CODEBOOK: A National Estimate of the Health and Cost Burden of *Escherichia coli* Bacteraemia & Antibiotic Resistance in the Hospital Setting - 2019

Author: Nichola R. Naylor 2019

This code was utilised to estimate the health and cost burden of resistant and susceptible E. coli bacteraemia using national datasets. The corresponding manuscript can be found on a pre-print server. [1]

All analyses of the data to be completed in R Studio using code adapted from code written by Nathan Green[2], Esther Van Kleef[3], and the Centre for Health Economics at York University (this code was in STATA format originally[4]). This code was written and adapted by Nichola Naylor, with the help of Koen Pouwels.

The corresponding Data Dictionary will help with understanding what some of the following code aims to do. This is an update of the 2017 codebook, in line with updates in the coding used in the 2018 analyses.

Pre-processing & Checks

Generally the following in the rules are true for created objects within the R code:

"df." prefix → object is a data.frame

"dt." prefix → object is a data.table

"fun." prefix → object is a function

Note some R scripts will have to run in order (e.g. 'Pre-processing cases-Git.R' before 'Pre-processing Cases and Controls-Git').

R script:

"Preprocessing-Cases-Git.R" for cleaning of dates, missing information, reformatting variables, creation of spells, modified Elixhauser comorbidity coding and trust type.

Some key points/features with this code:

- Removed those with an "invalid" coded admission date & discharge date.
- Removed those with "invalid" episodes dates (needed for spell creation).
- Spells were created using the Centre of Health Economics (CHE) Method[4].
- Deduplicating on hesid, epistart, epiorder, epiend and transit (from CHE methodology).

---

[1] Naylor, NR. et al (2017) 'A National Estimate of the Health and Cost Burden of Escherichia coli Bacteraemia in the Hospital Setting: The Importance of Antibiotic Resistance' *Biorxiv*. DOI: https://doi.org/10.1101/153775

[2] Green, N. et al (2015) 'Quantifying the Burden of Hospital-Acquired Bloodstream Infection in Children in England by Estimating Excess Length of Hospital Stay and Mortality Using a Multistate Analysis of Linked, Routinely Collected Data' *J Pediatric Infect Dis Soc.* DOI: 10.1093/jpids/piu073

[3] van Kleef, E. et al (2014) 'Excess length of stay and mortality due to Clostridium difficile infection: a multi-state modelling approach.' *J Hosp Infect.* DOI: DOI: 10.1016/j.jhin.2014.08.008

[4] Centre for Health Economics, University of York, 2015, course details available at <www.york.ac.uk/che/courses/patient-data/> [Accessed 04 September 2017].

- Dropped the middle spells – kept patient admission information from the first episode (e.g. age, comorbidities).
- Dropped the middle spells – kept patient admission information from the first episode (e.g. age, comorbidities). The spell creation is done by the function sourced from "spellcreation-Git.R"
- Removed those where discharged method was unknown/missing (or miscoded as baby still born).
- Comorbidities were coded using the modified Elixhauser index, ICD10 codes for the index were taken from Quan et al (2005)[5] and weights for the comorbidities were taken from Bottle & Aylin (2011)[6] [the 2008-9 HES weights were used – see Table 2 of Bottle & Aylin (2011)]. The coding of comorbidity is done by the function sourced from "elixcreation-Git.R"
- Estates Data to provide trust type.[7] Only kept trusts which were labelled as "acute" in the estates dataset.
- Created variables which were time differences between admission, specimen and discharge; see data dictionary for definitions.
- Removed spells where admission was more than 14 days after specimen date or where discharge was more than 14 days before specimen. This was chosen, to keep in line with definition used previously with similar data.[8]
- Created onset variables, using the following definitions:
    - HO (hospital onset) if [specimen date – admission date] >= 2 days
    - COHA (community onset hospital associated) if [specimen date – admission date] < 2 days and a previous spell discharge was < or = 14 days before the specimen date
    - COCA (community onset community associated) if [specimen date – admission date] < 2 days and a previous spell discharge was > 14 days before specimen date [if there was a previous spell discharge].
    - The code used here assumes that there is one specimen per person (since there was only one specimen per person used from Dataset A). You may need to change this if using the code over a few years' worth of data. **This should be checked and the code adjusted accordingly.**

"Preprocessing-Controls-Git.R" for cleaning of dates, missing information, reformatting variables, creation of spells, modified Elixhauser comorbidity coding and trust type

Note that depending on the size of your data, you may be best place to run in chunks, save and reload etc rather than attempting to run full script (prone to crashing due to memory issues).

Some key points/features with this code:

[5] Quan, H. et al (2005) 'Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data' .*Med Care*. PubMed ID (PMID): 16224307

[6] Bottle, A. and Aylin, P. (2011) 'Comorbidity scores for administrative data benefited from adaptation to local coding and diagnostic practices.' *J Clin Epidemiol.* DOI: 10.1016/j.jclinepi.2011.04.004

[7] NHS Digital, 'Hospital Estates and Facilities Statistics' <http://hefs.hscic.gov.uk/DataFiles.asp> [Accessed 20 August 2017]

[8] Abernethy, J. K., et al. (2015) 'Thirty day all-cause mortality in patients with Escherichia coli bacteraemia in England' *Clin Microbiol Infect.* DOI: 10.1016/j.cmi.2015.01.001

- Removed those with an "invalid" coded admission date & discharge date.
- Removed those with "invalid" episodes dates (needed for spell creation).
- Spells were created using the Centre of Health Economics (CHE) Method[9].
- Deduplicating on hesid, epistart, epiorder, epiend and transit (from CHE methodology)
- Dropped the middle spells – kept patient admission information from the first episode (e.g. age, comorbidities). The spell creation is done by the function sourced from "spellcreation-Git.R"
- Removed those where discharged method was unknown/missing (or miscoded as baby still born).
- Comorbidities were coded using the modified Elixhauser index, ICD10 codes for the index were taken from Quan et al (2005)[10] and weights for the comorbidities were taken from Bottle & Aylin (2011)[11] [the 2008-9 HES weights were used – see Table 2 of Bottle & Aylin (2011)]. The coding of comorbidity is done by the function sourced from "elixcreation-Git.R"
- Linked data with Estates Data to provide trust type.[12]
- **Removed case patients** (note this code removed case patient, rather than just a specific case spell; again might be important if have multiple years and want to maximise sample).

R script:

"Preprocessing Cases-controls-Git.Rmd" for creating of time independent and time dependent datasets to use in cox proportional hazards and multistate models.

Assumptions/Definitions:

- Combining the case and control datasets.
- Creating dummy & factor variables to be used in analyses.
- Age and Elixhauser continuous variables were centred at the mean
- Created time to event variables, where there were same day events. Note that if a same day event occurred, this was coded as 0.5 days.
- To reduce the impact of outliers, and to enable comparison with the previous literature utilising this type of methodology,[13] censoring occurred at day 45. Essentially if you didn't get the infection by day 45 you were treated as someone who was a non-case patient until day 45 (i.e. all infection related variables were set to the same as controls up to day 45), and if you weren't dead or discharged by this time you were coded as not experiencing the event of interest.

---

[9] Centre for Health Economics, University of York, 2015, course details available at <www.york.ac.uk/che/courses/patient-data/> [Accessed 04 September 2017].

[10] Quan, H. et al (2005) 'Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data' .*Med Care*. PubMed ID (PMID): 16224307

[11] Bottle, A. and Aylin, P. (2011) 'Comorbidity scores for administrative data benefited from adaptation to local coding and diagnostic practices.' *J Clin Epidemiol.* DOI: 10.1016/j.jclinepi.2011.04.004

[12] NHS Digital, 'Hospital Estates and Facilities Statistics' < http://hefs.hscic.gov.uk/DataFiles.asp> [Accessed 20 August 2017]

[13] Stewardson, AJ. et al (2016) 'The health and economic burden of bloodstream infections caused by antimicrobial-susceptible and non-susceptible Enterobacteriaceae and Staphylococcus aureus in European hospitals, 2010 and 2011: a multicentre retrospective cohort study.'*Eurosurveillance* DOI: http://dx.doi.org/10.2807/1560-7917.ES.2016.21.33.30319

- After running “Checks-Git.R” in the data cleaning and analysis processes – it was necessary to create “flags” which are utilised in creating a step function to be used in the cox proportional hazards models & creating different groupings for the resistance variables (susceptible was grouped with non-tested in some instances).
    - The flags highlight events that happen before/after day 2– this was identified as the time by looking at the Shoenfeld residual plots for infection when building the cox models [see “Checks.R” description below]. If this analysis is redone, the time point might be different, or the step function might not be needed at all. **This should be checked and the code adjusted accordingly.**
- Created the time dependent dataset, whereby hospital onset cases had 2 rows (the first being admission to infection, the second infection to discharge) and appropriate tstart/tstop variables. Flag variables are also created to know which row “type” (e.g. hospital onset patient but non-infected row) you are dealing with.
    - Infection related variables were added in a “time dependent” format – this just meant resistance variables for our analysis, but this should be done for other variables if you are utilising them in time dependent models. Therefore, **this should be checked and the code adjusted accordingly.**

The outputs of this script include dt.TimeIndep and dt.TimeDep, which are data.tables used in analyses where the data are arranged into time independent (1 row per patient) and time dependent (2 rows for hospital onset patients) structures respectively.

R script:

“**Checks-Git.R”** for grouping susceptible and non-tested based on descriptive statistics, testing variable inclusion into the cox models and checking the cox-proportional hazards assumptions.

This script runs code used to check the code/data was ok in the data cleaning and analysis sections. Note that viewing the data (e.g. for specific patients) before and after spell creation and Elixhauser was done, and is recommended, but was not coded explicitly in the following file. This file also contains some model fitting code used to construct the cox proportional hazards models. **This code should be adapted to your needs, and may lead to changes in the other scripts being needed.** This is simply added to give an idea of the types of checks one should perform when doing a similar analysis. If there are non-proportional hazards present (i.e. the line on the plots is not horizontal), step functions can be used to correct for this. See “StepFunctions.R” description below. This was performed for the “deadordischarged” outcome cox proportional hazards models.

This uses a function ggsurv loaded from “**ggsurv-Git.R**” – this code was written by Barret Schloerke as part of the GGally package. [14]

Analysis

“**StepFunctions.R**” utilised step functions that were used to create time dependent coefficients.

[14] RDocumentation, “GGally v1.3.2”. <https://www.rdocumentation.org/packages/GGally/versions/1.3.2> [Accessed 20 August 2017]

- For the "deadordischarged" models the step function split by 0-2 and 2-45 days inclusive (have in mind that "0" days is the first admission day).
- This is read in the literature by Therneau et al's right closed intervals in a sense, since if the person is discharged/died on day 2 they are only ever considered to be in the first time group (tgroup).[15]

MODELS ESTIMATING MORTALITY IMPACT

If your mortality outcome of interest/available is in-hospital mortality, then the competing risk of being discharged needs to be taken into account, making standard Cox proportional hazards models potentially inappropriate. The two ways this was done in our analysis was by presenting the cumulative incidence of in-hospital death, utilising the cumulative incidence functions in the etm package[16] and by calculating the hazard ratios using a subdistribution hazards approach.

"CIF-Git.R" was use to obtain cumulative incidence on in-hospital mortality utilising etmCIF in built into the etm package17 .

- This R script provides the functions used and an example of how they were used.
- The outcome is competing-risk adjusted cumulative incidence, that can be easily converted into transition probabilities for health economic models.

"Subdistribution-Git.R" was used to estimate in-hospital mortality-related hazard ratios that adjusted for covariates (such as patient age and sex).

Note that because there was no censoring within the dataset apart from the artificial right censoring at day 45, this code worked. If you have censoring throughout the dataset, then other methods/code should be utilised, such as those within the kmi package[18].

MODELS ESTIMATING LENGTH OF STAY IMPACT

"Length-of-Stay-Git.R" estimates excess length of stay utilising a multistate model approach.

- For infection versus no infection, the 3 states are 0 = non-infection, 1 = infection, 2 =deadordischarged. For the resistance groups, the 3 states are 0 = susceptible, 1 = resistant, 2 = deadordischarged, where left-truncation of the data means people enter 0 and 1 when they get the infection. [19]
- Functions used can be sourced from the following scripts:
    - "clos-Git.R" uses the msm[20] and etm[16] packages to length of stay based on "3-state" models.
    - "NG-boot-Git.R" includes a bootstrapping function developed by Nathan Green.[21]

[15] Therneneau et al (2017)" Using Time Dependent Covariates and Time Dependent Coefficients in the Cox Model". <https://cran.r-project.org/web/packages/survival/vignettes/timedep.pdf> [Accessed 20 August 2017].
[16] Allignol, A. (2015) "Package 'etm'". <https://cran.r-project.org/web/packages/etm/etm.pdf> [Accessed 20 August 2017]
[17] Allignol, A. (2015) "Package 'etm'". <https://cran.r-project.org/web/packages/etm/etm.pdf> [Accessed 20 August 2017]
[18] Allignol, A. (2018) "Package 'kmi'" https://cran.r-project.org/web/packages/kmi/kmi.pdf
[19] Allignol, A. (2015) "Package 'etm'". <https://cran.r-project.org/web/packages/etm/etm.pdf> [Accessed 20 August 2017]
[20] Jackson, C. (2016) "Package 'msm'". <https://cran.r-project.org/web/packages/msm/msm.pdf>

- This is then used in functions from "boot-clos-Git.R", to estimate the standard errors in length of stay estimates.
- "los-equation-Git.R" uses the below logic to calculate the excess length of stay associated with resistant and susceptible infections compared to no-infection; soliving for $P_S * X + P_R * (X + Y) = Z$, where P_S and P_R represent the proportion of susceptible and resistant cases respectively, Y represents the estimate for excess length of stay comparing resistant and susceptible cases, X represents the excess length of stay comparing susceptible cases to non-infected controls and Z represents the excess length of stay comparing all *E. coli* bacteraemia cases to non-infected controls. For standard error calculation, the bootstrapped results from infection vs non-infection and from resistant vs susceptible were combined and the equation was solved for each row, to get 1,000 estimates of the difference.
- "los-CO-Git.R" performs the excess length of stay and bootstrapping functions on the community onset cases that are investigated in the subgroup analysis.

"Coxph.R" performed cox proportional hazards analysis to the data.

- Resistance variables were treated as factor variables, however, they were coded not as "as.factor(*resistance*)" but rather split into dummy variables for resistant and susceptible. This is coded as the following:
  - Our current model is ;
    coxph(Surv(tstart,tstop,deadordischarged)~ cluster(ID) + RESDR:strata(tgroup) + RESDS:strata(tgroup) + <*Additional Variables*> , data= data)

    Where:
    Infection = 1 & Resistance =1 --> RESDR =1
    Infection=1 & Resistance = 0 --> RESDR=0
    Infection=0 & resistance = 1/0 --> RESDR =0
    We do this ourselves (i.e. we use RESDR instead of 2 separate dummies for infection and resistance) as otherwise we get correlation issues then between the infection and resistance dummies. Theoretically you should also be able to use the approach of "as.factor(RES) where RES is a 0 for no-infection, 1 for susceptible infection and 2 for resistant infection, however with the introduction of the step-function it was creating some practical issues with just running as.factor() into the model.
    This assumes that there are no differences in confounding factors across having the infection and having resistance, which is a limitation of this method.
- 'strata(tgroup)' allows for the calculation of time dependent coefficients. The dt.TimeDep.S data.table which is used is the same as dt.TimeDep but with coding for the tgroups (as described within StepFunctions.R description)

Though the Cox proportional hazards and subdistribution hazards code allows us to estimate the impact of exposures of interest whilst adjusting for covariates, the hazard ratio outcomes are harder to utilise in subsequent health economic models.

---

[21] Green, N. et al (2015) 'Quantifying the Burden of Hospital-Acquired Bloodstream Infection in Children in England by Estimating Excess Length of Hospital Stay and Mortality Using a Multistate Analysis of Linked, Routinely Collected Data' *J Pediatric Infect Dis Soc.* DOI: 10.1093/jpids/piu073

“Subgroup.Git” performs subgroup analyses to estimate excess length of stay in days and cumulative incidence of in-hospital mortality.