Data Management Plan: Predicting Antimicrobial Resistance Rates using R

Data for Q1 2019 onwards

Daniel Weiand

Tuesday, 7 November, 2023

# Data management plan (DMP)

# Introduction

A Data Management Plan (DMP) is a document that often accompanies research and digital projects. The document is used to outline how a project will manage data during the project and once it is completed [[1](#ref-MU-Intro-to-DMP)].

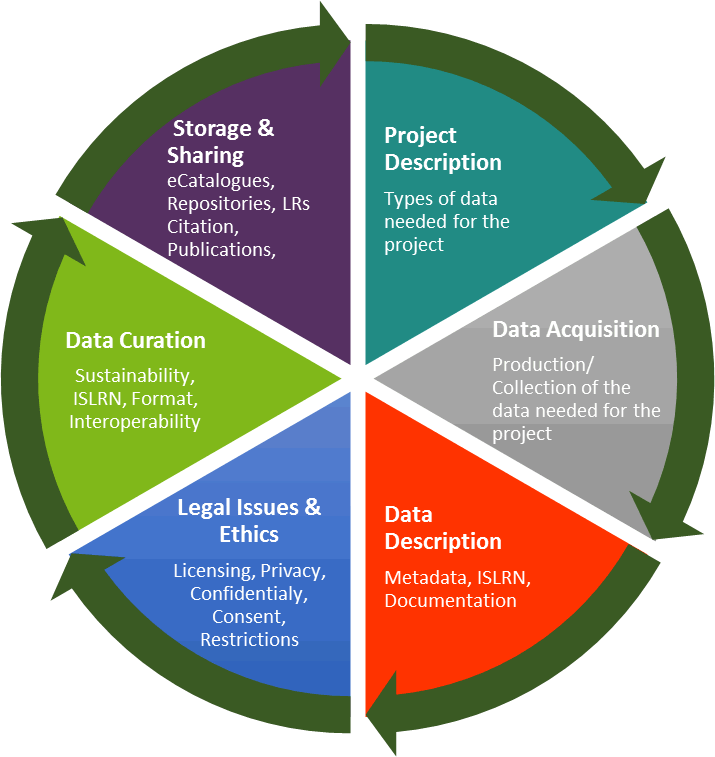
DMPs have become nearly a worldwide requirement for research funding. To meet these new funding agency expectations, information professionals across domains and the world have worked to create resources and services to successfully implement [[2](#ref-Bishop_2023)].

[[2](#ref-Bishop_2023)] posit that well-executed DMPs add to research transparency, accountability, and reproducibility, although more work needs to be done to validate these assumptions.

In the process of synthesising this DMP, all of the following resources have been been consulted.

## ELDA/ELRA

The [Evaluations and Language resources Distribution Agency (ELDA)](http://www.elra.info/en/services-around-lrs/dmp/) [[3](#ref-ELDA-DMP)] has formulated and published a more concise list of six distinct phases, encompassing the entire data life cycle, which need to be documented by DMPs. These are themselves based on 20 factors impacting sustainability, identified by [FLaReNet](http://www.elra.info/media/filer_public/2021/11/18/flarenet_final_book.pdf) [[4](#ref-FLaReNet_Book)]:



## ICPSR

In addition, the [Inter-university Consortium for Political and Social Research (ICPSR)](https://www.icpsr.umich.edu/web/pages/datamanagement/dmp/elements.html) [[5](#ref-ICPSR-DMP-Elements),[6](#ref-ICPSR-YouTube-Intro-DMP)] recommends that, where possible and appropriate, the following elements should be considered as part of a DMP:

* Data description
* Existing data
* Format
* Metadata
* Storage and backup
* Security
* Responsibility
* Intellectual property rights
* Access and sharing
* Audience
* Selection and retention periods
* Archiving and preservation
* Ethics and privacy
* Budget
* Data organization
* Quality Assurance
* Legal requirements

These elements have been mapped, by the [ICPSR](https://www.icpsr.umich.edu/web/pages/datamanagement/dmp/elements.html), to the U.S. [National Science Foundation’s (NSF)](https://new.nsf.gov/funding/data-management-plan) [[7](#ref-NSF-DMP)] policy on managing, disseminating and sharing research results.

## Online tools

Finally, online tools are available to assist with the creation of DMPs [[8](#ref-dmponline),[9](#ref-NCL-DMP)].

# Project Background

## Name

Data Management Plan: Using R to Predict Antimicrobial Resistance Rates using Data for all Isolates from Blood Cultures collected from Q1 2019 onwards

## Project description

### Aims and objectives

Antimicrobial resistance threatens the very core of modern medicine and the sustainability of an effective, global public health response to the enduring threat from infectious diseases [[10](#ref-WHO_AMR)].

In 2015, the World Health Assembly adopted a global action plan on antimicrobial resistance. Key objectives include: - strengthening our existing knowledge through improved surveillance; and - reducing the incidence of infection through effective infection prevention measures.

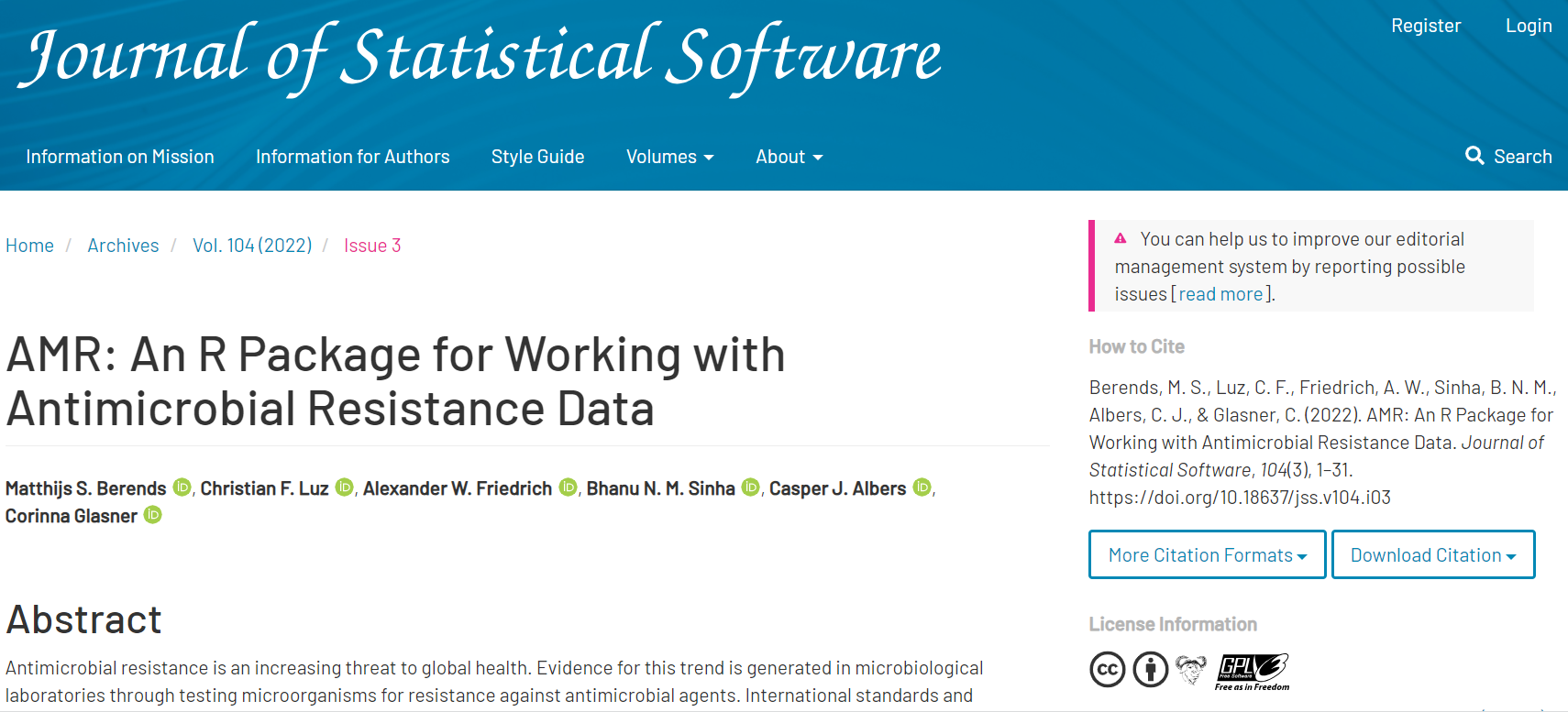
The aim of this project is to model predicted antimicrobial resistance (AMR) rates at The Newcastle upon Tyne Hospitals NHS Foundation Trust (NUTH). The integrated laboratory medicine directorate at NUTH is fully HTA, MHRA & UKAS accredited, employs hundreds of scientific and technical staff, and processes more than 11 million samples per annum.

AMR rates will be modeled using the [AMR](https://msberends.github.io/AMR/articles/AMR.html) package [[11](#ref-AMR2022),[12](#ref-R-AMR)] for [R](https://nhsrcommunity.com/installing-r-and-r-studio/) [[13](#ref-R-base)], including data on all organisms isolated from blood cultures collected at NUTH from Q1 2019 onwards.

Laboratory data will be triangulated with data from other sources (e.g. the Trust’s dialysis information management system, electronic health record, and bereavement ledger) to gain additional insights in relation to disease trends, potential risk factors and outcomes.

The synthesised models will be used to gain valuable insights, which will help to: - Inform clinical decision-making - Assist with updating of antimicrobial guidelines - Target (limited) infection prevention and control resources - Justify additional investment in antimicrobial stewardship

### The AMR package for R



The [AMR](https://msberends.github.io/AMR/articles/AMR.html) package [[11](#ref-AMR2022),[12](#ref-R-AMR)] is a free, open-source and independent package for [R](https://nhsrcommunity.com/installing-r-and-r-studio/) [[13](#ref-R-base)] that provides a standard for clean and reproducible analysis and prediction of Antimicrobial Resistance (AMR).

This package will be used to determine ‘first isolates’, as per Hindler et al [[14](#ref-hindler2007analysis)], for use in the final analysis; calculate and visualise AMR data; and predict future AMR rates using regression models.

### The concept of ‘first isolates’

To conduct an analysis of antimicrobial resistance, Hindler et al [[14](#ref-hindler2007analysis)] posit that one should only include the first isolate of every patient per episode. This approach avoids overestimating or underestimating AMR rates [[11](#ref-AMR2022),[12](#ref-R-AMR)]. The [AMR](https://msberends.github.io/AMR/articles/AMR.html) package was used to determine ‘first isolates’ for use in the final analysis.

### Predicting future AMR rates

The [AMR](https://msberends.github.io/AMR/articles/AMR.html) package [[11](#ref-AMR2022),[12](#ref-R-AMR)] includes functions which, based on a date column, calculates cases per year and uses a regression model to predict antimicrobial resistance.

The resistance\_predict() function creates a prediction model including standard errors (SE), which are returned as columns se\_min and se\_max.

Valid options for the statistical model (argument model) are: “binomial”, “poisson” and “linear”.

### The NHSRplotthedots package for R

The {[NHSRplotthedots](https://github.com/nhs-r-community/NHSRplotthedots/)} package [[15](#ref-R-NHSRplotthedots)] for [R](https://nhsrcommunity.com/installing-r-and-r-studio/) [[13](#ref-R-base)] is used to plot time series data and detect special cause variation by applying statistical process control (SPC) rules.

### Survival analysis

The {ggsurvfit} and {survival} packages [[16](#ref-R-ggsurvfit)–[18](#ref-survival-book)] for [R](https://nhsrcommunity.com/installing-r-and-r-studio/) [[13](#ref-R-base)] are used to conduct survival analyses.

## Domain

Clinical medicine

## Project start

Data collection for the purposes of this project started at the beginning of Q1 2019.

## Project end

Data analysis is undertaken quarterly.

# Data acquisition, storage and quality

## General considerations

### Format

The data required for this project are generated and maintained by different teams, in different formats. For each data set, the way in which it is generated, maintained, and made available is specified, below.

### Missing data

Missing data are typically grouped into three categories [[19](#ref-Mack2018_Missing_Data_Types)]: Missing Completely at Random (MCAR); Missing at Random (MAR); and Missing Not at Random (MNAR). For each data set, the expected type of ‘missingness’, if any, is specified, below.

The import of the MAR vs. MNAR distinction is to indicate whether bias can be fully removed in analysis [[19](#ref-Mack2018_Missing_Data_Types)].

In general terms, it’s known that sicker patients tend to have more complete records and healthier patients tend to have records that are less complete. Blind sampling of complete records within an EHR database may skew the sampled population towards sicker patients [[20](#ref-Weiskopf)].

### Metadata

Metadata is data about data. It means it is a description and context of the data. It helps to organize, find and understand data [[21](#ref-What-is-Metadata)]. Some typical metadata elements include: - File name and description; - Tags and categories; - Who created and when; - Who last modified and when; - Who can access or update.

Relational databases (the most common type of database) store and provide access to not only data but also metadata in a structure called data dictionary or system catalog, which hold information about: tables, columns, data types, constraints, table relationships, and so on [[21](#ref-What-is-Metadata)].

The NHS Data Model and Dictionary for England [[22](#ref-NHS-2023-10-12)] provides a reference point for approved Information Standards [Notices] to support health care activities

## Existing data

### LIMS blood culture data

#### Information

The LIMS will be interrogated using IBM COGNOS [[23](#ref-IBM-COGNOS)] by a specialist biomedical scientist to collect data on a quarterly basis on all culture-positive blood cultures collected from Q1 2019.

The underlying data is created through collaboration between large teams of laboratory staff, including administrative staff, medical laboratory assistants (MLAs), biomedical scientists, and medical microbiologists, according to the standards outlined in ISO 15189 [[24](#ref-UKAS_ISO)].

Accreditation against this standard underpins confidence in the quality of medical laboratories through a process that verifies their integrity, impartiality and competence. Assessments under UKAS accreditation ensure labs meet the relevant requirements including the operation of a quality management system and the ability to demonstrate that specific activities are performed within the criteria set out in the relevant standard [[24](#ref-UKAS_ISO)].

Data will also be collected on a quarterly basis on all culture-negative blood cultures collected from Q1 2019.

#### Missing data

Data is expected to be complete, as missing (critical) data would be a breach of the relevant standard [[24](#ref-UKAS_ISO)].

Of course, not all patients in hospital have blood cultures taken, nor do all patients with sepsis survive until hospital admission to have blood cultures collected in the first place. But when blood cultures are collected, the results of the investigation should be recorded according to the standards outlined in the relevant standard [[24](#ref-UKAS_ISO)].

Arguably, some participants may have missing laboratory values because a batch of lab samples was processed improperly. Whilst this is technically an example of data that is Missing Completely at Random (MCAR) [[19](#ref-Mack2018_Missing_Data_Types)], this is a very rare event, which would result in declaration of a DATIX [[25](#ref-DATIX)] and further investigation.

Some data may be “missing” because, whilst basic tests were undertaken, more advanced tests were considered unnecessary. For example, some blood culture isolates do not have full antimicrobial susceptibility testing (AST) undertaken because the isolates in question are thought to be possible contaminants. Arguably, these data could be said to be Missing Not at Random (MNAR) because it is more likely that clinically significant blood culture isolates are fully investigated than probable contaminants.

Data may also be Missing Not at Random (MNAR) for blood culture isolates that are fastidious in nature, or relatively more difficult to isolate in vitro, like pneumococci, which are known to autolyse and may be non-viable for AST, leading to the absence of AST.

#### Metadata, storage and backup

Interrogation of the LIMS results in the creation of a single .xls files; one for each quarter. These files are stored in unmodified format, locally, on an encrypted computer that belongs to the Trust, and saved to here::here(“02\_data”, “Original data”).

To facilitate data analysis in R [[13](#ref-R-base)], each file is modified to .xlsx format and saved to here::here(“02\_data”, “Modified data”).

The raw data are imported, in unmodified format, using R [[13](#ref-R-base)]. All data wrangling, analysis and write-up is undertaken in R [[13](#ref-R-base)].

As an example of metadata included with data files extracted from the LIMS, here is a table produced using the dir\_info() function of the {fs} R package, which outlines some of the key features of the raw LIMS data:

| path | type | size | permissions | modification\_time | user | group | device\_id | hard\_links | special\_device\_id | inode | block\_size | blocks | flags | generation | access\_time | change\_time | birth\_time |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/2019-20 New POS BC WITH SENS Q3.xlsx | file | 522.45K | rw- | 2022-08-18 15:40:42 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 1048 | 0 | 0 | 2023-11-06 10:30:48 | 2023-02-02 09:02:18 | 2023-10-26 10:31:26 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/2019-20 New POS BC WITH SENS Q4.xlsx | file | 478.98K | rw- | 2022-08-18 15:40:58 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 960 | 0 | 0 | 2023-11-06 10:30:48 | 2023-02-02 09:02:18 | 2023-10-26 10:31:26 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/New POS BC WITH SENS 19.20 (2).xlsx | file | 2.68M | rw- | 2022-08-10 11:10:04 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 5488 | 0 | 0 | 2023-11-06 12:16:50 | 2023-02-02 09:02:18 | 2023-10-26 10:31:26 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/POS BC 19.20 | directory | 0 | rw- | 2023-10-26 10:31:26 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 8 | 0 | 0 | 2023-11-07 17:00:47 | 2023-10-26 10:31:26 | 2023-10-26 10:31:26 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/POS BC 20.21 | directory | 0 | rw- | 2023-10-26 10:31:27 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 8 | 0 | 0 | 2023-11-07 17:00:47 | 2023-10-26 10:31:27 | 2023-10-26 10:31:26 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/POS BC 20.21.xlsx | file | 3.91M | rw- | 2022-08-10 11:11:22 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 8008 | 0 | 0 | 2023-11-06 12:16:50 | 2023-02-02 09:02:18 | 2023-10-26 10:31:26 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/POS BC 21.22 | directory | 0 | rw- | 2023-10-26 10:31:27 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 8 | 0 | 0 | 2023-11-07 17:00:47 | 2023-10-26 10:31:27 | 2023-10-26 10:31:27 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/POS BC 22.23 | directory | 0 | rw- | 2023-10-26 10:31:27 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 0 | 0 | 0 | 2023-11-07 17:00:47 | 2023-10-26 10:31:27 | 2023-10-26 10:31:27 |
| C:/Users/Weiandd2/Documents/BC AMR/02\_data/Modified data/BC data with sens/POS BC WITH SENS 21.22.xlsx | file | 4.07M | rw- | 2022-08-10 11:10:50 |  |  | 11279836 | 1 | 0 | 1.7e+15 | 4096 | 8344 | 0 | 0 | 2023-11-06 12:16:50 | 2023-02-02 09:02:18 | 2023-10-26 10:31:26 |

With regards to data retention periods, diverse regulatory requirements govern how long stored data should be stored/archived for in the LIMS. There is no plan to store raw data for long time periods outside the LIMS, as all required data are relatively easily extracted from the LIMS. More important than the raw data used in the course of this project are the insights gained, which will be shared in the form of parameterised reports using Quarto.

In its raw form, since Q1 2019, the blood culture data set comes to **160331 rows.**

The {skimr} package for R has been used to provide select summary information about the data, such as its shape, features, and number of records, including number of missing values.

# A tibble: 39 × 21  
 skim\_type skim\_variable n\_missing complete\_rate Date.min Date.max   
 <chr> <chr> <int> <dbl> <date> <date>   
 1 Date receive\_date 0 1 2019-04-01 2023-09-30  
 2 Date date\_of\_birth 0 1 1915-09-14 2023-09-13  
 3 character specimen\_no 0 1 NA NA   
 4 character order\_code 0 1 NA NA   
 5 character specimen\_type\_code 0 1 NA NA   
 6 character organism\_code 0 1 NA NA   
 7 character location\_code 0 1 NA NA   
 8 character hospital\_code 0 1 NA NA   
 9 character antibiotic\_code 7260 0.955 NA NA   
10 character patient\_full\_name 0 1 NA NA   
# ℹ 29 more rows  
# ℹ 15 more variables: Date.median <date>, Date.n\_unique <int>,  
# character.min <int>, character.max <int>, character.empty <int>,  
# character.n\_unique <int>, character.whitespace <int>, numeric.mean <dbl>,  
# numeric.sd <dbl>, numeric.p0 <dbl>, numeric.p25 <dbl>, numeric.p50 <dbl>,  
# numeric.p75 <dbl>, numeric.p100 <dbl>, numeric.hist <chr>

After some data wrangling, including ‘widening’ the data set, using the pivot\_wider() function of the {dplyr} package, so that AST results are not reported on separate lines, the blood culture data set comes to **10837 rows.**

### EHR diabetes data

#### Information

The electronic health record (EHR) will be interrogated by a data scientist working for the central IT team, to collect data on all patients with diabetes from Q1 2019.

Data include patient-level ICD-10 and SNOMED-CT codes implying a diagnosis of diabetes.

The Trust uses the Cerner Millennium EHR [[26](#ref-Cerner_Millennium)] and undertakes data extraction using the Nautilus data warehouse, which consolidates data from multiple sources into one or more query databases [[27](#ref-Nautilus)].

#### Missing data

ICD-10 and SNOMED-CT data relating to the diabetic status of individual patients is likely to be incomplete.

Data are likely to be Missing at random (MAR) because of incomplete coding of patients’ diagnoses at both the point the diagnosis being made and at the point of hospital discharge [[19](#ref-Mack2018_Missing_Data_Types)].

However, data may also be considered to be Missing not at random (MNAR) in the sense that I would expect more severe diabetic complications (e.g. new diagnosis of diabetic ketoacidosis, DKA, leading to emergency hospital admission) to be coded more completely than less severe forms of diabetes (e.g. diet-controlled diabetes, diagnosed incidentally).

#### Metadata, storage and backup

Interrogation of the EHR results in the creation of a single .xlsx file. This file is stored in unmodified format, locally, on an encrypted computer that belongs to the Trust, and saved to here::here(“02\_data”).

The raw data are imported, in unmodified format, using R [[13](#ref-R-base)]. All data wrangling, analysis and write-up is undertaken in R [[13](#ref-R-base)].

With regards to data retention periods, diverse regulatory requirements govern how long stored data should be stored/archived for in the EHR There is no plan to store raw data for long time periods outside the EHR, as all required data are relatively easily extracted from the EHR. More important than the raw data used in the course of this project are the insights gained, which will be shared in the form of parameterised reports using Quarto.

In its raw form, the diabetic data set comes to **70533 rows.**

The {skimr} package for R has been used to provide select summary information about the data, such as its shape, features, and number of records, including number of missing values.

# A tibble: 4 × 9  
 skim\_type skim\_variable n\_missing complete\_rate character.min character.max  
 <chr> <chr> <int> <dbl> <int> <int>  
1 POSIXct admission\_dt 0 1 NA NA  
2 POSIXct disch\_dt 0 1 NA NA  
3 character hospital\_number 0 1 8 8  
4 character diagnosis\_codes 0 1 5 506  
# ℹ 3 more variables: character.empty <int>, character.n\_unique <int>,  
# character.whitespace <int>

### Dialysis data

#### Information

The dialysis information management system will be interrogated by an administrative staff member of the dialysis team, Clinical Computing’s “clinicalvision” EHR [[28](#ref-Clinical-Vision)], to collect data on all patients undergoing dialysis from Q1 2019.

clinicalvision is used by health care organisations in the UK, Canada, Australia, New Zealand, and the USA to improve the flow of information between care professionals who are managing patients with kidney disease. The clinicalvision application can provide a full view of a patient’s progress through all stages of chronic kidney disease (CKD), all ESRD treatment modalities, including kidney transplantation [[28](#ref-Clinical-Vision)].

clinicalvision’s reporting solutions and interface capabilities allow users to harness the data collected to quickly focus on key kidney care information, improve team collaboration and fulfil regulatory reporting requirements [[28](#ref-Clinical-Vision)].

#### Missing data

Data is expected to be complete, as missing (critical) data would likely be a breach of regulatory reporting requirements [[28](#ref-Clinical-Vision),[29](#ref-MHRA_reg_req)].

Arguably, some participants may have missing values because the dialysis procedure was undertaken incorrectly. Whilst this is technically an example of data that is Missing completely at random (MCAR) [[19](#ref-Mack2018_Missing_Data_Types)], this is a very rare event, which would result in declaration of a DATIX [[25](#ref-DATIX)] and further investigation. Indeed, the MHRA publishes strict standards for the traceability of medical devices [[29](#ref-MHRA_reg_req)].

#### Metadata, storage and backup

Interrogation of clinicalvision results in the creation of many .xls files; one for each calendar year. These files are stored in unmodified format on the renal shared drive, with restricted access.

The raw data are imported, in unmodified format, using R [[13](#ref-R-base)]. All data wrangling, analysis and write-up is undertaken in R [[13](#ref-R-base)].

With regards to data retention periods, diverse regulatory requirements govern how long stored data should be stored/archived for in clinicalvision There is no plan to store raw data for long time periods outside clinicalvision, as all required data are relatively easily extracted from clinicalvision. More important than the raw data used in the course of this project are the insights gained, which will be shared in the form of parameterised reports using Quarto.

In its raw form, the dialysis data set comes to **298471 rows.**

The {skimr} package for R has been used to provide select summary information about the data, such as its shape, features, and number of records, including number of missing values.

# A tibble: 12 × 21  
 skim\_type skim\_variable n\_missing complete\_rate Date.min Date.max   
 <chr> <chr> <int> <dbl> <date> <date>   
 1 Date date\_of\_birth 0 1 1927-05-30 2005-08-02  
 2 Date session\_date 0 1 2018-04-01 2023-09-30  
 3 POSIXct month\_session\_date 0 1 NA NA   
 4 character ccg 6470 0.978 NA NA   
 5 character hospital\_number 4 1.00 NA NA   
 6 character postcode 0 1 NA NA   
 7 character gender 6589 0.978 NA NA   
 8 character ethnicity 6840 0.977 NA NA   
 9 character gp\_code 58 1.00 NA NA   
10 character access 0 1 NA NA   
11 character location 1 1.00 NA NA   
12 numeric nhs\_number 2 1.00 NA NA   
# ℹ 15 more variables: Date.median <date>, Date.n\_unique <int>,  
# character.min <int>, character.max <int>, character.empty <int>,  
# character.n\_unique <int>, character.whitespace <int>, numeric.mean <dbl>,  
# numeric.sd <dbl>, numeric.p0 <dbl>, numeric.p25 <dbl>, numeric.p50 <dbl>,  
# numeric.p75 <dbl>, numeric.p100 <dbl>, numeric.hist <chr>

### Bereavement ledger data

#### Information

Data from the Trust’s bereavement ledger, created and maintained by the bereavement office staff, will be interrogated to collect data on all patients who died in the Trust from Q1 2019.

#### Missing data

The Newcastle upon Tyne Hospitals NHS Foundation Trust’s bereavement ledger only includes information on patients who died in hospital, before hospital discharge and/or transfer to another healthcare setting. In effect, information is not available on the status of patients in the community and/or other healthcare settings. As a result, the mortality data outlined in the survival analysis are likely to underestimate patients’ true mortality rates.

As such, data is expected to be Missing Not At Random (MNAR), in the sense that I would expect that patients with a Newcastle postcode as more likely to be admitted to- and eventually die in a hospital operated by The Newcastle upon Tyne Hospitals NHS Foundation Trust.

#### Metadata, storage and backup

Bereavement data is stored in unmodified format on the a separate shared drive with restricted access.

The bereavement ledger exists in the form of several .xls files; one for each calendar year.

The raw data are imported, in unmodified format, using R [[13](#ref-R-base)]. All data wrangling, analysis and write-up is undertaken in R [[13](#ref-R-base)].

With regards to data retention periods, diverse regulatory requirements govern how long stored data should be stored/archived for in the bereavement ledger There is no plan to store raw data for long time periods outside the bereavement ledger, as all required data are relatively easily extracted from the bereavement ledger. More important than the raw data used in the course of this project are the insights gained, which will be shared in the form of parameterised reports using Quarto.

In its raw form, the bereavement ledger data set comes to **5329 rows.**

The {skimr} package for R has been used to provide select summary information about the data, such as its shape, features, and number of records, including number of missing values.

# A tibble: 5 × 13  
 skim\_type skim\_variable n\_missing complete\_rate Date.min Date.max   
 <chr> <chr> <int> <dbl> <date> <date>   
1 Date date\_of\_death 0 1 2020-01-01 2022-11-08  
2 POSIXct dttm\_date\_of\_death 0 1 NA NA   
3 character mrn 0 1 NA NA   
4 character cause\_of\_death 31 0.994 NA NA   
5 character hospital\_number 0 1 NA NA   
# ℹ 7 more variables: Date.median <date>, Date.n\_unique <int>,  
# character.min <int>, character.max <int>, character.empty <int>,  
# character.n\_unique <int>, character.whitespace <int>

## Novel data

No additional data collection is required.

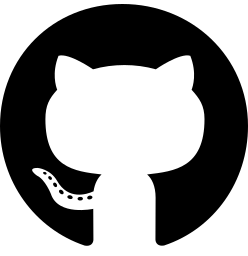
# Data transformation, modelling and processing

## Version control

For version control, Git Bash were used to iteratively upload all code to a private repository on GitHub [[30](#ref-ourcodingclub_github),[31](#ref-posit_git)].

## Quarto code

This whole document has been created using Quarto (markdown) [[32](#ref-Quarto)] for R [[13](#ref-R-base)].

The final code for this document has been uploaded (in .qmd file format) to my public  [GitHub](https://github.com/send2dan/public) repository, which can be accessed by following the link or scanning the following QR code:

|  |
| --- |
|  |

## ERD

### Background information regarding ERDs

An Entity relationship diagram (ERD) illustrates how “entities” relate to each other within a system [[33](#ref-lucid_ERD)]. ERDs are most often used to design tables in relational databases.

ERDs are comprised of 3 main categories: entities, attributes, and relationships. Entities are objects or concepts that are associated with important data—like a customer, order, or product. Attributes represent properties or traits of an entity, such as a customer’s phone number or home address. Relationships demonstrate the link between entities [[34](#ref-youtube_erd),[35](#ref-lucidchart_erd_symb)].

**Entities** are objects or concepts that represent important data. Entities are typically nouns such as product, customer, location, or promotion. There are three types of entities commonly used in entity relationship diagrams [[35](#ref-lucidchart_erd_symb)]:

* Strong entity: These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity.
* Weak entity: Weak entities depend on some other entity type. They don’t have primary keys, and have no meaning in the diagram without their parent entity.
* Associative entity: Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances.

Within entity-relationship diagrams, **relationships** are used to document the interaction between two entities. Relationships are usually verbs such as assign, associate, or track and provide useful information that could not be discerned with just the entity types [[35](#ref-lucidchart_erd_symb)].

ERD **attributes** are characteristics of the entity that help users to better understand the database. Attributes describe details of the various entities that are highlighted in a conceptual ER diagram [[35](#ref-lucidchart_erd_symb)]. Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship. Multivalued attributes are those that are can take on more than one value. Derived attributes are attributes whose value can be calculated from related attribute values. **Fields** represent the portion of a table that establish the attributes of the entity. **Types** refer to the type of data in the corresponding field in a table.

**Keys** are one way to categorise attributes [[35](#ref-lucidchart_erd_symb)]:

* *Primary keys* are an attribute or combination of attributes that uniquely identifies one and only one instance of an entity.
* *Foreign keys* are created any time an attribute relates to another entity in a one-to-one or one-to-many relationship.

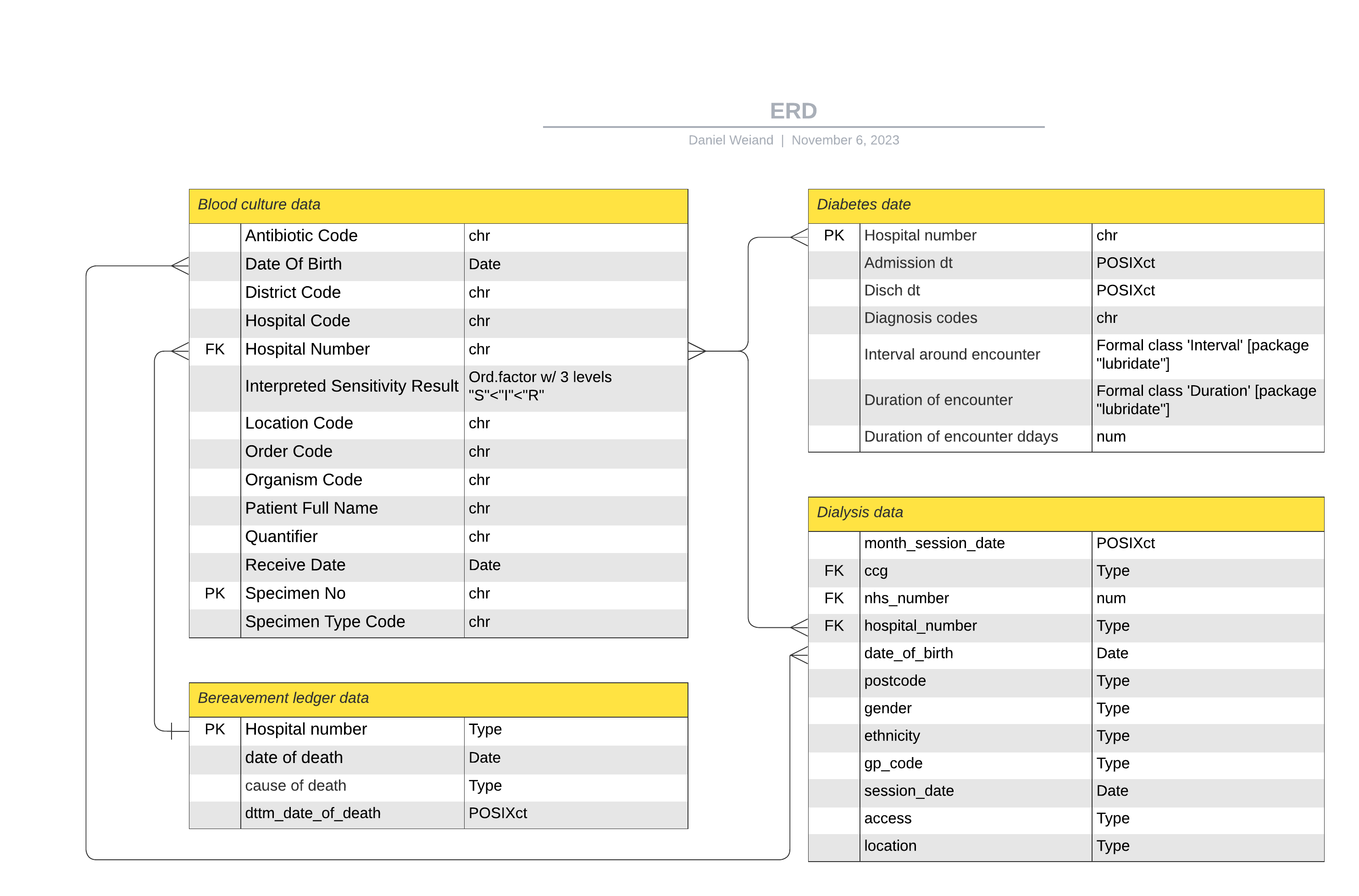
**Cardinality and ordinality** are shown by the styling of a line and its endpoint, according to the chosen notation style [[35](#ref-lucidchart_erd_symb)].

* *Cardinality* refers to the maximum number of times an instance in one entity can relate to instances of another entity.
* *Ordinality*, on the other hand, is the minimum number of times an instance in one entity can be associated with an instance in the related entity.



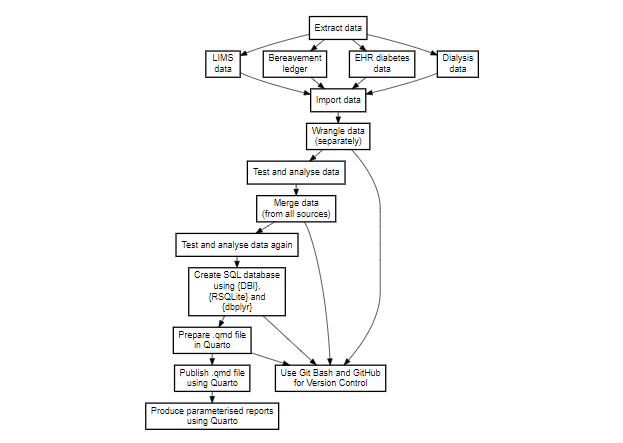
### ERD for this project

The following figure, synthesised using the Lucid app [[36](#ref-lucidapp)], gives the field names (attributes) and types for each entity.



## Data flow diagram

The following diagram provides a high-level summary of the steps involved in producing parameterised reports summarising trends in predicted AMR rates, stratified by variables including patient age and location.

 Running the final analysis, including all data import and wrangling steps, is a relatively processor-intensive procedure. Therefore, the {DBI}, {RSQLite}, and {dbplyr} are used to create a SQL database [[39](#ref-R-dbplyr)]. This database is ‘read’ when publishing .qmd files in order to speed up the whole process, including creation of parameterised reports.

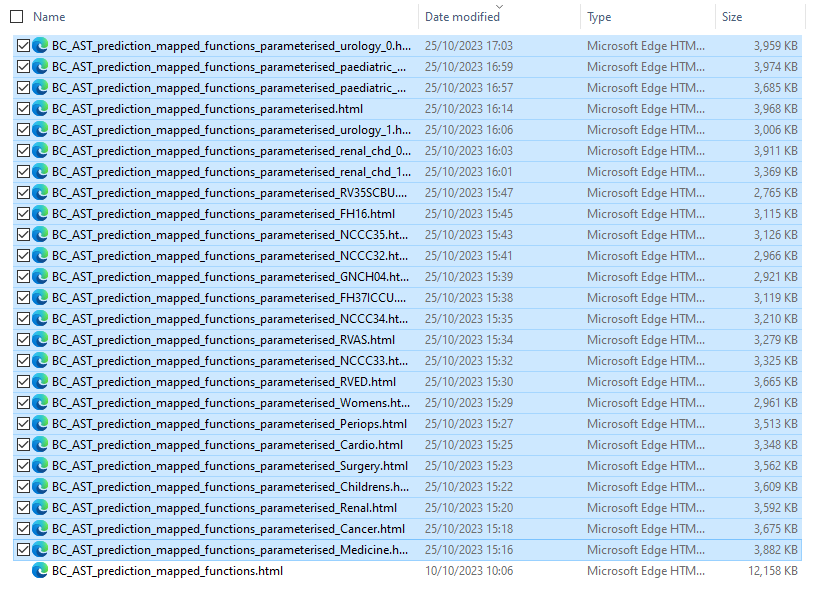
### SQL database creation, storage and justification

The SQL database, created using {DBI}, {RSQLite}, and {dbplyr} [[39](#ref-R-dbplyr)] is stored on an encrypted drive on a laptop that belongs to the Trust.

The SQL database is deleted after completion of all analyses, including publication of all parameterised reports summarising trends in predicted AMR rates, stratified by variables including patient age and location.

The size of SQL database files is significantly smaller than the sum of the size of all the other files (.xls, .xlsx and .csv files)

It is much, much quicker to import and wrangle all data once, and save it to a SQL database for repeated analysis, than it is to import and wrangle all data every time that a report is run. As can be seen from the screenshot below, it is possible to run 25 large, parameterised reports, each over 2-3mb in size, in less than 2 hours using this process.



## Examples of data wrangling (transformation) steps

### Naming conventions using the {janitor} package

The clean\_names() function of {janitor} package for R has been used to routinely transform the column names of imported data sets to snake\_case [[40](#ref-janitor_snake_case)]. Effectively, this function is called every time that data is read. I usually add “\_clean” to the end of the object name in order to make it clear that the data has been cleaned using {janitor}.

#import data and clean names  
  
data\_diabetic <- read\_excel(here("02\_data", "INC2577625 - Diabetes patients.xlsx"),   
 sheet = "Diabetes mellitus", col\_types = c("text",   
 "date", "date", "text")) |>   
 janitor::clean\_names() |>   
 rename(hospital\_number = mrn)

### Adding age\_group column using dplyr::case\_when()

#add age group  
  
data\_clean <- data\_clean %>%   
 mutate(age\_group = case\_when(age >= 100 & age <= 150 ~ "more than 100",  
 age >= 90 & age <= 99 ~ "90-100",  
 age >= 80 & age <= 89 ~ "80-90",  
 age >= 70 & age <= 79 ~ "70-80",  
 age >= 60 & age <= 69 ~ "60-70",  
 age >= 50 & age <= 59 ~ "50-60",  
 age >= 40 & age <= 49 ~ "40-50",  
 age >= 30 & age <= 39 ~ "30-40",  
 age >= 20 & age <= 29 ~ "20-30",  
 age >= 10 & age <= 19 ~ "10-20",  
 age >= 0 & age <= 9 ~ "0-10"))

### Calculate durations using {lubridate}

# calculate duration of encounters in ddays  
  
data\_df <- data\_df |>   
 mutate(interval\_around\_encounter = lubridate::interval(start = admission\_dt, end = disch\_dt),  
 duration\_of\_encounter = as.duration(interval\_around\_encounter),  
 duration\_of\_encounter\_ddays = duration\_of\_encounter / ddays(1))

### Use dplyr::pivot\_wider()

A huge amount of effort is spent cleaning data to get it ready for analysis [[41](#ref-tidy-data)]. Tidy datasets are easy to manipulate, model and visualize, and have a specific structure. The framework outlined by Hadley Wickham makes it easy to tidy messy datasets.

Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types. In tidy data: - Every column is a variable. - Every row is an observation. - Every cell is a single value.

In the course of this project, a diverse tools have been used to ‘tidy’ imported data sets, ready for further analysis.

One example of this is the use the dpylr::pivot\_wider() function to take the long “as\_ab” column and make it wider, in order to ensure that each antibiotic column can be formated into a S/I/R format.

# pivot wider to get a data set where the antibiotic susceptibility test results are wide (i.e. tidy)  
  
data\_clean\_sens\_wide <- data\_clean\_sens\_distinct |>   
 pivot\_wider(  
 id\_cols = NULL,  
 id\_expand = FALSE,  
 names\_from = as\_ab, #antibiotic codes to be pivot\_wider  
 names\_prefix = "",  
 names\_sep = "\_",  
 names\_glue = NULL,  
 names\_sort = FALSE,  
 names\_vary = "fastest",  
 names\_expand = FALSE,  
 names\_repair = "check\_unique",  
 values\_from = interpreted\_sensitivity\_result, #AST result to be pivot\_wider  
 values\_fill = NULL,  
 values\_fn = NULL,  
 unused\_fn = NULL  
)

### Using AMR::first\_isolate() to intelligently de-duplicate the blood culture data set

To conduct an analysis of antimicrobial resistance, it is recommended that one only includes the first isolate of every patient per episode [[14](#ref-hindler2007analysis)]. If this is not done, one could easily misjudge the AMR rate. The {AMR} package accounts for this methodology with the first\_isolate() function.

# find first isolates  
  
data\_clean\_sens\_wide\_2 <- data\_clean\_sens\_wide |>   
 mutate(first = first\_isolate(col\_mo = "as\_mo",  
 col\_date = "receive\_date",  
 col\_patient\_id = "hospital\_number",  
 info = TRUE))   
  
data\_1st <- data\_clean\_sens\_wide\_2 |>   
 filter(first == TRUE)

### Creating and reading a SQL database

As outlined above, the {DBI}, {RSQLite}, and {dbplyr} are used to create a SQL database [[39](#ref-R-dbplyr)]. This database is ‘read’ when publishing .qmd files in order to speed up the whole process, including creation of parameterised reports.

The following code shows how I create a SQL database

# connect data\_clean.db database to con -------------------------------------------------  
  
con <- DBI::dbConnect(drv = RSQLite::SQLite(), #use SQLite function to progress  
 dbname = "data\_clean.db",  
 #loadable.extensions = TRUE,  
 #default.extensions = loadable.extensions,  
 cache\_size = NULL,  
 synchronous = "off",  
 flags = SQLITE\_RWC,  
 vfs = NULL,  
 bigint = "integer64", #c("integer64", "integer", "numeric", "character")  
 extended\_types = TRUE) #!!!When TRUE columns of type DATE, DATETIME / TIMESTAMP, and TIME are mapped to corresponding R-classes, c.f. below for details. Defaults to FALSE. Use TRUE to keep correct date/time formats!!!  
  
# create data\_clean.db database using dbWriteTable ------------------------------------  
  
DBI::dbWriteTable( #Writes, overwrites or appends a data frame to a database table, optionally converting row names to a column and specifying SQL data types for fields.  
 con, #A DBIConnection object, as returned by dbConnect().  
 "data\_clean", #The table name, passed on to dbQuoteIdentifier().   
 data\_clean,   
 # row.names, (default: FALSE) #The interpretation of rownames depends on the row.names argument, see sqlRownamesToColumn() for details  
 # append (default: FALSE) #If the append argument is TRUE, the rows in an existing table are preserved, and the new data are appended. If the table doesn't exist yet, it is created.  
 field.types, #(default: NULL) #The field.types argument must be a named character vector with at most one entry for each column. It indicates the SQL data type to be used for a new column. If a column is missed from field.types, the type is inferred from the input data with dbDataType().  
 # temporary, (default: FALSE) #If the temporary argument is TRUE, the table is not available in a second connection and is gone after reconnecting. Not all backends support this argument. A regular, non-temporary table is visible in a second connection, in a pre-existing connection, and after reconnecting to the database.  
 overwrite = TRUE #If the overwrite argument is TRUE, an existing table of the same name will be overwritten. This argument doesn't change behavior if the table does not exist yet.  
)  
  
DBI::dbDisconnect(con)

The following code shows how I read a SQL database

# connect data\_clean.db database to con -------------------------------------------------  
  
con <- DBI::dbConnect(drv = RSQLite::SQLite(), #use SQLite function to progress  
 dbname = "data\_clean.db",  
 #loadable.extensions = TRUE,  
 #default.extensions = loadable.extensions,  
 cache\_size = NULL,  
 synchronous = "off",  
 flags = SQLITE\_RWC,  
 vfs = NULL,  
 bigint = "integer64", #c("integer64", "integer", "numeric", "character")  
 extended\_types = TRUE) #!!!When TRUE columns of type DATE, DATETIME / TIMESTAMP, and TIME are mapped to corresponding R-classes, c.f. below for details. Defaults to FALSE. Use TRUE to keep correct date/time formats!!!  
  
# read data\_clean.db database using dbReadTable -----------------------------------------  
  
#create new object to test how database entries look - are all formats correct?  
data\_clean <- DBI::dbReadTable(con, "data\_clean") #note that all data are character vectors  
  
#correct format of vectors in database, e.g. use AMR::as.() and lubridate::receive\_date()   
data\_clean <- data\_clean |>   
 dplyr::mutate(across(where(AMR::is\_sir\_eligible), AMR::as.sir),  
 across(c(date\_of\_birth, receive\_date), lubridate::as\_date)) |>   
 dplyr::glimpse()  
  
DBI::dbDisconnect(con)

## ISLRN

The International Standard Language Resource Number (ISLRN) is a new, unique and universal identification schema for Language Resources which provides Language Resources with unique names using a standardized nomenclature [[42](#ref-ELDA-ISLRN)].

The 13-digit ISLRN format is: XXX-XXX-XXX-XXX-X. It can be allocated to any Language Resource; its composition is neutral and does not include any semantics in reference to the type or nature of the Language Resource. The ISLRN is a randomly created number with a check digit that validates a Verhoeff algorithm [[42](#ref-ELDA-ISLRN)].

For the purposes of this project, no ISLRN’s have been used.

# Governance and security

## Security, ethics and privacy

### Data storage

All raw data is stored in the LIMS, EHR, clinicalvision or bereavement ledger, and requires special access priviledges/permissions to access (i.e. access is restricted).

As mentioned above, database (SQL) files are stored (only temporarily) on an encrypted drive of a Trust laptop.

### Data collection

No new data collection is- or will be undertaken as part of this project.

### Version control

As mentioned above, Git Bash is used to iteratively upload all code to a private repository on GitHub [[30](#ref-ourcodingclub_github),[31](#ref-posit_git)].

Git is configured to ignore all potentially confidential files when making a commit, using a .gitignore file in the repository’s root directory [[43](#ref-gitignore)].

GitHub maintains an official list of recommended .gitignore files for many popular operating systems, environments, and languages in the “github/gitignore” public repository [[44](#ref-gitignore_dot_io)]. This file was used as the basis for a more extensive .gitignore file, which aims to prevent any patient confidential information being published to the (private) project repository.

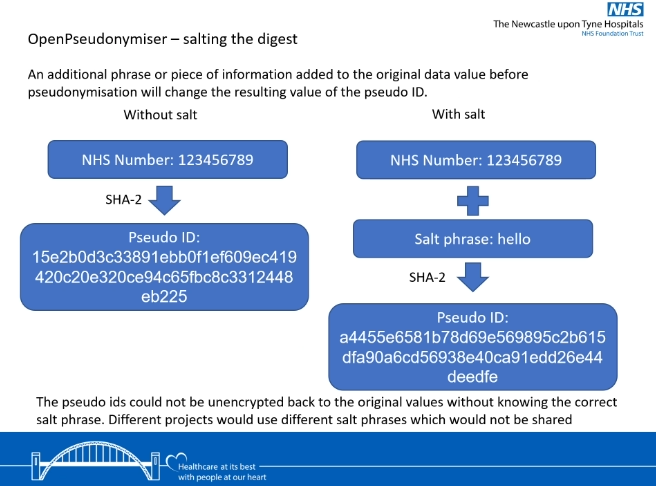
# Created by https://www.toptal.com/developers/gitignore/api/r  
# Edit at https://www.toptal.com/developers/gitignore?templates=r  
  
### R ###  
# History files  
.Rhistory  
.Rapp.history  
  
# Session Data files  
.RData  
.RDataTmp  
  
# User-specific files  
.Ruserdata  
  
# Example code in package build process  
\*-Ex.R  
  
# Output files from R CMD build  
/\*.tar.gz  
  
# Output files from R CMD check  
/\*.Rcheck/  
  
# RStudio files  
.Rproj.user/  
  
# produced vignettes  
vignettes/\*.html  
vignettes/\*.pdf  
  
# OAuth2 token, see https://github.com/hadley/httr/releases/tag/v0.3  
.httr-oauth  
  
# knitr and R markdown default cache directories  
\*\_cache/  
/cache/  
  
# Temporary files created by R markdown  
\*.utf8.md  
\*.knit.md  
  
# R Environment Variables  
.Renviron  
  
# pkgdown site  
docs/  
  
# translation temp files  
po/\*~  
  
# RStudio Connect folder  
rsconnect/  
  
### R.Bookdown Stack ###  
# R package: bookdown caching files  
/\*\_files/  
  
# End of https://www.toptal.com/developers/gitignore/api/r

### Caldicott principles

Caldicott principles are be observed throughout, with regards to data storage, data sharing and so forth [[45](#ref-Caldicott)].

### Data pseudonymisation

Furthermore, potentially confidential/identifiable information can be pseudonymised with, for example, OpenPseudonymiser, which is supported through the Trust [[46](#ref-OpenPseudononymiser)].



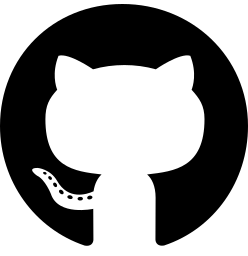
### Ethical approval

The privacy rights of all subjects have been observed throughout.

The final parameterised reports do not include any patient-level data in the body of the text, nor any tables or figures.

Formal ethics approval has not been sought given the service improvement nature of this project.

### IP rights

The final code for this document, which was created using Quarto, has been uploaded (in .qmd file format) to my public  [GitHub](https://github.com/send2dan/public) repository, under an MIT License

MIT License  
  
Copyright (c) 2023 Daniel Weiand  
  
Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:  
  
The above copyright notice and this permission notice shall be included in all  
copies or substantial portions of the Software.  
  
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.

I do not believe any IP needs to be protected, and there are no clear copyright and Intellectual Property Rights (IPR) issues, particularly given the fact I don’t own the original data, have no plans for its use/reuse by other parties, and have no plans to share the raw data.

## Responsibility and resources

### Primary Investigator (PI) and Co-Investigators

* Daniel Weiand, Consultant medical microbiologist [PI]
* Caroline Cullerton, Biomedical scientist [Co-Investigator; staff member with access to data stored in the Trust’s laboratory information management system (LIMS)]
* [Staff who have access to data stored in the Trust’s bereavement ledger]
* [Staff who have access to data stored in the Trust’s Nautilus data warehouse, which stores data from the Trust’s electronic health record (EHR)]
* [Staff who have access to data stored in the Trust’s dialysis information management system, clinicalvision]

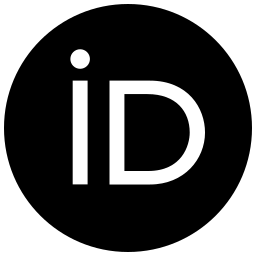
#### Contact details

Daniel Weiand, Consultant medical microbiologist ILM Directorate Clinical Informatics Lead Medical Examiner for Freeman Hospital Training Programme Director for Infection (Microbiology, Virology and Infectious Diseases) Newcastle upon Tyne Hospitals NHS Foundation Trust

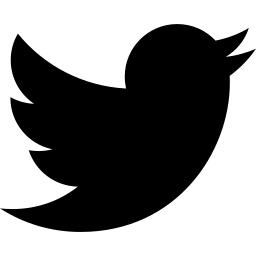
Email: dweiand@nhs.net

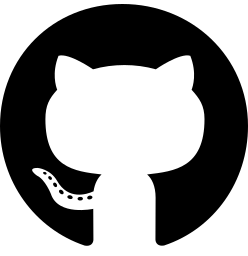
Tel (Direct): +44 (0)191 21 39054 Tel (Sec): +44 (0)191 22 37736 or +44 (0)191 22 31019

Website: <https://www.newcastlelaboratories.com>

 [ORCiD](https://orcid.org/0000-0001-5854-3452)

NHS-R community blog: <https://nhsrcommunity.com/author/daniel-weiand/>

 [send2dan](https://twitter.com/send2dan)

 [github.com/send2dan/public](https://github.com/send2dan/public)

|  |
| --- |
|  |

### Budget

Given the open-source nature of the software used to undertaken this project, there are no additional costs to account for with regards to data preparation, documentation or archiving, and there have been no requests for additional funding.

### Funder

There is no funder associated with this plan

### Grant number/url

N/A

# Limitations

Only phenotypic AST results are included (e.g. “R”, “I”, and “S”). In other words, data stored as disk diffusion zone sizes (e.g. 20mm) or MIC values (e.g. 1mg/L) are not included.

Phenotypic AST results recorded as “D” are recoded as “R” to facilitate analysis using the [AMR](https://msberends.github.io/AMR/articles/AMR.html) package [[11](#ref-AMR2022),[12](#ref-R-AMR)].

With regards to AMR rates, these are not easily predicted: for particular drug-bug combinations, the spread (i.e. standard error) can be very large.

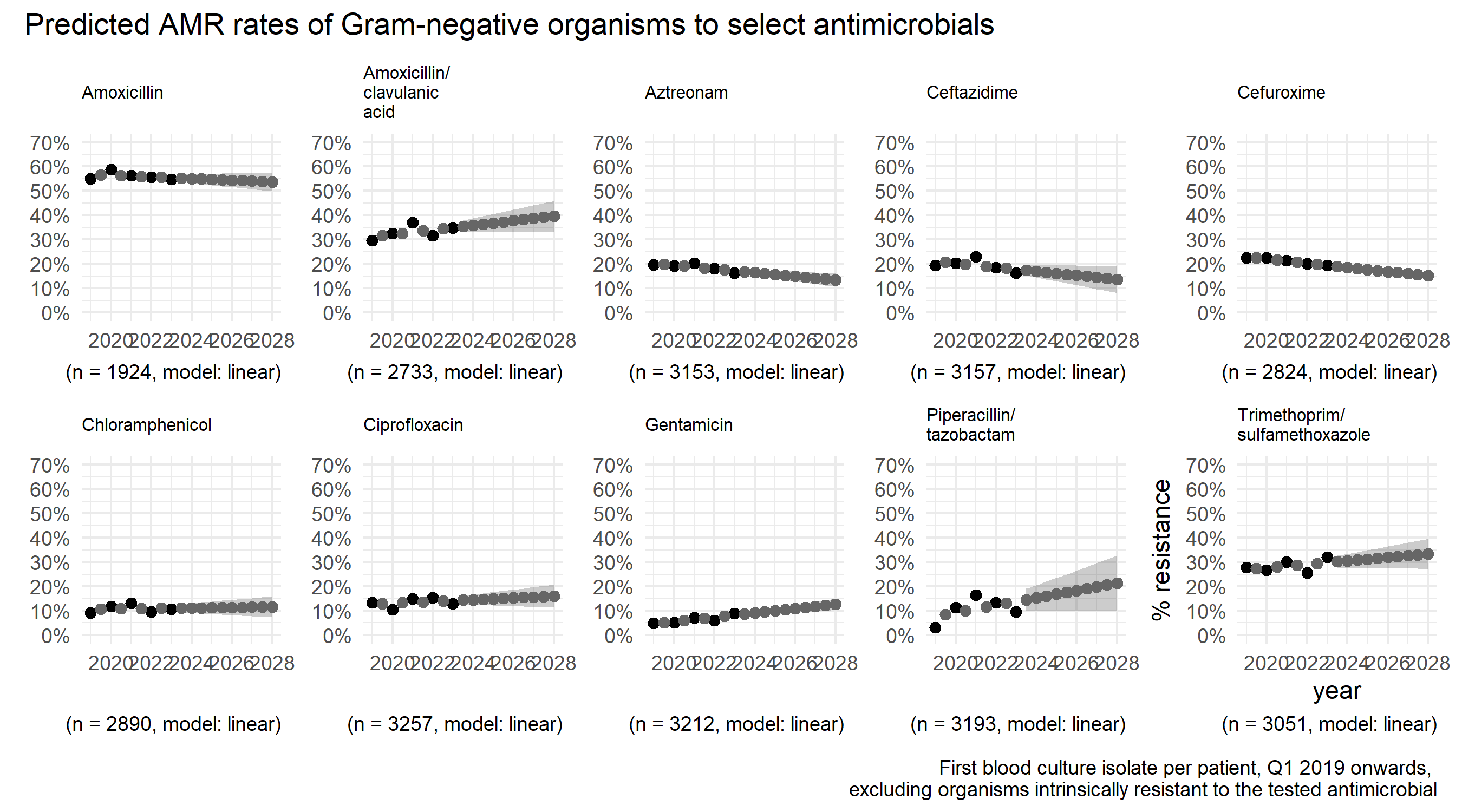
Generalised linear regression models using a binomial distribution assume that a period of zero resistance was followed by a period of increasing resistance leading slowly to more and more resistance. Where the standard error is very large, a linear model might be more appropriate.

The [AMR](https://msberends.github.io/AMR/articles/AMR.html) package [[11](#ref-AMR2022),[12](#ref-R-AMR)] facilitates modelling of drug resistance rates using either binomial or linear distributions. The chosen model is stated in each figure produced using the [AMR](https://msberends.github.io/AMR/articles/AMR.html) package.

As mentioned above, with regards to survival data, the Newcastle upon Tyne Hospitals NHS Foundation Trust’s bereavement ledger only includes information on patients who died in hospital, before hospital discharge and/or transfer to another healthcare setting. In effect, information is not available on the status of patients in the community and/or other healthcare settings. As a result, the mortality data outlined in the survival analysis are likely to underestimate patients’ true mortality rates.

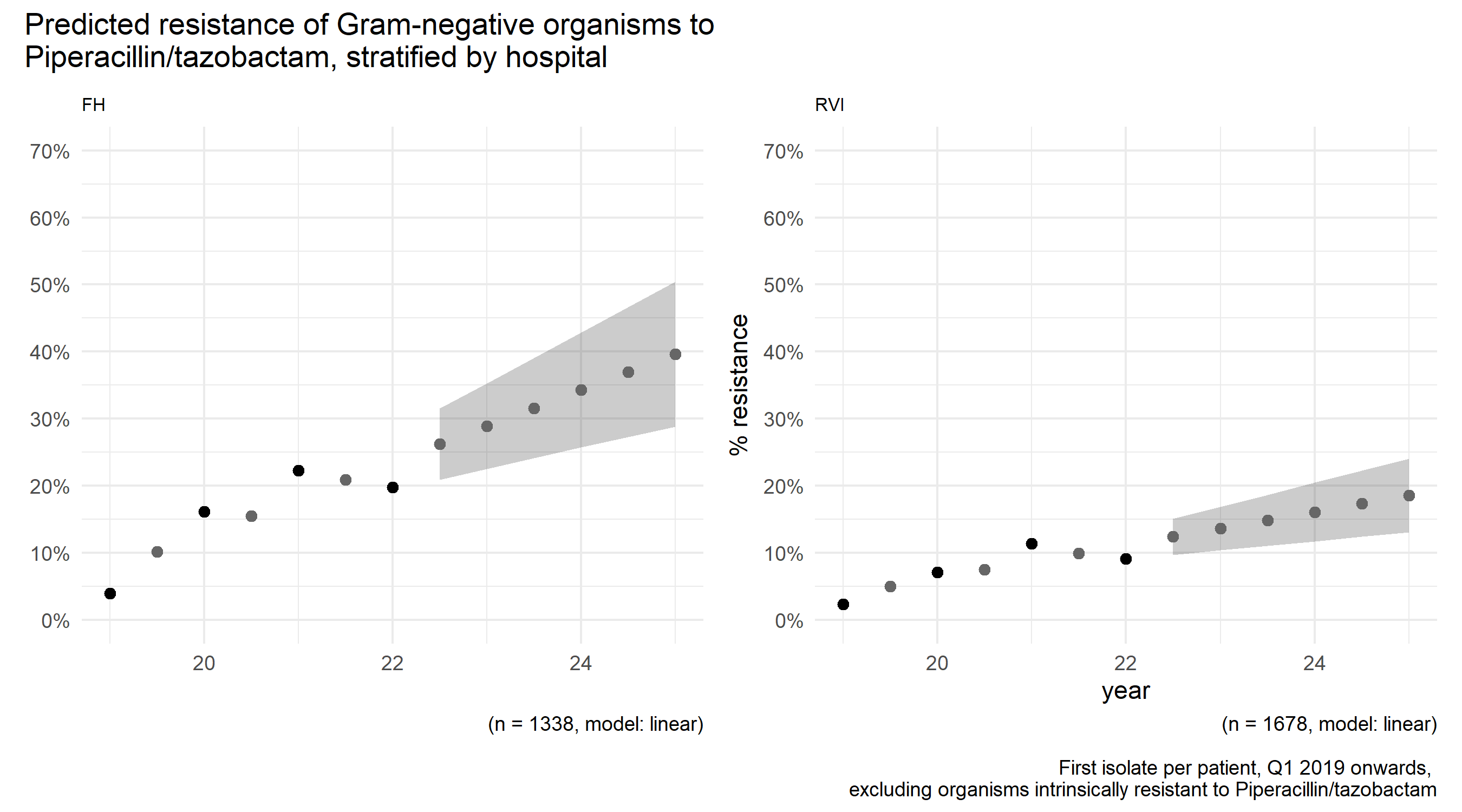
# Example output

## Prediction of future AMR trends

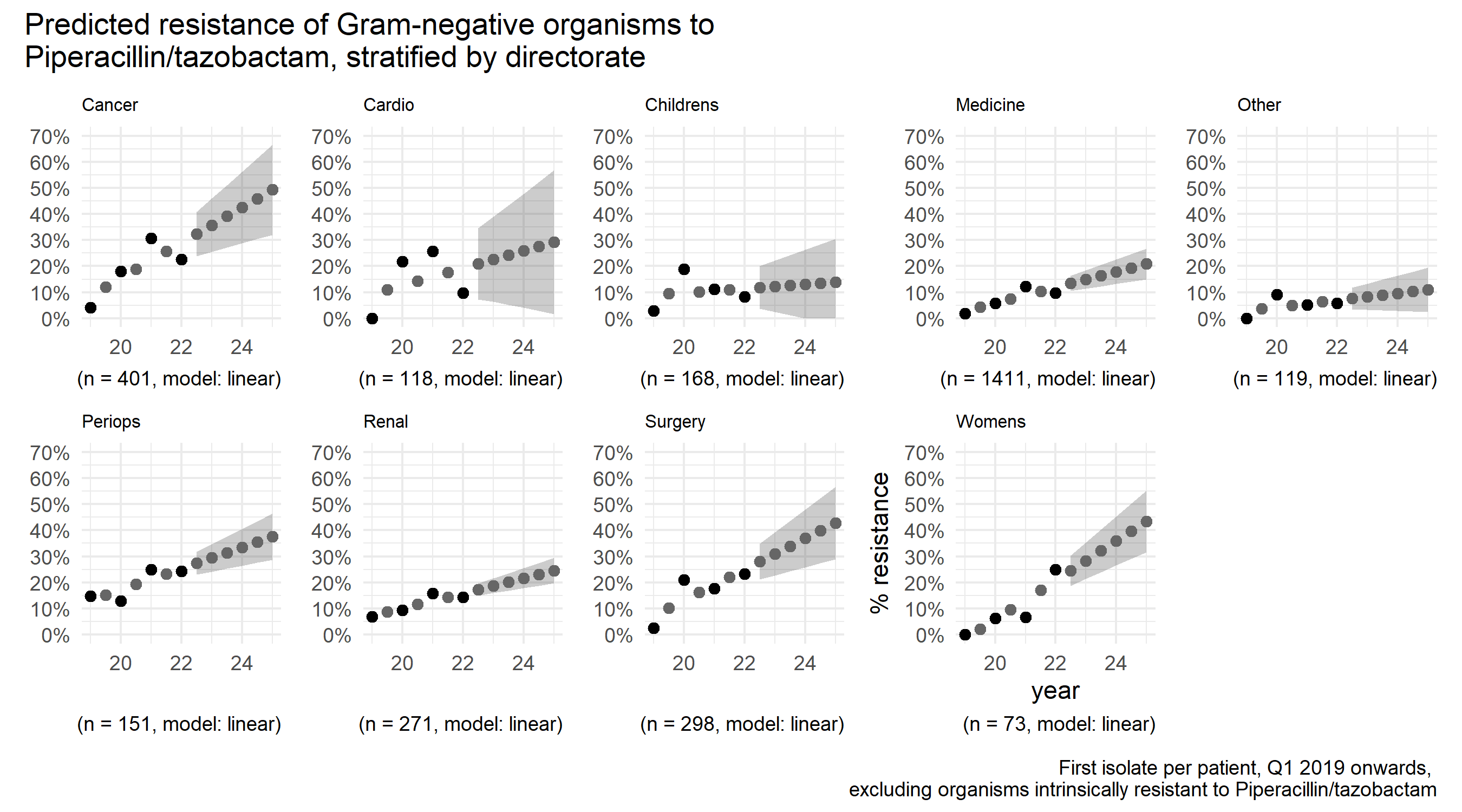


## Stratified reporting of predicted AMR rates

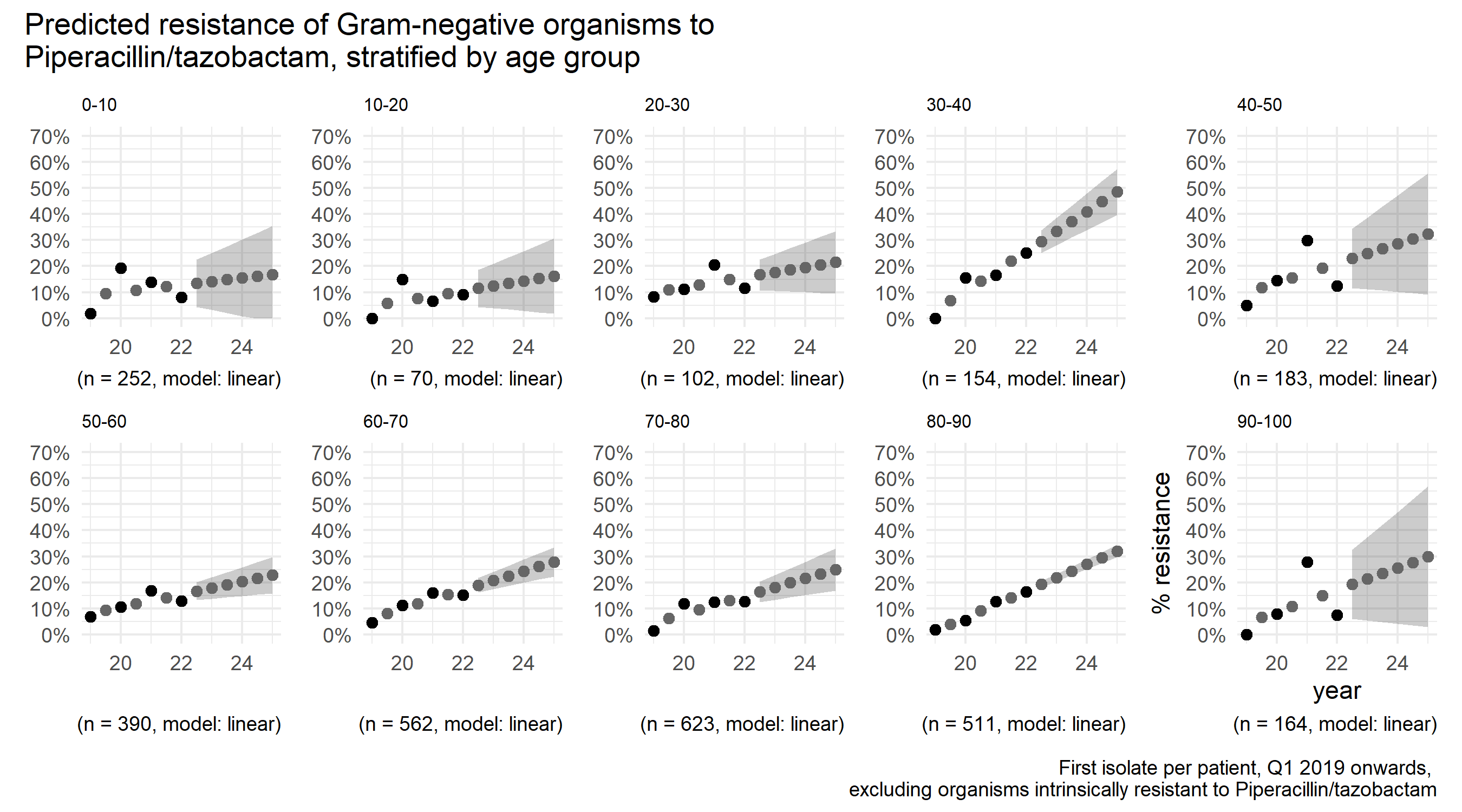
For example, tazocin resistance rates, stratified by hospital



Also, tazocin resistance rates, stratified by clinical directorate

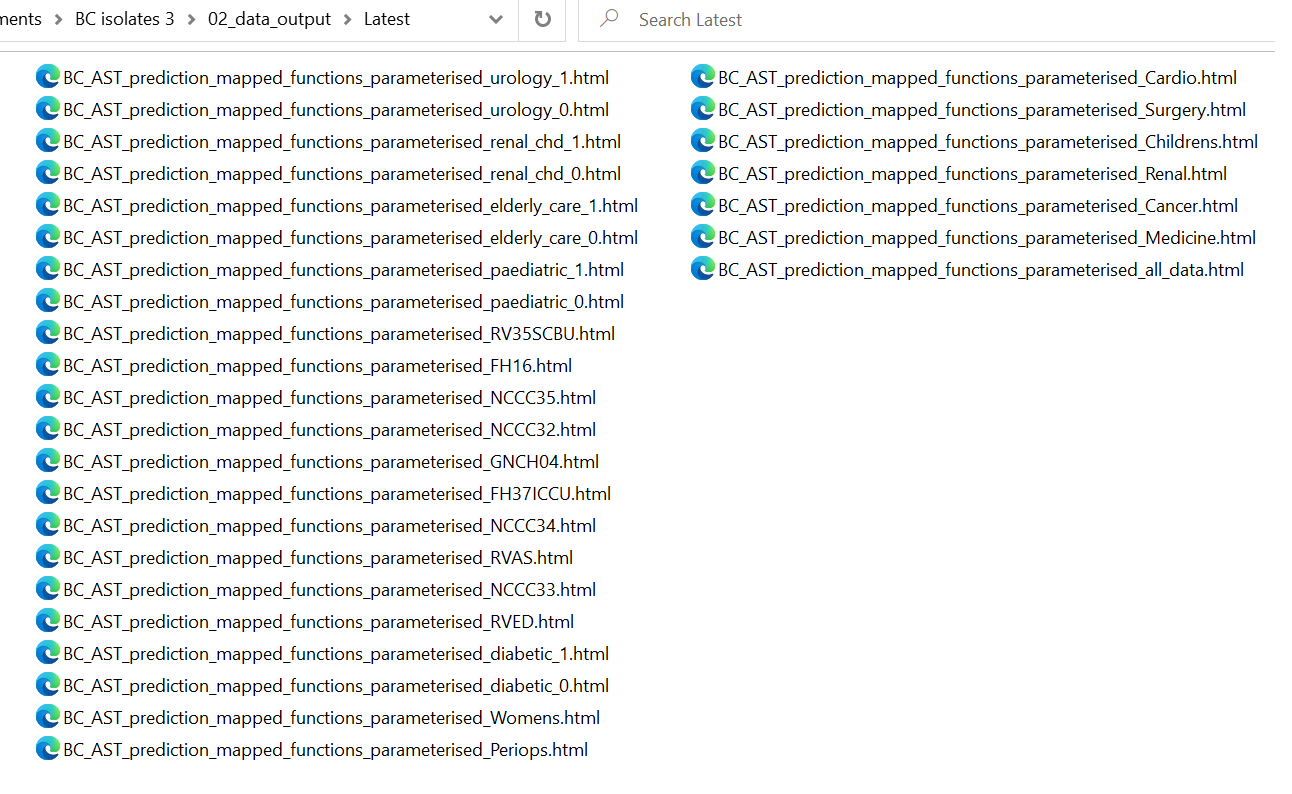


Also, tazocin resistance rates, stratified by age group



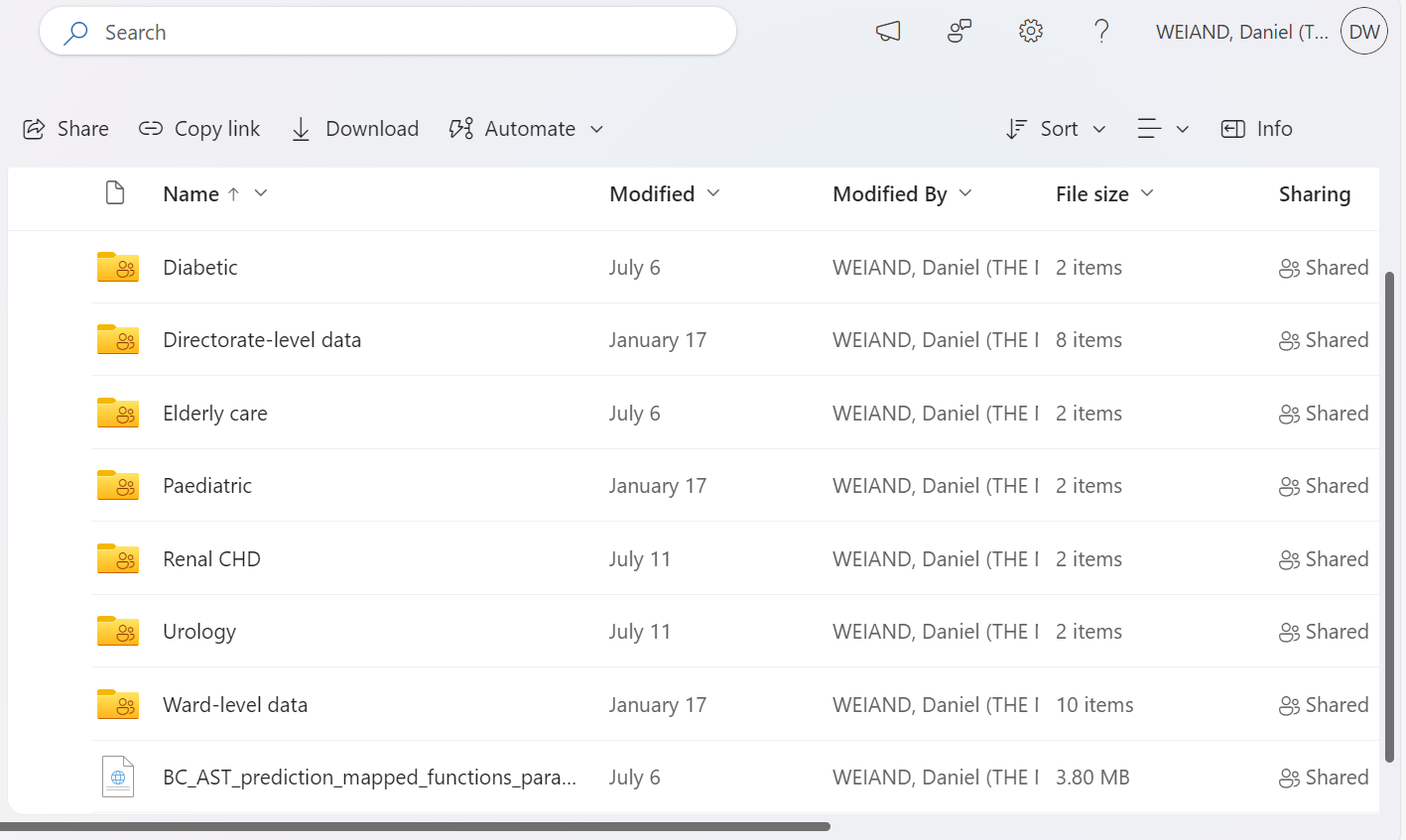
## Parameterised reporting of predicted AMR trends

Individual reports are prepared using Quarto



## Sharing the results

Parameterised reports are uploaded to the Trust’s OneDrive



# References

1 University M. Introduction to data management plans - data management plan. 2023.<https://elab5.its.manchester.ac.uk/content/mylearning/rise/3-3/content/index.html#/lessons/BVL99NDTf6ILraiIXwmhzL6sfMKZD9x0>

2 Bishop BW, Neish P, Kim JH, *et al.* Data management plan implementation, assessments, and evaluations: Implications and recommendations. *Data Science Journal* 2023;**22**. doi:[10.5334/dsj-2023-027](https://doi.org/10.5334/dsj-2023-027)

3 ELDA. Data management plan. 2023.<http://www.elra.info/en/services-around-lrs/dmp/>

4 ELRA. The FLaReNet book. 2023.<http://www.elra.info/media/filer_public/2021/11/18/flarenet_final_book.pdf>

5 ICPSR. Elements of a data management plan. 2023.<https://www.icpsr.umich.edu/web/pages/datamanagement/dmp/elements.html>

6 ICPSR. Intro to data management plans - YouTube. 2023.<https://www.youtube.com/watch?v=0m5kgYsPwe0&t=825s>

7 Foundation NS. Preparing your data management plan - funding at NSF | NSF - national science foundation. 2023.<https://new.nsf.gov/funding/data-management-plan>

8 dmponline. Dmponline dashboard. 2023.<https://dmponline.dcc.ac.uk/plans/new>

9 University N. DMP website. 2023.<https://dmp.ncl.ac.uk/plans/new>

10 WHO. Global action plan on antimicrobial resistance. 2023.<https://www.who.int/publications/i/item/9789241509763>

11 Berends MS, Luz CF, Friedrich AW, *et al.* AMR: An R package for working with antimicrobial resistance data. *Journal of Statistical Software* 2022;**104**:1–31. doi:[10.18637/jss.v104.i03](https://doi.org/10.18637/jss.v104.i03)

12 Berends MS, Luz CF, Souverein D, *et al.* *AMR: Antimicrobial resistance data analysis*. 2023. <https://CRAN.R-project.org/package=AMR>

13 R Core Team. *R: A language and environment for statistical computing*. Vienna, Austria: : R Foundation for Statistical Computing 2022. <https://www.R-project.org/>

14 Hindler JF, Stelling J. Analysis and presentation of cumulative antibiograms: A new consensus guideline from the Clinical and Laboratory Standards Institute. *Clinical infectious diseases* 2007;**44**:867–73.

15 Reading C, Wellesley-Miller S, Turner Z, *et al.* *NHSRplotthedots: Draw XmR charts for NHSE/i making data count programme*. 2021. <https://nhs-r-community.github.io/NHSRplotthedots/>

16 Sjoberg DD, Baillie M, Fruechtenicht C, *et al.* *Ggsurvfit: Flexible time-to-event figures*. 2023. <https://CRAN.R-project.org/package=ggsurvfit>

17 Therneau TM. *Survival: Survival analysis*. 2022. <https://github.com/therneau/survival>

18 Terry M. Therneau, Patricia M. Grambsch. *Modeling survival data: Extending the Cox model*. New York: : Springer 2000.

19 Mack C, Su Z, Westreich D. Types of missing data. 2018.

20 Weiskopf NG, Rusanov A, Weng C. Sick patients have more data: The non-random completeness of electronic health records. *AMIA Annual Symposium Proceedings*;**2013**:1472–2.

21 dataedo. What is metadata (with examples) - data terminology. 2023.<https://dataedo.com/kb/data-glossary/what-is-metadata>

22 NHS. NHS data model and dictionary. 2023.<https://www.datadictionary.nhs.uk/index.html>

23 IBM. IBM cognos analytics. 2023.<https://www.ibm.com/products/cognos-analytics>

24 UKAS. Medical laboratory accreditation - ISO 15189. 2023.<https://www.ukas.com/accreditation/standards/medical-laboratory-accreditation/>

25 England N. NHS england » LFPSE-compliant local risk management system (LRMS) suppliers. 2023.<https://www.england.nhs.uk/patient-safety/learn-from-patient-safety-events-service/lrms-suppliers/>

26 Cerner. Millennium | cerner. 2023.<https://www.cerner.com/se/en/solutions/millennium>

27 Nautilus. Nautilus systems - data warehousing. <https://www.nautilus-systems.com/dw1.html>

28 Group CK. Clinical vision by clinical computing. 2023.<https://www.constellationkidney.com/electronic-health-records/clinical-vision/>

29 MHRA. Managing medical devices. 2023.<https://assets.publishing.service.gov.uk/media/6089dc938fa8f51b91f3d82f/Managing_medical_devices.pdf>

30 ourcodingclub.github.io. Intro to github for version control. 2023.<https://ourcodingclub.github.io/tutorials/git/>

31 Posit. Version control with git and SVN. 2023.<https://support.posit.co/hc/en-us/articles/200532077>

32 Quarto. Quarto. 2023.<https://quarto.org/>

33 Lucidchart. ER diagram (ERD) - definition & overview. 2023.<https://www.lucidchart.com/pages/er-diagrams>

34 YouTube. Entity relationship diagram (ERD) tutorial - part 1. 2023.<https://www.youtube.com/watch?v=QpdhBUYk7Kk&t=41s>

35 Lucidchart. Entity-relationship diagram symbols and notation. 2023.<https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning>

36 Lucid. Https://lucid.app/. 2023.<https://lucid.app/documents#/documents?folder_id=recent>

37 R Special Interest Group on Databases (R-SIG-DB), Wickham H, Müller K. *DBI: R database interface*. 2022. <https://CRAN.R-project.org/package=DBI>

38 Müller K, Wickham H, James DA, *et al.* *RSQLite: SQLite interface for r*. 2023. <https://CRAN.R-project.org/package=RSQLite>

39 Wickham H, Girlich M, Ruiz E. *Dbplyr: A dplyr back end for databases*. 2023. <https://CRAN.R-project.org/package=dbplyr>

40 CRAN. Clean data.frame names with clean\_names(). 2023.<https://cran.r-project.org/web/packages/janitor/vignettes/janitor.html>

41 Wickham H. Tidy data. *The Journal of Statistical Software* 2014;**59**.<http://www.jstatsoft.org/v59/i10/>

42 ELDA. What is ISLRN? 2023.<http://www.elda.org/en/islrn/overview/>

43 Docs G. Ignoring files. 2023.<https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files>

44 gitignore.io. Gitignore.io - create useful .gitignore files for your project. 2023.<https://www.toptal.com/developers/gitignore>

45 GOV.UK. The caldicott principles. 2023.<https://www.gov.uk/government/publications/the-caldicott-principles>

46 OpenPseudononymiser. OpenPseudononymiser. 2023.<https://www.openpseudonymiser.org/>