake\_data.RData workspace. More detail on these data frames is given in section ‘Data information’.
* There is a batch file (AA\_run\_all\_batch.sh) that will run all the programs in order to carry out the analysis from the UNIX command line. However the working directory must be set at the top of each .R file in order to do this. I would not recommended running these programs in the R gui using the R project approach, as you need to keep Rstudio open and some programs take ~ 8 hours to run.
* I have simulated a historical cohort of size 500000, and a contemporary cohort of size 50000. For each individual, each covariate is simulated at random from the real distribution of that variable from the dataset used the real analysis. This means that no variable has an association with the outcome and everybody’s risk according to the population derived model is very similar. This risk happens to be in the 4-5% risk group, and means there is only one boxplot in each figure produced, as there are no patients in the other risk groups. Note this also messes with the ordering of the boxplots. Despite this, I hope being able to run the code with some data is still helpful.
* The figures that will be produced when running the code are already provided in the /figures/ directory. Code can be re-run to compare.
* I have just supplied code for female cohort, given the code for the male cohort is identical, all that was different was the contents of the datasets read in at the start, and that Nmin = 1405 instead of 1434, and Nepv10 = 2297 instead of 2954.
* The code for calculation of Nmin is exactly as was used in the main body of work. Given the loaded data is fake, it will not give a value of 1434, this file doesn’t have to be run to produce results. It is therefore just there for transparency.
* Programs were run using R.3.4.2.

# Packages

The package versions in the below table were used. These were the package versions installed on the remote computer I used for the analyses so some are quite old, however in most cases I see no reasons why the latest package versions wouldn’t work.

|  |  |
| --- | --- |
| Package | Version |
| foreach | 1.4.4 |
| doParallel | 1.0.11 |
| tidyverse | 1.3.0 |
| ggpubr | 0.1.8 |
| knitr | 1.20 |
| survival | 2.42-3 |
| reshape2 | 1.4.3 |
| Hmisc | 4.1-1 |
| matrixStats | 0.52.2 |
| dca | 0.1.0.9000 |

# Directory structure

The directory structure is as follows:

Code: This is where all the code is stored

Figures: Figures generated from running the code will be saved into this location

R\_out\_C4: This is where all R workspaces that are generated are saved to. Initially, it contains *chapter4\_fake\_data.RData,* which is a workspace containing the fake datasets provided to run this code on. After running all the provided code, this will contain multiple intermediate workspaces.

# Dataset information

Two datasets are provided in the workspace /R\_out\_C4/*chapter4\_fake\_data.RData.* I provide information about the variables in the datasets, in case you want to repeat the analysis using your own data. All variables are generated at random.

**1) data.2016**: This contains 50,000 patients in the format of the contemporary cohort, that were actively registered on the 1st Jan 2016. All date variables are in the format ‘01/01/1990’, apart from the first cvd event, which is in form ‘01/01/90’. The variables included are:

|  |  |
| --- | --- |
| Variable | Definition |
| patid | Patient id |
| cens\_fake | A dummy variable so that we can extract data from the dataset into a format that can be used to generate risk scores |
| age | Age |
| BMI | Body mass index |
| Cholesterol\_HDL\_Ratio | Cholesterol/HDL ratio |
| Famhis\_lstrict | Family history of coronary heart disease |
| Hypertension | Treated hypertension |
| SBP | Systolic blood pressure |
| Smoking | Smoking |
| Townsend | Townsend deprivation index quintiles |
| T2dia | Type 2 diabetes |

**2) data.historical**: This contains 500,000 patients in the format of the historical cohort, these are patients that are used to developed the risk prediction models. Variables are the same, except two extra variables are added:

|  |  |
| --- | --- |
| Variable | Definition |
| CVD\_time | Time until CVD event or censored |
| CVD\_cens\_R | 0 = censored, 1 = CVD event |

Figures are also provided in the

# Program information

The table below details what is done in each program.

As long as all the code and directories are placed within a root directory, the code can be run by setting the working directory to the parent directory of your choice at the start of each program. Alternatively, an Rproject file can be placed in this parent directory, which will automatically set the root directory when R is opened and ran. The code must be ran in numerical order to produce results.

The individual level data used in this project could not be shared. Therefore in the datasets provided, each individual’s covariates are generated at random from the real distributions of those covariates. This means there is no correlation between any variables, and the figures generated from running this code on the provided datasets make no sense. However hopefully providing this will allow the code to be checked for those interested, and the ability to run the code should aid that process.

One final note, the main bulk of the simulation is ran from programs 2.1 – 2.5. The simulation is then ‘re-run’ to calculate the net benefit for each model. The reason this is done separately is that this was a late addition after reviewer comments, and we did not want to re-run the whole simulation (i.e. calculating stability of risk scores, and the calibration, discrimination and the MAPE of each model) again, and so created separate programs to calculate the net benefit of each model. The same seed is used in these programs to ensure models in which the net benefits is calculated, are the exact same models used to calculate the stability of risks.

|  |  |
| --- | --- |
| Program | Definition |
| 0.1 | Install packages |
| 1.1 | Create the development and validation cohorts from the historical cohort |
| 1.2 | Generate risks of validation cohort using population derived model, and calculate discrimination and calibration of the population derived model |
| 1.3 | Code for calculating Nmin |
| 2.1 | Run simulation for N=Nmin |
| 2.2 | Run simulation for N = 10,000 |
| 2.3 | Run simulation for N = 50,000 |
| 2.4 | Run simulation for N = 100,000 |
| 2.5 | Run simulation for N = Nepv10 |
| 2.6 | Calculate net benefit for models when N=Nmin |
| 2.7 | Calculate net benefit for models when N = 10,000 |
| 2.8 | Calculate net benefit for models when N = 50,000 |
| 2.9 | Calculate net benefit for models when N = 100,000 |
| 2.10 | Calculate net benefit for models when N = Nepv10 |
| 3.1 | Generate Figure 2 (boxplots) |
| 3.2 | Generate Figure 3 (boxplots by discrimination, N = Nmin) |
| 3.2 | Generate Figure 3 (boxplots by discrimination, N = Nepv10) |
| 3.2 | Generate Figure 3 (boxplots by discrimination, N = 10000) |
| 3.2 | Generate Figure 3 (boxplots by discrimination, N = 50000) |
| 3.2 | Generate Figure 3 (boxplots by discrimination, N = 100000) |
| 3.3 | Generate Figure 4 (boxplots by calibration, N = Nmin) |
| 3.3 | Generate Figure 4 (boxplots by calibration, N = Nepv10) |
| 3.3 | Generate Figure 4 (boxplots by calibration, N = 10000) |
| 3.3 | Generate Figure 4 (boxplots by calibration, N = 50000) |
| 3.3 | Generate Figure 4 (boxplots by calibration, N = 100000) |
| 3.4 | Generate Figure 5 (boxplots by MAPE, N = Nmin) |
| 3.4 | Generate Figure 5 (boxplots by MAPE, N = Nepv10) |
| 3.4 | Generate Figure 5 (boxplots by MAPE, N = 10000) |
| 3.4 | Generate Figure 5 (boxplots by MAPE, N = 50000) |
| 3.4 | Generate Figure 5 (boxplots by MAPE, N = 100000) |
| 3.5 | Generate Table 2 (distribution of the model performance metrics for each N) |
| 3.6 | Generate data for Figure 6 (percentage over threshold) |
| 3.6 | Generate Figure 6 (percentage over threshold) |
| 3.7 | Generate Table 1 (baseline table) |