Data quality in health research: the development of methods to improve the assessment of temporal data quality in electronic health records

Nuffield Department of Population Health

T. Phuong Quan, BSc (Hons), MSc

St. Catherine's College, Oxford

Thesis submitted for the degree of Doctor of Philosophy

Michaelmas Term 2022

ABSTRACT

Background: Electronic health records (EHR) are increasingly used in medical research, but the prevalence of temporal artefacts that may bias study findings is not widely understood or reported. Furthermore, methods aimed at efficient and transparent assessment of temporal data quality in EHR datasets are unfortunately lacking.

Methods: 7959 time series representing different measures of data quality were generated from eight different EHR data extracts covering activity between 1986-2019 at a large UK hospital group. These time series were visually inspected and annotated via a citizen-science crowd-sourcing platform, and consensus labels for the locations of all change points (i.e. places where the distribution of data values changed suddenly and unpredictably) were constructed using density-based clustering with noise. The crowd-sourced consensus labels were validated against labels produced by an experienced data scientist, and a diverse range of automated change point detection methods were assessed for accuracy against these consensus labels using a novel approximation to a binary classifier. Lastly, an R package was developed to facilitate assessment of temporal data quality in EHR datasets.

Results: Over 2000 volunteers participated in the citizen-science project, performing 341,800 visual inspections of the time series. A total of 4477 distinct change points were identified across the eight data extracts, covering almost every year of data and virtually all data fields. Compared to expert labels, accuracy of crowd-sourced consensus labels identifying the locations of individual change points had high sensitivity 80.4% (95% CI 77.1, 83.3), specificity 99.8% (99.7, 99.8), positive predictive value (PPV) 84.5% (81.4, 87.2) and negative predictive value (NPV) 99.7% (99.6, 99.7). Automated change point detection methods failed to detect the crowd-sourced change points accurately, with maximum sensitivity 36.9% (35.2, 38.8), specificity 100% (100, 100), PPV 51.6% (49.4, 53.8), and NPV 99.9% (99.9, 99.9).

Conclusions: This large study of real-world EHR found temporal artefacts occurred with very high frequency, which could impact findings from analyses using these data. Crowd-sourced labels of change points compared favourably to expert labels, but currently-available automated methods performed poorly at identifying such artefacts when compared to human visual inspection. To improve reproducibility and transparency of studies using EHRs, thorough visual assessment of temporal data quality should be conducted and reported, which can be assisted by tools such as the new *daiquiri* R package developed as part of this thesis.

Acknowledgements

I would like to thank my NDM supervisors Sarah Walker and Tim Peto for supporting me over the years and for believing in me and in this project idea, and I would like to thank my NDPH supervisors Martin Landray and Ben Lacey for their support in making this project happen and getting it over the line. I would also like to thank everyone I have ever talked to about this project, whether it was to discuss ideas, offer feedback, or provide encouragement.

This work uses data generated via the Zooniverse.org platform. I would like to thank the Zooniverse team and all the Zooniverse volunteers who donated their time freely and generously.

This work uses data provided by patients and collected by the NHS as part of their care and support. We thank all the people of Oxfordshire who contribute to the Infections in Oxfordshire Research Database. Research Database Team: L Butcher, H Boseley, C Crichton, DW Crook, DW Eyre, O Freeman, J Gearing (community), R Harrington, K Jeffery, M Landray, A Pal, TEA Peto, TP Quan, J Robinson (community), J Sellors, B Shine, AS Walker, D Waller. Patient and Public Panel: G Blower, C Mancey, P McLoughlin, B Nichols.

This work was supported by the National Institute for Health Research (NIHR) Health Protection Research Unit in Healthcare Associated Infections and Antimicrobial Resistance (NIHR200915), a partnership between the UK Health Security Agency (UKHSA) and the University of Oxford, and by the NIHR Oxford Biomedical Research Centre.

Statement of contribution to work

I designed and conducted all analyses and data collection, drafted the text, and produced all tables and figures presented in the thesis. I conceived the idea with input from my supervisors Sarah Walker, Tim Peto and Martin Landray, who also provided advice on the design and methodology, and together with Ben Lacey reviewed the thesis for scientific merit. I wrote the code for the R software package, with the exception only of a small amount of specific code written by Jack Cregan to draw the aggregated data summary plots which I incorporated.

All analyses were conducted using R v3.6.3, and the code is available to download from https://doi.org/10.5281/zenodo.7327780

Table of Contents

List of abbreviations

ACHILLES	Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems (software tool)
BS	Binary Segmentation
CRAN	The Comprehensive R Archive Network
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DQ^{e}	Data Quality Explorer (software tool)
ED	Emergency Department
EHR	Electronic Health Record
HES	Hospital Episode Statistics
IDA	Initial Data Analysis
IORD	Infections in Oxfordshire Research Database
MMC	Matthew's Correlation Coefficient
NPV	Negative Predictive Value
OMOP CDM	Observational Medical Outcomes Partnership Common Data Model
OUH	Oxford University Hospitals NHS Foundation Trust
PELT	Pruned Exact Linear Time
PPV	Positive Predictive Value
RECORD	REporting of studies Conducted using Observational Routinely-collected health Data
SN	Segment Neighbourhood
SUS	Secondary Uses Service
WBS	Wild Binary Segmentation
WHO	World Health Organisation

Aggregation functions

distinct	number of distinct values
max	maximum value
maxlength	maximum string length
mean	mean value
meanlength	mean string length
median	median value
midnight_n	number of values with no time element
midnight_perc	percentage of values with no time element
min	minimum value
minlength	minimum string length
missing_n	number of missing values
missing_perc	percentage of missing values
n	number of values present
nonconformant_n	number of non-conformant values
nonconformant_perc	percentage of non-conformant values
nonzero_perc	percentage of records which had been duplicated
subcat_n	number of non-missing values within each subcategory
subcat_perc	percentage of non-missing values within each subcategory
sum	sum of duplicate records removed

List of Figures

List of Tables

CHAPTER 1: Introduction

1.1 Use of electronic health records in research

Electronic health records (EHRs) (also known as electronic medical records or electronic patient records) were first introduced for patient management in the US in the 1960s[1], and have developed since then to cover a wide range of activity from patient administration to medical interventions to laboratory test results. A survey conducted in November 2018 showed that 77% of acute NHS hospital Trusts in England were using full EHR systems in place of paper records[2].

The value of EHRs beyond immediate patient care has also been increasingly recognised, with their use in medical research growing enormously over the past 30 years, assisted by the availability of large administrative datasets such as Hospital Episode Statistics[3], which cover secondary care activity across all of England. A simple title and abstract search in PubMed for the phrase “electronic health record(s)” reveals an almost exponential increase since 1993 when one single study was published, compared to 1122 studies in 2018 (Figure 1-1).

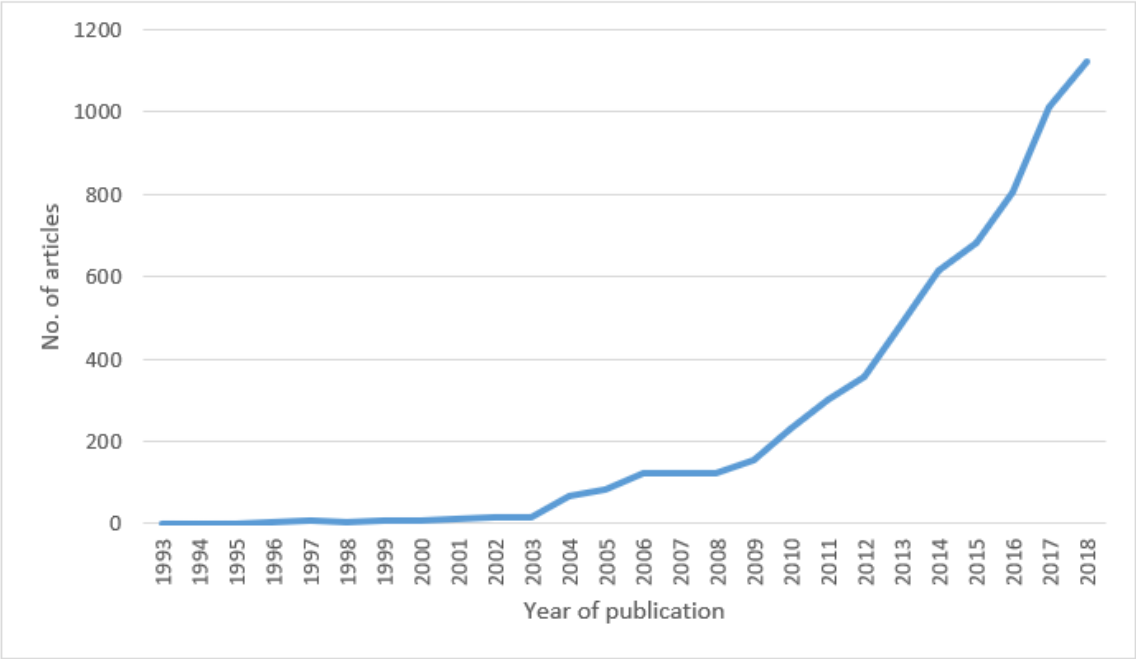


Figure 1-1. Number of articles in PubMed that contain the phrase "electronic health record(s)" in the title or abstract.

One of the main advantages that EHRs offer to researchers is the sheer volume and detail of data available, covering large numbers of patients and often over long time periods. For instance the Oxford University Hospitals has a linked EHR database which currently comprises over a billion records across ~7000 data fields and over 20 years of activity (described in more detail below).

However, there are limitations associated with using routinely-collected data for research. Since EHRs comprise data that were collected for a different purpose (i.e. operational rather than research), and usually at a great distance (both temporally and physically) from the researchers making use of it, there is a high risk of erroneous assumptions being made.

Unfortunately, it is not clear to what extent this is understood and taken into account by researchers who use this data.

1.2 Risk of systematic bias in EHR datasets

While it is generally acknowledged that data that has been typed in manually by humans will contain a certain amount of errors (for example, misspelled words), it is not well understood how the data that is entered during the healthcare encounter are then extracted and transformed into analytic datasets for use by researchers. This process will also have been designed and implemented by humans, and often involves manual steps, which means that

> *"[Data quality] issues can appear in any data source, even those generated by highly skilled technical staff." – Kahn et al.*[4]

Furthermore, there are systematic errors and biases that can arise through unexpected events such as changes in data classification (e.g. whether patients undergoing haemodialysis are recorded as inpatients or not), major and minor software updates (e.g. when the available values of a dropdown menu change), or when the meaning or content of a particular data field changes (e.g. when laboratory methods and/or units of measurement are altered), see Figure 1-2. These types of events have a temporal nature (i.e. they happen at a particular point in time and can either be long or short-lived), and if they are not taken into account when analysing EHR data, this can lead to erroneous results, incorrect conclusions being drawn, and ultimately result in poor decisions at a clinical or public health policy level.

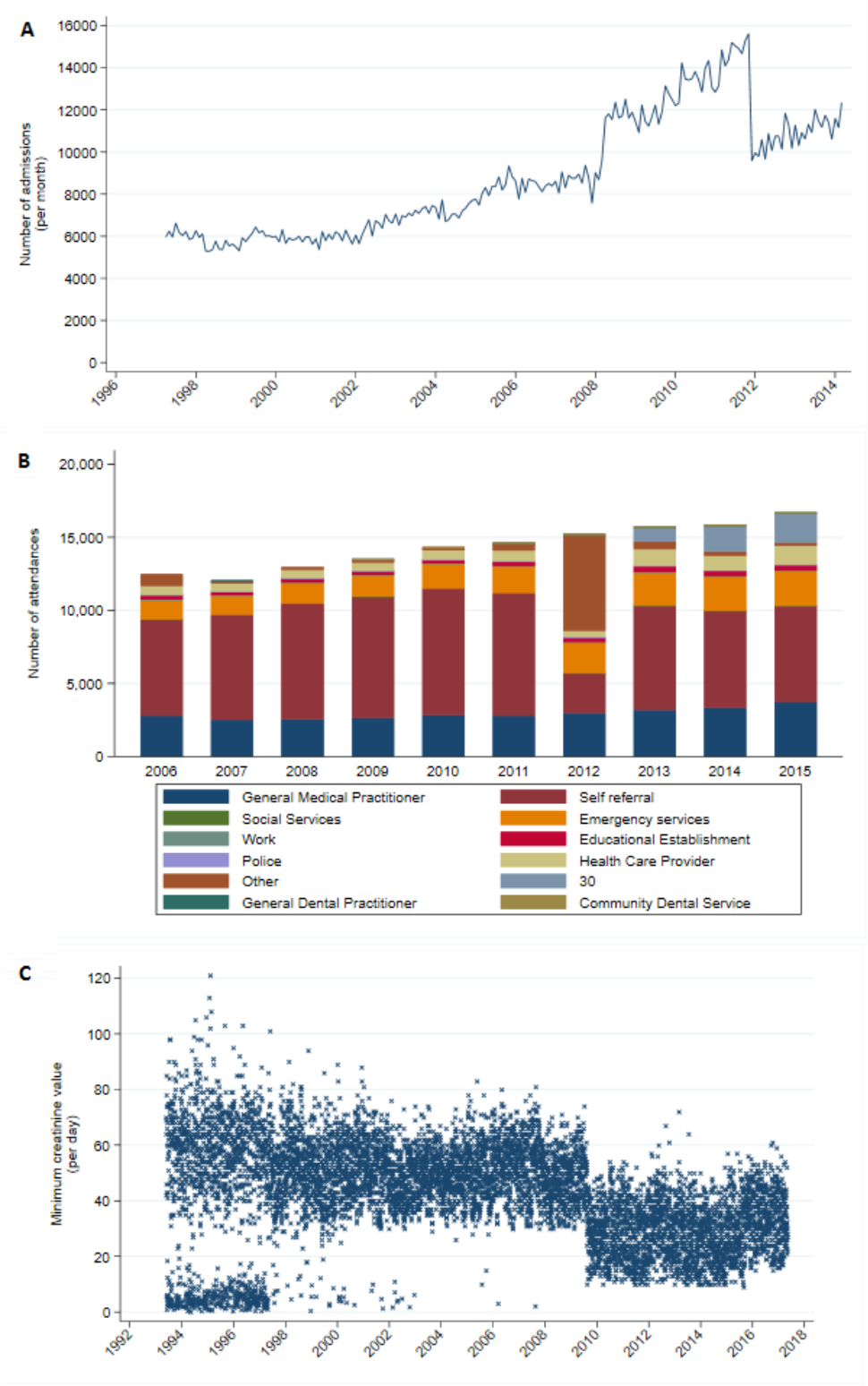


Figure 1-2. Examples of temporal changes in data caused by updates to infrastructure at Oxford University Hospitals.
(A) Total number of inpatient admissions containing multiple diagnosis codes. The sudden increase in records in 2008 was caused by the inclusion of dialysis day-case patients, which were then excluded again in 2012. (B) Emergency Department attendances by referral source. A change in computer systems in 2011 noticeably affected the data recorded, with the 'Other' category temporarily being overrepresented in 2012, and a new, undefined category of '30' appearing thereafter. (C) Lowest creatinine blood test result each day. The bimodal distribution up to 1997 was due to a mixture of units being used, and the drop in values in 2009 was due to a change in testing method and reference range.

While these unexpected changes in the data should in theory be identified by the diligent researcher, in practice it is not clear to what extent this is actually done, nor how often these kinds of events occur, since this initial data analysis stage is rarely, if ever, reported in published papers. Even when metadata exist about changes in practice or that highlight previously-identified data quality issues, it is unreasonable to expect these to cover the needs of every possible research question, or to expect that no further errors could have been introduced during the extraction process. It is therefore incumbent on each researcher to investigate and understand the specific limitations that their particular dataset may contain.

1.3 Different types of temporal changes that can occur in data

Temporal changes in data can either be smooth and predictable (such as trends and seasonality), or abrupt (such as outliers, or sudden changes in trend, level or variability), see Figure 1-3. From a data quality perspective, it is the abrupt changes that are of more interest, since artefacts caused by changes in data collection or processing are more likely to manifest as sudden, unnatural-looking changes, while real changes in the patient population are more likely to occur smoothly or predictably. These types of sudden changes are also known as ‘change points’.

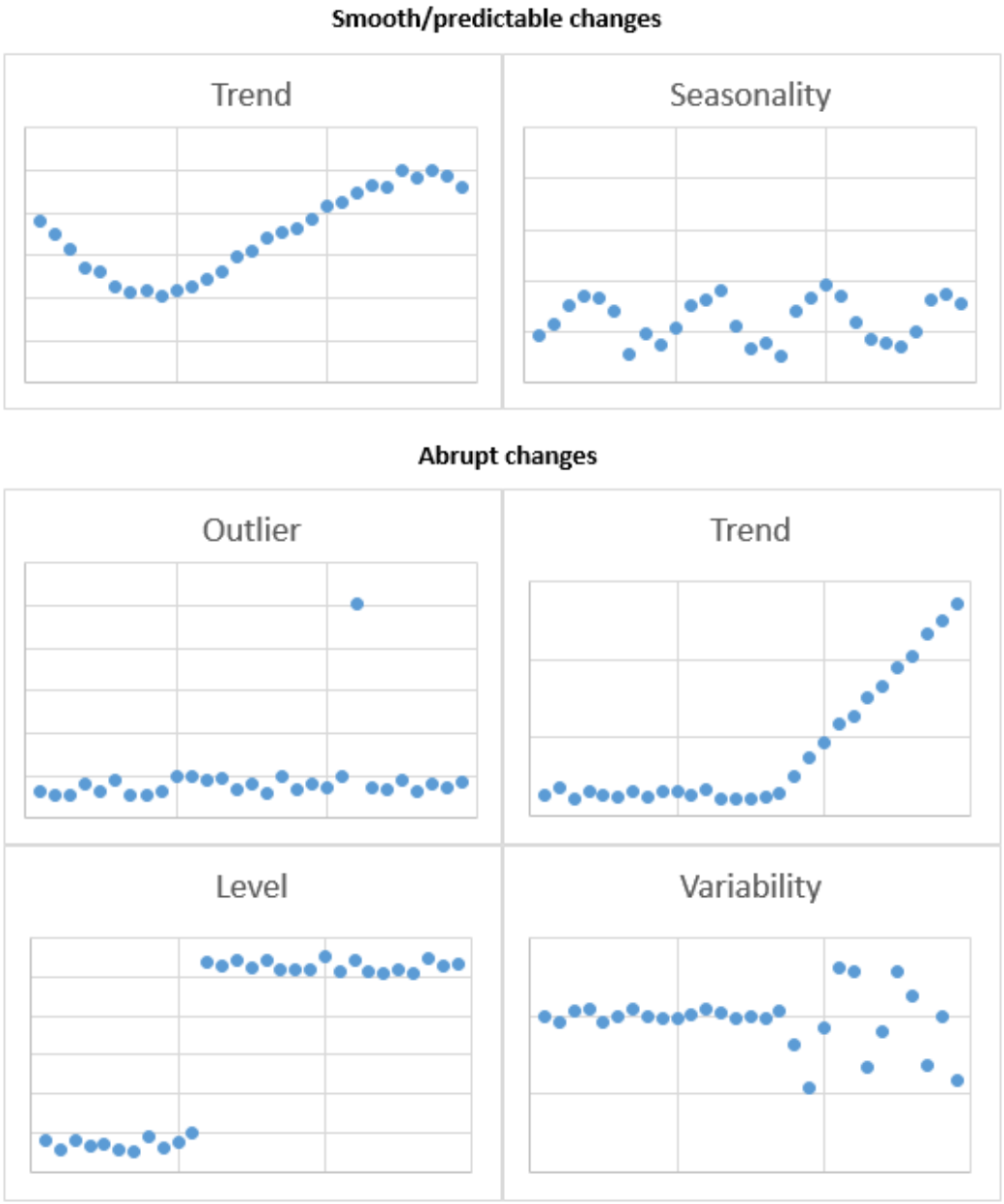


Figure 1-3. Illustration of different types of temporal changes

Research studies covering extended time periods are likely to benefit, in particular, from checking for change points, in order to identify any systematic biases in the data from the start versus the end of the study. This is particularly the case when the key exposure of interest relates to calendar time, e.g. the impact of the introduction of a new policy or system of management, which could then be completely confounded with any such systematic bias. However, a sudden change could in theory occur at any point in time (including in the middle of the day), and since even cross-sectional studies will almost certainly use data from more

than one day, this effectively applies to *all* epidemiological studies (only assuming that they contain data spanning more than one instant in time). Another potential use for these types of checks would be to help identify time periods where the data appears consistent enough to use for a particular analysis or sub-analysis.

A key question is how often these types of changes actually occur in EHR datasets, and what the consequences would be if they are not identified. Anecdotally within the Nuffield Department of Medicine at the University of Oxford, dozens of temporal change points have been found in the EHR data at Oxford University Hospitals during the course of conducting epidemiological studies, and there are almost certain to be many more, particularly as the dataset is constantly growing. The potential downstream effects on study results and interpretation if such change points are not identified will obviously be situation dependent, but it is important for researchers to at least try to identify and eliminate any such systematic biases in advance of their main analyses where possible.

1.4 Transparency and reproducibility of initial data analysis stage of studies

Standards such as STROBE[5] (Strengthening the Reporting of Observational Studies in Epidemiology) have been in use for over 10 years to help improve the quality of reporting of observational research, and the RECORD[6] (REporting of studies Conducted using Observational Routinely-collected health Data) statement was produced in 2015 to include further considerations when using routinely-collected health data such as EHRs. However, these standards focus on how to report what was done, rather than on what should have been done in the first place. An initiative to strengthen the actual process of analysing data in observational studies[7] (STRATOS - STRengthening Analytical Thinking for Observational Studies) was set up in 2011, however the outputs of this group have so far been limited to studies using primary-data collection. Even so, they note the following:

"As much as 80% of the time allocated to the statistical analysis process is spent on data cleaning and preparation; however, these first steps in the analysis of data are often neglected or disorganized, and decisions made during these steps, such as changing the methods used or the outcomes measured, are often unreported." – Huebner et al.[8]

Given the increasing drive towards greater transparency and reproducibility within the scientific community, it is inevitable that this essential, yet often-overlooked, part of the analysis process will soon begin to come under greater scrutiny. Furthermore, given the ability of EHR data to enable very large studies, both in terms of the number of subjects but also in the numbers of measurements and data fields available for each subject, it is possible that the traditional methods used to inspect the data for any potential issues may no longer be feasible, and that finding ways to automate some parts of this process would also enable greater consistency and reproducibility. Healthcare is becoming more (not less) complex with multiple, heterogeneous datasets, so errors, omissions and poor documentation will continue to accumulate as time and technology progresses, meaning that this is a task that will need to be repeated frequently. It is therefore important to increase our understanding of the extent of the problem, and to develop methods to improve the assessment and reporting of data quality issues in this context of large and ever-increasing amounts of data.

1.5 Thesis objectives

This thesis has 4 main objectives:

1. Summarise the published literature regarding the assessment of data quality in EHRs for research, in particular the detection of sudden, unexpected temporal change points. This is covered in Chapter 2.
2. Increase our understanding of the prevalence of temporal change points in real-world EHR datasets. This is covered in Chapter 3.

3. Assess a range of methods for identifying temporal change points in real-world EHR datasets, both visually and by automation. This is covered in Chapters 4 and 5.
4. Develop open-source tools for improving the efficiency, transparency, and reproducibility of identifying temporal data quality issues in EHRs for research, in particular by using automation where possible. This is covered in Chapter 6.

CHAPTER 2: Literature review

2.1 Chapter summary

The aim of this chapter is to identify what guidance and support is currently available to EHR researchers for assessing data quality, both in general and in relation to temporal artefacts.

A literature review was conducted in the abstracting databases Embase and PubMed up to 19 June 2019. A first, broad search used keywords relating to “data quality”, “EHRs”, and “research”. A second search focussing on temporal changes used keywords relating to “data quality”, “EHRs”, and “temporal”.

Data quality frameworks found in the literature categorised elements of data quality in different ways, sometimes using the same term to mean different things. Features of *intrinsic* data quality, i.e. those that do not depend on any external requirements, are particularly suited for automation since they are generalisable to different EHR datasets and study designs. Intrinsic data quality can be conceptualised using the categories of Completeness, Conformance, and Plausibility. The most common methods suggested for assessing intrinsic data quality were calculating simple summary statistics and visual inspection of graphs.

Only two articles reported on the prevalence of temporal artefacts, though these were limited to only two types of variables (i.e. categorical codes and numeric lab test results) rather than applying to EHR data in general. Only one article employed a method which appeared promising as a way to automate checks for abrupt temporal changes.

In conclusion, there is very little in the literature to mid-2019 regarding the general prevalence of temporal artefacts or in effective methods to assess temporal data quality in large medical datasets, so there are many opportunities to make a difference in this area.

2.2 Aims

The aims of this literature review fall into two sections. Firstly, to identify and summarise the published literature regarding the assessment of data quality in EHRs for research in general, particularly what guidance, methods and tools are available to researchers to ensure that the data they are using is fit for purpose. Secondly, focussing on the detection of sudden, unexpected temporal changes in EHR data (referred to here as 'change points'), to find out what is already known about the prevalence of such change points in EHR data (in order to understand whether it is common enough to routinely recommend checking for them), as well as to identify what methods are currently used to address their detection.

This review is not concerned with methods to improve data quality at the point of data entry, as this will normally not be within the control of the individual researcher working with EHRs, but rather its focus is on how best to assess data quality issues once the researcher has received their EHR data extract, particularly when it involves large datasets.

2.3 Overall search strategy

A keyword search was conducted in two large medical abstracting databases: PubMed and Embase. Specific keywords and inclusion criteria are described in the relevant subsequent sections. Inclusion was done by title and abstract review, followed by full text screening. References made to articles that did not appear in the search were followed manually, and any relevant further publications included.

2.4 General methods and guidelines for data quality assessment of EHRs

2.4.1 Specific search methodology

For this section a broad literature search was conducted to identify the current methods and guidelines for assessing data quality in research studies that use routinely-collected

healthcare data. All articles which described ontologies/frameworks, guidelines/toolkits/software, or implementation/practice were included. Any articles that focussed on improving data quality at the point of data entry were excluded, as this would normally not be within the control of the individual researcher working with EHRs.

PubMed (title/abstract) and Embase (title/abstract/keyword) were searched for terms relating to "**data quality**" AND "**EHRs**" AND "**research**" for all articles up to 19th June 2019. Other healthcare-related data sources such as disease registries and biomedical data were included within the umbrella term of "EHRs" as they may have similar data quality challenges for secondary use. Clinical trials were not included as their data is generally collected as primary data. A list of synonyms for each of the three umbrella terms was constructed in collaboration with a librarian, and can be found in Appendix A.1.

2.4.2 Results

The search returned a total of 2209 results, see Figure 2-1. A title and abstract review reduced these further to 233 results, and these underwent full text screening. References within articles were followed manually where relevant, which led to the further inclusion of 3 articles. This resulted in a final collection of 6 articles describing ontologies/frameworks, 15 describing guidelines/toolkits/software, and 38 relating to implementation/practice.

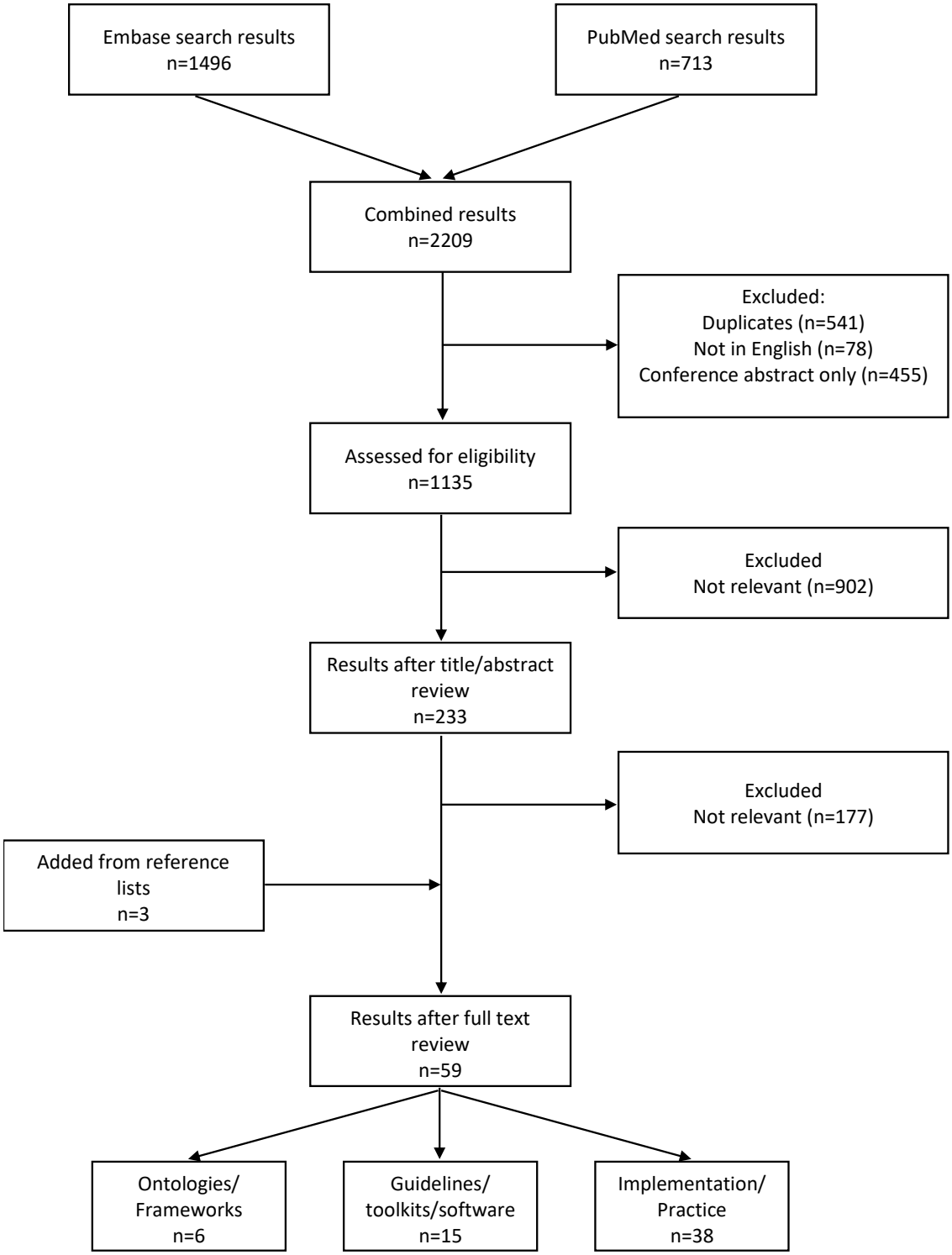


Figure 2-1. Selection procedure for articles relating to methods and expectations for assessing data quality in EHRs for research

2.4.2.1 The major categories of data quality

The literature search identified six ontologies/frameworks that have been created for assessing data quality in EHRs for secondary uses such as research[4 9-13]. These were variously developed using literature reviews, interviews with stakeholders, or as modifications of existing frameworks. While the exact terminology varied, and there were some differences in meaning, the overall elements of each of these frameworks could broadly be distributed into 6 different categories (Table 2-1).

Table 2-1. Allocation of elements from published data quality frameworks for EHRs into 6 common categories, ordered by date of publication

Framework	Completeness	Consistency	Correctness	Plausibility	Timeliness	Usability
Smith 2017[12]	Completeness*	Internal validity	Accuracy	External validity	Timeliness	Interpretability
Kahn 2016[4]	Completeness	Conformance		Plausibility		
Johnson 2015[9]	Completeness	Consistency	Correctness		Currency	
Weiskopf 2013[11]	Completeness	Concordance	Correctness	Plausibility	Currency	
Liaw 2013[13]	Completeness	Consistency	Correctness Security†		Timeliness	Usability Relevance
Kahn 2012[10]	Appropriate amount of data		Accuracy Objectivity	Believability	Timeliness	

* Completeness was not listed as a distinct element but was included as a subcategory of Accuracy

† Security was used here to mean “lack of data corruption”

The only category that appeared in all the different frameworks was Completeness. This was typically defined in terms of the occurrence of missing values in the data, though Weiskopf et al.[11] defined Completeness as whether or not the EHR contained all truths about a patient.

The widest category (and most open to interpretation) was that of Consistency, which ranged from concepts such as strict conformance to pre-defined technical specifications (Kahn et al. 2016[4]), to uniformity of values (Liaw et al.[13]), to agreement between different but related values (Weiskopf et al.[11], Smith et al.[12], Johnson et al.[9]).

Surprisingly, Correctness did not feature in one case, with Kahn et al. 2016[4] excluding it deliberately on the basis that “terms such as accuracy, validity, or correctness” are “used in widely differing ways across the [data quality] literature”. Instead, they selected “terms that captured our intended concept that represents the trueness of data values”, and concluding that Consistency, Correctness and Plausibility can be used as proxies for each other.

The least common category was Usability, with only two authors including it in their final framework. The further category of Redundancy[14] (also known as Duplicity[15]) was identified by two non-EHR-specific but related frameworks, though this could reasonably be included under the category of Plausibility.

2.4.2.1.1 Categories of intrinsic data quality

A particular point of interest was that Kahn et al. 2016[4] focussed exclusively on *intrinsic* features of the data, i.e. features that do not depend on any external requirements, whereas all other authors included contextual factors such as Timeliness (i.e. how up-to-date the data is). Since EHRs potentially cover all conditions and all patients (within a particular region), and different researchers will have different needs depending on their specific domain and study designs, focussing on intrinsic data quality allows any elements to be applied more generally. Intrinsic data quality was categorised by Kahn et al. 2016[4] in terms of:

- **Completeness** (i.e. features that describe the frequencies of data attributes present in a dataset),
- **Conformance** (i.e. the compliance of the representation of data against internal or external formatting, relational, or computational definitions), and
- **Plausibility** (i.e. the believability or truthfulness of data values).

2.4.2.2 Current guidelines and tools for assessing data quality problems

In addition to theoretical frameworks, the literature review also aimed to identify practical guidelines and tools, i.e. how to proceed with identifying and assessing data quality problems in a particular dataset. For a guideline to be useful it ideally needs to cover both

what needs to be done as well as *how* to do it. This is because without guidance on *how* to perform the checks, ad-hoc methods will prevail, leading to a lack of consistency and rigour.

2.4.2.2.1 Guidelines for researchers

In addition to the frameworks described in the previous section, four guidelines[6 16-18] were identified which were aimed specifically at researchers using EHRs. However, three of these were focussed on how to *report* what was done in a study, rather than how to *conduct* it in the first place, and only one of these three[16] included any recommendations or examples of actual checks to perform. The most novel guideline was by Weiskopf et al. (2017)[18], who developed a 2-dimensional formulation, with three data quality constructs (Complete, Correct, Current) along one dimension, and three data dimensions (Patients, Variables, Time) along another dimension. This provided a simple yet fuller framework for performing data quality assessments on EHRs than the more typical one-dimensional frameworks.

Expanding the search to include wider healthcare data, the STRATOS[7] (STRengthening Analytical Thinking for Observational Studies) initiative set up in 2011 proposes to fill this gap in guidance for *conducting* studies, with its Initial Data Analysis (IDA) subgroup responsible for making recommendations regarding data quality, including checking it for consistency and accuracy. The only substantial output so far has been for studies using primary data collection, where they developed a conceptual framework for initial data analysis,[19] however much of the framework is still applicable to studies using EHR data. The framework includes ‘Data cleaning’ as the process of “identifying and correcting data errors”; ‘Data screening’ as the process of “reviewing and documenting the properties and quality of the data that may affect future analysis and interpretation”; ‘Initial data reporting’ as the process of “informing all potential collaborators about all relevant insights obtained from the previous steps”; and ‘Reporting IDA in research papers’. Additionally, its ‘Metadata setup’ step can be made applicable to EHRs by revising it from “systematically setting up all background information” (e.g. data dictionaries and data collection methods) to “systematically compiling all background information”.

2.4.2.2.2 Guidelines for data providers

The WHO produced a toolkit in 2017 for assessing data quality in health facilities[20], and numerous guidelines exist for disease registries[21-25], but these are more focussed on assessing and maintaining data quality as an ongoing exercise, and as an institution, rather than specifically for secondary use of the data in research. These have therefore not been considered further.

2.4.2.2.3 Publicly available data quality tools

Two open source data quality tools were identified in the literature search. Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems (ACHILLES)[26 27] produces descriptive statistics and interactive visualisations to inform data quality assessments at an institutional level. It requires data to be stored in a database that conforms to their common data model but provides a user-friendly web interface for visualisations such as pie and bar charts.

Data Quality Explorer (DQ^e) is a suite of interactive, database-agnostic tools for assessing data quality, so far with 2 components: completeness (DQ^e-c)[28], and variability across site location and time (DQ^e-v)[29]. DQ^e-c can operate on more than one database model (unlike ACHILLES), and produces an HTML report that contains summary statistics (frequencies of rows, unique values, missingness, and percent missingness for each column and table) and simple graphs of counts and proportions. DQ^e-v is suitable for any database model (as it works with text files extracted from the source database), however the data extracts still need to conform to a pre-specified data model (with 6 items), and pre-aggregated by site, time unit, and condition(s) of interest. It then produces interactive box and scatter plots, density plots and regression-based analysis (for smoothing and outlier identification). This means it is good for user-friendly visualisations but it still requires some manual effort to create extracts in the appropriate format as well as to decide what conditions to check.

Commercial data quality assessment tools may also exist, but none were found in the literature search. However, they would typically be aimed at business users, and produced

using opaque and proprietary methods, which do not support the aims of transparency and reproducibility in the scientific community. They have therefore not been considered further.

2.4.2.3 Types of checks used to assess data quality

All articles that mentioned conducting checks to assess *intrinsic* data quality (as defined by Kahn et al. 2016[4]), namely Completeness, Conformance and Plausibility, were included and were separated into 3 categories: guidelines (including frameworks discussed earlier), research studies, and descriptions of institutional databases. For guidelines, papers that related to EHRs and general observational research were included. For research studies and institutions, only those that related to EHRs were included. A publication from the Canadian Institute for Health Information (CIHI), and documentation regarding data quality in Hospital Episode Statistics (HES) and Secondary Uses Service (SUS) published by the Health and Social Care Information Centre (now known as NHS Digital) were also included, despite not appearing in the literature search.

Out of 38 publications that mentioned checking for intrinsic data quality in some form, 33 mentioned checking for Completeness, 29 mentioned checking for Conformance, and 27 for Plausibility (Table 2-2). The most common check for Completeness was to measure the percentage of missing values per field (n=12), followed by measuring the percentage of missing values per time unit or event (n=7). The most common check for Conformance was to check that values belonged to a predefined set of codes e.g. Male/Female (n=15), followed closely by checking that values conformed to a particular format e.g. date (n=14) or fell within a particular domain e.g. maximum age (n=12). The most common check for Plausibility was that there were no duplicate values (n=12) followed by a check for outliers (n=8).

Table 2-2. Suggested checks to assess intrinsic data quality categories, ordered by most commonly mentioned

Type of check	**Guidelines (EHRs) n=8**	**Guidelines (general) n=2**	**Research studies (EHRs) n=19**	**Institutional databases (EHRs) n=9**	**Total n=38**
COMPLETENESS (n=33)					
Missingness per data field	Kahn 15[16] Liaw 13[13] Kahn 12[10]		Hoeven 17[30] Johnson 16[31] Newton-Dame 16[32] Taggart 15[33] Dugas 01[34]	Sengupta 19[35] Smith 17[12] Knowlton 17[36] HES 16[37]	12
Missingness per time unit or event	Weiskopf 17[18] Kahn 16[4]		Johnson 16[31] Taggart 15[33]	Sengupta 19[35] Smith 17[12] Walker 14[38]	7
Missingness over entire dataset		Rajan 19[14]	Johnson 16[31] Dugas 01[34]		3
No. of values present/expected no. of values* (e.g. where the presence of a value is conditional on other data (such as data fields that are only completed after discharge); or where it is known that there is not complete coverage of all sites)	Kahn 16[4]	Rajan 19[14]	Johnson 16[31]		3
No. of values present/expected no. of values, per time unit or event*	Kahn 16[4]	Rajan 19[14]			2
Default values representing missing* (e.g. a code of 99 used to designate ‘Unknown’)	Kahn 12[10]				1

Type of check	**Guidelines (EHRs) n=8**	**Guidelines (general) n=2**	**Research studies (EHRs) n=19**	**Institutional databases (EHRs) n=9**	**Total n=38**
General (i.e. mentions checking for Completeness, but no specific *intrinsic* checks given)	CIHI 17[39] Lee 17[40] Johnson 15[9]	Huebner 18[19]	Ouedraogo 19[41] Rogers 19[42] Wanyonyi 19[43] Kurniati 19[44] Tran 17[45] Dziadkowiec 16[46] Chan 10 (review)[47] Jones 07[48] Sayer 03[49]	Callahan 17[50] Khare 17[51] Chen 14[52] Liaw 11[53]	17
CONFORMANCE (n=29)					
To codes in a data dictionary* (e.g. Male/Female)	Kahn 16[4] Johnson 15[9] Liaw 13[13] Kahn 12[10]	Rajan 19[14] Huebner 18[19]	Rogers 19[42] Hoeven 17[30] Dziadkowiec 16[46] Johnson 16[31]	Knowlton 17[36] Smith 17[12] HES 16[37] Walker 14[38] Liaw 11[53]	15

Type of check	**Guidelines (EHRs) n=8**	**Guidelines (general) n=2**	**Research studies (EHRs) n=19**	**Institutional databases (EHRs) n=9**	**Total n=38**
To format (e.g. date)	Kahn 16[4] Kahn 15[16] Johnson 15[9] Kahn 12[10]	Rajan 19[14] Huebner 18[19]	Rogers 19[42] Kim 18[54] Hoeven 17[30] Johnson 16[31]	Sengupta 19[35] Smith 17[12] HES 16[37] Liaw 11[53]	14
To domain* (e.g. age greater than zero)	Kahn 16[4] Johnson 15 Kahn 12[10]	Rajan 19[14] Huebner 18[19]	Rogers 19[42] Goldberg 10[55] van Vlymen 05[56]	Smith 17[12] HES 16[37] Walker 14[38] Liaw 11[53]	12
Inconsistencies in data model (e.g. data fields added, removed, or renamed over time; or different data fields supplied by different sites)	Kahn 16[4]		Wanyonyi 19[43] Tran 17[45] Berndt 01[57]		4
To precision* (e.g. to a certain number of decimal places)			Hoeven 17[30] Berndt 01[57]		2
General (i.e. mentions checking for Conformance, but no specific *intrinsic* checks given)	CIHI 17[39] Lee 17[40]		Kurniati 19[44]	Callahan 17[50] Khare 17[51] Chen 14[52]	6

Type of check	Guidelines (EHRs) n=8	Guidelines (general) n=2	Research studies (EHRs) n=19	Institutional databases (EHRs) n=9	Total n=38
PLAUSIBILITY (n=27)					
Duplicates	Kahn 16[4] Kahn 15[16] Kahn 12[10]	Rajan 19[14] Huebner 18[19]	Rogers 19[42] Wanyonyi 19[43] Tran 17[45] Hoeven 17[30] Taggart 15[33] Hodgson 12[58]	HES 16[37]	12
Outliers	CIHI 17[39]	Huebner 18[19]	Ouedraogo 19[41] Wanyonyi 19[43] Dziadkowiec 16[46] van Vlymen 05[56] Berndt 01[57]	Smith 17[12]	8
Smooth trend over time (e.g. sudden jumps in numbers of records may indicate a problem with ascertainment)	Kahn 15[16]		Ouedraogo 19[41] Rogers 19[42] Berndt 01[57]	Smith 17[12] HES 16[37]	6
Unexplained time patterns (e.g. a steep decrease in numbers of admissions when it is known that the population is growing)	Kahn 15[16]	Rajan 19[14]	Hoeven 17[30] Tran 17[45]		4
Gaps between records (e.g. time periods where there are no records)	Kahn 15[16] Kahn 12[10]				2

Type of check	**Guidelines (EHRs) n=8**	**Guidelines (general) n=2**	**Research studies (EHRs) n=19**	**Institutional databases (EHRs) n=9**	**Total n=38**
Gaps in distributions (e.g. different measurement units used)		Huebner 18[19]	van Vlymen 05[56]		2
Dates in the future (e.g. when the record relates to an event in the past)				Knowlton 17[36] Qualls 18[59]	2
New (previously unseen) values (e.g. this may indicate a corruption of data or error in data entry)				HES 16[37]	1
Exact same values consecutively over time (e.g. this may indicate that a value has been carried forward from a previous date instead of being calculated for the new date)				Smith 17[12]	1
General (i.e. mentions checking for Plausibility, but no specific *intrinsic* checks given)	Lee 17[40] Weiskopf 17[18]		Kurniati 19[44]	Sengupta 19[35] Khare 17[51] Chen 14[52] Callahan 17[50]	7

* Checks that may arguably not be truly intrinsic to the data (as they require some element of external information) but that could still reasonably be conducted by someone without any expert domain knowledge (e.g. conformance to data dictionaries that may or may not be supplied as metadata) have been included here

Within these same publications, the most common *methods* suggested for assessing data quality were calculating simple summary statistics such as percentages, counts and averages (n=19)[8 10 14 16 17 28-33 35 36 41 44 47 51 55 57] and visual inspection of graphs (n=12)[8 10 12 30 31 33-35 37 49 55 56]. This is in line with textbook advice for conducting preliminary analyses, which also mainly recommend summary statistics and visual inspection of graphs.[60-62] Three publications suggested performing visual inspection of tables[12 30 37], four suggested fitting statistical models[8 12 16 55], and four mentioned using existing tools (such as ACHILLES and WHO toolkit described earlier) or previously-developed in-house processes (such as SAS macros)[10 35 41 44]. Eight provided no suggestions[4 11 13 15 39 40 50 52] and nine were unclear[36 44 45 47 48 51 54 57 58].

2.5 Identification of temporal change points in EHRs

The most common types of checks found in the earlier general literature search which were specifically aimed at *temporal* data quality issues, were completeness per time unit (n=7) and smooth trend over time (n=6). Less common checks were for unexplained time patterns (n=4), inconsistencies/changes in the data model (such as data fields being added/removed) (n=4), gaps in timestamps between records (n=2), appearance of new (previously unseen) values (n=1), and exact same values consecutively over time (n=1). However, while there is some acknowledgment of the need to make these checks, this may not actually be happening very often in practice; for example, in their review of the data quality checks used by six different EHR data sharing networks[50], Callahan et al. found that only one network checked for temporal completeness. While the already-existing Data Quality Explorer (DQ^e) tool could potentially be used to perform most of these checks, it is designed for interactive use by researchers and so is not directly aligned with this thesis' goals regarding automation and transparency/reproducibility.

2.5.1 Specific search methodology

In order to understand the literature relating specifically to temporal data quality, a second search was conducted, again in PubMed (title/abstract) and Embase (title/abstract/keyword) for terms relating to "**EHRs**" AND "**data quality**" AND "**temporal**" for all articles up to 19th September 2019. As before, other healthcare-related data sources such as disease registries and biomedical data were included under the umbrella term "EHRs" as they may have similar data quality challenges. Again, a list of synonyms for each of the three umbrella terms was constructed in collaboration with a librarian, and can be found in Appendix A.1. Articles were included if they covered either of 2 topics: prevalence of temporal data quality issues (in intrinsic data quality measures); or methods for detecting temporal data quality issues.

2.5.2 Results

The search returned 975 results in total, see Figure 2-2. A title and abstract review reduced these to 64 results, which underwent full text screening, except for two articles (from 1996 and 1993) for which the full text could not be located. References within the articles were followed manually where relevant, which led to the further inclusion of 1 article. This resulted in the final inclusion of 6 articles related to the prevalence of temporal change points and 14 articles related to methods for detecting temporal change points.

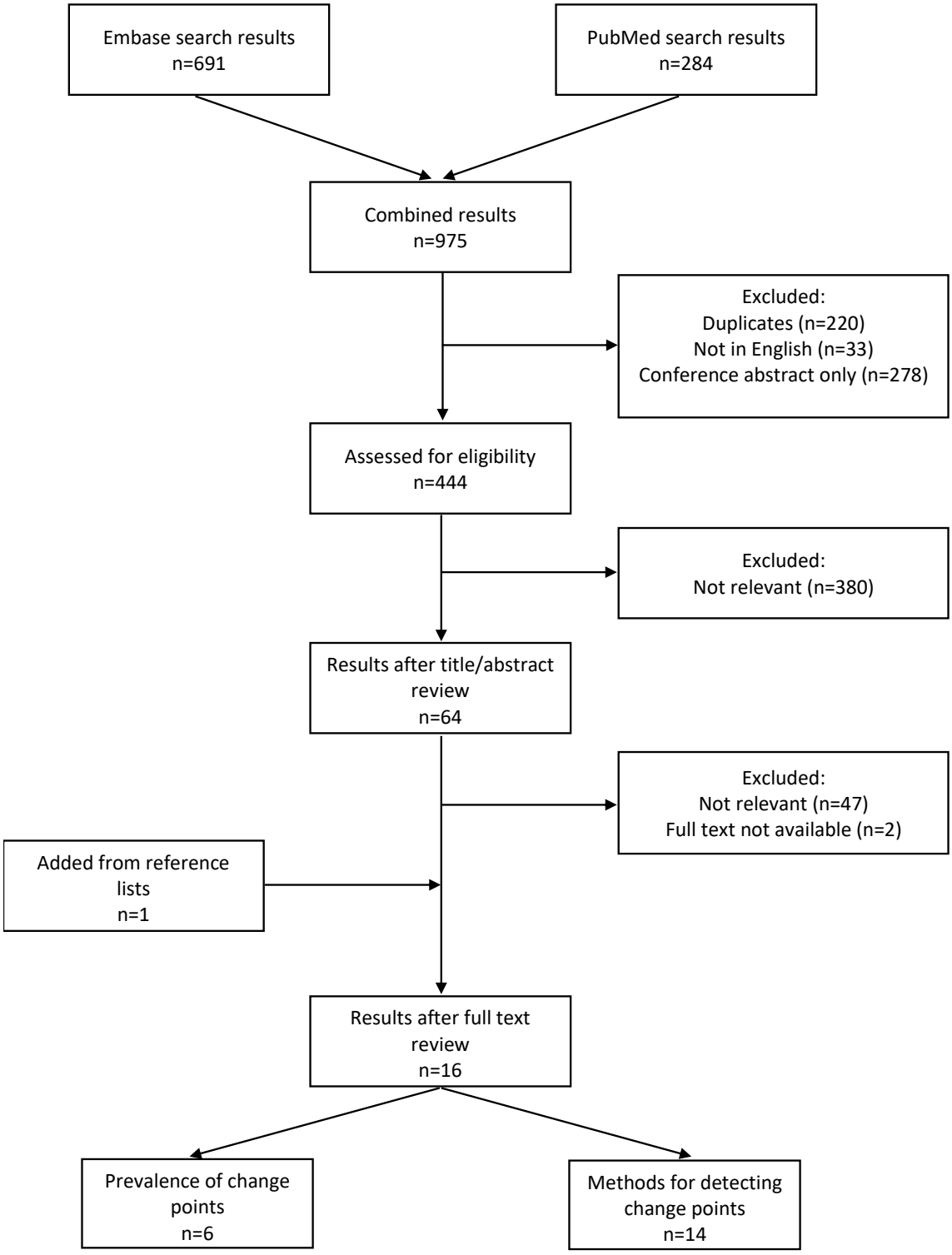


Figure 2-2. Selection procedure for articles relating to temporal data quality in EHRs

2.5.2.1 Prevalence of change points in EHR data

There is very little in the medical literature describing intrinsic data quality over time. Only five articles[63-67] reported on temporal data quality in EHR datasets (as opposed to other healthcare-related data sources), and all concluded that temporal consistency was a factor that should be taken into account when reusing EHR data.

Only one of these articles attempted to quantify the prevalence of change points. Garcia-de-Leon-Chocano et al.[66] reported that 9/46 categorical variables from EHR data related to infant feeding in the perinatal period (in a Spanish hospital between 2009 and 2011), were affected by temporal stability, i.e. where the coded values used in the categorical variables were updated at some point. While these were not described as change points in their article, they would be expected to manifest as sudden change points in the data.

Expanding the search to include disease registries and biomedical data, one further article was found which attempted to quantify the prevalence of change points. This was a study by Looten et al.,[68] who investigated the biochemistry/haematology laboratory test results from a Paris hospital, where they found 49 changes in level and 79 changes in discretization across 192 biological parameters over 17 years.

2.5.2.2 Methods for identifying temporal change points

As mentioned in Section 2.4.2.3, the typical methods suggested in the broad literature search for identifying temporal changes in data quality were visual inspection of graphs, followed by the use of statistical models.

Within the second literature search, 14 articles described the methods they used for investigating temporal data quality (seven based on EHR data, five on biomedical data, and three on registry data), see Table 2-3. Eight used visual inspection of some form of graph over time (whether of counts or percentage differences or more complicated heat maps). Four used process control charts[69] (which originate from the manufacturing industry); three used sophisticated multivariable methods which looked for clusters or temporal variation but

not specifically for change points; and one used a simple test from the WHO toolkit (i.e. that the reported value for the year is within ±33% of the mean value for the preceding three years) which is similar to the idea behind the process control charts. Only one method appeared suitable for automation, since it was designed to detect multiple abrupt change points and could be run without human intervention. This was the use of the *changepoint* package in R on a moving daily median (with a rolling window of 60 days) by Looten et al.[68].

Table 2-3. Methods used to assess temporal data quality in different types of healthcare data sources

Method	**EHRs**	**Biomedical test results**	**Disease registries**
Graphs over time	Ta 2019[65] Kahn 2012[10] Haynes 2011[63] Sayer 2003[49]	Looten 2018[68] Pukkala 2011[22]	Saez 2016[70] Pollock 1995[71]
Process control charts		Ali 2019[72] Yasaka 1979[73]	Saez 2016[70] Giunta 2011[74]
WHO toolkit	Bhattacharya 2019[75]		
Sophisticated multivariable models	Ta 2019[65] Perez-Benito 2019[67]	Saez 2018[76]	
Simple change point model		Looten 2018[68]	

In terms of the aggregation functions used for producing the time-based graphs (either for visual inspection or for process control monitoring), numeric values were typically calculated as simple counts or rates of concepts of interest (such as patients, mortality, prescriptions, medical encounters, and glucose tests)[10 22 49 63 65 68 70 74]. Where the data field being assessed was already numeric (such as for laboratory test results), the actual values[72], mean values[73], moving daily median values[68], or moving daily ratio of last digits[68] were used. Other aggregation functions used were the number of unique concepts[65], and the mean frequency across concepts[65]. Aggregated values were calculated at an annual, monthly or daily granularity.

There was very little discussion of how these aggregation methods were selected, beyond a simple statement explaining their relevance and the types of issues they should be able to reveal. The only article which mentioned comparing the performance of different aggregation methods was by Looten et al.[68] who experimented with rolling windows of 15, 30, or 60 days when creating their daily time series, though they settled on a 60-day window without conducting a formal assessment. However, they still used only a single aggregation function for each type of change point being detected (namely the median value for changes in level, and a ratio of last digits for changes in discretisation), and the resulting time series were generated at a single granularity of one (aggregated) value per day.

2.6 Discussion

Elements of data quality found in frameworks relevant to EHRs were able to be broadly split into six categories: Completeness, Consistency, Correctness, Plausibility, Timeliness, and Usability. However, there were still substantial differences and overlap in meaning between the terms used by different frameworks, which can lead to confusion if it is not made clear which definitions one is using.

Intrinsic data quality, i.e. features that do not depend on any external requirements, is of particular interest for automation, and hence for this doctoral thesis. This is due to its generalisability to different EHR datasets and study designs without need for specific domain knowledge. Intrinsic data quality was categorised by Kahn et al. 2016[4] as Completeness, Conformance, and Plausibility.

Of the six frameworks and four guidelines which discussed intrinsic data quality, only four suggested specific checks to conduct within all the three categories, and three of these four were by the same first author. The most common methods suggested for assessing data quality were calculating simple summary statistics and visual inspection of graphs. One open source tool was found which was aimed at EHR researchers for use during the initial data

analysis stage (DQ^e) (the other tool identified was designed for use at the institutional level, i.e. could only be used on data stored in a database conforming to a particular data model).

In terms of articles describing what EHR researchers do in practice, out of 19 articles that discussed some aspect of intrinsic data quality, 14 considered Completeness, 12 Conformance, and 12 Plausibility, and again the predominant methods used were calculating simple summary statistics and visual inspection of graphs.

Simple checks may be effective enough for traditional research studies where there are a limited number of variables of interest as well as a researcher with appropriate domain knowledge, but with the increasing volume of data being collected in EHRs, and across multiple sites (each with their own idiosyncratic processes), these checks will become more and more onerous and therefore less likely to be conducted thoroughly and consistently. Therefore, automation of checks that would otherwise be labour-intensive and repetitive will be of value, since this would reduce the risk of user error or omission (as well as saving researcher time).

One opportunity for improvement therefore, is in identifying changes in data over time, since if done thoroughly, this would involve the generation and visual inspection of multiple graphs per data field. Therefore, a second literature search was conducted to focus on this area.

Only two articles were found in the literature which described the prevalence of temporal change points in their datasets. Garcia-de-Leon-Chocano et al.[66] found change points in 9/46 categorical variables from EHR data related to infant feeding in a Spanish hospital between 2009 and 2011, while Looten at al.[68] found 49 changes in level and 79 changes in discretization in laboratory test results from a Paris hospital, covering 192 biological parameters over a period of 17 years. If this is at all representative of other hospital systems, there is clearly an imperative to check for change points before any analysis of EHR data, in order to understand any impact they may have, and to make any appropriate adjustments. However, these results are very limited in their scope, covering only two types of variables

from only two datasets, and did not touch on the most recognised data quality category of Completeness. It would therefore be valuable to know if this is also true in other types of EHR data and across other data quality categories, and therefore generalisable.

The same article by Looten at al.[68] was the only article identified which employed a method which appeared highly promising as a way to automate checks for abrupt changes (as it is abrupt changes that would be more likely to represent data quality artefacts rather than real changes in the underlying population). This used the *changepoint* package in R on a moving daily median (with a window of 60 days), though this was applied to biomedical test results rather than to EHR data in general. Other methods such as process control charts could potentially also be used, though these are typically employed to generate alerts for gradual drifts in a mean as well as for abrupt changes, so are not directly suitable for identifying data quality artefacts alone.

There were no articles which analysed the relative performance of different aggregation methods for their effectiveness for visual inspection of change points. Since EHR data contains largely categorical data fields, in order to identify change points by visual inspection of graphs (i.e. the typical method found in the literature for identifying temporal changes in data quality), these data need to be converted to a numeric format by using aggregation functions such as total counts, percentages or averages. It is also necessary to choose an appropriate level of granularity for these aggregations (e.g. by day, month or year), which can further affect the visibility of change points. This is a completely unexplored area and so it will be valuable to start to narrow down the currently open-ended number of options. And while it is clearly impossible to provide an exhaustive list of checks that will cover every eventuality in every research project, it must still be desirable to find a core set of generic checks that can be applied systematically to the majority of projects that use EHR data.

Given the broad question, a scoping review was conducted as opposed to a full systematic review. A scoping review can be appropriate for a number of reasons, including when the goal is to examine the extent, range and nature of research activity, to clarify key

concepts/definitions in the literature, or to identify research gaps in the existing literature[77 78]. In contrast, a systematic review is often more appropriate when the goal is to evaluate the current evidence supporting current practice to inform clinical decision-making.

In terms of limitations, it is possible that there are further relevant articles that were not captured with the chosen search terms. However, the synonyms were selected in collaboration with a librarian, and reference lists were searched, so this risk should have been minimised.

Ultimately, even though a few guidelines do exist regarding the types of data quality checks researchers should conduct at the initial data analysis stage, it is not clear to what extent individual researchers actually do this, as there is little to no reporting of this stage in published papers[8 79]. In the past, limited word counts would likely have influenced this lack of reporting, but with online publishing now being the standard, the use of supplementary files means that there is unlimited space available. Habit and culture are other likely reasons, and since historically publications have not included this information, a researcher would need to make a conscious pro-active choice to include it. However, with the increasing push for transparency and reproducibility in scientific research, it would help researchers to fulfil this if it were made as simple as possible for them to conduct and report on this highly overlooked part of the research process.

2.7 Conclusion and research questions

There is little published literature up to mid-2019 relating to the assessment of data quality in EHRs for research. The most common methods suggested for assessing data quality were calculating simple summary statistics and visual inspection of graphs.

Regarding the identification of temporal change points in EHRs, there were particular gaps in the understanding of their prevalence as well as in efficient methods for identifying them

either visually or automatically. Therefore, the following research questions will be investigated in this thesis:

How often do change points occur in EHRs?

Two articles were found which described the occurrence of (a narrow set of) change points in their study datasets, but no articles were found which measured the prevalence of change points in EHRs in general. Understanding the frequency at which change points occur over different types of EHR data will provide stronger evidence for the need (or not) to conduct temporal data quality checks on EHR data prior to conducting analyses.

Which aggregation methods work best for identifying change points visually in EHRs?

No articles were found which formally compared different aggregation methods for generating time series for visual inspection of change points. Identifying a clear set of methods which have been shown to be effective at revealing change points will help researchers to target the use of their time efficiently.

How well do automated change point detection methods work for identifying change points in EHRs?

Only one change point detection method was found in the literature which appeared suitable for automation, but the change points it identified were simply accepted without any formal validation. There are many change point detection methods that have been developed within other scientific fields, and so testing the accuracy of a range of these methods on EHRs will help to determine whether or not this job could be done effectively without time-consuming human intervention.

CHAPTER 3: Prevalence of change points in EHRs

3.1 Chapter summary

There is very little in the literature regarding the occurrence of change points in EHRs and therefore it is not currently clear to researchers, particularly those who are new to the field, whether or not it is important for them to check for change points in their data in order to avoid bias in their results.

This chapter assesses the prevalence of change points in a range of 8 different EHR data extracts from a large UK hospital group, covering patient, laboratory, and clinician activity. It does this by using the predominant method currently used by researchers for identifying change points, namely visual inspection of graphs.

Using annotations made on a citizen-science crowd-sourcing platform, consensus labels for change points were generated using vote thresholds and density based clustering, and their accuracy compared against labels made by an expert.

A total of 341,800 visual inspections were conducted across 7959 images by >2000 volunteers. Accuracy of crowd-sourced labels identifying whether or not there were *any* change points in an image was very high overall (when compared to expert labels), with sensitivity 94.4% (96% CI 89.9, 96.9), specificity 93.6% (87.3, 96.9), positive predictive value (PPV) 96.0% (91.9, 98.0), and negative predictive value (NPV) 91.1% (84.3, 95.1). Accuracy of crowd-sourced labels identifying the precise *locations* of individual change points was also very good, with sensitivity 80.4% (95% CI 77.1, 83.3), specificity 99.8% (99.7, 99.8), PPV 84.5% (81.4, 87.2) and NPV 99.7% (99.6, 99.7).

The prevalence of change points identified by crowd-sourced visual inspection was incredibly high, with 4477 change points detected across the eight data extracts examined, covering almost every year of data and virtually all the data fields that each extract covered.

3.2 Background

EHR data can cover a range of activities related to patient care. The most common type of EHR data used in research studies are administrative data relating to appointments and admissions (such as arrival and discharge dates), any procedures performed, and diagnoses made. In some cases, further clinical data are also available, such as laboratory test results, medicines administered/prescribed, or clinical observations taken (e.g. blood pressure readings).

There are many different ways in which routinely-collected data can change suddenly over time, whether in rates of missing values, changes in calibration of laboratory values, or updates to dictionaries of coded values to mention just a few. Only two studies were found during the literature review which described the prevalence of temporal change points in their datasets, but these were very limited in scope, covering only two types of variables from only two datasets, and did not touch on the most recognised data quality category of Completeness. It would therefore be valuable to know if these findings are generalisable to other types of EHR data and across other data quality categories.

The most commonly used method for identifying temporal change points is by visual inspection of graphs. While it is fairly straightforward to plot a graph showing how a numeric value (e.g. the neutrophil counts of blood samples sent to a laboratory) varies over time, it is not so straightforward to do this for other types of data fields such as categorical values (e.g. whether a hospital admission was due to an emergency or was elective). Therefore, there is a large amount of data preparation required in order to convert the largely text-based data fields in EHRs into a format that will allow change points to be identified by visual inspection. Furthermore, in order to gain a broad and comprehensive view of the prevalence of change points in EHRs, it will be necessary to inspect large numbers of graphs, and so a scalable method for identifying the locations of change points will also be required.

3.3 Data preparation methods

3.3.1 Data sources

EHR data is collected from all patients attending the four hospitals within the Oxford University Hospitals NHS Foundation Trust (OUH), which provide all acute care and all microbiology and pathology services in the region (~600,000 individuals). Much of this data is automatically fed into a linked database for use in surveillance and service activities within the OUH (which currently comprises over a billion records across ~7000 data fields), and is periodically extracted into a partially-curated, anonymised, research database (~700 data fields), the Infections in Oxfordshire Research Database (IORD). This data goes back to the 1980s and is known to cover multiple periods of change in the hospital computer and laboratory systems.

IORD has Research Ethics Committee and Health Research Authority approval as a generic de-identified electronic research database without individual patient consent (19/SC/0403, 19/CAG/0144).

3.3.1.1 Study sample

Research-ready EHR data is typically structured in table format, with one row (or record) per event (such as an inpatient admission or a blood sample taken), with a range of columns (or data fields) describing the event (such as the date the sample was taken, or the result of the test).

To achieve a sample of EHR data which could be generalised to other data sources, data from all four major component datasets of IORD (patient administration, antibiotic prescribing, haematology/biochemistry laboratory tests, and microbiology tests) were included, which together comprise a total of ~250 data fields and >500 million records. These four components can be seen as representing four different aspects of healthcare activity: patient activity, clinician activity, “quick turnaround” laboratory activity, and “slow turnaround” laboratory activity.

In practice, a data extract created for a particular research study would only contain a fraction of the data contained in the source datasets (e.g. emergency admissions only, or specific laboratory tests only). Since it is not possible to test every possible subset of the data, a large and representative sample was selected as follows.

Within the patient administration dataset, there are separate tables for inpatient, outpatient, and emergency department (ED) episodes, along with further tables for diagnoses, procedures, and ward movements within those patient episodes. All records for inpatient, outpatient, and ED episodes, were included as separate data extracts.

The antibiotic prescribing dataset is relatively small and self-contained and so was included in its entirety. It only includes prescriptions for systemic antibiotics.

Within the haematology/biochemistry laboratory dataset there are thousands of different tests with widely differing result ranges that would not be meaningful to aggregate, and too many to visually inspect separately. Therefore, two of the commonest tests were chosen, i.e. all biochemistry creatinine tests (a common biomarker of renal function which can be affected by infection) and all haematology neutrophil counts (a standard test requested for most patients, also affected by infection), and included as separate data extracts.

Within the microbiology laboratory dataset, data extracts are typically filtered by type of test (e.g. blood culture) or type of organism (e.g. *Esherichia coli*), or a combination of these. While IORD contains the results for many different types of tests (including viral polymerase chain reaction, and microscopy), only the bacterial culture tests have results recorded categorically (as opposed to free text), so these were focussed on. All blood culture tests, and all tests that identified *E. coli* regardless of specimen type, were included as separate data extracts.

One data field from each data extract was chosen to be the ‘timepoint’ field, and this was used to represent the date of the record (patient administration data used the discharge date, laboratory data used the specimen collection date, and antibiotic data used the

prescription date). Data was included across the maximum time span available for each data extract, excluding any leading period where data was sparse, and excluding any records after 30 June 2019 which was the last complete month of data in IORD. A very small number of records which contained a missing or invalid datetime value in the timepoint field were necessarily excluded. Also, all records which were exact duplicates of other records were removed, and the number of removed records stored as a calculated field. In total, the eight data extracts comprised 253 data fields and around 3.7 million records.

3.3.2 Converting EHR data into time series

EHR data contains a variety of often non-numeric data fields, and so needs to be consistently converted into a time series format that can be plotted on graphs for visual inspection. To achieve this, the 'timepoint' field for each data extract (i.e. the data field that represents the date of the record, e.g. the discharge date for an inpatient admission), was first divided into regular intervals, from now on referred to as 'aggregate time points'. Then the records in each data extract were partitioned based on the aggregate time point they belonged to, and numeric values generated for each data field and aggregate time point by applying a range of aggregation functions (such as mean value or percentage of missing values) on each group of values. The set of aggregation functions that could be applied to each data field necessarily depended on its underlying data type, and the aggregation functions that were noted during the literature search, both for generating time series as well as for calculating summary statistics, were considered for relevance.

3.3.2.1 Data field types

Computer software typically categorises data into the following basic types: numeric, character, datetime, or binary (non-human readable). However, by reviewing the data fields present in the chosen data extracts, more conceptually-meaningful categories were able to be specified as follows:

Numeric – These are fields which contain continuous values (such as blood cell counts) or discrete integers (such as the episode number within an admission spell).

Datetime – The values in datetime fields may have been entered manually by a person, or else be automatically-generated timestamps created by the system. They may or may not contain a time element (or may be purely the time element).

UniqueIdentifier – These are identifiers that are computer generated (such as a local hospital number for a patient), and may be based on either a numeric or a character data type, but are conceptually different to fields that relate to real-world attributes of the object being recorded.

Categorical – The majority of data fields in EHRs are categorical, and these may be stored either as character strings or coded as integers.

Freetext – EHR systems often include fields for entering free text, sometimes limited in length, e.g. for patient names, but also sometimes for large amounts of text, such as unstructured comments.

The full list of data fields from the chosen data extracts and their designated data field type are provided in Appendix B.1.

3.3.2.2 Aggregation granularity

Each data extract was partitioned into 'aggregate time points' using the chosen timepoint field (see above) by day (midnight to midnight), as well as by week (Monday to Sunday) and by calendar month. After viewing examples of plots of monthly time series, it was decided that any larger aggregation granularities, such as by year, would remove too much of the information and so were not considered further.

3.3.2.3 Aggregation functions

The following aggregation functions were applied at each aggregate time point, either on the values in a particular data field or across all values in the data extract as a whole, to produce

a collection of time series. These functions were selected to cover the intrinsic data quality dimensions of Completeness, Conformance, and Plausibility (see Chapter 2).

Calculated for each individual data field (excluding the timepoint field as this is being used to generate the aggregate time points, except for where indicated below)**:**

Completeness

- Number of missing values
- Percentage of missing values

Conformance

- Number of non-conformant values (i.e. non-numeric values in a (supposedly) numeric data field, and non-date values in a (supposedly) datetime field)
- Percentage of non-conformant values (i.e. number of non-conformant values in the data field divided by the total number of (missing and non-missing) values in the data field)

Plausibility (depending on the specified data field type)

- All data field types:
    - Number of values present (for the timepoint field this is equivalent to the number of records in the data extract)
- Numeric types:
    - Mean value
    - Median value
    - Minimum value
    - Maximum value
- Categorical types:
    - Number of distinct values

    - Number of values in each category (if number of categories<20), with one time series created per subcategory
    - Percentage of values in each category (if number of categories<20) , with one time series created per subcategory
- Datetime types:
    - Minimum value
    - Maximum value
    - Number of values that contain a (non-midnight) time portion (on the assumption that midnight is used as a default when time is not available) (also calculated for the timepoint field)
    - Percentage of values that contain a (non-midnight) time portion (on the assumption that midnight is used as a default when time is not available) (also calculated for the timepoint field)
- Uniqueidentifier types:
    - Mean string length
    - Minimum string length
    - Maximum string length

Calculated overall for the data extract:

Completeness

- Number of missing values (i.e. the total number of missing values across all the data fields in the extract, excluding the timepoint field as this is being used to generate the aggregate time points)
- Percentage of missing values (i.e. the total number of missing values across all the data fields in the extract divided by the total number of values (both missing and non-missing) in the extract, excluding the timepoint field as above)

Conformance

- Number of non-conformant values (i.e. the total number of non-conformant values across all numeric or datetime fields in the extract (excluding the timepoint field as above))
- Percentage of non-conformant values (i.e. the total number of non-conformant values across all numeric or datetime fields in the extract, divided by the total number of (both missing and non-missing) values in numeric or datetime fields in the extract (excluding the timepoint field as above))

Plausibility

- Number of values present
- Number of duplicate records that were removed
- Percentage of records which were duplicated (i.e. the number of records which had duplicates divided by the total number of records present (after removal of duplicates))

3.3.2.4 Number of time series generated

Applying the above aggregation functions on the 8 data extracts produced 1842 time series at each granularity, resulting in 3*1842=5526 time series altogether.

3.3.3 Visual representation of time series

Each time series was plotted on a separate graph (with time on the x-axis and the aggregation function value on the y-axis), see Figure 3-1 for an example. The axis scales were chosen to show the data points within their full context (i.e. the x-axis covered the full time span), and the y-axis was scaled based on the minimum and maximum aggregation function values. Frequency-based aggregation functions were plotted on a scale always starting at zero and ending no earlier than 10. Percentages were always plotted on a 0-100 scale, and frequencies of subcategories were plotted on the same scale as frequencies for the data field as a whole. The y-axis was deliberately not labelled in order to avoid biasing

the person inspecting the graph. All graphs were saved as png files of the same size, i.e. 1000px wide by 666px tall, at a resolution of 96 dpi.

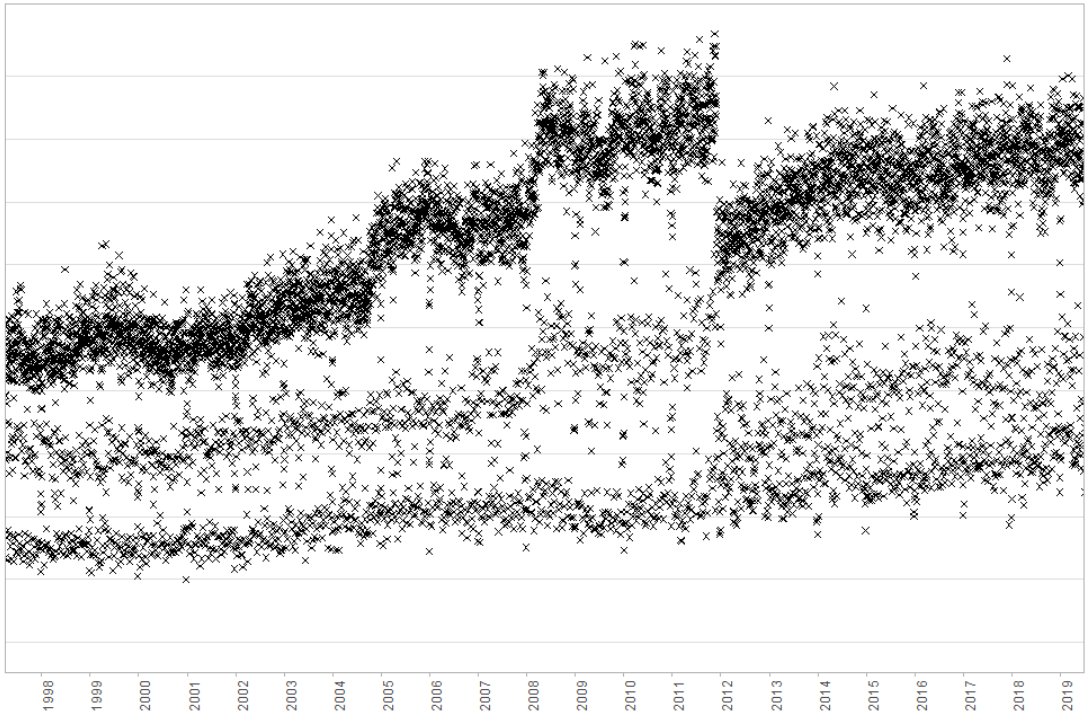


Figure 3-1. Example of a graph generated for visual inspection of change points.
The y-axis is deliberately not labelled in order to reduce the risk of biasing the person inspecting the graph

3.4 Data collection methods for crowd-sourcing of change point labels

Since thousands of images needed to be inspected, and the assessment of whether or not a change point is present is a subjective one, an investigation was conducted to see if it would be possible to crowd-source the creation of these labels using existing online tools.

The Zooniverse (https://www.zooniverse.org) is a free, popular, and well-established online platform for public involvement in research, and has over 2 million registered volunteers who review and participate in multiple projects from astronomy to wildlife surveys to historical transcriptions. It includes a flexible project-building interface with a variety of options for collecting data from volunteers, from multiple-choice questions to image annotation tools.

Volunteers are first shown a project-specific tutorial which explains the task expected of them, before being shown a random series of images for them to classify. Images are "retired" once a pre-specified number of volunteers have classified them.

3.4.1 Pilot study

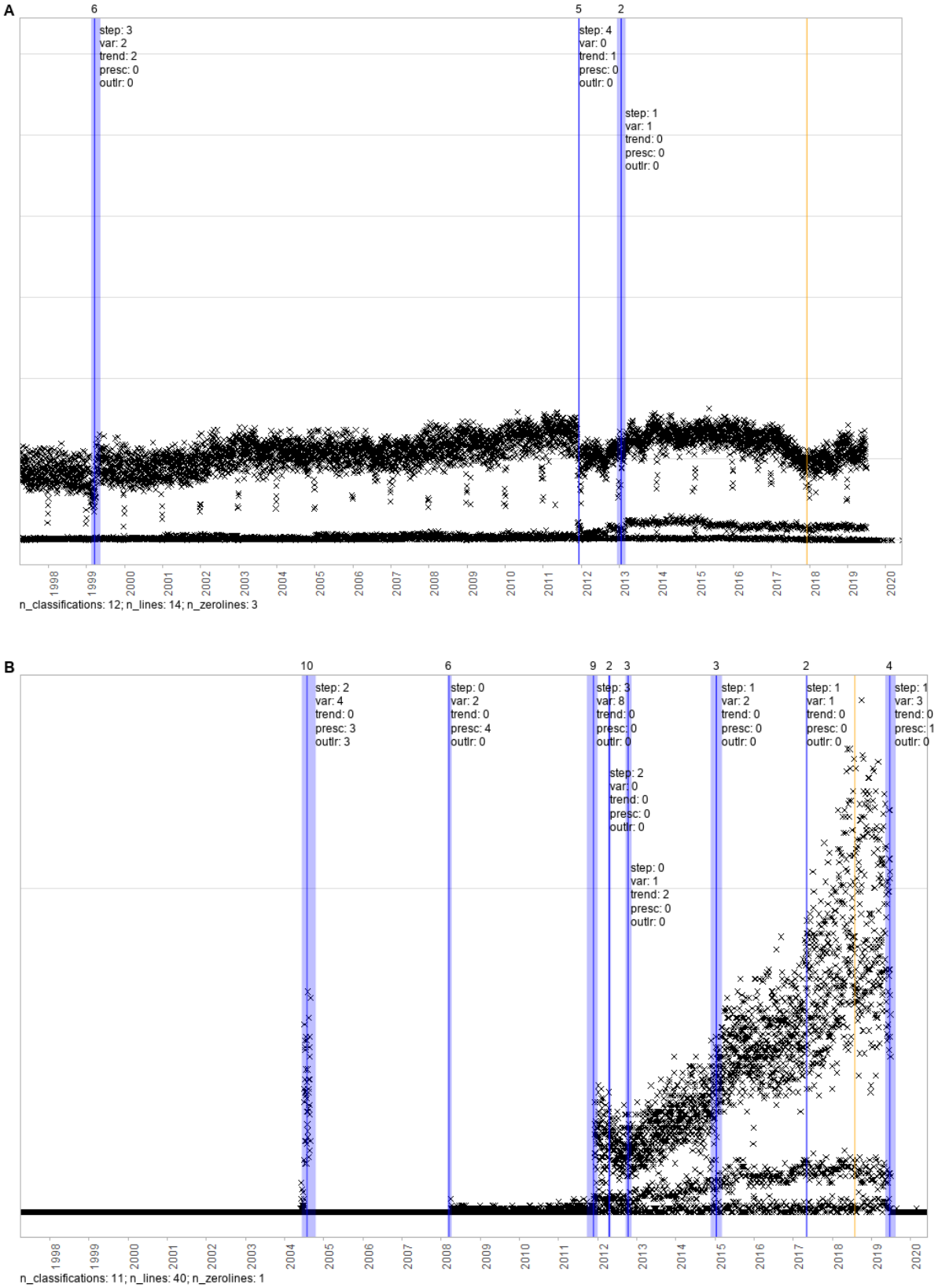
A pilot Zooniverse project was conducted using 210 images of outpatient data to see whether or not the platform could potentially produce useful labels for change points. The pilot workflow asked the volunteers to draw a vertical line wherever they saw a sudden change in the data, and then to (optionally) select one or more values for the "type" of change point they thought it was, i.e. a step change, trend change, change in presence/absence of data points, change in (vertical) variability, or an outlier (note: volunteers were instructed to only draw lines for unexpected outliers and to ignore regularly-repeating outliers).

The pilot lasted for one week and the volunteers consisted of experienced Zooniverse participants who had signed up to review new projects. The 144 volunteers completed 1387 classifications (i.e. visual inspections) across the 210 images, with a maximum of 13 different volunteers classifying a single image.

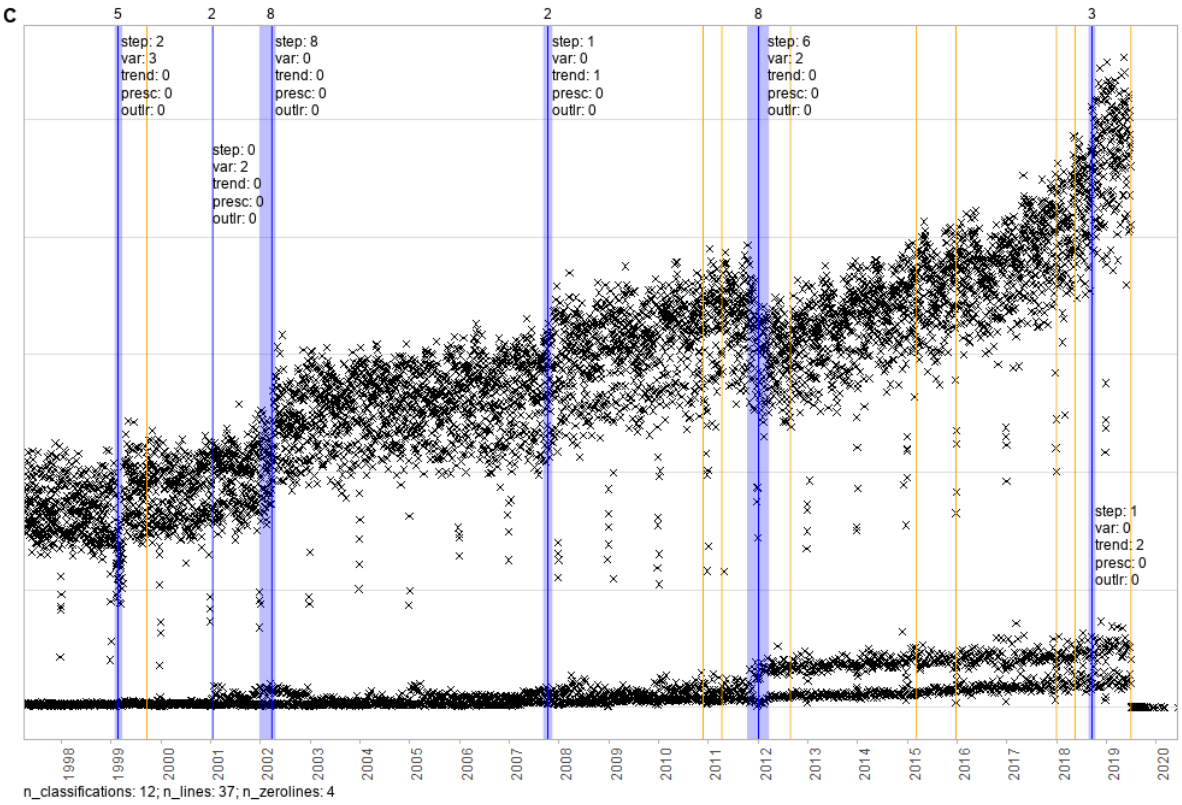
The volunteers also completed 46 feedback forms. Slightly under half of respondents found the task somewhat difficult (5 "Very easy", 20 "Moderately easy", 19 "Somewhat hard", 2 "Very hard"), and about half of respondents said they would take part in future (6 "No", 18 "Not sure", 14 "Yes", 8 "Yes and I'll bring friends").

The locations of any clusters of lines drawn by volunteers were identified using density-based spatial clustering of applications with noise (DBSCAN)[80], which is the method recommended by the Zooniverse team for projects that use the line drawing tool. Following instructions in the R package *dbscan* (v1.1-5), the parameters used to produce clusters were set based on a minimum of 2 lines being drawn within an 11-pixel epsilon-neighbourhood. The resulting clusters for images which had been seen by more than 10 different volunteers

(n=12), alongside the “types” of changes that they had selected for each change point, were plotted and reviewed by eye. A selection of these are shown in Figure 3-2.

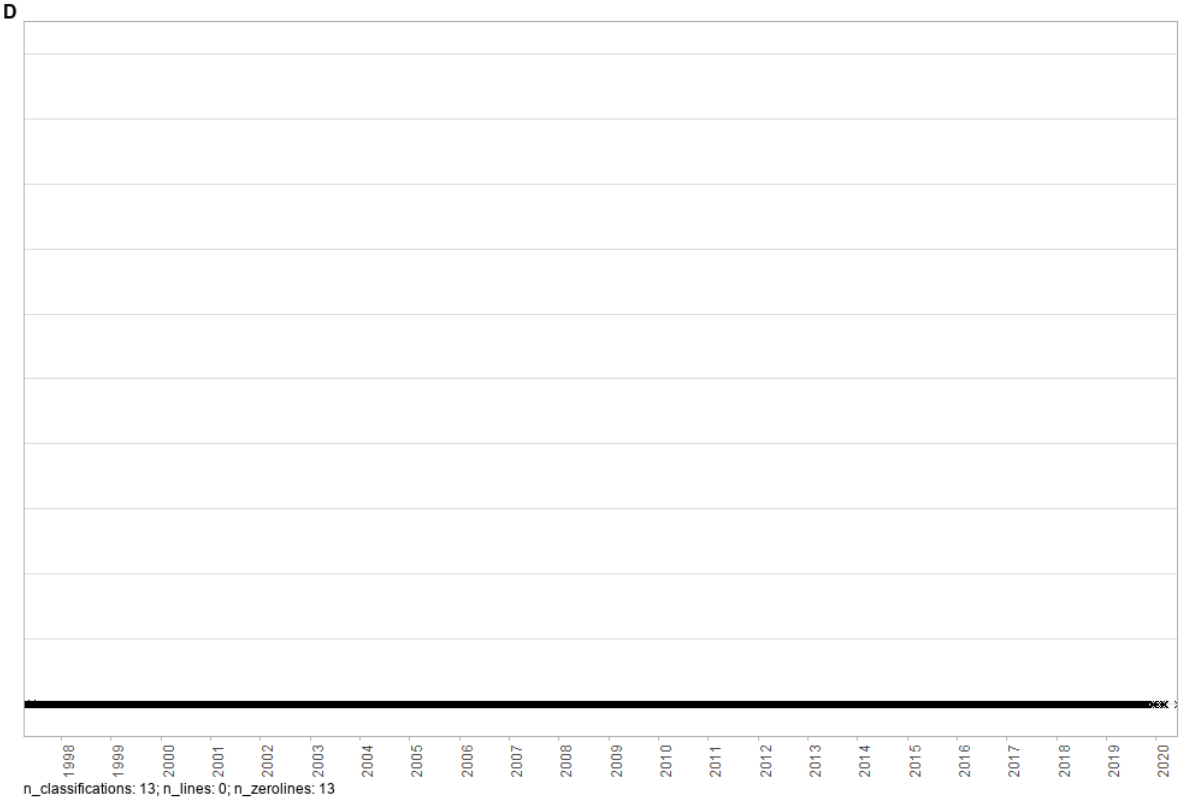


C



n_classifications: 12; n_lines: 37; n_zerolines: 4

D



n_classifications: 13; n_lines: 0; n_zerolines: 13

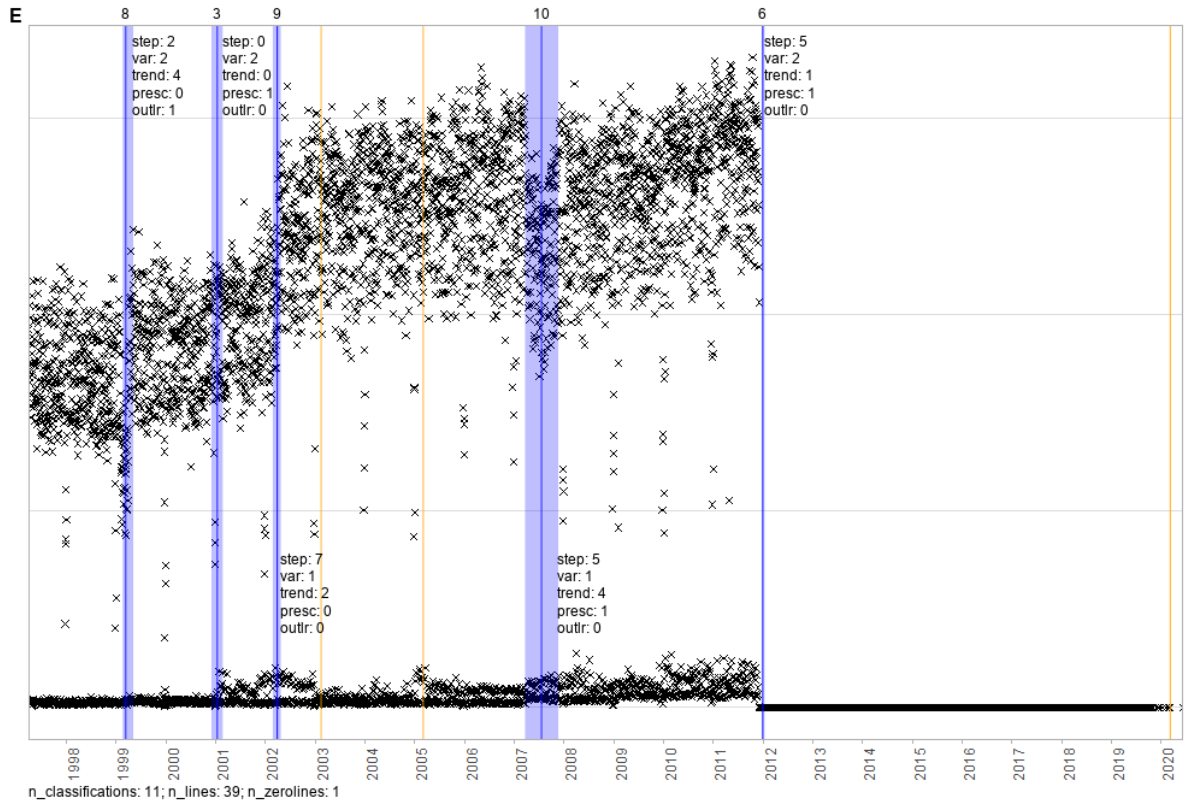


Figure 3-2. A selection of images showing the results from clustering the lines drawn by Zooniverse volunteers during the pilot.
The blue bars represent the width of a cluster, with the mean position in darker blue, and the number of lines contributing to the cluster along the top. Orange lines are volunteers' lines which have not clustered, and hence are considered as noise. Alongside each blue cluster is a summary of the number of times each change point "type" was selected by the volunteers for that cluster, where step = step change, var = change in variability, trend = trend change, presc = change in presence/absence, outlr = outlier. Along the bottom of each image is a summary of the total number of volunteers contributing to the image (n_classifications), the total number of lines drawn (n_lines), and the total number of volunteers who drew no lines at all on the image (n_zerolines).

There were 57 clusters of volunteers' lines, and 43 of these appeared clearly to be genuine change points, while 2 clear change points were missed. Surprisingly, up to 1/3 of classifications had no lines drawn even when the image had obvious change points e.g. in Figure 3-2A, 3/12 volunteers drew no lines, and in Figure 3-2C, 4/12 volunteers drew no lines. Occasionally, nearby but distinct change points were clustered together e.g. Figure 3-2E.

The identification of different types of change was very poor, with little consistency between similar-looking change points. For example, the leftmost cluster in Figure 3-2A/C/E appear very similar except for the y-axis scale, but were given a label of "trend" by 2/6, 0/5 and 4/8 volunteers respectively. Change points were also frequently mislabelled, for example in

Figure 3-2B, the second-from-left cluster has been labelled 4 times as a change in presence/absence when in fact there are data points both to the left and right of it (this is likely due to volunteers missing the fact that the solid black line to the left is a line of data points and not merely an axis line). This task was unlikely to be improved substantially by updating the instructions (since >90% of feedback said instructions were adequate) nor by more classifications (as this would probably lead to as many extra "wrong" classifications as "right" ones).

Three conclusions were able to be inferred from this pilot study. Firstly, that the Zooniverse platform would likely be able to produce adequate labels for the location of change points, but not for the specific "type" of change point. Secondly, that the required number of classifications per image should be increased in order to gain greater confidence in the resulting clusters. And thirdly that the clustering algorithm would need to be examined further to avoid clustering of nearby but distinct change points.

3.4.2 Final Zooniverse project

The "Health Record Hiccups" Zooniverse project showed volunteers one image at a time, and asked them to draw a vertical line on the image wherever they saw an abrupt change in the distribution of values, see Figure 3-3 for a screenshot. They were initially presented with a tutorial which included multiple examples of the types of changes they were being asked to identify – namely changes in level, trend, variability, presence/absence of values, or (unpredictable) outliers – and they could re-access this tutorial, as well as additional examples, in the "field guide" at any time. They were asked to draw a green line wherever they saw a *clear* change, and a yellow line where they were uncertain. If they did not see any change points, they clicked "Done" to be taken to the next image. To reduce risk of bias, no metadata was visible at the point of classification. The full tutorial can be seen in Appendix B.2.

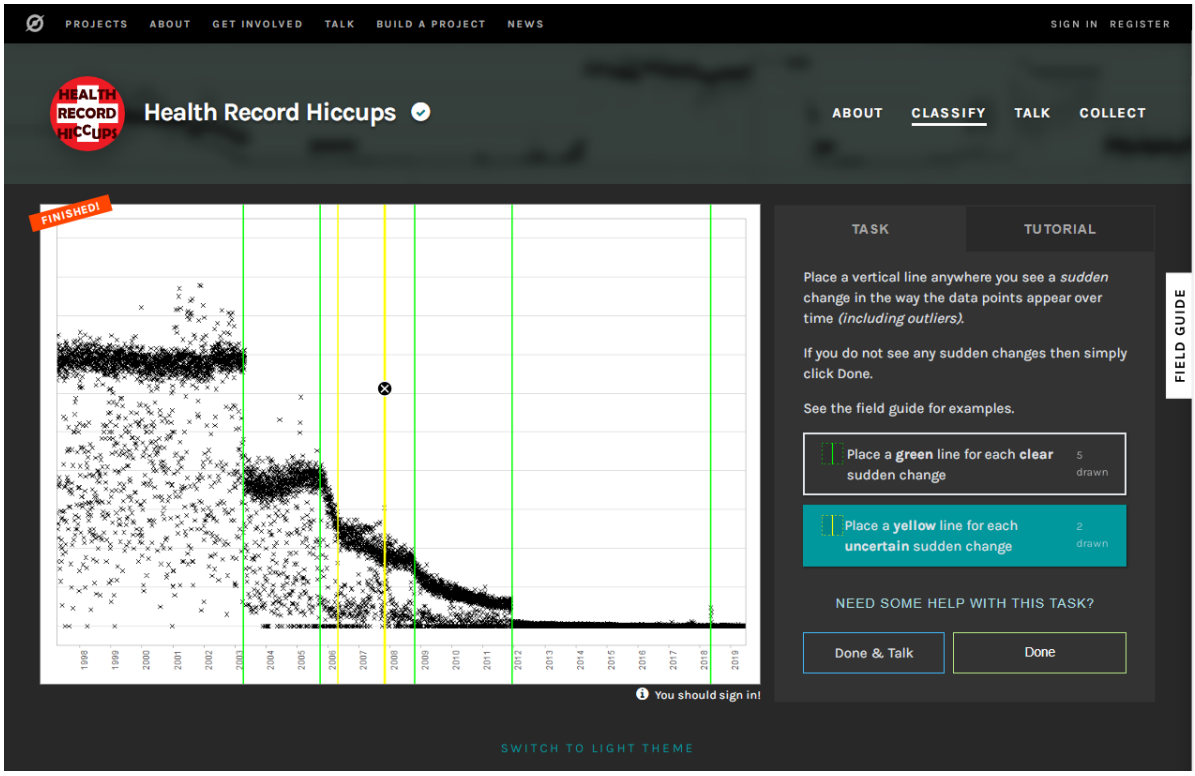


Figure 3-3. Screenshot of Zooniverse project interface

Careful consideration was given to whether volunteers should be asked to label outliers as change points. If each point on the graph were an individual measurement then outliers might be considered to be qualitatively different to change points, but in this case each point on the graph is an aggregation over a period of time, and therefore whether it appears as a single point or as a series of points depends on the granularity of the aggregation period. For example, a whole calendar month of outliers would appear as two change points if data were aggregated by day, but only as a single outlier if aggregated by month. Also, given that hospital activity is seasonal, in particular with patient attendance considerably lower on Christmas day (see Figure 3-2A), volunteers were asked to only label unpredictable outliers.

As part of the project set-up, a classification limit had to be chosen (i.e. the number of times an image should be assessed by the volunteers before it is marked as complete and ”retired” from view). The pilot project had produced some promising results based on a maximum of 13 classifications per image, but it appeared likely that more classifications would be needed in order to be confident (except for trivial images where there are clearly no change points).

A review was conducted of the full list of research papers published using data from the Zooniverse (where links were available on the Zooniverse website in June 2020, n=308), but this did not reveal any methods for calculating a suitable classification limit for projects that use a similar classification interface (or even any other projects that used the vertical-line drawing tool). Therefore, the classification limit was arbitrarily set at 31 to begin with (double the default value used for the pilot, plus one to create an odd number in order to avoid ties when calculating the majority vote) with a view to reviewing this after the project had gone live. If the limit of 31 did not produce a stable set of change point labels then it would be increased. Alternatively, if it was found that fewer classifications produced the same results then it would be reduced (in order to avoid wasting volunteers' time and effort).

3.4.3 Post-launch review

Even if consensus labels can be generated by crowd-sourcing, this does not necessarily mean that the labels that have been produced are "correct", so to assess their accuracy they were validated against labels I produced (based on >8 years' experience compiling and analysing EHR data, denoted "expert" labels below), using the same interface as the volunteers, but blinded to any of their results.

The minimum level of utility for visually-inspected labels is the binary decision of whether or not *any* change points are present in the image. A crowd-sourced sensitivity and specificity of 90% compared to expert labels was targeted, since even experts will disagree. In order to calculate these with a sufficient level of confidence, the number of images to generate expert labels for was determined as follows. Within the pilot data extract, 14/224 (7%) time series were not converted to images because they were constant across the entire time period, and 1/12 (8%) images with >10 classifications contained no change points. Therefore, it was estimated that roughly 15% (i.e. ((224-14)*(1/12)+14)/224) of images would contain no change points (since all time series would be included for the live project, including constant ones). Using a sample size calculation for diagnostic test studies[81], with an expected sensitivity and specificity of 90%, a minimal acceptable lower confidence limit of 70%, and a

probability that the estimated 95% lower confidence limit is above the minimal acceptable value is 0.95, an estimated 41 cases would be required (where a case is an image without any change points and a control is an image with change points – due to symmetry it does not matter if one considers cases to be images *with* change points or images *without* change points). Given an estimated prevalence of 15%, a total sample size of 341 would be required to achieve a high probability (95%) that the sample contains at least 41 cases. See Appendix B.3 for full details of the sample size calculation.

In the end, given the speed and convenience of the Zooniverse interface, the entire first batch of 956 images were classified before downloading any volunteer data. Furthermore, since the volunteers were completing the images so quickly (>4000 classifications the first evening, and 12000 classifications the following day), this preliminary analysis was conducted as quickly as possible, in order to make any necessary adjustments to the interface as soon as possible.

3.4.3.1.1 Reviewing the accuracy of the crowd-sourced labels

Firstly, the sensitivity and specificity of whether or not there were *any* change points present in the image (decided by simple majority vote and only including green lines (i.e. clear changes)) was calculated. This was conducted on the second day after launch, at which point there was an average of 18 classifications per image. There were two ties (i.e. images where there was an equal number of volunteers who classified it as containing change points versus containing no change points), both of which had been classified by the expert as containing change points, so ties were treated as positives. This resulted in a preliminary sensitivity of 88.7% (95% CI 85.6, 91.2) and specificity of 98.9% (97.1, 99.6).

To investigate why the sensitivity was below 90%, a sample of images where there had been disagreement was visually inspected, and the discrepancies were found to be mostly due to volunteers missing outliers or changes that were small in magnitude. In addition, there had been a small number of comments on the discussion boards of the Zooniverse project querying how to deal with outliers, and about the inability to zoom into the image (because

the zoom function did not work correctly at the time). Therefore, additional text was added to the instructions and Help pages to make the need to include outliers more prominent (on day 2), and additional images were generated which did not enforce the restriction on the y-axis to show the data in its full context (i.e. percentages were previously always plotted on a 0-100 scale, and frequencies of subcategories plotted on the same scale as frequencies for the category as a whole).

Secondly (on the fifth day after launch, when all 956 images in the first batch had already been retired), the *locations* of the crowd-sourced change points were estimated for 70% of the images, using density-based clustering of all lines drawn by volunteers. This was done over a grid search of potential parameters (include/exclude yellow lines, minimum-lines-in-cluster between 3 and 20, epsilon-neighbourhood between 2 and 20, see main Analysis methods below for details) and the results compared to the expert labels. In this preliminary analysis, a crowd-sourced label was judged as a true positive if an expert label fell within the range of the cluster, although it was subsequently determined that this would unfairly benefit wide and “loose” clusters, and so the final analysis uses a different method, see below. Sensitivity peaked at around 85%, but only when using wide epsilon-neighbourhood values.

3.4.3.1.2 Reviewing the classification limit

Since the stability of labels for the *locations* of change points is likely to require more volunteer classifications than the simple determination of the existence or not of *any* change points, only the former was considered when reviewing the classification limit. To assess whether or not the chosen limit of 31 was appropriate, crowd-sourced consensus labels were created using only the lines available at each classification increment between 5 and 31, using *dbscan* clustering parameters that were selected to achieve a balance between optimising sensitivity versus creating overly-wide clusters (i.e. include yellow lines, minimum-lines-in-cluster = 5, epsilon-neighbourhood = 8). Since sensitivity compared to expert labels had not quite plateaued after 31 classifications, the classification limit was increased to 41 and used for the remainder of the project.

3.5 Analysis methods

3.5.1 Assessing accuracy of crowd-sourced labels

For the first batch of 956 images (inpatient episodes, antibiotic prescriptions, creatinine tests and blood culture tests, aggregated by day), expert labels were created for the locations of change points using the same interface as the volunteers, but blinded to any of their results. Accuracy of the crowd-sourced labels was assessed against these expert labels. Tuning of the algorithms to create consensus change points from the crowd-sourced data was done using a random sample of 70% of the 956 images, balanced across the 4 data extracts, with the remaining 30% reserved for final testing of the performance of the algorithms.

3.5.1.1 Existence of change points

By considering each image as a subject, a simple consensus decision of whether or not there were *any* change points in an image, was constructed by using simple majority vote, i.e. if >=50% of volunteers drew at least one line on the image, the subject would be classed as positive, otherwise it would be classed as negative.

Further tuning of this algorithm was conducted via a grid search of potential values for the consensus threshold, at 1% intervals, either including or excluding yellow lines from volunteers, and comparing them to *all* lines from the expert. The optimal parameters were selected based on Youden's Index (i.e. treating sensitivity and specificity as equally important).

3.5.1.2 Locations of individual change points

Theoretically there could be a change point at any date in the time series used to produce a graph, but there were only a limited number of pixels (1000) upon which the volunteers could draw lines, and this would also vary depending on each individual volunteers' screen size. In order to guard against lines representing two different but nearby change points from interfering with each other (as in the fourth-from-left cluster in Figure 3-2E), the minimum distance (in pixels) was calculated between lines drawn on the same image by each

volunteer, and plotted to find a generalisable ‘minimum distance cut-off’ between distinct change points (i.e. any lines closer together than this should be assumed to represent the same change point). This ‘minimum distance cut-off’ was also used to estimate the maximum number of change points that could possibly be identified on a single image (i.e. the number of ‘minimum distance cut-off’ intervals on the x-axis).

Nevertheless, it was still possible for a single volunteer to draw multiple lines very close to (or even on top of) each other, so these needed to not be counted as equivalent to multiple different people marking the same location. Therefore, any lines that were drawn by the same person inside a ‘minimum distance cut-off’ interval were combined and replaced with a single line located at the mean position of the contributing lines. If any of the combined lines was green (certain), the resulting line was also considered to be green.

The *dbscan* (Density based clustering with noise, v1.1-5) package in R (v3.6.3) was used to find zero or more clusters of lines within an image, with the mean cluster location assigned to be the crowd-sourced consensus label for the change point. Any lines that were deemed to be “noise” by the package, were ignored. A grid search of three parameters was conducted:

- include or exclude yellow (uncertain) lines,
- minimum-lines-in-cluster (i.e. the minimum no. of lines needed to create a cluster) between 2 and 20,
- epsilon-neighbourhood (i.e. the maximum distance between two lines in a cluster) between 1px and the ‘minimum distance cut-off’.

To calculate the accuracy of the crowd-sourced consensus labels compared to expert labels, a binary classifier was approximated as follows (this time taking each *line* to be a subject, as opposed to each *image* as was done earlier):

- **true positive**: a crowd-sourced label is within the ‘minimum distance cut-off’ of an expert line

- **false positive**: a crowd-sourced label is present, but no expert line lies within the ‘minimum distance cut-off’ of it
- **false negative**: an expert line is present, but no crowd-sourced label lies within the ‘minimum distance cut-off’ of it
- **true negative**: total estimated as the number of ‘minimum distance cut-off’ intervals in an image minus the sum of above 3 categories

In order to avoid double-counting, the following additional rules were enforced:

- when there were ‘x’ crowd-sourced labels close to one expert line, this counted as one true positive and zero false positives
- when there were two expert lines close to one crowd-sourced label, this counted as two true positives and zero false negatives

Since there is no gold standard for measuring the accuracy of crowd-sourced line labels, an alternative, stricter double-counting rule was also implemented, which ensured each label was counted once and only once, to see if this affected the results, i.e.:

- when there were 'x' crowd-sourced clusters close to one expert line, this counted as one true positive and 'x-1' false positives
- when there were two expert lines close to one crowd-sourced cluster, this counted as one true positive and one false negative

Given the imbalanced distribution of positive versus negative calls, Matthew’s Correlation Coefficient[82] (MCC) was used to select the highest performing parameters,

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

3.5.2 Calculating prevalence of change points

The optimal algorithm derived from the initial 956 images was then applied to generate consensus crowd-sourced change points for all 8 data extracts aggregated at the daily granularity (total 1842 images).

Since each data field was represented by multiple images (one for each aggregation function), the results were combined to produce an "overall" classification per data field and per data extract as follows. If *any* of the images representing a data field contained a crowd-sourced change point then the data field itself was counted as positive. If any of the data fields in a data extract was counted as positive, then the data extract itself was counted as positive.

In order to assess how *often* change points occur, each calendar year for each data field in each data extract was assessed for the existence or not of any change points.

3.6 Results

3.6.1 Data collection

The "Health Record Hiccups" Zooniverse project was opened to the public on the afternoon of 2nd September 2020, and was formally announced in an email to the Zooniverse community on 8th September. A total of 7959 images were loaded to the project: 5526 images representing all aggregation functions at all three aggregation granularities for all eight data extracts (1842 at each granularity, including 956 in the first batch representing inpatient episodes, antibiotic prescriptions, creatinine tests and blood culture tests, aggregated by day); followed by a further 2433 images which were repeats generated on a smaller y-axis scale following the post-launch review. The images were scheduled for retirement once 41 classifications had been completed on them. All 7959 images were completed by 23rd October (7.5 weeks), see Figure 3-4, with a total of 341,800 classifications by 2549 volunteers. Of note, it is possible to participate in Zooniverse projects without

logging in to the website, in which case a temporary user ID is assigned, and these temporary IDs cannot be linked across sessions nor to existing users, which results in a likely overestimate of the number of volunteers. The total number of logged-in users was 1895, suggesting a true value somewhere between the two.

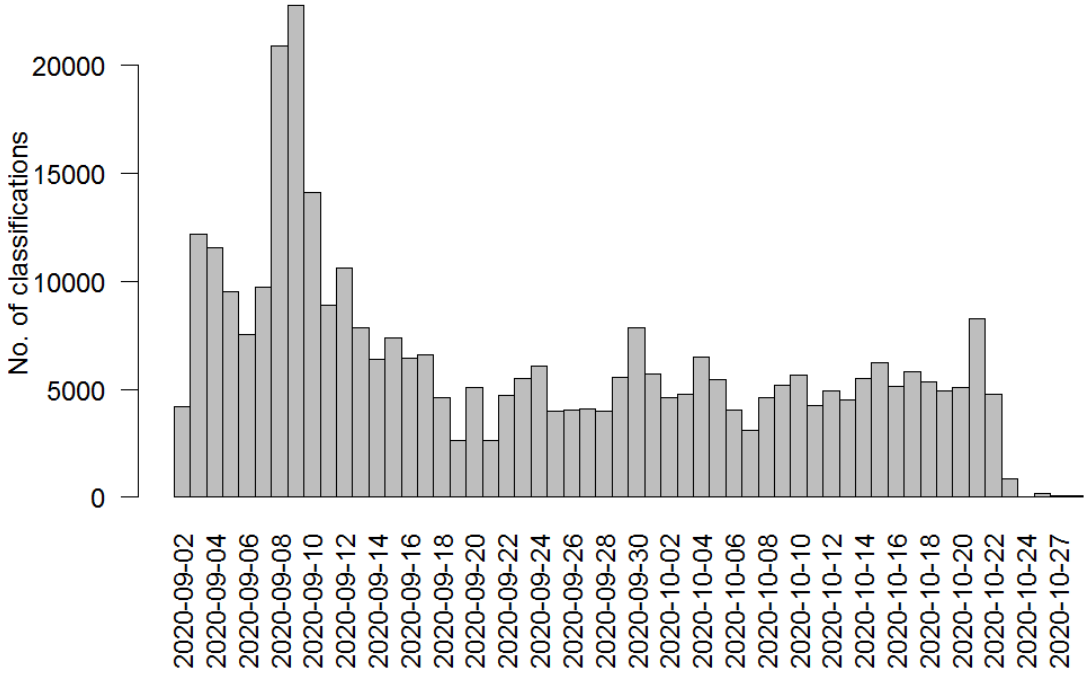


Figure 3-4. Number of classifications completed per day since launch of the Zooniverse project

3.6.2 Data cleaning

Due to the way the Zooniverse platform randomises and supplies images to its volunteers, it was possible for the same person to be served the same image more than once, and for those "repeated" classifications to count towards retirement. Since it was important for each classification to be independent, only the first attempt per person was kept, resulting in the removal of 8268/341,800 classifications. This led to a minimum of 37 volunteer classifications per image.

It was also possible for images to have more than the specified number of 41 classifications, for instance there was one image which had 156 distinct classifications. This was because images continued to be shown to volunteers for classification (albeit with an "already-seen"

or “workflow complete” banner) even when all images had already been completed. In order for the optimisation of the clustering algorithm parameters to be consistent across all images, it was important that each image be represented by a similar number of classifications. The results were therefore restricted to include only classifications from the first 41 people viewing each image, resulting in the removal of 7677 classifications.

This led to a final number of 325,855 distinct classifications across 7959 images.

3.6.3 Accuracy of crowd-sourced labels

Accuracy of crowd-sourced labels was assessed using the initial batch of 956 images (inpatient episodes, antibiotic prescriptions, creatinine tests and blood culture tests, aggregated by day), for which expert labels had also been assigned. For this set of images, a total of 48,533 classifications were completed by at least 543 different volunteers, and after data cleaning, there were 43,502 distinct classifications. 840/956 (88%) images had the full complement of 41 classifications each.

3.6.3.1 Existence of change points

The expert labels determined that 550/956 (58%) images contained at least one clear change point (green lines), and a further 36 (4%) images contained only equivocal evidence of a change point (yellow lines).

Using the naïve classifier of a simple majority vote on all 956 images, and including only green (clear) lines (from both volunteer and expert classifications), the sensitivity of crowd-sourced vs expert-based classification of any change point was 89.1% (95% CI 86.2, 91.4) and specificity of 98.3% (96.5, 99.2). Further including yellow (equivocal) lines (from both volunteer and expert classifications) resulted in a sensitivity of 90.8% (88.2, 92.9) and specificity of 97.8% (95.8, 98.9).

Using the tuning set (random selection of 70% of images) to compare the performance of different consensus thresholds against all lines from the expert, and ranking these based on Youden’s Index, the optimal parameters for an image to be called a positive were to exclude

yellow lines from volunteers, and to apply a threshold of 23% of volunteers drawing at least one green line (rather than 50% in the naïve classifier), see Table 3-1 and Figure 3-5.

Table 3-1. Top 10 parameters for the binary classifier on the tuning set, when including all lines from the expert

Include yellow lines	Threshold (%)	Sensitivity	Specificity	PPV	NPV	Youden's Index
FALSE	23	0.944	0.958	0.972	0.916	0.902
FALSE	24	0.944	0.958	0.972	0.916	0.902
FALSE	20	0.949	0.950	0.968	0.922	0.899
FALSE	21	0.949	0.950	0.968	0.922	0.899
FALSE	22	0.944	0.954	0.970	0.915	0.898
FALSE	18	0.954	0.943	0.963	0.928	0.896
FALSE	19	0.954	0.943	0.963	0.928	0.896
FALSE	25	0.932	0.962	0.974	0.900	0.893
FALSE	26	0.929	0.962	0.974	0.896	0.891
FALSE	30	0.912	0.977	0.984	0.876	0.889

Note: thresholds of 23 and 24 gave the same accuracy. 23 was selected as "the best" since the next "top" thresholds were at lower values. Results presented as proportions.

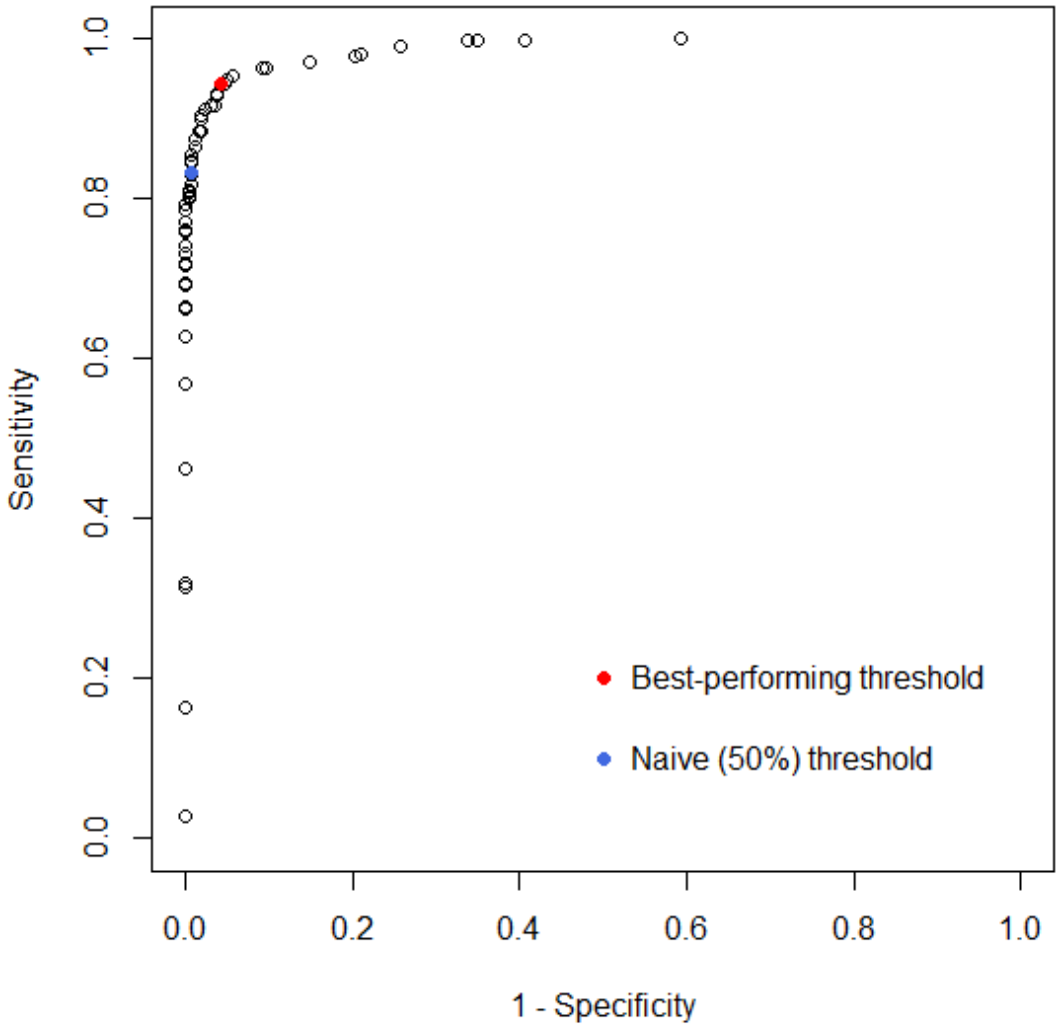


Figure 3-5. Receiver operating characteristic curve for different threshold values on the tuning set, when including all lines from the expert and excluding yellow lines from volunteers.

Applying this optimal threshold to the remaining 30% test set gave a final sensitivity of 94.4% (89.9, 96.9), specificity of 93.6% (87.3, 96.9), positive predictive value (PPV) of 96.0% (91.9, 98.0), and negative predictive value (NPV) of 91.1% (84.3, 95.1) for crowd-sourced vs expert-based classification of any change point. This was from 167 true positives, 102 true negatives, 7 false positives, and 10 false negatives.

Of the false positive images, 6/7 had volunteer lines spread widely across the image, with change points arguably in only 2 of these images. Notably, 5 of the 6 images contained highly discretised values on the y-axis, see Figure 3-6 for an example. The remaining false

positive image contained an outlier that had been missed by the expert. Of the false negative images, 6/10 contained outliers of varying magnitude (see Figure 3-7 for an example), 2 contained change points which were very small in magnitude, and 2 contained only an equivocal change (as assessed by the expert) in trend or variability.

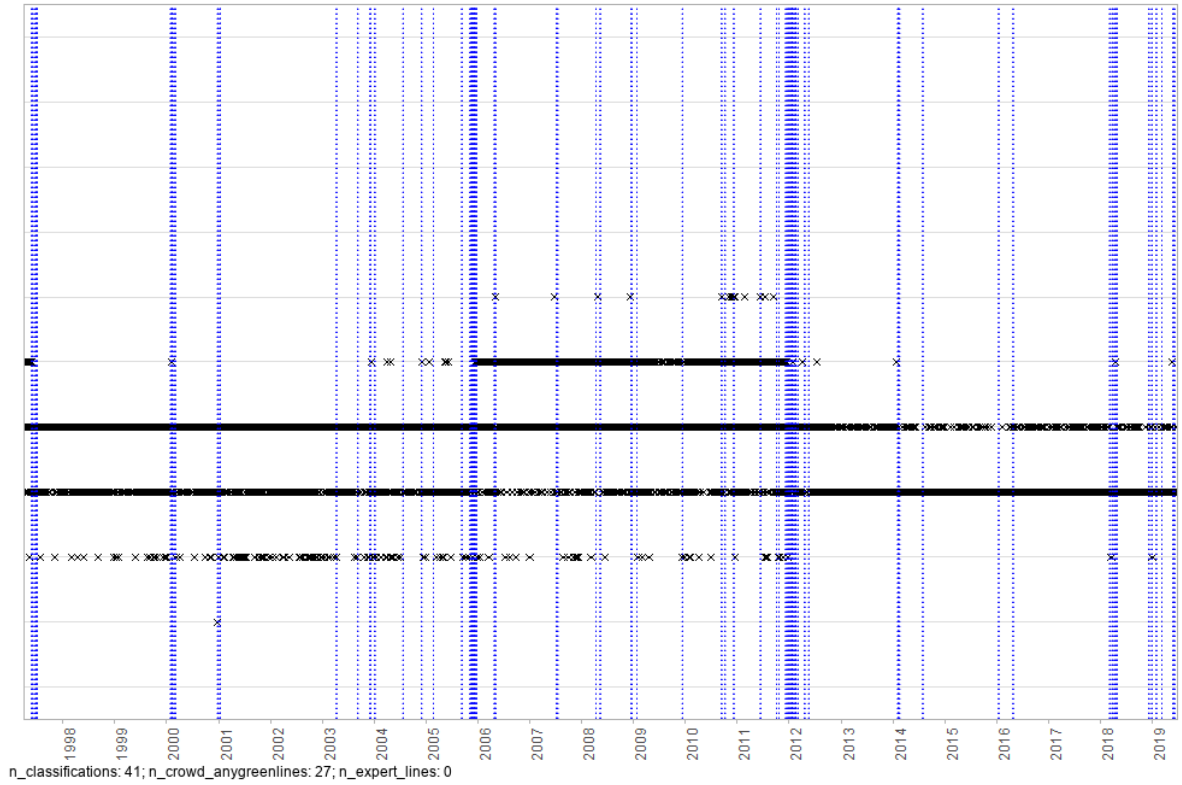


Figure 3-6. Example of a false positive image where the aggregation function values are highly discretised, and with change points arguably around 2006 and 2012 (not identified by the expert). Volunteers' lines are shown as dotted lines.

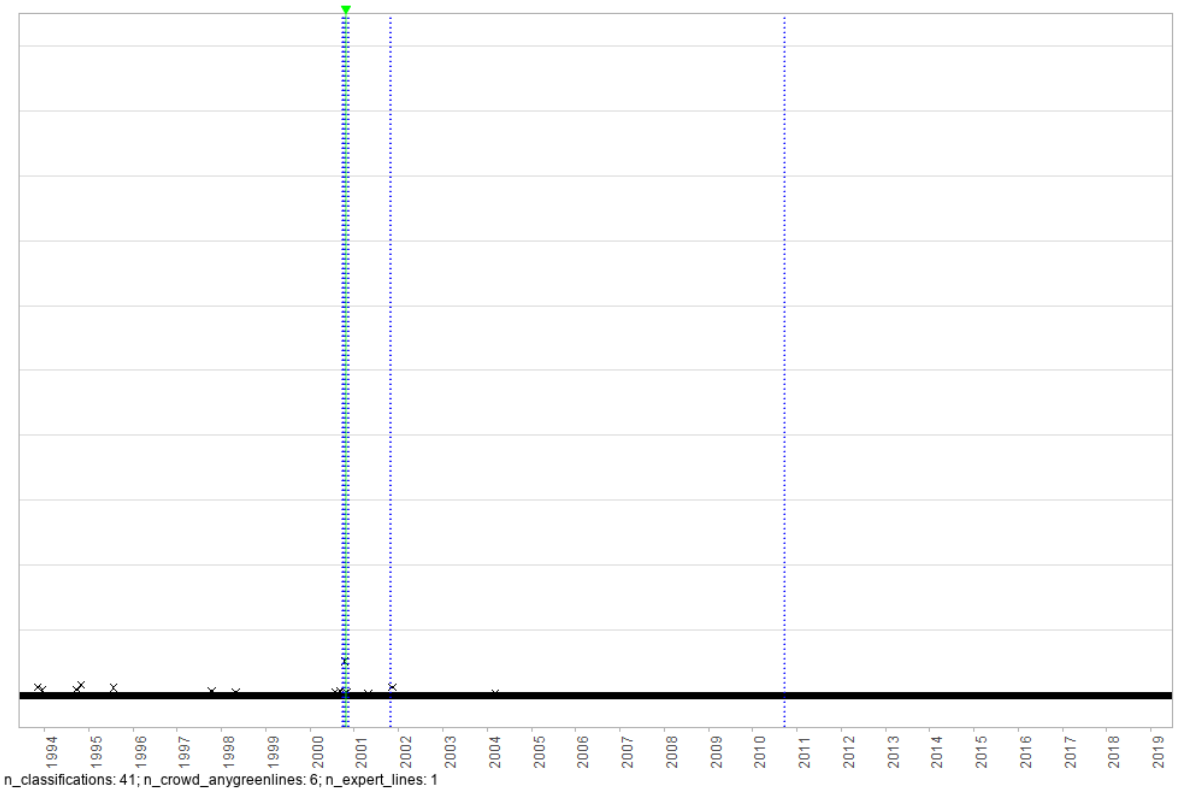


Figure 3-7. Example of a false negative image where the expert identified an outlier but insufficient numbers of volunteers did the same.
Volunteers' lines are shown as dotted lines. Expert lines are shown as a solid line with an inverted triangle on top.

340/956 (36%) images were regenerated using a smaller y-axis scale (in response to volunteer feedback and the false negatives related to small-in-magnitude changes), of which 248 were in the tuning set of 670 images. Substituting the volunteers' results from the rescaled images (but still comparing against the expert labels from the original images), the optimal parameters based on the tuning set changed to *include* the yellow (uncertain) lines from the volunteers, and a vote threshold of 81%. This led to a much poorer final performance on the test set of sensitivity 83.1% (95% CI 76.8, 87.9), specificity 93.6% (87.3, 96.9), PPV 95.5% (90.9, 97.8), NPV 77.3% (69.4, 83.6), from 147 true positives, 102 true negatives, 7 false positives, and 30 false negatives. These rescaled images were therefore not used any further.

3.6.3.2 Locations of individual change points

The expert identified 1992 clear change points plus a further 163 equivocal change points in the initial 956 images.

The minimum distance between two lines drawn on a single image by the same volunteer was below 1px, see Figure 3-8. Since there was a visible threshold at 7px, this was chosen to be the 'minimum distance cut-off' to define two distinct change points (as opposed to one diffusely located change point). This resulted in the removal of 96 (0.1%) volunteer lines (with distance <7px), and for consistency, the removal of 12 (0.6%) expert lines. An example of a 7px distance between two lines is shown in Figure 3-9.

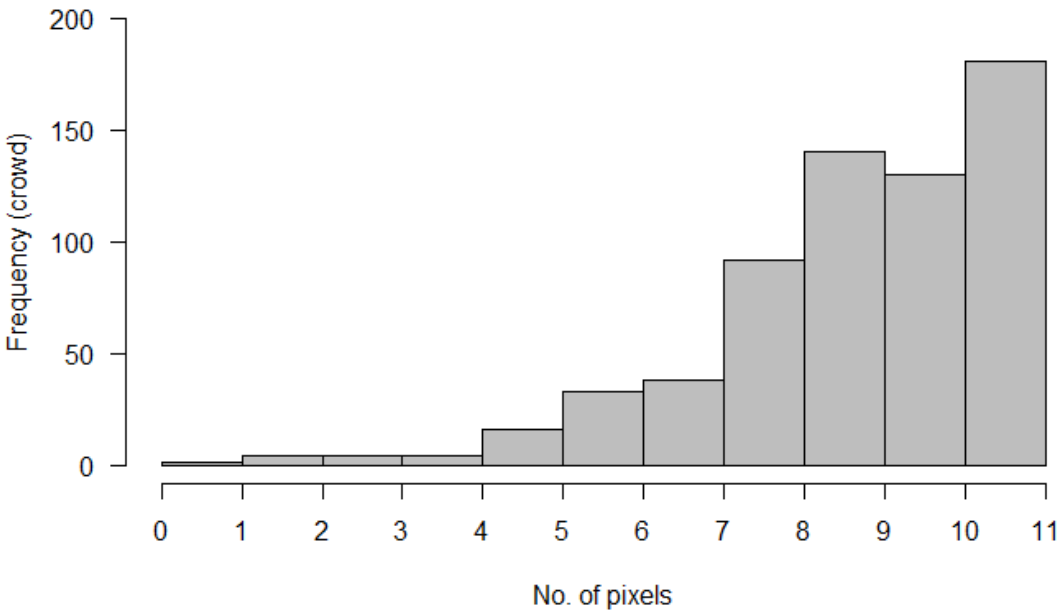


Figure 3-8. Minimum distance between two lines drawn on an image by the same volunteer, shown up to a maximum of 10px.
Intervals are closed on the left and open on the right, i.e. when the minimum distance is an integer, this is included in the bar to the right.

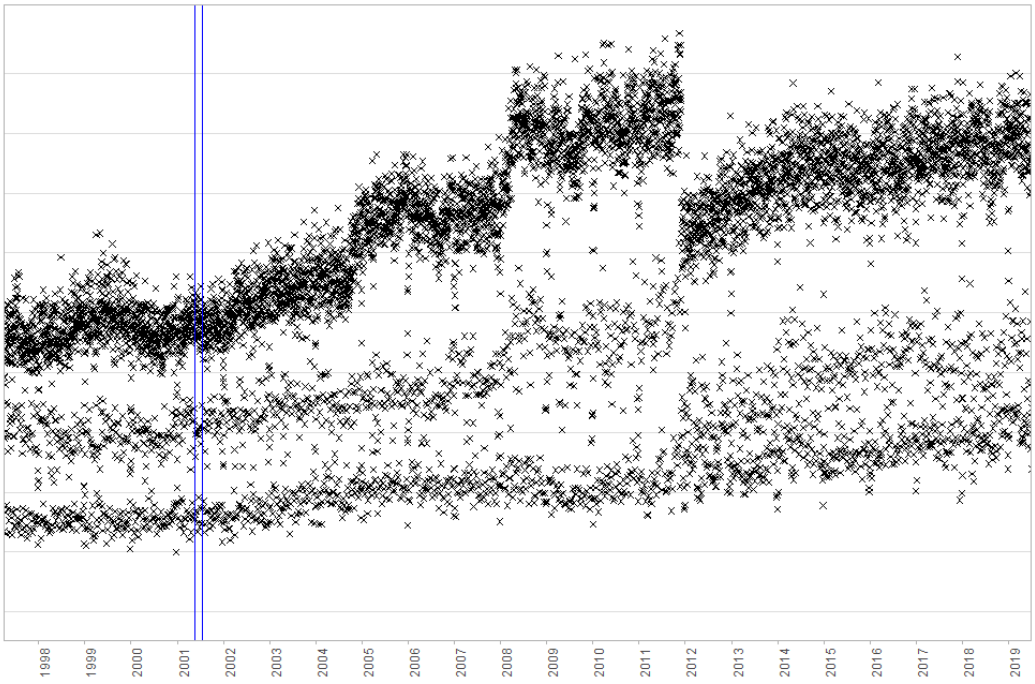


Figure 3-9. Example of two lines drawn 7px apart. Any lines drawn closer together than this were considered to represent the same change point.

Given that each image was 1000px wide, with a left margin of 22px and a right margin of 20px, and assuming that change points could be no closer than 7px from each other, the maximum possible number of change points in each image was calculated to be 136, i.e. floor((1000-22-20)/7).

Based on the MCC and only using the tuning set of images, the optimal parameters to identify the locations of individual change points by density-based clustering were: exclude yellow lines, minimum-lines-in-cluster = 5 and epsilon-neighbourhood = 3, see Table 3-2. Applying the alternative, stricter double-counting rule (where each cluster is counted once and only once) resulted in the same optimal parameters, although it should be noted that several different parameter combinations gave very similar performance.

Table 3-2. Top 10 parameters for the all-round performance of the density-based clustering algorithm based on the tuning set.

Include yellow lines	Minimum no. of lines in cluster	Epsilon distance	Matthews Correlation Coefficient	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value
FALSE	5	3	0.851	0.806	0.999	0.903	0.997
TRUE	7	2	0.850	0.796	0.999	0.913	0.997
TRUE	6	2	0.850	0.826	0.998	0.878	0.997
TRUE	8	3	0.849	0.796	0.999	0.911	0.997
TRUE	7	3	0.848	0.826	0.998	0.875	0.997
FALSE	6	3	0.845	0.769	0.999	0.933	0.996
FALSE	5	2	0.844	0.780	0.999	0.918	0.996
TRUE	6	3	0.844	0.855	0.997	0.838	0.998
FALSE	4	3	0.843	0.843	0.997	0.848	0.997
TRUE	9	3	0.843	0.767	0.999	0.930	0.996

Note: results presented as proportions

Using these parameters on the test set of images resulted in a final sensitivity for *locations* of change points of 80.4% (95% CI 77.1, 83.3), specificity of 99.8% (99.7, 99.8), PPV of 84.5% (81.4, 87.2), NPV of 99.7% (99.6, 99.7), and MCC of 0.822, (for crowd-sourced vs expert-based classification). This was from 492 true positives, 38194 true negatives, 90 false positives (in 42 distinct images), and 120 false negatives (in 70 distinct images).

Of the 120 false negatives, 78 (65%) had been classed as clear change points by the expert, and 42 (35%) as equivocal. In a random sample of 20 images which contained discrepancies (10 which contained at least one false positive and 10 which contained at least one clear false negative), there were 30 false positives and 14 false negatives. Similarly to the discrepancies found when investigating discrepancies in classifications of any vs no change points, 25/30 of the false positives were in images where the aggregation function values were highly discretised. 17/30 could be argued to be change points (13 in variability, 3 in trend, 1 outlier), although were not classified as change points by the expert, and one was in between two nearby (true positive) clusters and so potentially was merely

comprised of border points that could have belonged to either of the nearby clusters. 12 had no explanation beyond the discretisation. Of the 14 false negatives, 7 could be argued to not be change points although were classified as change points by the expert (5 in trend, 1 in variability, 1 outlier), and the other 7 were clear outliers (3 of which were very small in magnitude). See Figure 3-10 to Figure 3-13 for examples.

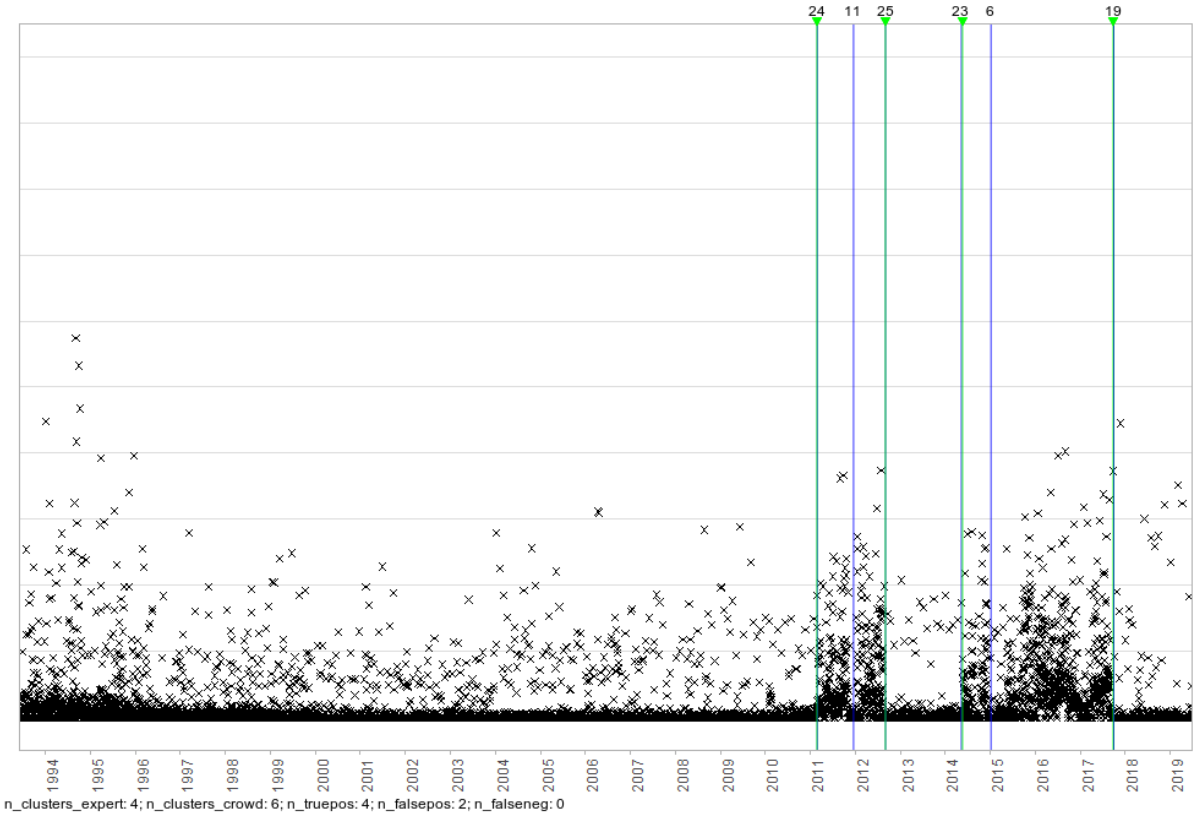


Figure 3-10. Examples of false positive clusters identified by the volunteers (blue lines only). The two at 2012 and 2015 could arguably be changes in variability although were not identified as such by the expert.
Vertical lines denote positions of volunteer clusters and expert labels; blue lines with numbers above indicate the number of volunteers contributing to the cluster, green lines with inverted triangles indicate lines drawn by the expert.

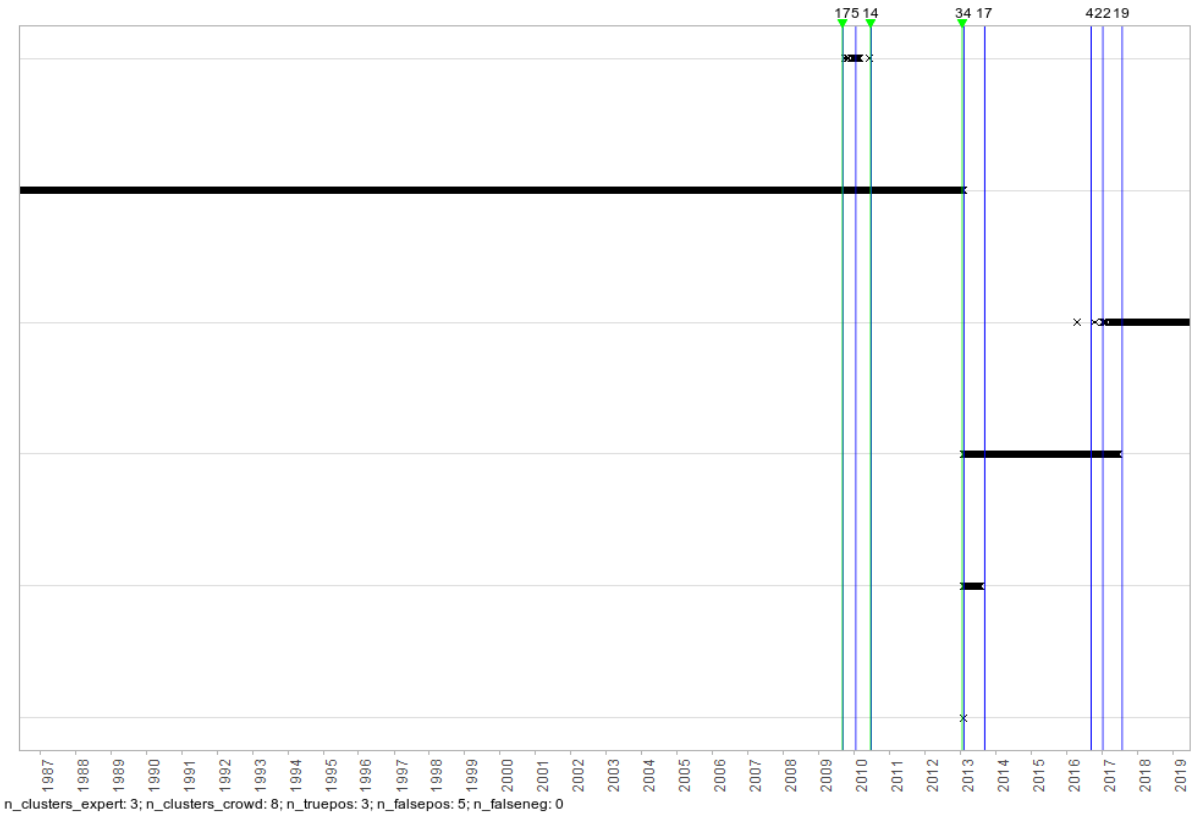


Figure 3-11. Examples of false positive clusters identified by the volunteers (blue lines only). The one at 2010 is potentially just comprised of border points for the two adjacent clusters, while the 4 on the far right are likely only related to discretisation.
Vertical lines denote positions of volunteer clusters and expert labels; blue lines with numbers above indicate the number of volunteers contributing to the cluster, green lines with inverted triangles indicate lines drawn by the expert.

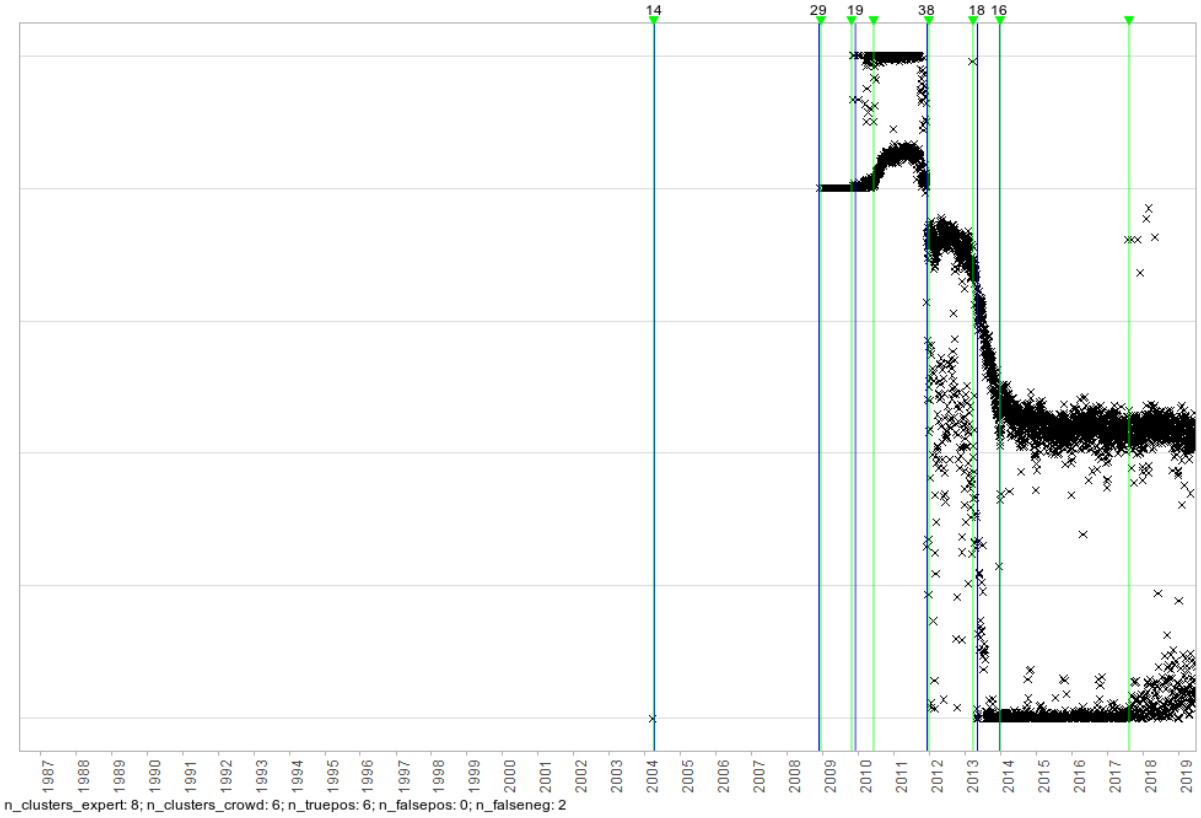


Figure 3-12. Examples of false negative clusters not identified by the volunteers (green lines only). The two in 2010 and 2017 identified by the expert could arguably be changes in trend or variability.
Vertical lines denote positions of volunteer clusters and expert labels; blue lines with numbers above indicate the number of volunteers contributing to the cluster, green lines with inverted triangles indicate lines drawn by the expert.

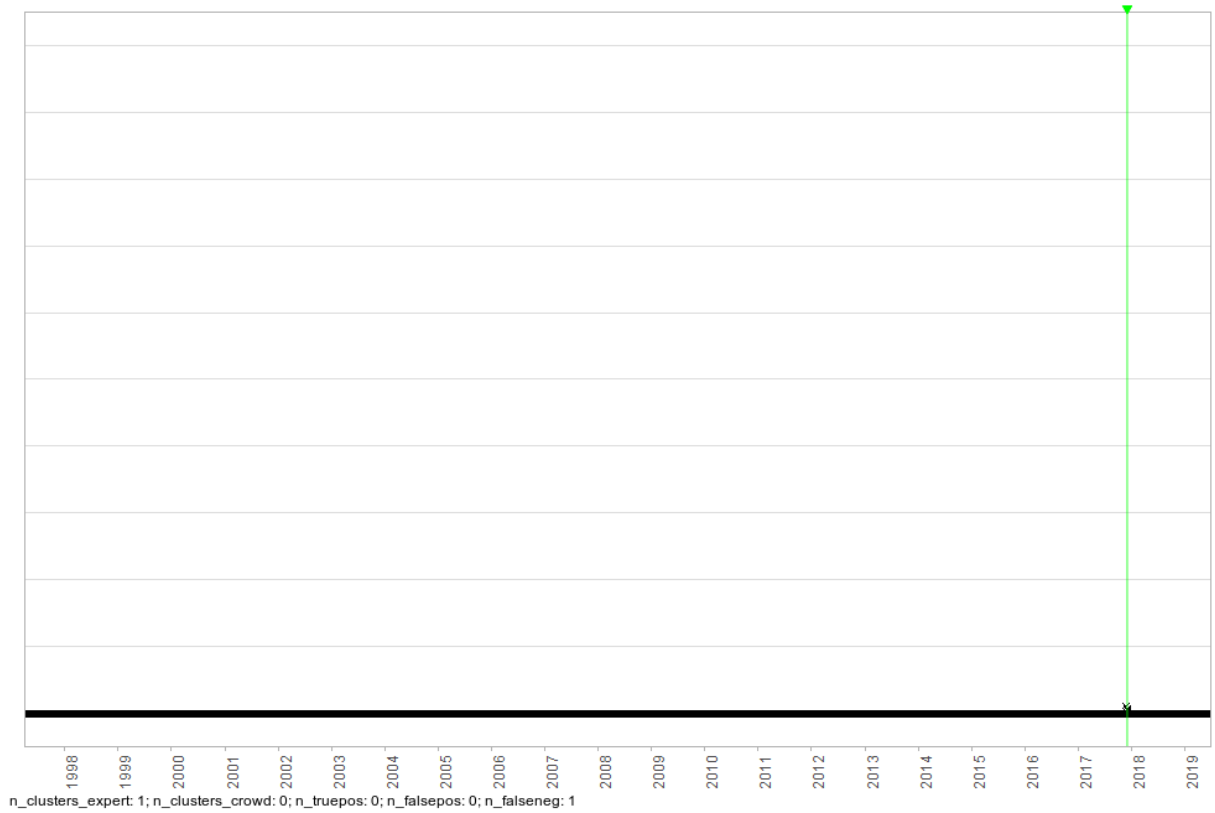


Figure 3-13. Examples of false negative clusters not identified by the volunteers (green line only). This is an outlier that was small in magnitude, and not identified by the volunteers.
Vertical lines denote positions of volunteer clusters and expert labels; blue lines with numbers above indicate the number of volunteers contributing to the cluster, green lines with inverted triangles indicate lines drawn by the expert.

3.6.3.3 Overall performance of the various different consensus methods

Lastly, the performance of the density-based clustering method for determining the simple existence of *any* change points in an image, was calculated by comparing whether or not any change points were found in a particular image (using the optimal parameters above), to whether or not the expert identified any change points in that image. This was done on the test set of 286 images and gave a final sensitivity for *any* change points of 96.0% (95% CI 92.1, 98.1), specificity of 89.9% (82.8, 94.3), PPV of 93.9% (89.4, 96.6), and NPV of 93.3% (86.9, 96.7), (for crowd-sourced vs expert-based classification). This was from 170 true positives, 98 true negatives, 11 false positives, and 7 false negatives.

Overall results comparing the various different consensus methods used for volunteer classifications are shown in Table 3-3. For determining the simple existence of *any* change points in an image, the performances of the vote threshold method using images showing

the full y-axis scale and the density clustering method were similar, with point estimates of sensitivity/specificity/PPV/NPV between 90-96% and overlapping confidence intervals. Therefore, either of these methods would be suitable for identifying whether or not change points exist in an image. For identifying where change points are located, the clustering method using *dbscan* parameters of minimum-lines-in-cluster = 5 and epsilon-neighbourhood = 3, and excluding yellow lines drawn by the volunteers, performed the best, with good levels of sensitivity/specificity/PPV/NPV all over 80%.

Table 3-3. Final performance of the different consensus methods against the expert labels, based on the test set (286 images).

Consensus method	**Sensitivity (95% CI)**	**Specificity (95% CI)**	**Positive Predictive Value (95% CI)**	**Negative Predictive Value (95% CI)**
For existence of *any* change points in the image (compared to the expert)				
By vote threshold, images using full y-axis scale	94.4% (89.9, 96.9)	93.6% (87.3, 96.9)	96.0% (91.9, 98.0)	91.1% (84.3, 95.1)
By vote threshold, images using smaller y-axis scale	83.1% (76.8, 87.9)	93.6% (87.3, 96.9)	95.5% (90.9, 97.8)	77.3% (69.4, 83.6)
By density-based clustering	96.0% (92.1, 98.1)	89.9% (82.8, 94.3)	93.9% (89.4, 96.6)	93.3% (86.9, 96.7)
For *locations* of individual change points in each image (compared to the expert)				
By density-based clustering	80.4% (77.1, 83.3)	99.8% (99.7, 99.8)	84.5% (81.4, 87.2)	99.7% (99.6, 99.7)

3.6.4 Prevalence of change points in EHRs

To estimate the prevalence of change points across different types of EHRs, the optimal clustering algorithm for calculating the locations of crowd-sourced change points (developed from comparing expert vs crowd-sourced classifications) was applied to all images from the 8 daily-aggregated data extracts classified by volunteers (total 1842 images). This resulted in a total of 4477 distinct change points.

All 8 data extracts contained change points, and in virtually all the data fields, see Table 3-4. The only data fields that did not contain any change points were the BugCode, BugResult, CollectionDateTime, Deathdate and DrugResult data fields in the microbiology blood cultures data extract (see Appendix B.1 for more details of data fields), and the 'duplicate records' calculated fields in the antibiotic prescribing, ED attendances and inpatient episodes data extracts.

Table 3-4. The total number of data fields assessed for each data extract, along with the number of data fields that were classed as containing change points based on clustering of volunteer classifications.

Dataset type	Data extract	Data from	Data to	Total no. of data fields	No. of data fields containing change points
Antibiotics	Antibiotic prescribing	10/06/2008	30/06/2019	27	26
Biochemistry	Creatinine tests	02/06/1986	30/06/2019	24	24
Haematology	Neutrophil counts	01/04/1987	30/06/2019	24	24
Microbiology	Blood cultures	04/06/1993	30/06/2019	37	32
Microbiology	E. coli isolations	17/05/1993	30/06/2019	37	37
Patient administration	ED attendances	01/04/2005	30/06/2019	28	27
Patient administration	Inpatient episodes	01/04/1997	30/06/2019	41	40
Patient administration	Outpatient episodes	01/04/1997	30/06/2019	35	35

Further, change points were detected in virtually every calendar year of every data extract, see Figure 3-14. The only exceptions were in the haematology (neutrophils) and biochemistry (creatinine) tests, where no change points were found in the final calendar year of 2019 (however this year also only contained 6 months of data since the data only ran to June 2019).

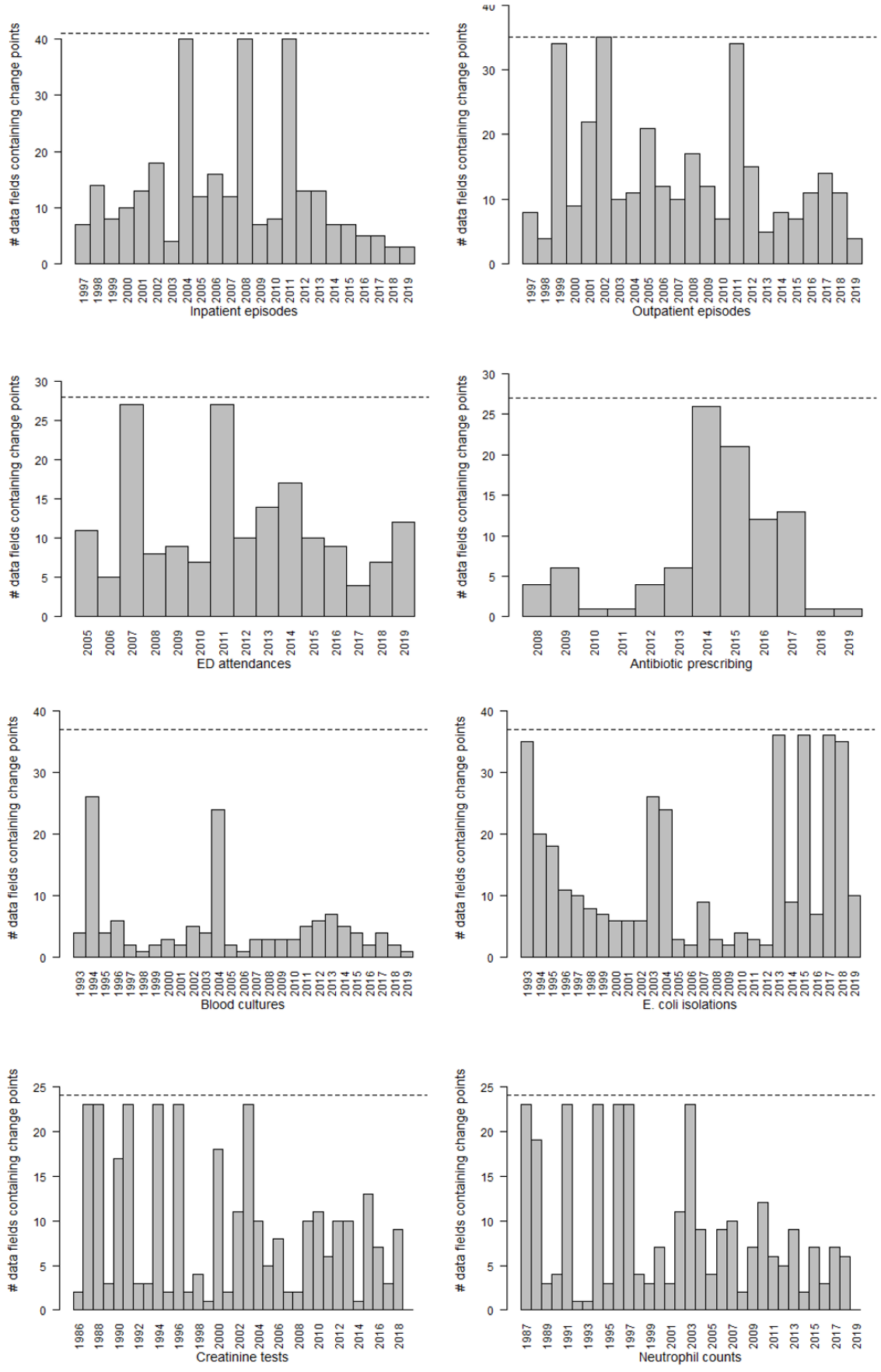
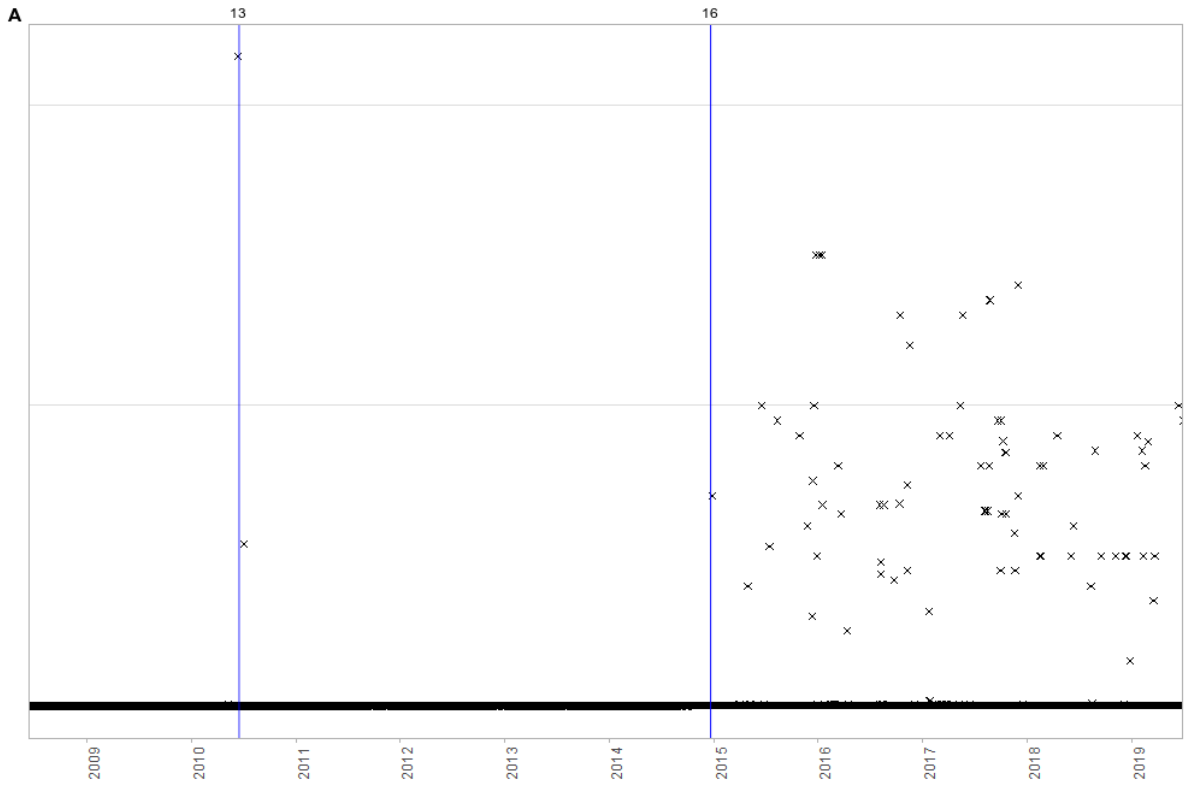


Figure 3-14. Number of data fields in each data extract which contained change points, by calendar year. Horizontal dotted lines denote the total number of data fields in the extract

Within individual data extracts, in some years change points were visible in only one data field and in other years change points were present across many data fields. For example, for antibiotics, year 2010 contained change points only in the Dose field, years 2011 and 2019 only in the PrescriptionID field, and in 2018 only in the ScheduledStopDateTime field, whereas in 2014, 26 of the 27 data fields contained change points (see Figure 3-15 for examples).



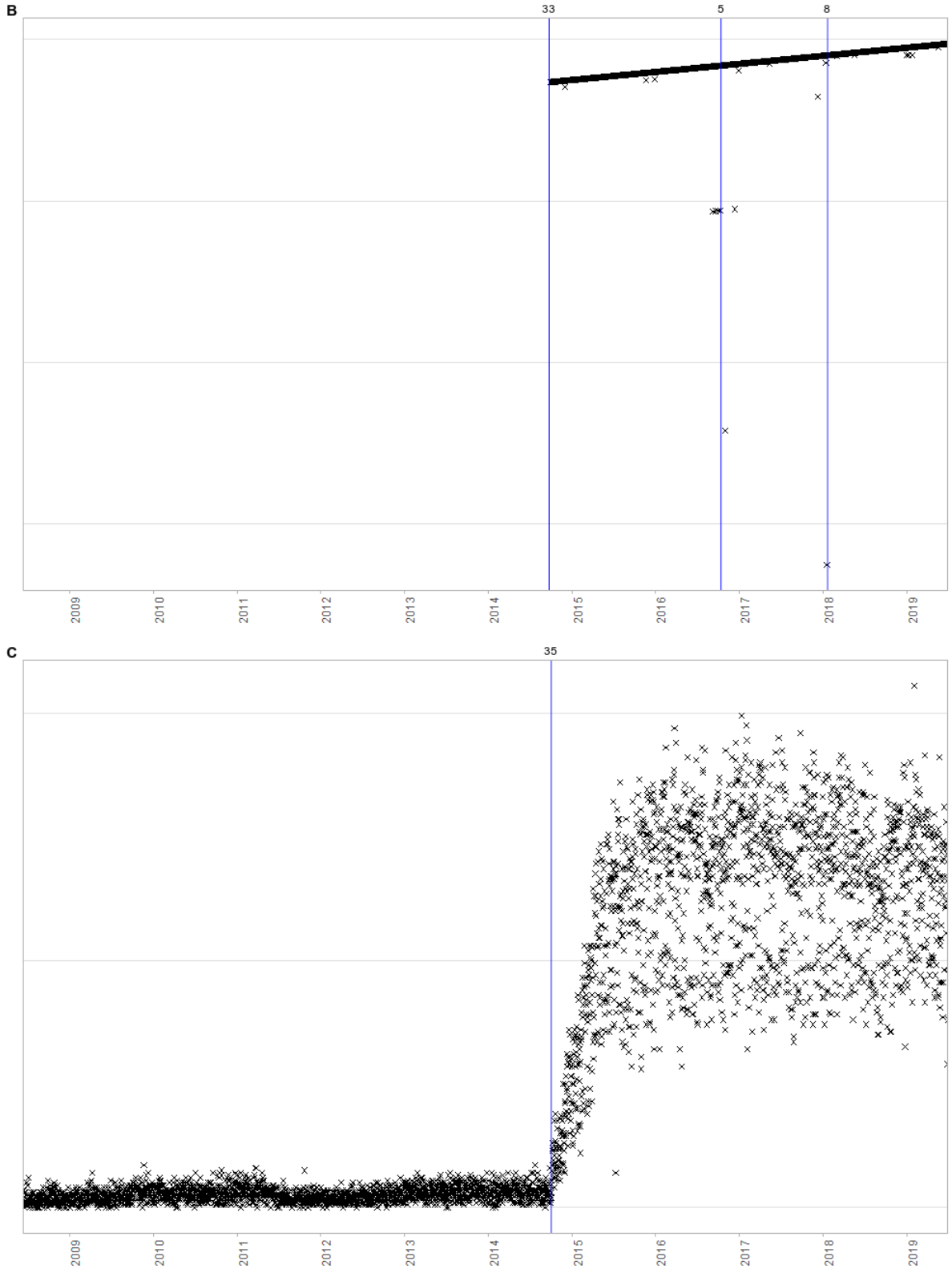
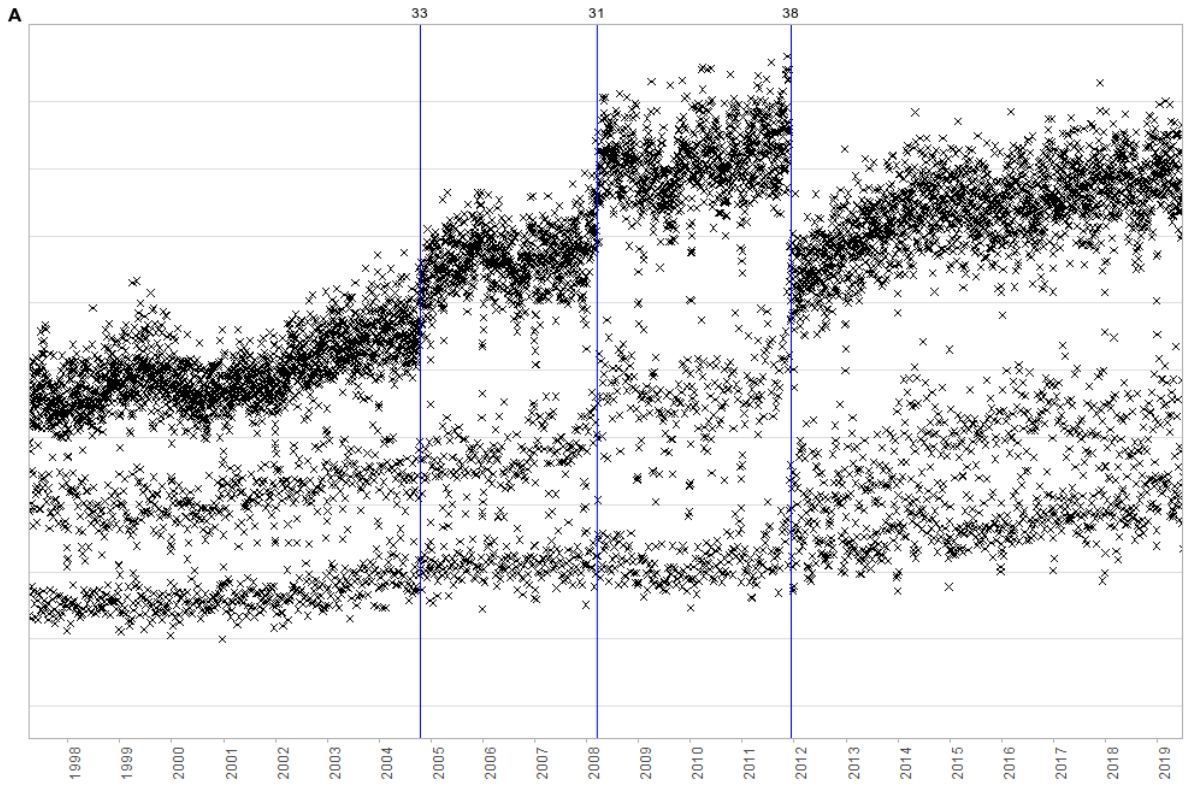


Figure 3-15. Examples of change points found in different data fields within the antibiotics data extract. A = Dose (maximum value), B = ScheduledStopDateTime (minimum value), C = Formulation (number of values present). All three contained change points in 2014, whereas only the Dose data field contained change points in 2010 and only the ScheduledStopDateTime data field contained change points in 2018.

When looking across related data extracts at years where change points were present in many different data fields, some years were consistent across different extracts while others were extract-specific. For example, 2011 contained change points for almost every field across all three data extracts from patient administration, while years 2004 and 2008 were additionally notable for change points in inpatient episodes, 1999 and 2002 for outpatient episodes, and 2007 for ED attendances (see Figure 3-16 for examples).



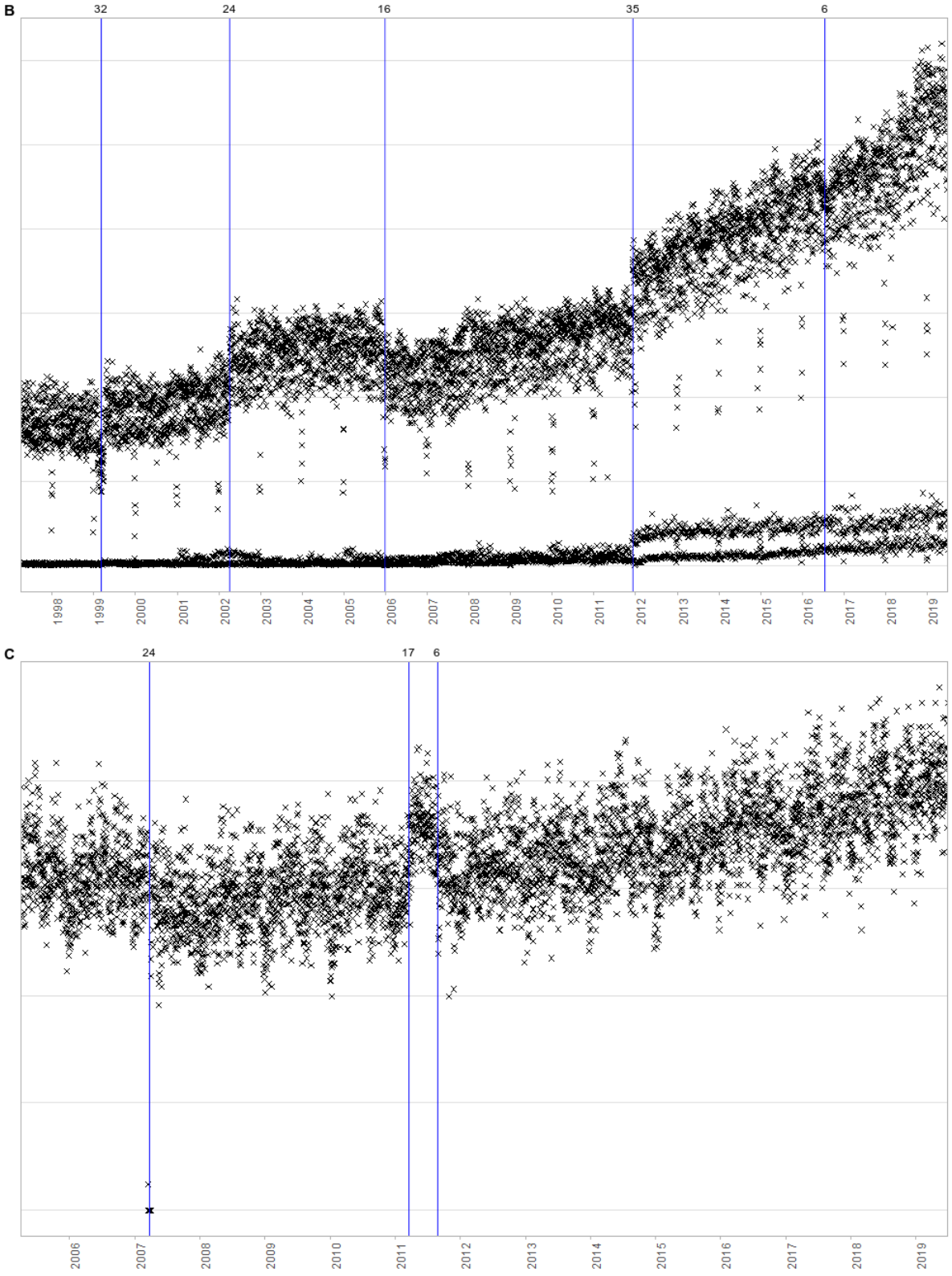


Figure 3-16. Examples of change points found in the three different patient administration data extracts. A = inpatient episodes, B = outpatient episodes, C = ED attendances, showing the number of values present for the PatientClassificationCode, ClinicCode , and IncidentLocationTypeCode data fields respectively. All three contained change points in 2011, whereas change points were also frequently found in 2004 and 2008 for inpatient episodes data fields, in 1999 and 2002 for outpatient episodes, and in 2007 for ED attendances.

3.7 Discussion

Crowd-sourced labels identifying whether or not there were *any* change points within an EHR time series were highly accurate when compared to labels made by an experienced data scientist (sensitivity/specificity/PPV/NPV >=90%), with both the density-based clustering method and the vote threshold method performing similarly. When comparing the actual *locations* of individual change points, the sensitivity was still quite high at ~80%, with PPV at ~85%, and specificity/NPV at >99%. Given that visual inspection is always going to be a subjective measure, even when performed by an expert, this level of accuracy suggests that crowd-sourcing is a satisfactory method for identifying change points in EHR time series. It was not possible to assess if the accuracy level achieved here is typical or not, since a review of the full list of research papers published using data from the Zooniverse did not reveal any projects which used the same classification method of drawing vertical lines.

The types of change points which were most often missed by the volunteers were “outliers”, and to a lesser extent, change points which were small in magnitude. This is potentially acceptable since arguably, outliers are less likely to have a significant impact on a study’s results than persistent change points, owing to them occurring for only a small number of records. Similarly, change points that are small in magnitude are less likely to have large consequences. Conversely, the volunteers tended to label change points more often than the expert on images based on highly discretised values, which means that certain aggregation functions (such as the number of distinct values in a categorical field) will likely result in more false positive calls than others, and hence may require more careful scrutiny when being used for validating automated methods. On review of discrepancies, despite the experience of the expert, many of the differences between crowd-sourced and expert classifications of the presence of a change point could have been argued either way. This subjectivity means that if these labels are to be used as a “gold standard” for testing automated methods, one can never expect those automated methods to perform perfectly

against the labels, and so perhaps one would need to accept a lower accuracy rate than one otherwise would.

Given the unexpected popularity of this project in the Zooniverse (with ~350,000 classifications completed by >2000 volunteers over 7.5 weeks), there is a clear opportunity for increasing public engagement in the medical sciences by utilising these types of avenues. Many of the established Zooniverse projects use their volunteer classifications as an integral part of their methodology, whether as a screening method for identifying potential objects of interest (such as looking for signals for planetary bodies), or to create end points (such as the prevalence of different mammal species within a terrain). These types of projects engage the public in a very concrete way, as the volunteers know that their classifications are directly determining the outcomes of real-world studies, rather than the more "meta" purpose of improving methodology as this project is doing.

Would it be inconceivable to have citizen scientists actively participating in scientific analyses within the field of healthcare? Most of the biomedical-related projects currently in the Zooniverse appear to use their volunteer classifications for the purpose of training machine-learning algorithms to do the same job (such as "Etch a Cell – Fat Checker", "Dental Disease Detection", and "Eye for Diabetes"), however the "Etch a Cell – Powerhouse Hunt" project used the segmentations produced by their volunteers directly to identify mitochondria in their images. For the "Bash the Bug" Zooniverse project, volunteers looked at images of bacterial growth under different concentrations of antibiotics and identified the point at which they believed the antibiotic successfully stopped the bacteria from growing. The researchers found that between 11-17 volunteer classifications were sufficient to be able to determine the correct concentration reproducibly and accurately[83], and concluded that "there is a potential role for crowds to support (but not supplant) the role of experts in antibiotic susceptibility testing". If volunteers can be as good as experts at certain tasks, why shouldn't they be utilised as an addition (or even alternative) to experts or algorithms? Clearly there are bigger ethical implications for health-related studies compared

to those in astrophysics, but this is not a suggestion for using volunteers to make clinical decisions on patient care, only that they could potentially be utilised for large-scale analysis of aggregated and fully anonymised (not individual record level) research data. The results would still be reviewed by experienced scientists. More specifically, if automated change point detection is not a near-future prospect, it could be argued that volunteers could be asked to detect any change points in the meantime.

The prevalence of change points identified by crowd-sourced visual inspection was incredibly high, far higher than was initially expected, with change points detected in all eight data extracts examined, and in almost every year of data that each extract covered. This suggests that poor or unknown data quality could be hampering research in many fields, and is a strong argument for increased transparency around the data-cleaning and initial data analysis phase of a study. The "reproducibility crisis" which originally came to prominence in the field of psychology[84] is certainly not confined to it, with 90% of ~1500 survey respondents from a range of scientific disciplines (including medicine) stating that they believed that there is a "slight" or "significant" crisis in their field[85]. While the respondents judged the biggest causes to be selective reporting and pressures to publish, around 45% said that they believed that unavailability of methods or code "always/often" contributed to irreproducibility. Whether or not there truly is a "crisis", given there appears to be a widespread lack of trust in the current system, this is something that needs to be addressed, and an increase in transparency and openness can help to achieve this.

One question is how representative this data from OUH is amongst all EHRs. Only two studies were found during the literature review which described the prevalence of change points in EHR-related data. The study from Looten et al.[68] found 49 changes in level and 39 changes in discretisation across 192 different biochemistry/haematology laboratory test values from a Paris hospital over 17 years. The study from Garcia-de-Leon-Chocano et al.[66] investigating EHR data related to infant feeding in a Spanish hospital reported that 9/46 categorical variables contained changes in their coded values over a period of only 3 years.

Whilst direct comparison is difficult, and arguably overall rates were somewhat lower than found here, these studies support the suggestion that change points occur frequently in EHRs from all hospitals.

Since it is highly likely that any data extract obtained from EHRs will contain temporal change points in metrics relating to data quality, it is important that all researchers who use these data are aware of this and take appropriate steps to assess and manage their impact. However, in order to do this thoroughly, researchers would have to invest the time and effort to generate and then visually inspect a potentially large number of images, which is something that they may not be inclined to do unless there is a change in the norms, incentives or policies around the levels of reporting of the initial data analysis phase of studies. There is some progress in this area already, with the Centre for Open Science (https://www.cos.io) actively trying to promote an open research culture by (amongst other initiatives) providing infrastructure such as the Open Science Framework (https://osf.io/) where all research-related materials and documents can be deposited and accessed publicly, by creating the Transparency and Openness Promotion guidelines[86] to encourage journals and organisations to adopt open science policies, and providing digital badges for researchers to signal to their peers that they have made aspects of their work openly available. There are further pragmatic elements that can increase researchers' willingness to adopt these practices, one being making it as easy and painless as possible. If the barriers to implementation can be lowered, for example by providing tools to assist researchers in doing certain steps easily and efficiently, this should make it an easier "sell" and hopefully increase uptake. How this might be achieved in the area of change points and data quality will be explored in the remaining chapters.

3.8 Conclusion

A huge number of temporal change points were found in the EHRs from a large UK hospital group, appearing in every year and data field from data extracts covering patient, laboratory,

and clinician activity. Given the high probability of change points occurring in *any* data extract from EHRs, it is important that researchers check their data for these types of artefacts before proceeding with their main analyses. To encourage this practice, it would be useful to raise awareness of this type of problem, and to provide researchers with more guidelines and tools for checking for data quality issues, in particular around checking for temporal change points.

CHAPTER 4: Optimal aggregation methods for visual identification of change points

4.1 Chapter summary

Temporal change points can occur very frequently in EHRs, so it is important that researchers check their data for the presence of change points in order to reduce the risk of generating erroneous results.

This chapter assesses how different data aggregation methods perform at enabling change points to be identified visually. It does this by comparing the change points identified within each of 8 different EHR data extracts from a large UK hospital group covering patient, laboratory, and clinician activity, when the data are aggregated in different ways.

The data in each data field and extract was aggregated at three different granularities (i.e. day, week or month) and by a variety of different functions (e.g. the percentage of missing values or number of distinct values). The resulting time series were plotted and change points were identified by crowd-sourced visual inspection. The change points identified using the different aggregation methods were compared for abundance and uniqueness.

Change points were found within every single aggregation function and every aggregation granularity used, though more change points were found using certain aggregation methods than others. In particular, more change points were found when using a daily granularity than a weekly or monthly granularity, and functions measuring frequencies were generally more effective than functions measuring proportions.

4.2 Background

In the previous chapter it was shown that temporal change points (i.e. sudden, unexpected changes in data at particular points in time) can occur very frequently in EHRs, with change points detected in all eight data extracts examined, and in almost every data field and every year of data that each extract covered. This means that it is highly likely that any data extract obtained from EHRs will contain temporal change points in metrics relating to data quality, and therefore it is important that all researchers who use these data are aware of this and take appropriate steps to assess and manage their impact.

In an ideal world, researchers would aggregate their data in as many ways as possible in order to find any potential change points, but in the real world, this may not be feasible depending on the size of the dataset and the time available. There are many different ways in which a data extract can be aggregated to produce time series for visual inspection, and it is not clear which methods are most effective at revealing change points.

While crowd-sourcing has been used to visually identify change points in the data used for this study, researchers do not typically have access to thousands of volunteers to identify change points for them, so there is a balance to be found between checking every possible aggregation of the data, and checking those most likely to reveal change points.

Therefore, this chapter will look to identify aggregation methods that are particularly effective at revealing change points, as well as those that are potentially unnecessary (either because they don't reveal many change points or that they only reveal change points that have already been identified using other aggregation methods).

4.3 Methods

4.3.1 Data sources

Eight data extracts were taken from IORD: inpatient, outpatient, and emergency (ED) episodes from the patient administration dataset; antibiotic prescribing records; creatinine

tests and haematology neutrophil counts from the haematology/biochemistry laboratory dataset; and blood culture tests and all tests that identified *E. coli* from the microbiology laboratory dataset. Across the eight data extracts, there were 253 data fields and around 3.7 million records.

4.3.2 Aggregation methods

The data from each data extract was aggregated in multiple ways to produce a collection of time series for comparison. Aggregation methods fell into 3 different categories:

- By **aggregation granularity** – records were grouped on a daily (midnight to midnight), weekly (Monday to Sunday) or calendar month basis, using the data field identified as representing the 'timepoint' for each record
- By **aggregation function** – numeric summary values were calculated at each timepoint for the (often non-numeric) data by applying simple functions (e.g. percentage of missing values, number of distinct values, or median value). Each aggregation function demonstrated a measure of one of the intrinsic data quality categories that arose from Chapter 2, namely Completeness, Conformance, and Plausibility. Different functions were used depending on the type of data field (i.e. numeric, datetime, timepoint, uniqueidentifier, categorical, or freetext[1]) or if applied to the data extract as a whole (e.g. calculating the number of duplicate records). See Table 4-1 for the complete list of aggregation functions
- By **individual data field or all data combined** – while most aggregation functions were applied to the data in an individual data field, some aggregation functions were also applied to the combined data values across all relevant fields in the data extract (namely numbers of values present, number/percentage of missing values, and number/percentage of non-conformant values)

[1] see Section 3.3.2.1 for a detailed explanation of the data field types

Table 4-1. Details of the aggregation functions applied to each type of data field, and across the data extract as a whole.

Aggregation function (*shorthand label*)	**Individual data field type**						**Across data extract as a whole**	
	numeric	**datetime**	**timepoint**	**uniqueidentifier**	**categorical**	**freetext**	**All data combined**	**Duplicate records**
COMPLETENESS								
Number of missing values (*missing_n*)	x	x		x	x	x	x	
Percentage of missing values (*missing_perc*)	x	x		x	x	x	x	
CONFORMANCE								
Number of non-conformant values* (*nonconformant_n*)	x	x					x	
Percentage of non-conformant values* (*nonconformant_perc*)	x	x					x	
PLAUSIBILITY								
Sum of duplicate records removed (*sum*)								x
Percentage of records which had been duplicated (*nonzero_perc*)								x
Number of values present (*n*)	x	x	x	x	x	x	x	
Minimum value (*min*)	x	x						
Maximum value (*max*)	x	x						
Mean value (*mean*)	x							
Median value (*median*)	x							

Aggregation function (*shorthand label*)	**Individual data field type**						**Across data extract as a whole**	
	numeric	**datetime**	**timepoint**	**uniqueidentifier**	**categorical**	**freetext**	**All data combined**	**Duplicate records**
Number of values with no time element† (*midnight_n*)		x	x					
Percentage of values with no time element† (*midnight_perc*)		x	x					
Minimum string length (*minlength*)				x				
Maximum string length (*maxlength*)				x				
Mean string length (*meanlength*)				x				
Number of distinct values (*distinct*)					x			
Number of values within each subcategory‡ (*subcat_n*)					x			
Percentage of values within each subcategory‡ (*subcat_perc*)					x			

* Non-conformance was deemed as a non-numeric value in a (supposedly) numeric data field, or a non-date value in a (supposedly) date field

† These were only calculated for fields that were known to contain a time element, and where midnight would be used as the default when no time element was available

‡ These were only calculated for fields with fewer than 20 subcategories (with the additional inclusion of DischargeDestinationCode in the inpat_episode data extract, which contained 23 subcategories, and was included for consistency with the other coded fields in the data extract)

4.3.3 Identification of change points

Individual time series were generated for each individual data field and overall for each data extract, using each aggregation function listed in Table 4-1, and repeated for all three granularities. Scatter plots of each time series were visually inspected by Zooniverse citizen-science volunteers for the locations of any sudden and unexpected changes in the data (including step changes, sharp changes in trend, changes in presence/absence of data points, changes in (vertical) variability, and unexpected outliers) (see Chapter 3). Consensus labels were derived by density-based clustering (using the *dbscan* package in R) and validated against expert labels. Positions of change points were recorded as pixel positions on each image.

4.3.4 Analysis methods

There is no real-world "truth" available for whether or not an apparent sudden change in the data should be classed as a change point or not. Therefore, the change points detected within a particular data field when using one aggregation method versus another, were compared to see whether certain methods resulted in more change points being identified than when using other methods, and whether any methods were unique in uncovering change points that no other method revealed. In addition to comparing methods for detecting change points within a particular data field, two methods were compared for detecting change points at the data extract level.

4.3.4.1 Comparisons between data aggregation functions

Different sets of aggregation functions were relevant to the different categories of data quality (i.e. Completeness, Conformance, and Plausibility). For each related set of aggregation functions, all images that were based on the same data field and aggregation *granularity* were grouped into units, and a set of distinct change points was generated by combining the change points from the different images if they were less than 7px from each other. The change points identified using each *specific* aggregation function were then compared either to each other or to this combined set, and were judged to represent the

same underlying change point if they were less than 7px from each other (as was done previously in Chapter 3 when comparing volunteer classifications to expert classifications).

For the data quality category of Completeness, the locations of change points identified when using either the number or the percentage of missing values as the aggregation function were compared. For the data quality category of Conformance, the locations of change points identified when using either the number or the percentage of non-conformant values as the aggregation function were compared. For the data quality category of Plausibility, since different aggregation functions were used for different data field types, the locations of change points identified using each aggregation function versus each other aggregation function for that data field type, as well as versus all distinct change points identified across all the aggregation functions combined, were compared.

See Table 4-1 for the full list of aggregation functions per data field type, along with their shorthand label which will sometimes be used for convenience.

4.3.4.2 Comparisons between data aggregation granularities

To compare change points across the 3 different granularity levels, each set of 3 images from the same data field and aggregation function were grouped, and the change points identified across each group were judged to represent the same underlying change point if they were less than 11px from each other.

The reason for using 11px, rather than the 7px distance previously used to match change points from different images, was because the data points on an image generated using daily-aggregated data would not be expected to align perfectly on the x-axis to the points on an image generated using weekly or monthly-aggregated data (given the differing number of data points in each time series). It was therefore necessary to determine a suitable distance to use to match change points across different granularities. For images based on the same data field and aggregation function, the minimum distances in pixels between daily-aggregated change points and monthly-aggregated change points were calculated (as these

would be the least expected to align), and the resulting pair of histogram plots was visually inspected for a threshold in the distances that could indicate a difference between labels for the same change point versus labels for different change points, see Figure 4-1. Since numbers appeared to level from 11px onwards to an approximately constant background level, change points from different granularities were considered to be a match if they were <11px away from each other.

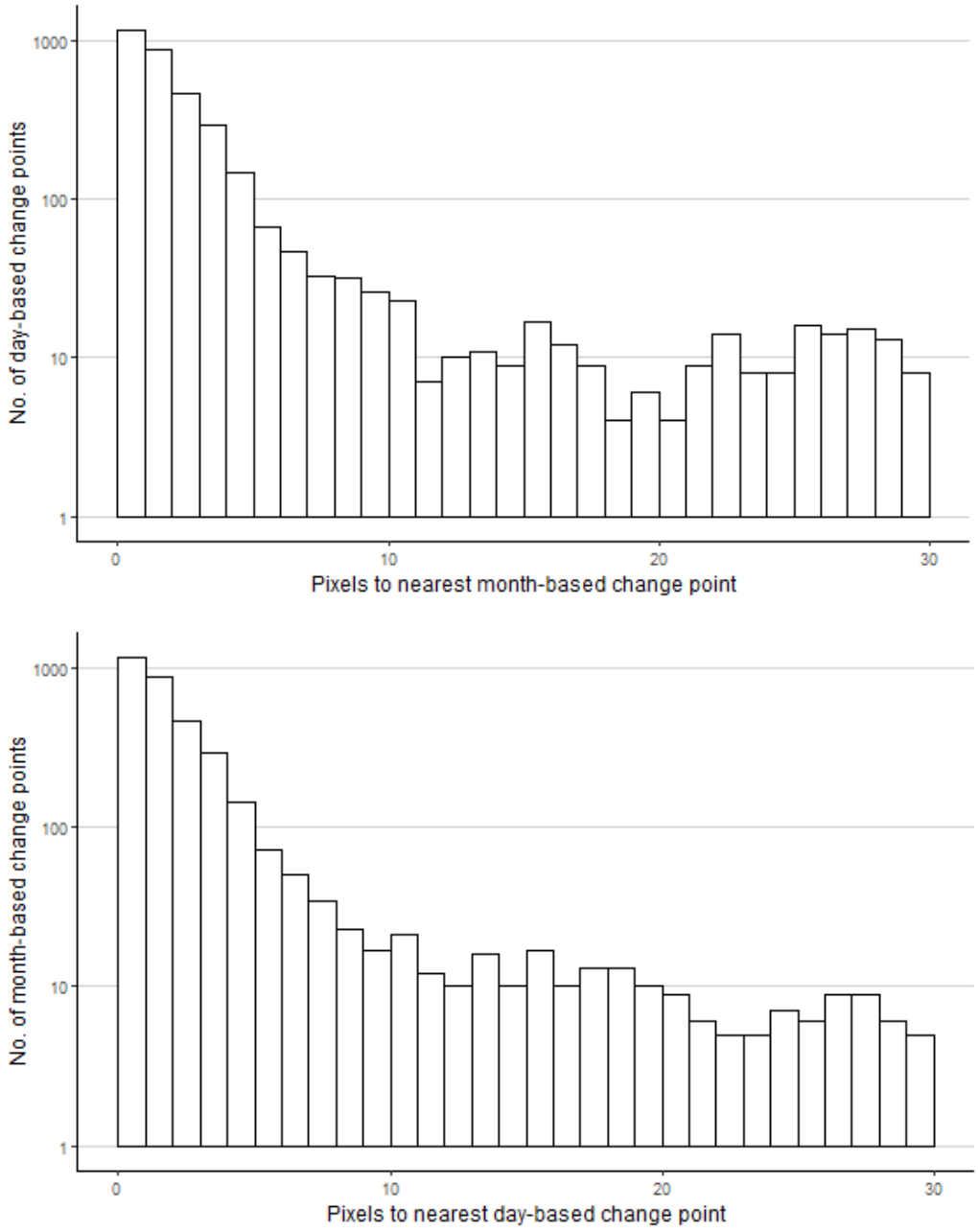


Figure 4-1. Minimum distances between daily-aggregated change points and monthly-aggregated change points for images based on the same data field and aggregation function.
Intervals are closed on the left. Y-axis is shown on a log scale.

4.3.4.3 Comparisons at data extract level

Two different methods were compared for detecting change points across a data extract as a whole. For a particular aggregation function and granularity, change points were identified using the *individual fields method* by creating individual time series images for each data field in the data extract, then combining the crowd-sourced change points across the different images if they were within 7px of each other. For the *combined data method*, the data values across all relevant fields in the data extract were combined first, before producing a single time series image for crowd-sourced inspection.

This was done for the 5 aggregation functions that were applicable to multiple different types of data fields, namely the number of values present and the number and percentage of missing values (n/missing_n/missing_perc, applicable to all data fields), and the number and the percentage of nonconformant values (nonconformant_n/nonconformant_perc, applicable to all numeric and datetime fields).

Since comparisons were made across images aggregated at the same granularity (as was done in Section 4.3.4.1), change points from the two different methods were judged to represent the same underlying change point if they were less than 7px from each other.

4.4 Results

A total of 5526 images were created from the 8 data extracts (1842 images at each of the 3 granularities), and 3867 of these images contained at least one change point. 12,745 crowd-sourced change points were identified altogether.

4.4.1 Comparisons between aggregation functions

4.4.1.1 Completeness

Completeness was measured using the number of missing values (missing_n) and percentage of missing values (missing_perc) aggregation functions. They were calculated both for individual data fields and on the combined data values across a data extract.

There were 711 (data field/aggregation granularity) units where completeness over time was measured, of which 524 (74%) contained at least one change point, see Table 4-2. Far more change points were identified when using the 'missing_n' aggregation function than with 'missing_perc' (2114 vs 1316), suggesting that the frequency of missing values is more effective at identifying change points in Completeness than the percentage of missing values. Despite this, 385 (15%) change points would have been missed if only 'missing_n' had been used. The pattern was similar across different data field types, except for numeric types where even fewer change points were identified using 'missing_perc' than with 'missing_n', see Figure 4-2.

Table 4-2. Change points identified in aggregation functions measuring Completeness, grouped by data field type.
missing_n = number of missing values, missing_perc = percentage of missing values

	Data field type						
Measure	**numeric**	**datetime**	**unique identifier**	**categorical**	**freetext**	**All data combined**	**Total**
No. of units	27	135	81	354	90	24	711
Units containing change points (%)	19 (70)	115 (85)	39 (48)	246 (69)	81 (90)	24 (100)	524 (74)
No. of distinct change points	40	504	269	1046	453	187	2499
Change points in missing_n only (%)	32 (80)	221 (44)	164 (61)	498 (48)	211 (47)	57 (30)	1183 (47)
Change points in missing_perc only (%)	1 (2)	87 (17)	39 (14)	124 (12)	88 (19)	46 (25)	385 (15)
Change points in both (%)	7 (18)	196 (39)	66 (25)	424 (41)	154 (34)	84 (45)	931 (37)

Figure 4-2. Percentage of change points seen in different aggregation functions measuring Completeness, grouped by data field type.
Width of bars corresponds to number of units that contained change points. missing_n = number of missing values, missing_perc = percentage of missing values.

4.4.1.2 Conformance

Non-conformance was defined as a non-numeric value in a (supposedly) numeric data field, or a non-date value in a (supposedly) date field. It was measured using the number of non-conformant values (nonconformant _n) and percentage of non-conformant values (nonconformant _perc) aggregation functions. Again, they were calculated both for individual data fields and on the combined data values across a data extract.

There were 186 (data field/aggregation granularity) units where non-conformance over time was measured, of which 18 (10%) contained at least one change point, see Table 4-3. Compared with Completeness, proportionately more change points were identified when using the ‘nonconformant_n’ aggregation function than with ‘nonconformant_perc’ (91 vs 11), and only 1 (1%) change point would have been missed if only ‘nonconformant_n’ had been used. No change points were identified in datetime fields, nor when combining all (numeric and date) fields together and using the ‘nonconformant_perc’ function, see Figure

4-3. Therefore, the frequency of non-conformant values may be sufficient on its own to identify change points in Conformance.

Table 4-3. Change points identified in aggregation functions measuring Conformance, grouped by data field type. nonconformant_n = number of non-conformant values, nonconformant_perc = percentage of non-conformant values

	Data field type			
Measure	**numeric**	**datetime**	**All data combined**	**Total**
No. of units	27	135	24	186
Units containing change points (%)	9 (33)	0 (0)	9 (38)	18 (10)
No. of distinct change points	46	0	46	92
Change points in nonconformant_n only (%)	35 (76)	0	46 (100)	81 (88)
Change points in nonconformant_perc only (%)	1 (2)	0	0 (0)	1 (1)
Change points in both (%)	10 (22)	0	0 (0)	10 (11)

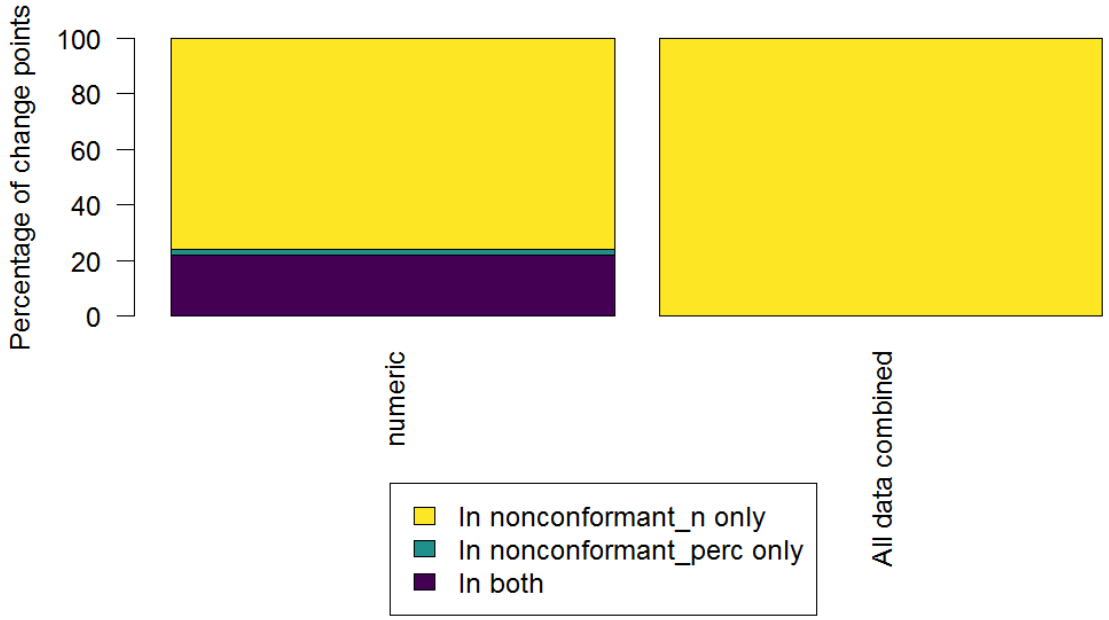


Figure 4-3. Percentage of change points seen in different aggregation functions measuring Conformance, grouped by data field type.
Width of bars corresponds to number of subjects that contained change points. There were no change points within datetime fields. nonconformant_n = number of non-conformant values, nonconformant_perc = percentage of non-conformant values.

4.4.1.3 Plausibility

For the data quality category of Plausibility, different aggregation functions were relevant to different types of data fields (or to the data extract as a whole) and so each type has been dealt with separately here.

4.4.1.3.1 Duplicate records

Two different aggregation functions were used to measure duplicate records over time – the total number of duplicate records removed from the timepoint (sum) and the percentage of the remaining records in the timepoint which had duplicates removed (nonzero_perc). Since this was a data-extract-level rather than a data-field-level measure, there were always 3 images per data extract (one for each granularity of day/week/month) relating to each aggregation function. Images from the same data extract and aggregation granularity were grouped into units and the change points identified between the two different aggregation functions were compared.

There were 24 data extract/aggregation granularity combinations, in which a total of 45 distinct change points were identified. All the change points were identified when using the 'sum' aggregation function, and only 10 (22%) when using 'nonzero_perc' (see Table 4-4, Figure 4-4). Since no additional change points were identified when using 'nonzero_perc' compared to when using 'sum', 'sum' might be sufficient on its own to identify change points in duplicate records.

Table 4-4. Change points identified in aggregation functions measuring duplicate records.
sum = sum of duplicate records removed, nonzero_perc = percentage of records which had been duplicated

	Aggregation function	
Measure	**nonzero_perc**	**sum**
Across all aggregation functions:		
- No. of units	24	24
- No. of distinct change points	45	45
Per aggregation function:		
- Units containing change points (%)	8 (33)	15 (62)
- No. of change points (%)	10 (22)	45 (100)
Compared to distinct change points:		
- No. only in this aggregation function (%)	0 (0)	35 (78)
- No. in this and other aggregation functions (%)	10 (22)	10 (22)
- No. not in this aggregation function (%)	35 (78)	0 (0)
Compared to other aggregation functions:		
- No. also in nonzero_perc	-	10
- No. also in sum	10	-

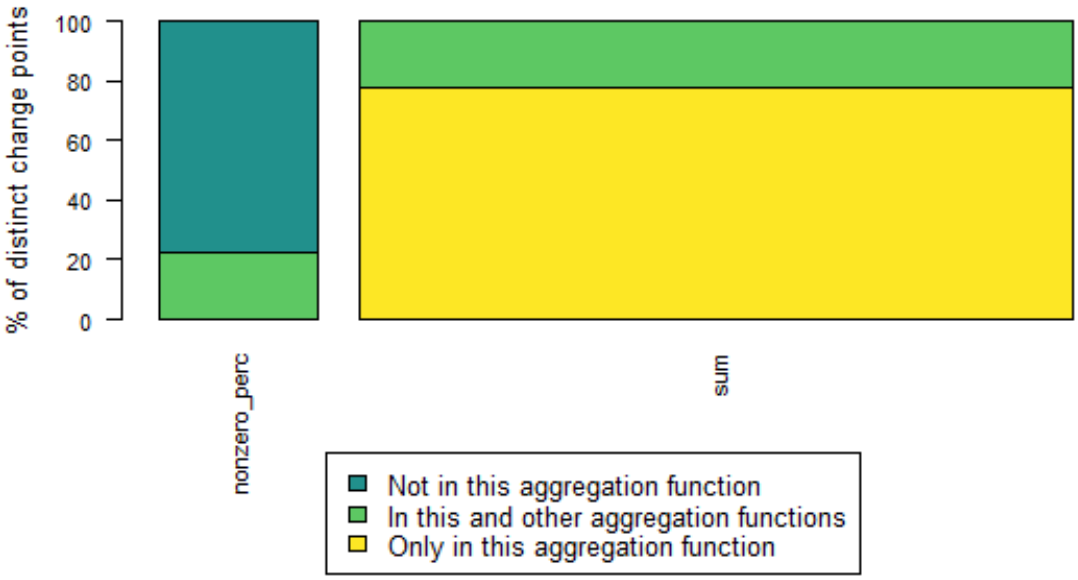


Figure 4-4. Percentage of the total distinct change points identified for duplicate records, based on different aggregation functions.
Width of bars corresponds to the total number of change points identified using that aggregation function. sum = sum of duplicate records removed, nonzero_perc = percentage of records which had been duplicated.

4.4.1.3.2 Numeric fields

Five different aggregation functions were used to assess Plausibility in numeric data fields – the number of values present (n), the minimum value (min), maximum value (max), mean

value (mean), and median value (median). There were 9 numeric data fields across the various data extracts, resulting in 27 data field/aggregation granularity units, in which a total of 274 distinct change points were identified. Between 16-41% of the total distinct change points were identified using any single aggregation function (see Table 4-5, Figure 4-5), and between 7-25% of the distinct change points would have been missed if any one of the aggregation functions had been omitted. Together, the 'n' and 'median' aggregation functions exposed a majority (68%) of the distinct change points, but without the additional 'min', 'max' and 'mean' functions, about a third of the change points would have been missed.

Table 4-5. Change points identified in numeric fields, for aggregation functions relating to Plausibility. n = number of values present, min = minimum value, max = maximum value, mean = mean value, median = median value.

	Aggregation function				
Measure	**n**	**min**	**max**	**mean**	**median**
Across all aggregation functions:					
- No. of units	27	27	27	27	27
- No. of distinct change points	274	274	274	274	274
Per aggregation function:					
- Units containing change points (%)	27 (100)	22 (81)	20 (74)	25 (93)	22 (81)
- No. of change points (%)	87 (32)	44 (16)	69 (25)	87 (32)	112 (41)
Compared to distinct change points:					
- No. only in this aggregation function (%)	56 (20)	21 (8)	29 (11)	20 (7)	68 (25)
- No. in this and other aggregation functions (%)	31 (11)	23 (8)	40 (15)	67 (24)	44 (16)
- No. not in this aggregation function (%)	187 (68)	230 (84)	205 (75)	187 (68)	162 (59)
Compared to other aggregation functions:					
- No. also in n	-	16	10	22	14
- No. also in min	16	-	13	15	14
- No. also in max	10	13	-	37	18
- No. also in mean	22	15	37	-	37
- No. also in median	14	14	18	37	-

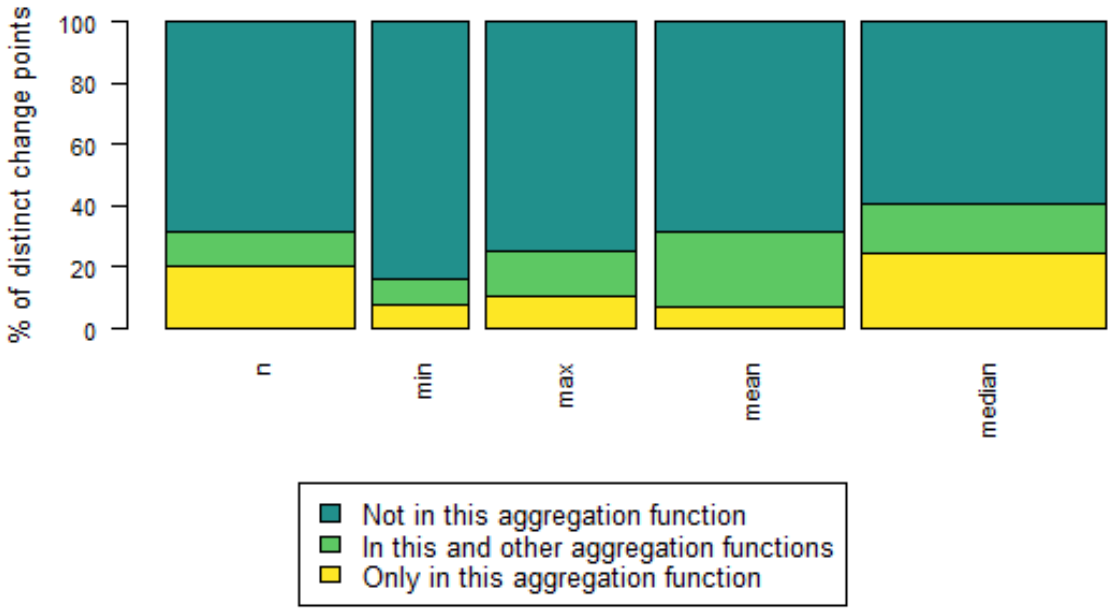


Figure 4-5. Percentage of the total distinct change points identified in numeric fields, for different aggregation functions relating to Plausibility.
Width of bars corresponds to the total number of change points identified using that aggregation function. n = number of values present, min = minimum value, max = maximum value, mean = mean value, median = median value.

4.4.1.3.3 Datetime fields

Five different aggregation functions were used to assess Plausibility in datetime data fields – the number of values present (n), the minimum value (min), maximum value (max), the number of values with no time element (midnight_n), and the percentage of values with no time element (midnight_perc). There were 45 datetime data fields altogether, resulting in 135 data field/aggregation granularity units. The ‘n’, ‘min’, and ‘max’ aggregation functions were calculated for all 135 units. The ‘midnight_n’ and ‘midnight_perc’ aggregation functions were only calculated if there was a time element in the data field values (comprising 81 data field/aggregation granularity units), and can be considered to have contributed zero change points for those data fields which did not have a time element (since they would have produced a constant time series value of zero). A total of 1139 distinct change points were identified. Between 16-40% of the distinct change points were identified using any single aggregation function (see Table 4-6, Figure 4-6), and between 3-24% of the change points would have been missed if any one of the aggregation functions had been omitted. The ‘n’

and ‘max’ functions appeared most effective, together finding 65% of the distinct change points. Perhaps unsurprisingly, there was a lot of overlap between the change points identified using the ‘midnight_n’ and ‘midnight_perc’ functions (122 change points in common), with the ‘midnight_perc’ function appearing to add the least value overall, finding only 3% of change points that were not also identified using other aggregation functions.

Table 4-6. Change points identified in datetime fields, for different aggregation functions relating to Plausibility. n = number of values present, min = minimum value, max = maximum value, midnight_n = number of values with no time element, midnight_perc = percentage of values with no time element

	Aggregation function				
Measure	**n**	**min**	**max**	**midnight_n**	**midnight_perc**
Across all aggregation functions:					
- No. of units	135	135	135	81	81
- No. of distinct change points	1139	1139	1139	1139	1139
Per aggregation function:					
- Units containing change points (%)	122 (90)	97 (72)	112 (83)	71 (88)	63 (78)
- No. of change points (%)	459 (40)	273 (24)	330 (29)	314 (28)	187 (16)
Compared to distinct change points:					
- No. only in this aggregation function (%)	274 (24)	183 (16)	247 (22)	99 (9)	33 (3)
- No. in this and other aggregation functions (%)	185 (16)	90 (8)	83 (7)	215 (19)	154 (14)
- No. not in this aggregation function (%)	680 (60)	866 (76)	809 (71)	825 (72)	952 (84)
Compared to other aggregation functions:					
- No. also in n	-	55	53	119	52
- No. also in min	55	-	51	31	39
- No. also in max	53	51	-	24	41
- No. also in midnight_n	119	31	24	-	122
- No. also in midnight_perc	52	39	41	122	-

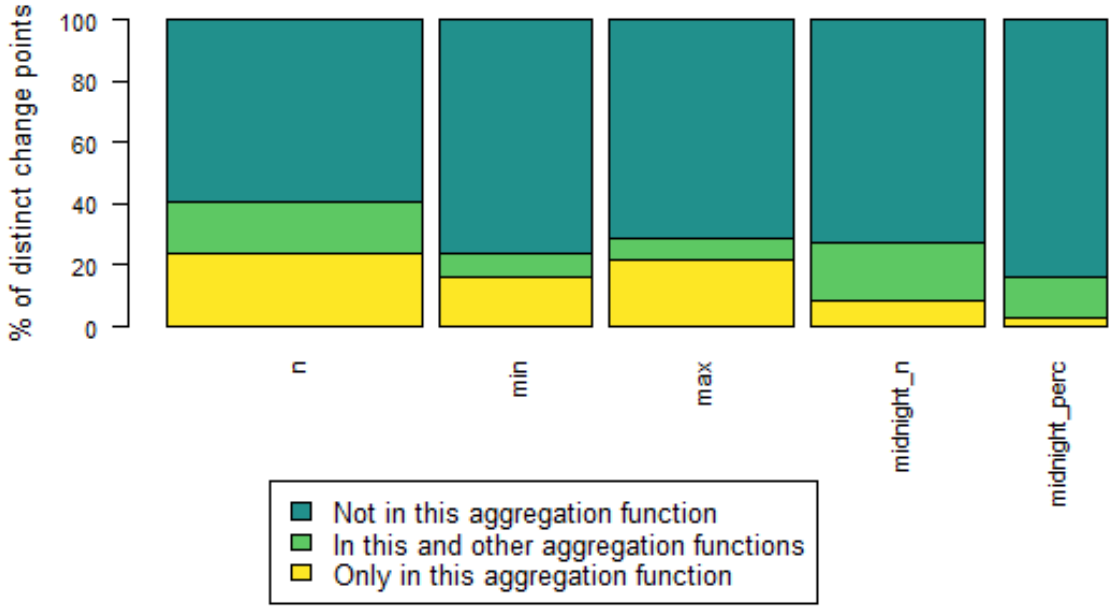


Figure 4-6. Percentage of the total distinct change points identified in datetime fields, for different aggregation functions relating to Plausibility.
Width of bars corresponds to the total number of change points identified using that aggregation function. n = number of values present, min = minimum value, max = maximum value, midnight_n = number of values with no time element, midnight_perc = percentage of values with no time element.

4.4.1.3.4 Uniqueidentifier fields

Four different aggregation functions were used to assess Plausibility in uniqueidentifier data fields – the number of values present (n), the minimum string length (minlength), maximum string length (maxlength), and mean string length (meanlength). Across 27 different data fields (comprising 81 data field/aggregation granularity units), a total of 915 distinct change points were identified. Between 27-44% of the total distinct change points were identified using any single aggregation function (see Table 4-7, Figure 4-7), and between 10-28% of the change points would have been missed if any one of the aggregation functions had been omitted. The ‘maxlength’ function identified the fewest change points.

Table 4-7. Change points identified in aggregation functions in uniqueidentifier fields, for different aggregation functions relating to Plausibility.

	Aggregation function			
Measure	**n**	**minlength**	**maxlength**	**meanlength**
Across all aggregation functions:				
- No. of units	81	81	81	81
- No. of distinct change points	915	915	915	915
Per aggregation function:				
- Units containing change points (%)	76 (94)	74 (91)	55 (68)	69 (85)
- No. of change points (%)	313 (34)	400 (44)	243 (27)	335 (37)
Compared to distinct change points:				
- No. only in this aggregation function (%)	252 (28)	213 (23)	87 (10)	105 (11)
- No. in this and other aggregation functions (%)	61 (7)	187 (20)	156 (17)	230 (25)
- No. not in this aggregation function (%)	602 (66)	515 (56)	672 (73)	580 (63)
Compared to other aggregation functions:				
- No. also in n	-	31	31	38
- No. also in minlength	31	-	98	170
- No. also in maxlength	31	98	-	138
- No. also in meanlength	38	170	138	-

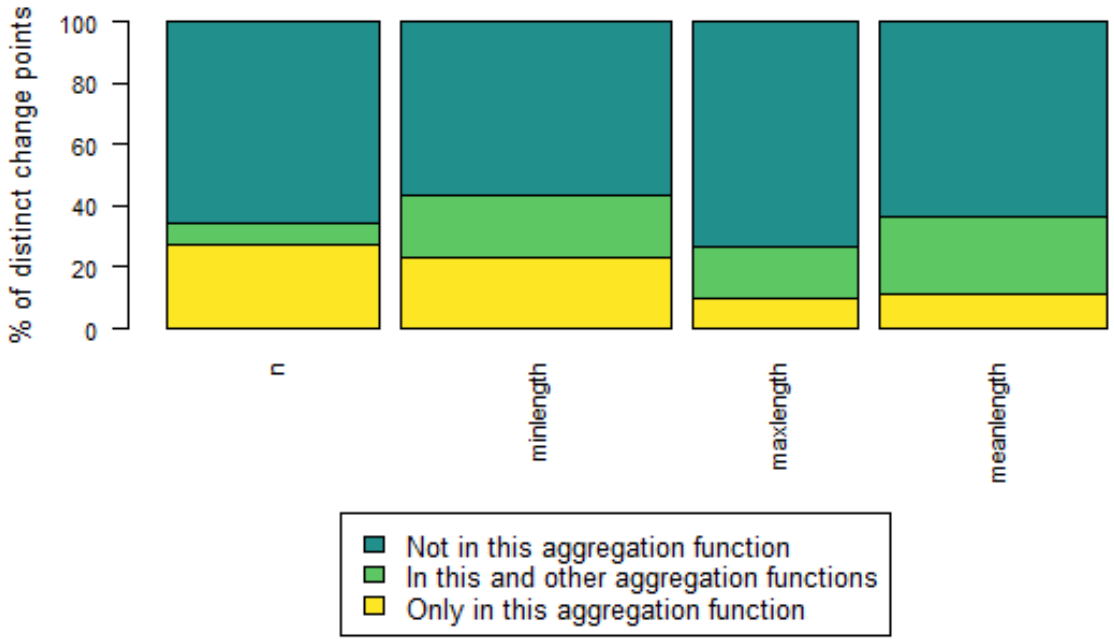


Figure 4-7. Percentage of the total distinct change points identified in uniqueidentifier fields, for different aggregation functions relating to Plausibility.
Width of bars corresponds to the total number of change points identified using that aggregation function.

4.4.1.3.5 Categorical fields

Four different aggregation functions were used to assess Plausibility in categorical data fields – the number of values present (n), the number of distinct values (distinct), the number of values within each subcategory (subcat_n), and the percentage of values within each subcategory (subcat_perc). There were 118 categorical data fields altogether, and the 'n' and 'distinct' aggregation functions were applied to all 354 data field/aggregation granularity units. The 'subcat_n' and 'subcat_perc' aggregation functions were only calculated for data fields which contained fewer than 20 different subcategory values (57 data fields containing 295 subcategories in total), and the change points identified across the various subcategories of each data field were combined to form a set of overall change points for the aggregation function. A total of 2575 distinct change points were identified. Between 34-46% of the distinct change points were identified using any single aggregation function (see Table 4-8, Figure 4-8), and between 5-26% of change points would have been missed if any one of the aggregation functions had been omitted. Together, the 'n' and 'distinct' aggregation

functions identified 77% of the change points. Despite the extra effort required to visually inspect the additional 295*3=885 images for each of the subcategory functions (versus the 354 images inspected for each of the ‘n’ and ‘distinct’ functions alone), there were relatively few additional change points identified.

Table 4-8. Change points identified in aggregation functions in categorical fields, for different aggregation functions relating to Plausibility.
n = number of values present, distinct = number of distinct values, subcat_n = number of values within each subcategory, subcat_perc = percentage of values within each subcategory

	Aggregation function			
Measure	**n**	**distinct**	**subcat_n**	**subcat_perc**
Across all aggregation functions:				
- No. of units	354	354	171*	171 *
- No. of distinct change points	2575	2575	2575	2575
Per aggregation function:				
- Units containing change points (%)	310 (88)	325 (92)	168 (98)	165 (96)
- No. of change points (%)	1100 (43)	1190 (46)	1011 (39)	863 (34)
Compared to distinct change points:				
- No. only in this aggregation function (%)	403 (16)	680 (26)	121 (5)	241 (9)
- No. in this and other aggregation functions (%)	697 (27)	510 (20)	890 (35)	622 (24)
- No. not in this aggregation function (%)	1475 (57)	1385 (54)	1564 (61)	1712 (66)
Compared to other aggregation functions:				
- No. also in n	-	308	489	195
- No. also in distinct	308	-	274	281
- No. also in subcat_n	489	274	-	586
- No. also in subcat_perc	195	281	586	-

* There were 57 data fields (containing a total of 295 subcategories) where the ‘subcat_n’ and ‘subcat_perc’ functions were calculated. For each of these data fields, the change points identified across all the subcategories were combined to form a set of overall change points for the aggregation function

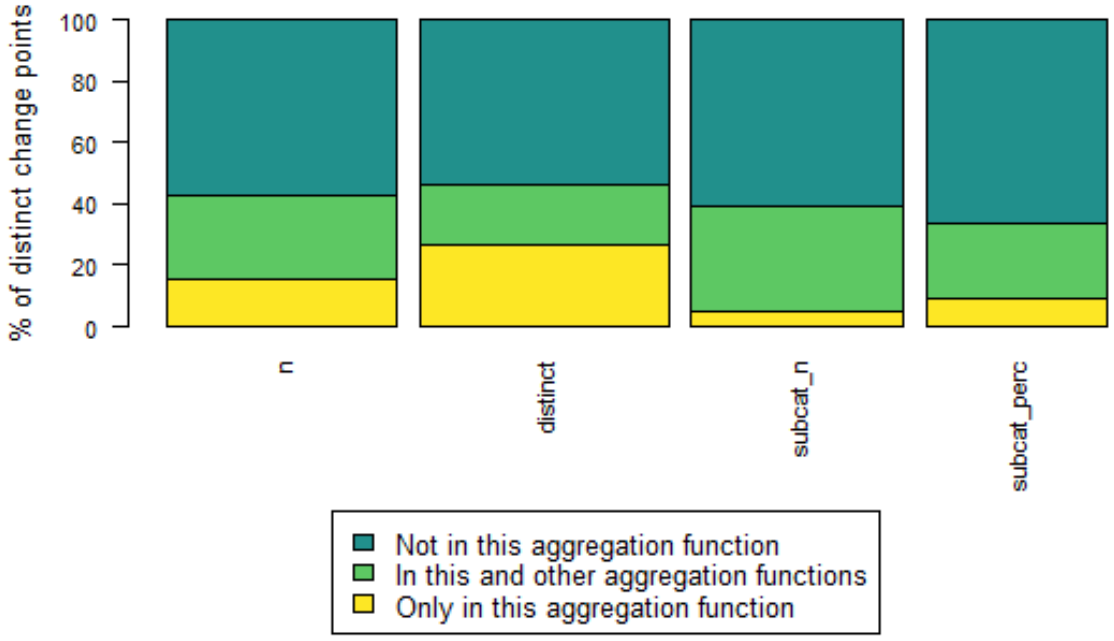


Figure 4-8. Percentage of the total distinct change points identified in categorical fields, for different aggregation functions relating to Plausibility.
Width of bars corresponds to the total number of change points identified using that aggregation function. Change points identified in different subcategories of a data field were combined into a single set of "overall" change points for the 'subcat' aggregation functions. n = number of values present, distinct = number of distinct values, subcat_n = number of values within each subcategory, subcat_perc = percentage of values within each subcategory.

4.4.2 Comparisons between aggregation granularities

There were 1842 images for each aggregation granularity of day, week or month. 4477 crowd-sourced change points were identified in 1260 (68%) images when the data was aggregated by day, 4277 change points in 1218 (66%) images when aggregated by week, and 3991 change points in 1209 (66%) images when aggregated by month.

Change points identified when using different granularities were judged to represent the same underlying change point if they were less than 11px from each other (see Methods Section 4.3.4.2). By doing this, 5799 distinct change points were identified.

Half of the change points (n=2918, 50%) were seen at all three granularity levels (see Figure 4-9), a sixth seen at two different granularities (960, 17%), and a third (1921, 33%) seen only in one granularity. While the daily granularity identified more change points than either the

weekly or monthly granularities, 1361 (23%) change points would have been missed if it had been used alone.

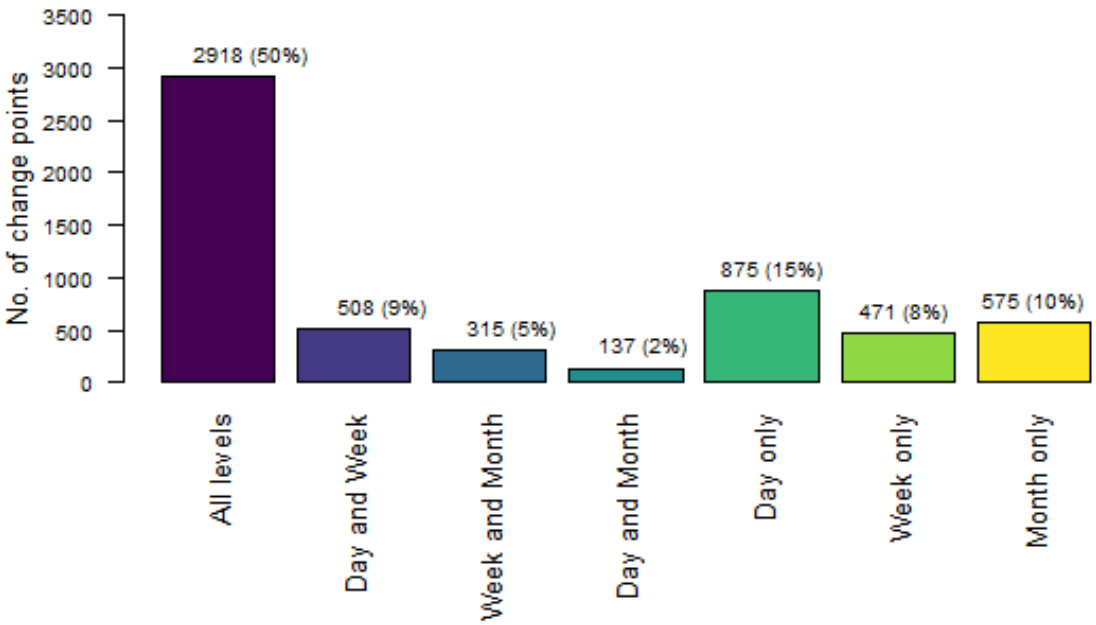


Figure 4-9. Number of change points identified at different granularity levels. Percentages in brackets are out of the 5799 distinct change points identified across all images.

The distribution of change points seen at the various granularity levels was fairly similar across the aggregation functions ‘n’, ‘missing_n’, ‘missing_perc’, ‘sum’, ‘midnight_n’, ‘midnight_perc’, ‘maxlength’, ‘subcat_n’ and ‘subcat_perc’, where the greatest proportion of change points were identified at either all three aggregation granularities or at the daily granularity, see Figure 4-10. For the ‘min’, ‘max’, ‘mean’, ‘median’, ‘minlength’, and ‘distinct’ aggregation functions, change points were more visible at the monthly than the daily granularity.

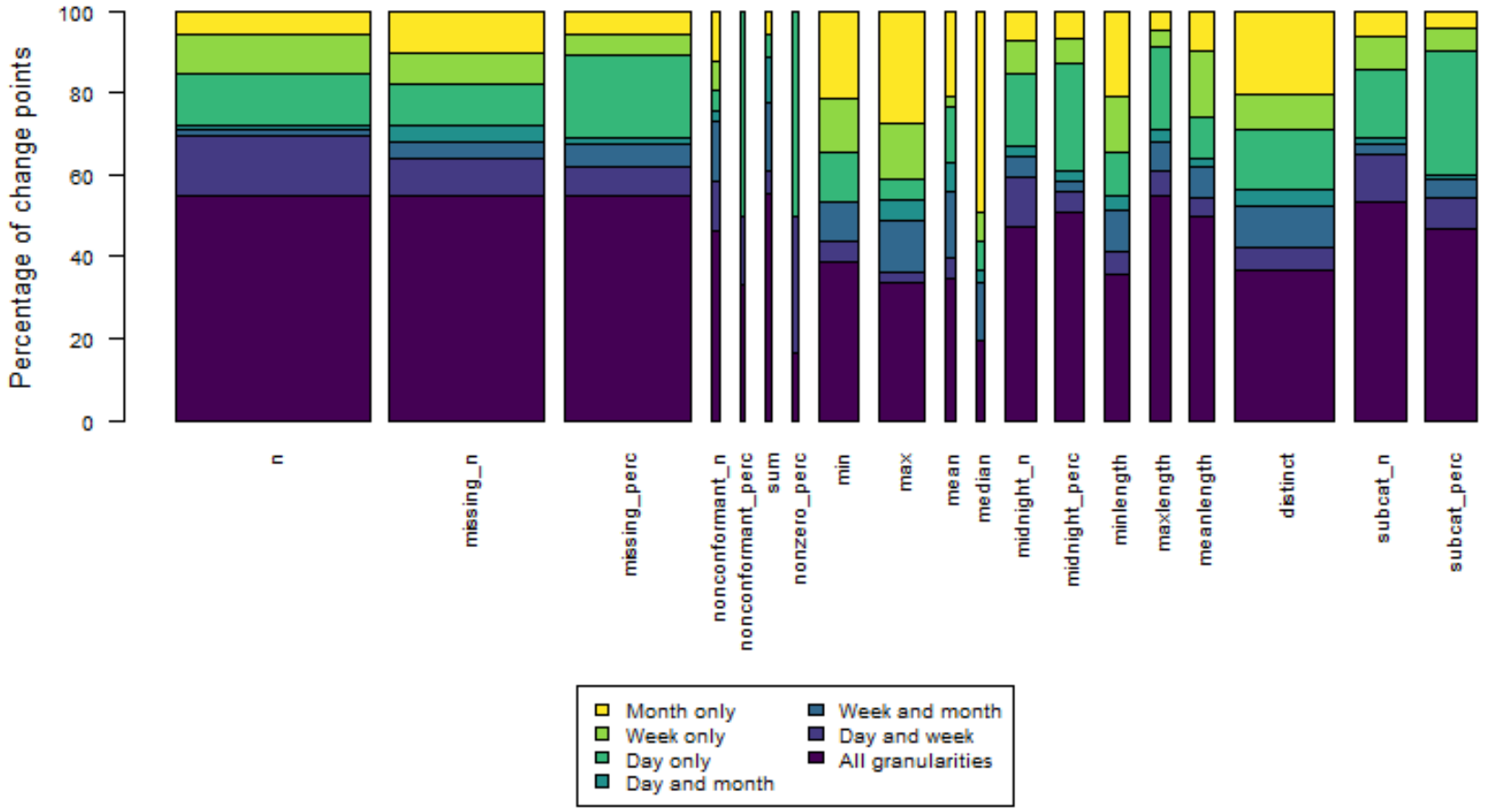


Figure 4-10. Percentage of change points identified at different aggregation granularities, by aggregation function.
Width of bars corresponds to the number of data fields containing change points identified using that aggregation function. n = number of values present, missing_n = number of missing values, missing_perc = percentage of missing values, nonconformant_ n = number of non-conformant values, nonconformant_perc = percentage of non-conformant values, sum = sum of duplicate records removed, nonzero_perc = percentage of records which had been duplicated, min = minimum value, max = maximum value, mean = mean value, median = median value, midnight_n = number of values with no time element, midnight_perc = percentage of values with no time element, minlength = minimum string length, maxlength = maximum string length, meanlength = mean string length, distinct = number of distinct values, subcat_n = number of values within each subcategory, subcat_perc = percentage of values within each subcategory

4.4.3 Comparisons at data extract level

Considering each data extract/aggregation granularity combination to be a unit, there were 24 units where aggregation functions were calculated using both the individual fields method (i.e. combining change points identified across multiple images representing each data field) and the combined data method (i.e. the change points from the single image representing the combined data across the data extract). There were 1595 distinct change points identified. 405 (25%) of these change points were detected using the combined data method, and 1583 (99%) were detected using the individual fields method (see Table 4-9, Figure 4-11). Only 12/1595 (1%) change points were visible using the combined data method but not the individual fields method, suggesting that visually inspecting time series representing individual data fields is far more effective at uncovering change points than inspecting the combined values of all data fields.

Table 4-9. Change points identified in aggregation functions applied at the data extract level, either by inspecting images from individual data fields or in a single image of combined data.
n = number of values present, missing_n = number of missing values, missing_perc = percentage of missing values, nonconformant_ n = number of non-conformant values, nonconformant_perc = percentage of non-conformant values

	Aggregation function					
Measure	**n**	**missing_n**	**missing_perc**	**nonconformant_n**	**nonconformant_perc**	**Total**
No. of units	24	24	24	24	24	120
Units containing change points (%)	24 (100)	24 (100)	24 (100)	9 (38)	6 (25)	87 (72)
No. of distinct change points	385	689	460	50	11	1595
Change points in individual fields only (%)	297 (77)	548 (80)	330 (72)	4 (8)	11 (100)	1190 (75)
Change points in combined data only (%)	0 (0)	6 (1)	1 (0)	5 (10)	0 (0)	12 (1)
Change points in both (%)	88 (23)	135 (20)	129 (28)	41 (82)	0 (0)	393 (25)

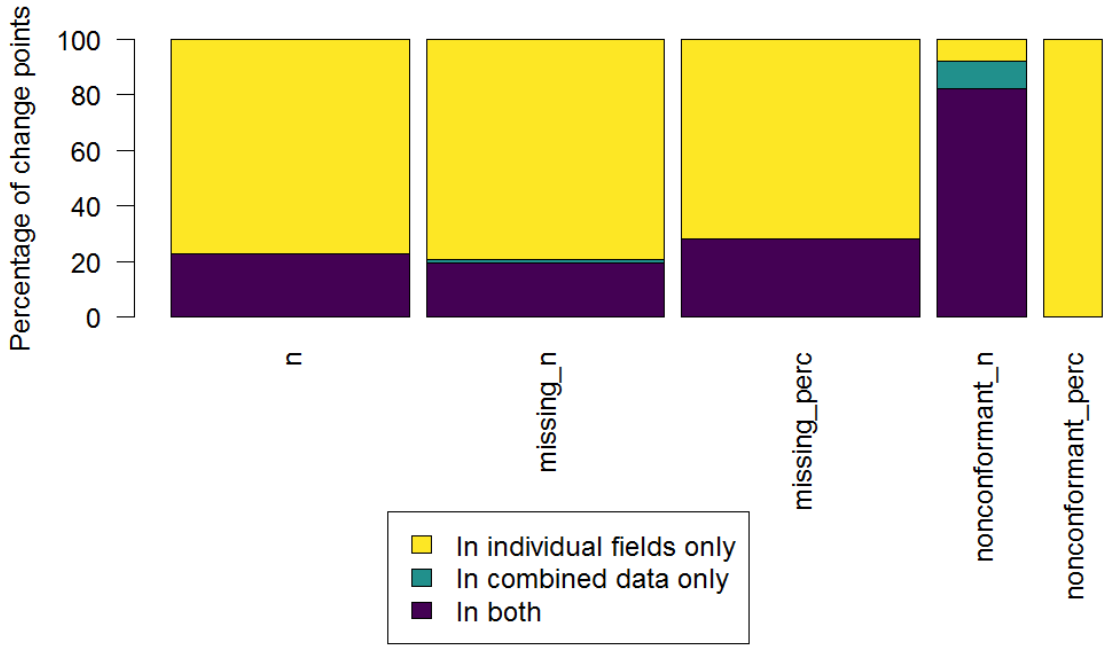


Figure 4-11. Percentage of change points seen in images from individual data fields versus when values from all data fields are combined into a single image.
Width of bars corresponds to the number of units that contained change points. n = number of values present, missing_n = number of missing values, missing_perc = percentage of missing values, nonconformant_ n = number of non-conformant values, nonconformant_perc = percentage of non-conformant values.

4.4.4 Best performing methods

Based on the findings above, the aggregation methods which would likely offer the best chance of identifying any change points across each type of data field and data quality dimension in an EHR dataset are:

- Aggregating data from individual data fields (as opposed to combining all the values across the data extract)
- Using a daily granularity
- Using the following aggregation functions:
    - Completeness: ‘missing_n’
    - Conformance: ‘nonconformant_n’
    - Plausibility:
        - Duplicate records: ‘sum’

- All data field types: ‘n’
- Numeric fields: ‘median’
- Datetime fields: ‘max’
- Uniqueidentifier fields: ‘minlength’
- Categorical fields: ‘distinct’

For a data extract containing 15 data fields, this limited set of aggregation methods would result in 31-58 different images to generate and visually inspect, versus up to 1872 different images if every method considered in this study was used. For each additional aggregation function included for inspection, one additional image per data field it related to would need to be generated and inspected (except for the ‘subcat’ aggregation functions which would add an image for each subcategory of the data field).

4.5 Discussion

Change points were identified within every single aggregation function and every aggregation granularity used in the study, with every aggregation method except the percentage of records which had been duplicated (nonzero_perc) finding at least one change point that was not identified using any other method.

More change points were identified when using a daily granularity than a weekly or monthly granularity, though 23% of change points would have been missed if the data had only been aggregated by day.

More change points were identified using certain aggregation functions compared to others, and in general, functions measuring frequencies were more effective than functions measuring proportions. For example, 85% of all change points relating to missingness were identified using the number of missing values (missing_n), versus 53% identified using the percentage of missing values (missing_perc). The sum of duplicate records removed (sum) and the number of non-conformant values (nonconformant_n) aggregation functions

appeared to unilaterally capture almost all the change points related to duplicate records and conformance respectively. However, there were relatively few change points present in these two categories, and the actual number of duplicate records/non-conformant values in the data were very small, so it is not clear whether this performance is generalisable to other data sources which might have larger numbers of duplicate records and/or non-conformant values.

When looking within specific data field types, a maximum of 47% of change points were identified using any single aggregation function. The percentage of values with no time element (midnight_perc) and the number of values within each subcategory (subcat_n) aggregation functions were arguably the least effective, as they only identified 3% and 5% of the total change points (in datetime and categorical fields respectively) that were not also identified using any of the other aggregation functions. Slightly surprisingly, more change points were identified using the median numeric value (median) and the distinct number of categories (distinct) than in the other aggregation functions within their respective data field types, though this could potentially be explained by a higher level of discretisation of the resulting summary values, and hence the chance that a significant proportion of these change points are false positives, as was seen in Chapter 3. Similarly, some of the change points seen in the minimum, maximum, and mean string length (minlength/maxlength/meanlength) aggregation functions for uniqueidentifier data types could merely be the point when an incremental integer identifier goes from e.g. 999 to 1000, and hence should be ignored.

Only 1% of change points would have been missed if data values had not been combined across entire data extracts, suggesting that doing this adds little value.

There is very little literature relating to the assessment of temporal data quality in EHRs, and of the five articles found during the literature search in Chapter 2, none compared different aggregation methods for their effectiveness at revealing change points. While Looten et. al[68] experimented with different rolling windows of 15, 30, or 60 days when creating their daily

time series, they settled on a rolling window of 60 days without conducting a formal assessment, and the resulting time series were generated at a single granularity of one (aggregated) value per day. In addition, only a single aggregation function was used for each type of change point being detected (namely the median value for changes in level, and a ratio of last digits for changes in discretisation). There has clearly been very little attention given to the practicalities of temporal data quality assessment, even when it has been acknowledged that temporal data quality is important enough to be addressed. This study is therefore a much-needed first step in helping researchers to make decisions on how to approach the task.

While there were some aggregation methods which appeared less effective than others at detecting change points, it is not necessarily true that this means it is reasonable to exclude them altogether, since different aggregation methods will inevitably be more or less relevant to different study designs. For instance, where it is known in advance that a particular transformation or measure will be used to create a variable or end point (e.g. where a statistical model will be used that includes the daily number of inpatient admissions as a denominator), these measures should of course be checked explicitly. Also, while the most common types of aggregation functions for checking data quality have been included, there may be other data transformations that could be relevant to particular study designs, and arguably a further screening for change points in a final “cleaned and transformed” dataset would be beneficial to ensure that no new artefacts have been introduced.

While a minimal set of aggregation methods has been proposed based on the 8 EHR data extracts studied here, and which could potentially be used if researcher time is a limiting factor, it remains up to the individual researcher to decide how many images they are willing to generate and inspect. To reduce this grey area, it would be useful if guidelines were created that suggest a minimum set of checks that should be relevant to all studies (such as checks for missingness), as well as more targeted checks depending on how the data will be used.

If the identification of change points can be automated then one might as well use all the methods, as arguably the more change points that are found the better, so long as this is balanced against the manual effort required to filter out any false positives. It is not yet known how well change point detection might be able to be automated, but at the very least, the simple automation of the generation of images for visual inspection would reduce much of the effort required on behalf of the researcher, and if these were compiled in a single report that could be included as a supplementary file in a publication, that would aid transparency considerably.

An open question is what the researcher should do when they find change points in their data, and if they end up finding thousands of them, as happened here, how might they decide which ones are important and which are not. Ultimately, the answer will be subjective based on the study design, but if there are very large numbers of change points, there is a risk that a researcher will be overwhelmed by the scale of the task and decide not to investigate further. Nevertheless, this is not a reason not to do it, and at least by reporting their assessment of temporal data quality, alongside any published results, this would make any limitations of the data clearer.

4.6 Conclusion

Change points were found within every single aggregation function and every aggregation granularity used, though more change points were found using certain aggregation methods than others. More change points were found using a daily granularity than a weekly or monthly granularity, and functions measuring frequencies were generally more effective than functions measuring proportions. However, if these aggregation methods were used alone, many change points would be missed. To maximise the number of aggregated time series a researcher is willing to inspect for data quality issues, it would be useful if the task could be automated in some way. Ideally, there would be an automatic identification of the change points themselves, but if not, then at least the simple generation of images for visual

inspection would make a difference. Further, if the results could be automatically compiled in a single report that could be included as a supplementary file in a publication, that would aid transparency considerably.

CHAPTER 5: Automated methods for change point detection

5.1 Chapter summary

Change points can be identified very easily by the human eye, but this task can be prohibitively labour-intensive when datasets are large or are being updated frequently.

This chapter assesses if change points can be identified successfully without human intervention, by testing a range of automated change point detection methods against change points identified by human visual inspection.

All change point detection methods found in the CRAN library of R packages were assessed for suitability for automation and for diversity of methodology. Ten methods were shortlisted and tested for speed and accuracy against change points identified by crowd-sourced visual-inspection from 8 different EHR data extracts from a large UK hospital group, covering patient, laboratory, and clinician activity. Parameter tuning was conducted on 2890 time series and final performance assessed on a reserved test set of 1234 time series.

Four of the short-listed methods failed initial implementation. Performance of the remaining six methods was poor, with maximum sensitivity of 36.9% (95% CI 35.2, 38.8), specificity of 100% (100, 100), PPV of 51.6% (49.4, 53.8), and NPV of 99.9% (99.9, 99.9).

Currently-available change point detection methods do not perform well on real-world EHR data. There is a lot of potential scope for the development of new methods.

5.2 Background

There has been very little previous research on detecting change points in EHRs, let alone on detecting them automatically (see Section 2.5 for details). However, many different change point detection methods exist in the wider scientific literature, covering fields such as

finance, ecology, genomics, as well as general statistical methods. These methods can be categorised in a number of different ways, including:

- **Type(s) of change point detected**: e.g. change in level, trend, or variability. (As a reminder, the Zooniverse volunteers were asked to identify 5 different types of "unnatural-looking" changes, i.e. sudden changes in level, variability, trend, presence/absence of data points, or irregular outliers, and these types of changes may not map perfectly onto the types of changes that existing change point detection methods are designed to look for)
- **Statistical data model:** the model used to approximate the data-generating process. This can be based on probability distributions or on autoregressive time series models, and is related to the types of change point detected e.g. a piecewise-constant mean model will detect sudden changes in level assuming zero trend between change points, while a piecewise-linear model will detect changes in trend assuming a linear trend between change points. There may further be an error model for the "noise" around the signal e.g. independent identically distributed Gaussian/Poisson/nonparametric distribution
- **Segmentation method:** this is the method used for detecting multiple change points, e.g. Binary Segmentation (BS), Segment Neighbourhood (SN), Pruned Exact Linear Time (PELT), Wild Binary Segmentation (WBS)
- **Online/Offline**: Online methods are designed for use on fast streams of incoming data, where the emphasis is usually on early detection and where historic data points are ignored. Offline methods try to find the best set of change points within the context of the entire dataset
- **Frequentist/Bayesian**: Frequentist methods use classical statistics to model the data and find change points. Bayesian methods typically employ stochastic methods such as Markov Chain Monte Carlo to explore the parameter space and approximate

the underlying distribution. This means they depend on appropriate choice of start-up parameters and sufficient “mixing” in the burn-in period, and can be non-deterministic

- **Univariate/Multivariate**: As well as looking for change points in univariate time series, some change point detection methods can look for change points across multiple time series simultaneously

In order to ascertain which change point detection methods might work on EHRs, various factors need to be considered. EHR data is high in volume, multidimensional, may contain missing data or outliers, includes nonlinear cycles and trends (as populations and disease aetiology change), and can have multiple change points (in more than one direction). This makes it a challenge for traditional change point detection methods which typically assume a piecewise-constant mean and often only a single change point. Furthermore, many papers which describe new methods do not publish any software code to go with them, making it extremely difficult for the average researcher to implement them.

However, if an automatable method could be found, this would make the task of identifying change points much less onerous on researchers, as well as make the process more consistent and reproducible. In order for a method to be adopted, it will need to be shown to perform adequately compared to human visual inspection.

5.3 Methods

5.3.1 Package selection process

The R software environment includes a long-established online repository[87] for user-contributed packages (CRAN) with already dozens of packages related to change point detection. Some of these have been developed with a specific field of application in mind, but currently none of them are specifically aimed at EHR data quality. The repository was searched for the following keywords – “change”, “homogenization/homogenisation”, “harmonization/harmonisation”, “break”, “segment”, “edge”, “jump”, “switch”, “interrupted”, “anomaly/anomalies” – to cover the terms: change point, edge detection, jump detection,

break detection, interrupted time series analysis, segmented regression analysis, Markov switching, data homogenization, and anomaly detection. The term “outlier” was not included since outlier detection methods would not be expected to detect other types of change point (whereas change point detection methods may potentially detect outliers as well). The titles and descriptions were reviewed manually to confirm if the package was relevant or not.

To be eligible for inclusion, the change point detection methods needed to satisfy the following essential criteria:

- Identifies multiple (0+) unknown change points
- Can be implemented in an automated fashion (i.e. not designed to be used interactively)
- Does not require specific domain knowledge (e.g. accepts a simple time series as opposed to a domain-specific format, and suggests suitable default parameters where appropriate)
- Univariate, so that any change points can be compared directly to visually-inspected consensus labels from the Zooniverse
- Deterministic, since one of the eventual purposes is reproducibility

The ability to handle trended data can potentially be dealt with by applying appropriate transformations to the time series prior to change point detection so was not considered essential.

Since it was important to test a broad range of different methods rather than multiple similar methods, eligible packages were classified according to the following categories (as described in Section 5.2):

- Type(s) of change point they claim to detect
- Statistical model
- Segmentation method
- Online/Offline

Each package was rated for suitability for inclusion based on covering a range of methods (above), and also taking into account the following:

- Quality of documentation, e.g. is the methodology described clearly, is there a corresponding published paper
- Package appears to be actively used and maintained, in order to be able to rely upon the package in the future

5.3.2 Assessment of performance of automated methods

The 5526 crowd-sourced labelled time series obtained from Chapter 3 were split randomly into a tuning and a test set, at a 70/30 ratio, balanced across data extracts and aggregation granularities. All exploration, implementation, and tuning of the various R packages was done using the tuning set.

Some time series contained NAs (a special datatype within R which denotes a “missing” value), which are created when there are no records in the relevant day/week/month. Since most change point detection methods are unable to handle NAs, these time series were excluded. Therefore, the final tuning set contained 2890 time series, and the final test set contained 1234 time series.

The accuracy of the detection methods was assessed by comparing the locations of the change points they identified to the Zooniverse labels created in Chapter 3, using the same method previously used to assess the accuracy of the density-based clustering algorithm against the expert labels, namely:

- **true positive**: an auto-detected change point is within the ‘minimum distance cut-off’ of a crowd-sourced label
- **false positive**: an auto-detected change point is present, but no crowd-sourced label lies within the ‘minimum distance cut-off’ of it
- **false negative**: a crowd-sourced label is present, but no auto-detected change point lies within the ‘minimum distance cut-off’ of it

- **true negative**: total number of time points minus the sum of above 3 categories,

where the ‘minimum distance cut-off’ (i.e. the minimum distance allowed between two change points for them to be considered as *distinct* change points) was calculated to be equivalent to the 7-pixel distance used previously.

Again, in order to avoid double-counting, the following additional rules were enforced:

- when there were ‘x’ auto-detected change points close to one crowd-sourced label, this counted as one true positive and zero false positives
- when there were ‘x’ crowd-sourced labels close to one auto-detected change point, this counted as ‘x’ true positives and zero false negatives

These values were then used to calculate Matthew’s Correlation Coefficient[82] (MCC), a summary metric which takes into account the imbalanced distribution of positive versus negative calls. In addition, the sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV) were calculated.

A representative subset of 300 time series was selected from the tuning set to use for initial screening of potential detection methods. This initial set was constructed by selecting one time series at random from each aggregation granularity/aggregation function combination, then filling the remaining spaces with a random draw from the remaining time series.

This initial set of 300 was used to test the speed of each shortlisted detection method and to record any errors produced. For the detection methods that ran without error and within a reasonable amount of time, these same 300 time series were used to explore the parameter space, narrowing down a suitable range of parameters to use in a grid search to find the optimal parameters for the entire tuning set. This same set of 300 time series was also used to explore potential transformations that might reduce model violations (e.g. inappropriately using a piecewise-constant model when there is an underlying trend) and improve accuracy.

A sample of 25 time series was taken completely at random from the 300, and the discrepancies between the crowd-sourced labels and each automated method (using the parameters which performed the best for the set of 300 time series, selected based on MCC) was analysed.

While in Chapter 3 the accuracy of the volunteers at identifying the binary existence of *any* change points in an image was also assessed (in addition to the *locations* of change points), it was not considered appropriate in this evaluation. This is because it is possible that an automated method could identify change points in completely the wrong places but still get the overall binary call correct, so while it might be useful to be able to flag to a researcher whether or not a particular image likely contains a change point in an automated manner, it is nevertheless still important that the automated flag is based on identifying the locations of the change points correctly.

A grid search of the adjustable parameters for each automated method was conducted on the full tuning set of 2890 time series, over the range of values identified using the 300 time series, both for the raw time series values and after any appropriate transformations identified. MCC was used to select the best parameters and/or transformations (where appropriate) for each method. This was done overall, by aggregation granularity, and by aggregation function. Final performance of each method was assessed on the reserved test set of 1234 time series, using the best parameters identified from the full tuning set.

5.4 Results

5.4.1 Package selection

The package titles in the CRAN repository (n=17,484) were searched on 29 April 2021 for keywords relevant to change point detection. This resulted in 273 matches. After reviewing the titles, 130 different packages appeared potentially relevant. Reviewing the package documentation reduced this further to 25 that satisfied the ‘Essential’ criteria (see Appendix C.1).

The majority of packages (14/25) only looked for changes in level. Four packages looked for changes in trend, though none of them distinguished between smooth and abrupt changes. Six looked for changes in variability and two for outliers. There were no methods that looked for changes in presence/absence of data points.

Ten different methods from eight different packages were shortlisted for implementation, and were selected to cover a range of change point types, statistical data models and segmentation methods (see Table 5-1, with more detail about individual methods given later). All 10 methods were run on the initial set of 300 time series, using default parameters where appropriate. Two of the methods (from the *bfast* and *strucchangeRcpp* packages) produced errors and so were not investigated further. Of the remaining eight methods, three processed all 300 time series in under a minute (cpt.meanvar, cpt.var, DeCAFS), three within 10 minutes (cpt.np, OcpPewma, OcpTsSdEwma), and two took over 2 hours (offlineChange, trendsegmentR) (Table 5-2). All but one of the eight methods (OcpTsSdEwma) reported vastly more change points than the Zooniverse volunteers.

Table 5-1. Summary of 10 shortlisted change point detection methods from the CRAN repository of R packages

Package Name and Title	Method name	Type(s) of change	Statistical data model	Segmentation method	Online/Offline
bfast - Breaks For Additive Season and Trend (BFAST), v1.5.7	bfast[88]	Trend and/or Seasonality	Piecewise linear + piecewise seasonality + Gaussian noise	Iterative decomposition + Bai & Perron (2003)[89]	Both
changepoint - Methods for Changepoint Detection, v2.2.2	cpt.meanvar[90]	Level and/or Variability	Gaussian distribution with piecewise constant mean and variance	PELT (Pruned Exact Linear Time)	Offline
changepoint - Methods for Changepoint Detection, v2.2.2	cpt.var[90]	Variability only	Gaussian distribution with constant mean and piecewise constant variance	PELT (Pruned Exact Linear Time)	Offline
changepoint.np - Methods for Nonparametric Changepoint Detection, v1.0.3	cpt.np[91]	Level and/or Variability	Piecewise constant nonparametric distribution	ED-PELT (Empirical Distribution - Pruned Exact Linear Time)	Offline
DeCAFS - Detecting Changes in Autocorrelated and Fluctuating Signals, v3.2.3	DeCAFS[92]	Level	Piecewise locally-fluctuating mean (random walk plus autocorrelated noise)	DeCAFS	Offline

Package Name and Title	Method name	Type(s) of change	Statistical data model	Segmentation method	Online/Offline
offlineChange - Detect Multiple Change Points from Time Series, v0.0.4	ChangePoints[93]	Level	Piecewise autoregressive time series	Multiwindow	Offline
otsad - Online Time Series Anomaly Detectors, v0.2.0	OcpTsSdEwma[94]	Level/ Outlier	Nonparametric	TSSD-EWMA (Two-Stage Shift Detection based on Exponentially Weighted Moving Average)	Both
otsad - Online Time Series Anomaly Detectors, v0.2.0	OcpPewma[95]	Level/ Outlier	Nonparametric	PEWMA (Probabilistic Exponentially Weighted Moving Average)	Both
strucchangeRcpp - Testing, Monitoring, and Dating Structural Changes: C++ Version, v1.5-3-1.0.2	breakpoints[96 97]	Level Trend	Linear regression with piecewise Gaussian noise	Bai & Perron (2003)[89]	Offline
trendsegmentR - Linear Trend Segmentation and Point Anomaly Detection, v1.0.0	trendsegment[98]	Trend Outlier	Piecewise linear signal Gaussian errors	4-step wavelet-transform based	Offline

Table 5-2. Initial performance results for 8 methods which ran without error (across 300 time series). The total number of change points identified by the Zooniverse volunteers was 742.

Method	Total time taken (s)	No. of change points reported (using default parameters)
changepoint - cpt.meanvar	1.74	26101
changepoint - cpt.var	1.76	2290
changepoint.np - cpt.np	485	2141
DeCAFS - DeCAFS	32	105597
offlineChange - offlinechange	9753	2270
otsad - OcpPewma	285	28410
otsad - OcpTsSdEwma	205	288
trendsegmentR - trendsegment	7979	101653

Since one of the main purposes of automation is to save researcher time, the two methods which took over 2 hours to run were not considered further, particularly since their initial performance appeared no better than the fast-running methods. The remaining six automated methods were then investigated for opportunities to improve their performance.

A sample of 25 time series was taken completely at random from the 300, to use in a discrepancy analysis between the crowd-sourced labels and each automated method. Labels for the automated methods were produced using the parameters found to perform best during the exploration of the parameter space (based on MCC, see Methods). Results are shown in Table 5-3. False positive calls were commonly found where there was a smooth change in level (for the methods cpt.meanvar, cpt.var, cpt.np, and OcpTsSdEwma), see Figure 5-1A, or where the time series consisted of only a small number of discrete values which resulted in small variations appearing like outliers (for the methods cpt.meanvar and OcpPewma), see Figure 5-1B. For the methods cpt.meanvar, cpt.var, DeCAFS and OcpPewma, there were a large number of false positives with no clear explanation, Figure 5-1C, and in particular the DeCAFS method sometimes called change points consecutively for entire blocks of time, leading to the very high number of false positives for this method, Figure 5-1D. False negative calls were most often from missed

changes in level or outliers, Figure 5-1E, except for the DeCAFS method which produced relatively few false negatives.

Table 5-3. Discrepancies between crowd-sourced labels and change points reported by six automated methods, based on a random sample of 25 time series.

	cpt.meanvar	**cpt.var**	**cpt.np**	**DeCAFS**	**OcpTsSdEwma**	**OcpPewma**
FALSE POSITIVES	**85**	**91**	**61**	**14844**	**29**	**118**
Smooth change in level	24	23	16	0	21	0
Smooth change in variability	0	0	5	0	0	0
Smooth change in level and variability	2	6	35	8	0	0
Small variations appearing like outliers	28	3	0	0	0	27
No clear reason	27	55	3	14836	8	91
Arguable match to nearby label	1	1	2	0	0	0
Arguably a change point missed by volunteers	3	3	0	0	0	0
FALSE NEGATIVES	**36**	**40**	**40**	**5**	**57**	**47**
Sudden change in level	19	17	13	2	27	26
Sudden change in variability	0	0	1	0	1	0
Sudden change in trend	0	1	3	0	3	3
Outlier	9	12	17	1	17	8
Arguable match to a nearby change point	1	1	2	0	0	0
Arguably not a real change point	7	9	4	2	9	10

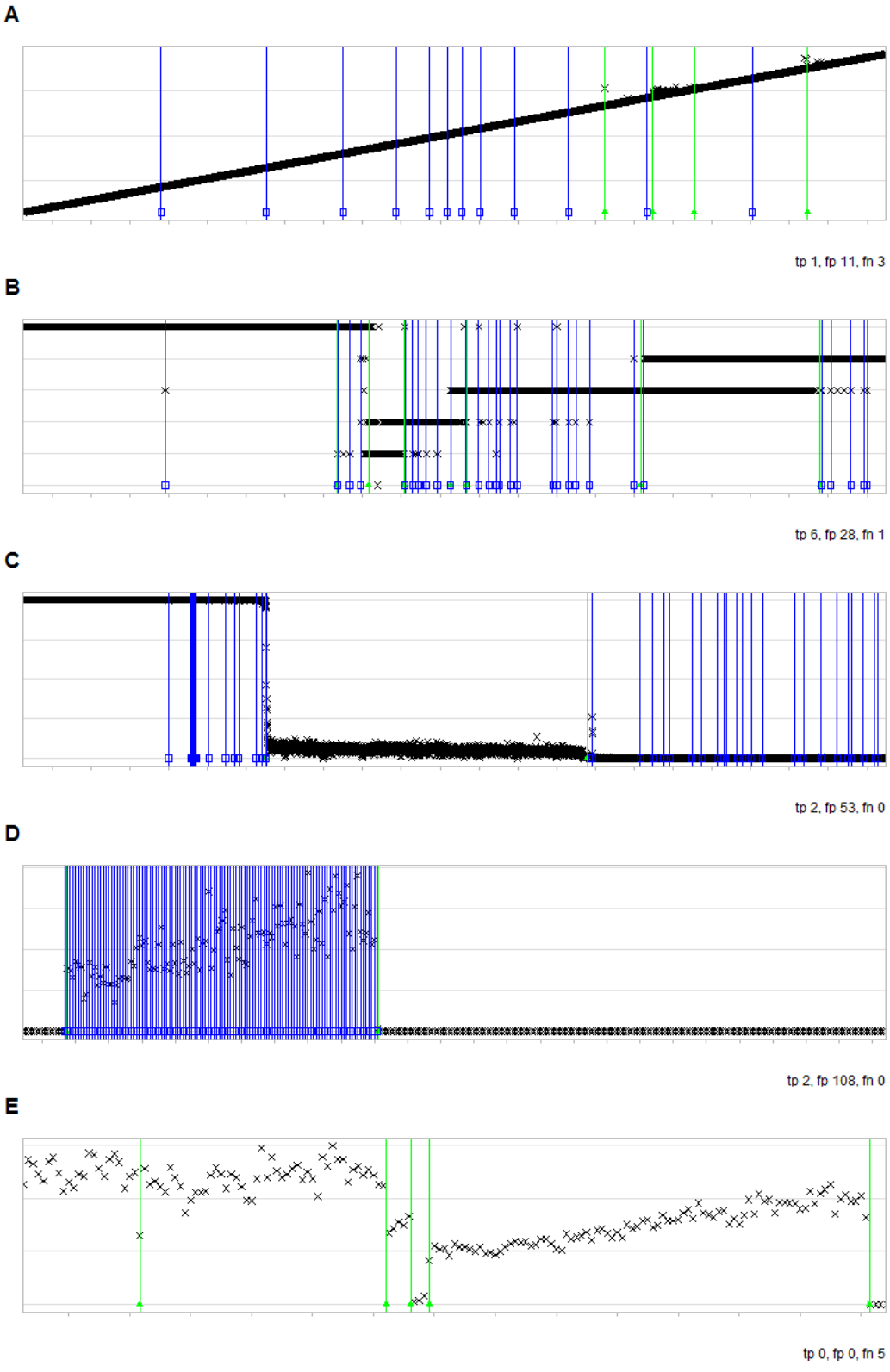


Figure 5-1. Examples of common errors in identifying change points using automated methods. Crosses denote the time series values, green lines with triangles are labels drawn by the volunteers, blue lines with squares are change points called by the automated methods. tp = true positives, fp = false positives, fn = false negatives. A. False positive calls where there was a smooth change in level (cpt.var), B. False positive calls where the time series consisted of only a small number of discrete values which led to small variations appearing like outliers (cpt.meanvar), C. False positive calls with no clear explanation (OcpPewma), D. The DeCAFS method sometimes called change points consecutively for entire blocks of time, E. False negative calls were most often missed changes in level or outliers (OcpPewma)

5.4.2 Exploration of potential transformations

There were 3 types of false positive calls which had the potential to be mitigated by finding a suitable transformation prior to applying change point detection, namely: smooth changes in level, smooth changes in variability, and smooth changes in both level and variability. These formed 140 of the false positive calls investigated in detail above. The most common of these was smooth changes in level (84/140 (60%)), which affected the methods cpt.meanvar, cpt.var, cpt.np, and OcpTsSdEwma. The simplest transformation for removing smooth changes in level is to take the first difference (i.e. subtracting the n-1th value from the nth value). Another option is to subtract a smoothing function from the raw values (such as a moving average), though this can introduce new artefacts in the data such as "stretching" the presence of a change in level and causing a single change point to appear like multiple change points (see Figure 5-2). While taking the first difference performs better in the above situation in that it isolates the change point to the immediate vicinity, in other situations it can remove features of the data, including sudden changes in level (see Figure 5-3), thus trading off a decrease in false positives with an increase in false negatives.

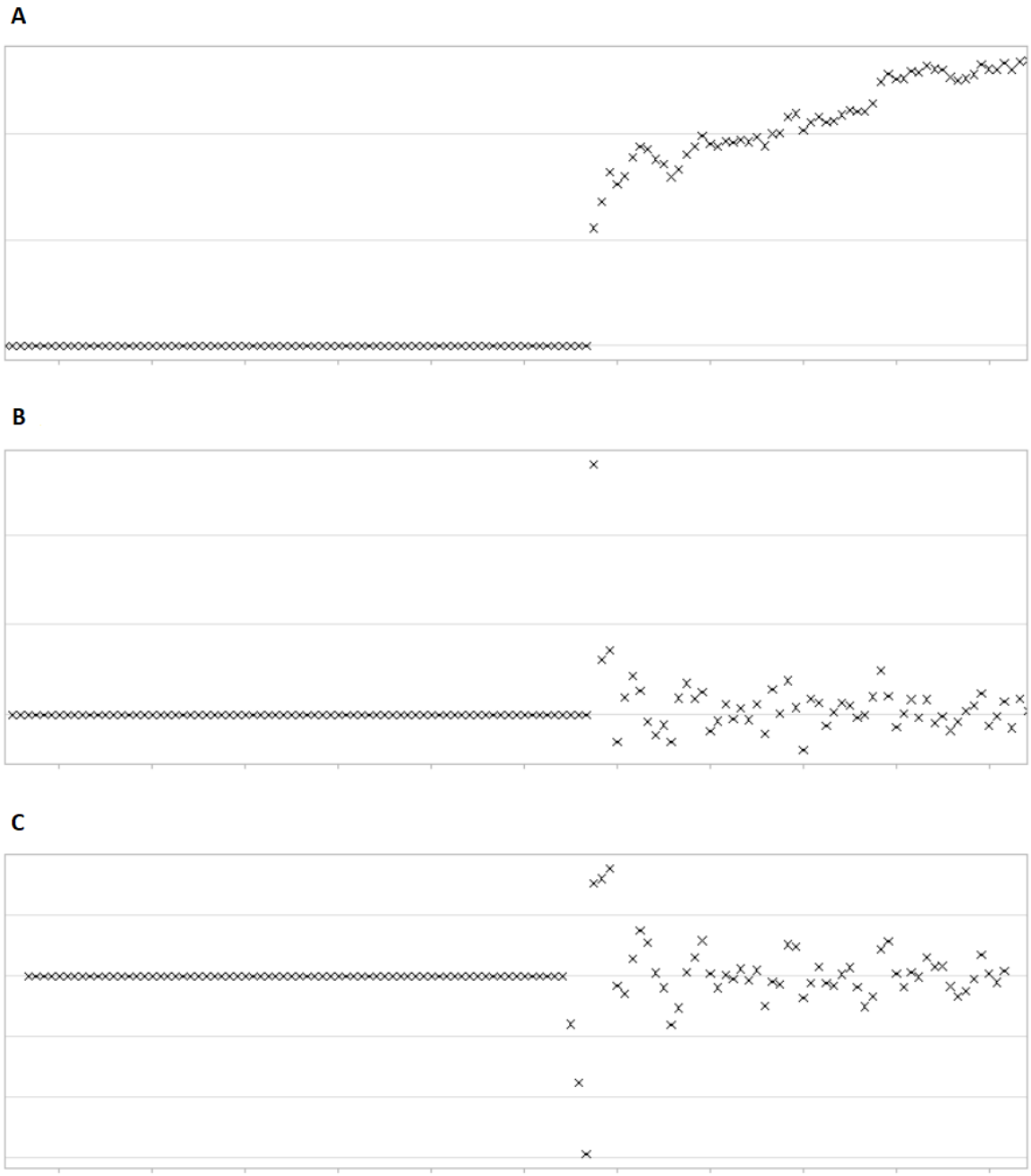


Figure 5-2. Effects of applying transformations to remove smooth changes in level.
A. *Raw time series. B. Taking the first difference of the time series in A neatly removes the smooth change in level seen on the right hand side of the plot. C. Subtracting a simple moving average of span 7 from the time series in A also removes the smooth change in level on the right hand side of the plot, but "stretches" the sudden change in level seen in the middle across 6 time points.*

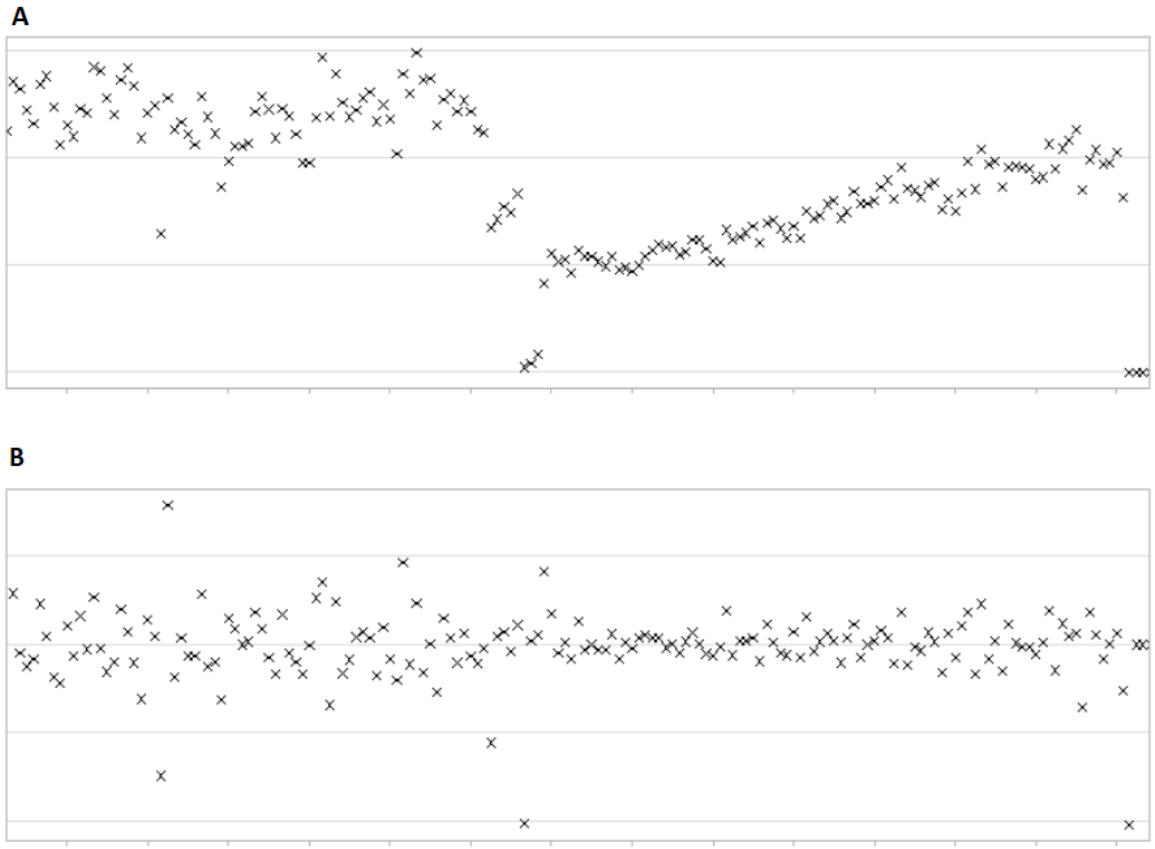


Figure 5-3. Loss of artefacts after applying a transformation to remove smooth changes in level.
A. Raw time series. B. Taking the first difference of the time series in A removes the smooth change in level seen on the right hand side, but also removes the 3 distinct changes in level slightly left of the centre

There were also many false positives caused by a smooth change in both level and variability (51/140 (36%)), though the majority of these were for just one method – the cpt.np method (35/51 (69%)). Changes in variability are harder to adjust for since typical transformations such as taking the logarithm or the square root can result in either over or under-adjustment depending on the amount of heteroskedasticity present in a particular time series, as well as requiring an additional transformation to ensure that all values are greater than (or equal to) zero. A Box-Cox transformation could potentially be used, specific to each time series, but this is less predictable, still does not guarantee a perfect transformation, and also adds to the run-time, see Figure 5-4.

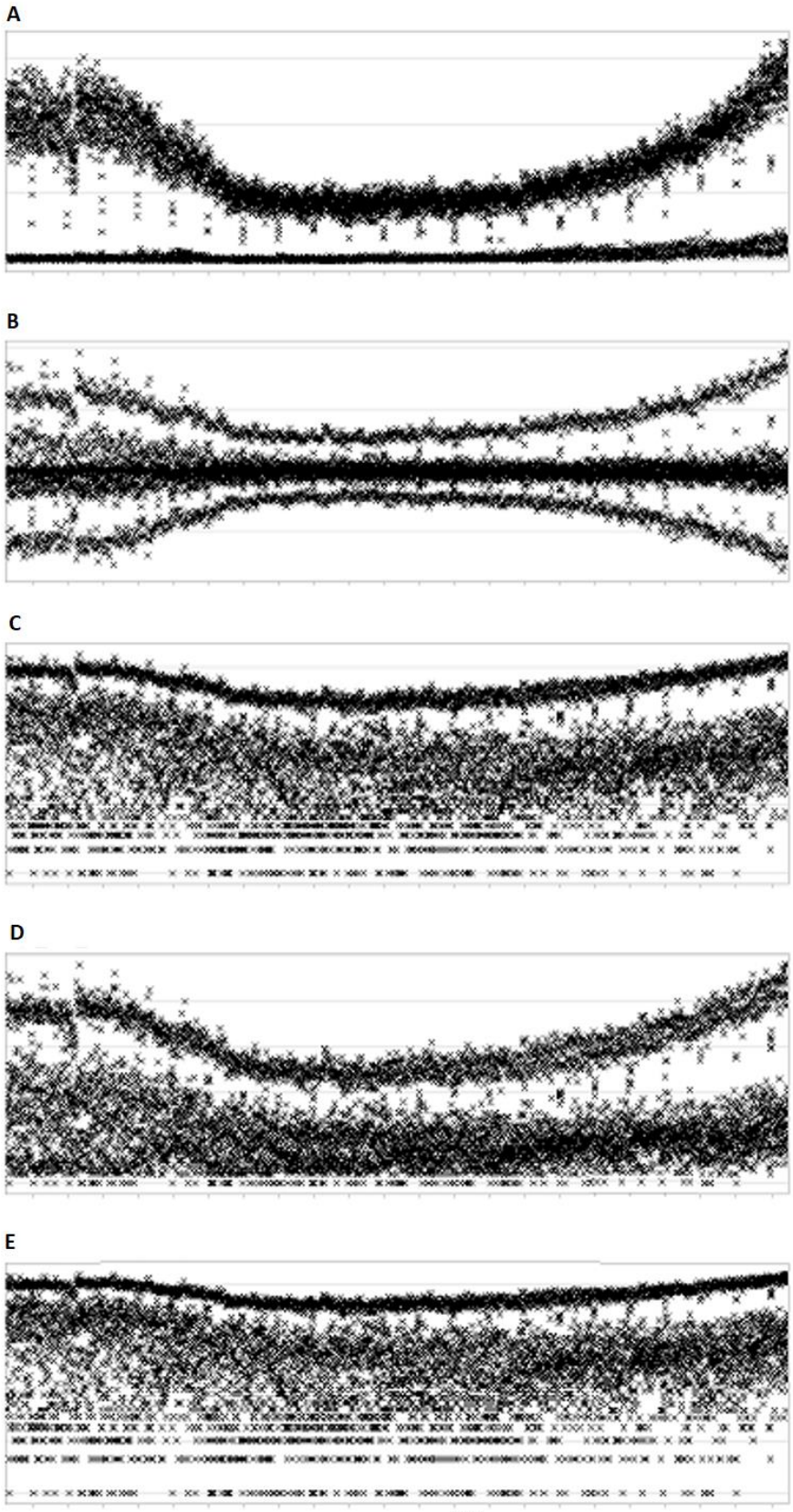


Figure 5-4. Effect of applying a variety of transformations to remove a smooth change in both level and variability. Combinations of trend and variance-stabilising transformations can reduce though not completely eliminate the smooth change. A. Raw time series. B. Transformed using first difference. C. Transformed using first difference, absolute value, plus one, then natural logarithm. D. Transformed using first difference, absolute value, then square root. E. Transformed using first difference, absolute value, plus one, then Box Cox transformation.

5.4.3 Tuning of automated methods

Following identification of a range of plausible parameters from the initial 300 time series, a grid search was applied across all 2890 images in the tuning set, to identify the best parameters for each of the six remaining automated methods, based on MCC. For the four methods where the discrepancy analysis identified false positives due to smooth changes in level, the method was also run on the first-differenced values.

5.4.3.1 changepoint – cpt.meanvar

The cpt.meanvar method from the *changepoint* package detects changes in level and/or variability. It assumes that a time series is comprised of a number of segments, where each segment contains independent, identically distributed values from a named parametric distribution, whose mean and variance is constant within a segment but can vary between segments (also referred to as “piecewise constant”). It offers 4 different distributional choices: Gaussian, Exponential, Gamma and Poisson, and 3 different segmentation methods for detecting an unknown number of change points: Binary Segmentation (an approximate method which tests for a single change point at a time, repeatedly bisecting the time series until no further change points are identified), Segment Neighbourhood (an exact method which identifies the optimum number and position of segments based on maximum likelihood), and PELT (an exact method which also identifies the optimum number and position of segments based on maximum likelihood, but is more computationally efficient than Segment Neighbourhood). The method was run using a Gaussian distribution (the default) and the PELT algorithm. The penalty term for adding an additional change point was an adjustable parameter. Following the example in the documentation for setting manual penalties, a grid search of penalties was used, of the form x*ln(n), where n is the length of the individual time series, and x set to each integer value between 2 and 100 inclusive.

Based on MCC, Table 5-4 shows the top five values of x when compared to the Zooniverse labels. For the best-performing value of x, sensitivity and PPV were only 37.7% and 30.2% respectively, based on 2490 true positives, 7,057,535 true negatives, 5747 false positives,

and 4117 false negatives. While specificity and NPV were both very close to 1, this was to be expected given the unbalanced nature of positive-labelled versus (negative) unlabelled time points in each time series. To try to reduce false positives caused by smooth changes in level, the method was also run with a range of penalties (between 2 and 150 inclusive) on the first difference of the data values. This actually resulted in a poorer performance overall.

Table 5-4. Top 5 parameter settings for the cpt.meanvar method, using the raw and first-differenced times series values in the tuning set of 2890 images

Parameter value (penalty)	**Matthews Correlation Coefficient**	**Sensitivity**	**Specificity**	**Positive Predictive Value**	**Negative Predictive Value**
Raw values					
89	0.337	0.377	0.999	0.302	0.999
90	0.337	0.375	0.999	0.303	0.999
88	0.336	0.378	0.999	0.300	0.999
82	0.335	0.389	0.999	0.290	0.999
87	0.335	0.380	0.999	0.298	0.999
First-differenced values					
137	0.256	0.224	1.000	0.295	0.999
144	0.256	0.219	1.000	0.300	0.999
132	0.256	0.226	0.999	0.291	0.999
143	0.256	0.220	1.000	0.299	0.999
130	0.256	0.227	0.999	0.289	0.999

Note: results presented as proportions

5.4.3.2 changepoint – cpt.var

The cpt.var method from the *changepoint* package works similarly to the cpt.meanvar method above, but is designed to detect changes in variability only, i.e. the variance is allowed to differ between segments but the mean is assumed to be the same in all segments. It offers two distributional choices: Gaussian or non-parametric, and the same 3 segmentation methods as before. The method was run using a Gaussian distribution (the default) and the PELT algorithm. While it is clear that the statistical model of a constant mean and piecewise constant variance does not reflect the nature of the raw data, it could

potentially reflect the nature of the first-differenced data. Again, the penalty term for adding an additional change point was an adjustable parameter, for which a grid search of penalties of the form x*ln(n) was used, where n is the length of the individual time series, and x set to each integer value between 2 and 20 inclusive for the raw values, and between 2 and 100 inclusive for the first-differenced values.

The results of the top 5 parameter settings based on MCC are shown in in Table 5-5. Performance on the raw values was poor overall, with sensitivity and PPV of 33.5% and 21.1% respectively, based on 2216 true positives, 7,055,003 true negatives, 8279 false positives, and 4391 false negatives. Slightly surprisingly, the performance was marginally weaker on the first-differenced values than on the raw values.

Table 5-5. Top 5 parameter settings for the cpt.var method, using the raw and first-differenced time series values in the tuning set of 2890 images

Parameter value (penalty)	**Matthews Correlation Coefficient**	**Sensitivity**	**Specificity**	**Positive Predictive Value**	**Negative Predictive Value**
Raw values					
7	0.265	0.335	0.999	0.211	0.999
8	0.265	0.318	0.999	0.221	0.999
9	0.262	0.300	0.999	0.231	0.999
6	0.262	0.349	0.999	0.198	0.999
15	0.260	0.244	0.999	0.279	0.999
First-differenced values					
94	0.247	0.235	0.999	0.260	0.999
95	0.246	0.234	0.999	0.261	0.999
96	0.246	0.233	0.999	0.262	0.999
98	0.246	0.231	0.999	0.264	0.999
93	0.246	0.235	0.999	0.259	0.999

Note: results presented as proportions

5.4.3.3 changepointnp – cpt.np

The cpt.np method from the *changepointnp* package employs a nonparametric model based on a piecewise constant (unknown) distribution, to detect *any* changes in the distribution

between segments. The only segmentation method available is a variation on the PELT method (called ED-PELT, referring to the fact that it uses the empirical distribution). Similar to the changepoint.meanvar method, the penalty term for adding an additional change point was an adjustable parameter, which was set as before using integers between 2 and 20 inclusive. Again, to try to reduce false positives caused by smooth changes in level, the method was also run with a range of penalties on the first difference of the data values.

The results of the top 5 parameter settings based on MCC are shown in Table 5-6. While performance was better than for the cpt.meanvar and cpt.var methods, it was still only modest, with maximum sensitivity of 50.7% and PPV of 35.0%, based on 3348 true positives, 7,057,070 true negatives, 6212 false positives, and 3259 false negatives.

Table 5-6. Top 5 parameter settings for the cpt.np method, using the raw and first-differenced times series values in the tuning set of 2890 images

Parameter value (penalty)	**Matthews Correlation Coefficient**	**Sensitivity**	**Specificity**	**Positive Predictive Value**	**Negative Predictive Value**
Raw values					
7	0.421	0.507	0.999	0.350	1.000
8	0.419	0.488	0.999	0.361	1.000
6	0.418	0.528	0.999	0.332	1.000
9	0.418	0.473	0.999	0.370	1.000
10	0.412	0.454	0.999	0.376	0.999
First-differenced values					
4	0.370	0.336	1.000	0.409	0.999
5	0.367	0.295	1.000	0.459	0.999
3	0.365	0.401	0.999	0.333	0.999
6	0.362	0.271	1.000	0.486	0.999
7	0.350	0.244	1.000	0.503	0.999

Note: results presented as proportions

5.4.3.4 DeCAFS – DeCAFS

The DeCAFS method from the *DeCAFS* package detects changes in level in a time series with a piecewise locally-fluctuating mean. It is designed to reduce overfitting of change

points when the data are not independent and identically distributed within segments (the assumption made in the methods from the previous packages). It does this by modelling each segment as a random walk (i.e. a path that consists of a succession of random steps) with autocorrelated noise (i.e. that the distance from the mean of a particular value is correlated with the immediately-preceding distance from the mean, but is not correlated with any older values). The segmentation method was developed specifically for the package and finds the best segmentation by maximum likelihood. Similar to the changepoint.meanvar method, the penalty term for adding an additional change point was an adjustable parameter, which was set as before using integers between 1 and 20 inclusive. Results are shown in Table 5-7. While the sensitivity was fairly high at 80.6%, this was offset by the extremely low PPV of 0.9%, based on 5328 true positives, 6,496,692 true negatives, 566,590 false positives, and 1279 false negatives. There was no obvious transformation that could be applied in order to reduce the huge numbers of false positive calls made by the method despite parameter tuning.

Table 5-7. Top 5 parameter settings for the DeCAFS method, using the raw values in the tuning set of 2890 images

Parameter value (penalty)	**Matthews Correlation Coefficient**	**Sensitivity**	**Specificity**	**Positive Predictive Value**	**Negative Predictive Value**
Raw values					
3	0.081	0.806	0.920	0.009	1
4	0.081	0.777	0.925	0.010	1
2	0.081	0.851	0.910	0.009	1
5	0.080	0.747	0.928	0.010	1
1	0.079	0.918	0.891	0.008	1

Note: results presented as proportions

5.4.3.5 otsad – OcpTsSdEwma

The OcpTsSdEwma method from the *otsad* package detects sudden changes in level as well as outliers. Unlike the previous methods, instead of identifying the best segmentations by modelling the time series as a whole, it is based on statistical process control theory,

which was developed originally to monitor manufacturing processes, and which provide an alert as and when a shift in mean is detected in a measurement of a process that is running continuously. This means that it could potentially be incorporated into a system that monitors EHR data streams continuously rather than only be used on static datasets. It works in two stages. In the first stage, it detects potential change points using the SD-EWMA (Shift detection based on exponentially weighted moving average) algorithm, i.e. upper and lower control limits are calculated based on predicted errors from an exponentially weighted moving average of the time series, and if a value falls outside of the control limits, a potential change point is identified. In the second stage, it checks the veracity of the potential change point using the Kolmogorov-Smirnov test (i.e. taking a subsequence of values either side of the potential change point and testing if they have equal means and variances) to reduce false alarms. While the TSSD-EWMA algorithm is primarily designed to be an online method, the *otsad* package implements an optimised version that can be run offline. The default value of 5 “training observations” (i.e. the first 5 values of each time series) was used to initialise the method, and the method optimised using the following 3 tuning parameters:

- threshold – Error smoothing constant, must be between 0 and 1, recommended 0.01 or 0.05, default=0.01
- l – Control limit multiplier, default=3.
- m – Length of the subsequences for applying the Kolmogorov-Smirnov test, default = 5.

When running the method on the raw values, all combinations of values with ‘threshold’ between 0.01 and 0.15 with a step of 0.01, ‘l’ between 2 and 7 inclusive, and ‘m’ between 5 and 10 inclusive were considered. While in theory the algorithm should be robust to smooth changes in level, the method was run again (using the same range of penalties as before except for going up to a maximum `threshold’ of 0.11) on the first difference of the data values given that a high number of false positives had been found during the discrepancy analysis where there was a smooth change in level. The results of the top 5 parameter

settings based on MCC are shown in Table 5-8. Performance was modest, with maximum sensitivity of 38.1% and PPV of 49.0%, based on 2520 true positives, 7,060,659 true negatives, 2623 false positives, and 4087 false negatives.

Table 5-8. Top 5 parameter settings for the OcpTsSdEwma method, using the raw and first-differenced times series values in the tuning set of 2890 images

Parameter values	Matthews Correlation Coefficient	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value
Raw values					
threshold = 0.13, m = 7, l = 3	0.432	0.381	1	0.490	0.999
threshold = 0.11, m = 7, l = 3	0.432	0.369	1	0.506	0.999
threshold = 0.1, m = 7, l = 3	0.431	0.365	1	0.510	0.999
threshold = 0.09, m = 7, l = 3	0.431	0.364	1	0.511	0.999
threshold = 0.12, m = 7, l = 3	0.430	0.375	1	0.495	0.999
First-differenced values					
threshold = 0.06, m = 9, l = 6	0.172	0.050	1	0.597	0.999
threshold = 0.05, m = 9, l = 6	0.170	0.049	1	0.593	0.999
threshold = 0.09, m = 9, l = 5	0.170	0.055	1	0.525	0.999
threshold = 0.04, m = 9, l = 7	0.170	0.046	1	0.624	0.999
threshold = 0.01, m = 9, l = 6	0.169	0.052	1	0.557	0.999

Note: results presented as proportions

5.4.3.6 otsad – OcpPewma

The *OcpPewma* method from the *otsad* package detects sudden changes in level and outliers, while ignoring gradual changes in level. It employs a probabilistic method of EWMA

(Exponentially Weighted Moving Average) which dynamically adjusts the weighting parameter based on the probability of the current observation. This is so it can recover more quickly after large anomalies. Again, while the PEWMA algorithm is primarily designed to be an online method, the *otsad* package implements an optimised version that can be run offline. The default value of 5 “training observations” was used to initialise the method, and the method was optimised using the following 3 tuning parameters:

- alpha0 – Maximal weighting parameter, must be between 0 and 1 exclusive, default=0.2
- beta – Weight placed on the probability of the given observation, must be between 0 and 1 inclusive, default=0
- l – Control limit multiplier, default=3.

Based on the results of the earlier exploration of suitable parameter values using the initial set of 300 time series, all combinations of values with ‘alpha0’ between 0.5 and 0.9 with a step of 0.1, ‘beta’ between 0 and 0.5 with a step of 0.1, and ‘l’ between 2 and 18 inclusive were tested. Since there were no false positives found during the discrepancy analysis caused by smooth changes in trend, the method was only applied to the raw time series. The results of the top 5 parameter settings based on MCC are shown in Table 5-9. Performance was poor with maximum sensitivity of 34.9% and PPV of 13.3%, based on 2306 true positives, 7,048,216 true negatives, 15,066 false positives, and 4301 false negatives.

Table 5-9. Top 5 parameter settings for the OcpPewma method, using the raw times series values in the tuning set of 2890 images

Parameter values	Matthews Correlation Coefficient	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value
Raw values					
alpha0 = 0.9, beta = 0, l = 6	0.214	0.349	0.998	0.133	0.999
alpha0 = 0.9, beta = 0, l = 7	0.213	0.317	0.998	0.144	0.999
alpha0 = 0.9, beta = 0, l = 5	0.211	0.387	0.997	0.117	0.999
alpha0 = 0.9, beta = 0, l = 8	0.206	0.285	0.998	0.150	0.999
alpha0 = 0.9, beta = 0, l = 11	0.205	0.242	0.999	0.175	0.999

Note: results presented as proportions

5.4.4 Overall performance on reserved test set

Each method was run on the reserved test set (excluding time series containing NAs), using the best parameters and transformations identified using the tuning set, which were all raw values rather than differences. The test set comprised 1234 images and a total of 2788 crowd-sourced consensus labels for change points. Results for each method are shown in Table 5-10.

Table 5-10. Overall performance of each automated method on the reserved test set of 1234 images. These images contained a total of 2788 crowd-sourced consensus labels for change points

Method	Matthews Correlation Coefficient	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value
OcpTsSdEwma	0.436	36.9% (35.2, 38.8)	100% (100, 100)	51.6% (49.4, 53.8)	99.9% (99.9, 99.9)
cpt.np	0.425	51.7% (49.8, 53.5)	99.9% (99.9, 99.9)	35.1% (33.6, 36.6)	100% (100, 100)
cpt.meanvar	0.359	36.2% (34.4, 38)	99.9% (99.9, 99.9)	35.7% (33.9, 37.5)	99.9% (99.9, 99.9)
cpt.var	0.293	35.3% (33.5, 37.1)	99.9% (99.9, 99.9)	24.4% (23.1, 25.8)	99.9% (99.9, 99.9)
OcpPewma	0.227	34.0% (32.3, 35.8)	99.8% (99.8, 99.8)	15.3% (14.5, 16.3)	99.9% (99.9, 99.9)
DeCAFS	0.084	80.6% (79, 82)	92.6% (92.6, 92.6)	0.984% (0.944, 1.03)	100% (100, 100)

Based on MCC, the best performing automated method was OcpTsSdEwma from the *otsad* package, run on the raw time series values, with parameters: threshold = 0.13, m = 7, l = 3. The method had a sensitivity of 36.9% (95% CI 35.2, 38.8), specificity of 100% (100, 100), PPV of 51.6% (49.4, 53.8), and NPV of 99.9% (99.9, 99.9), based on 1030 true positives, 3,058,313 true negatives, 966 false positives, and 1758 false negatives.

5.4.5 Performance of automated methods for different aggregation methods

Lastly, given the vast variation in behaviour of the different time series (see Figure 5-1), it was investigated if performance might be improved by using different tuning parameters and/or transformations for different aggregation methods.

5.4.5.1 Performance of automated methods by aggregation granularity

Aggregating the data by daily, weekly or monthly timepoints produced time series with slightly different properties, see Figure 5-5. Daily time series were longer and often contained a mixed distribution of weekday and weekend values, while monthly time series

were shorter and smoother, with fewer obvious change points. Similarly to before, the best parameters per method were selected using a grid search, but separately based on only the daily, weekly, or monthly time series from the tuning set. Performance was not assessed on the first-difference of the time series since this had been weaker for all of the methods when applied at the overall level. The tuning parameters selected for each method are shown in Appendix C.2.1.

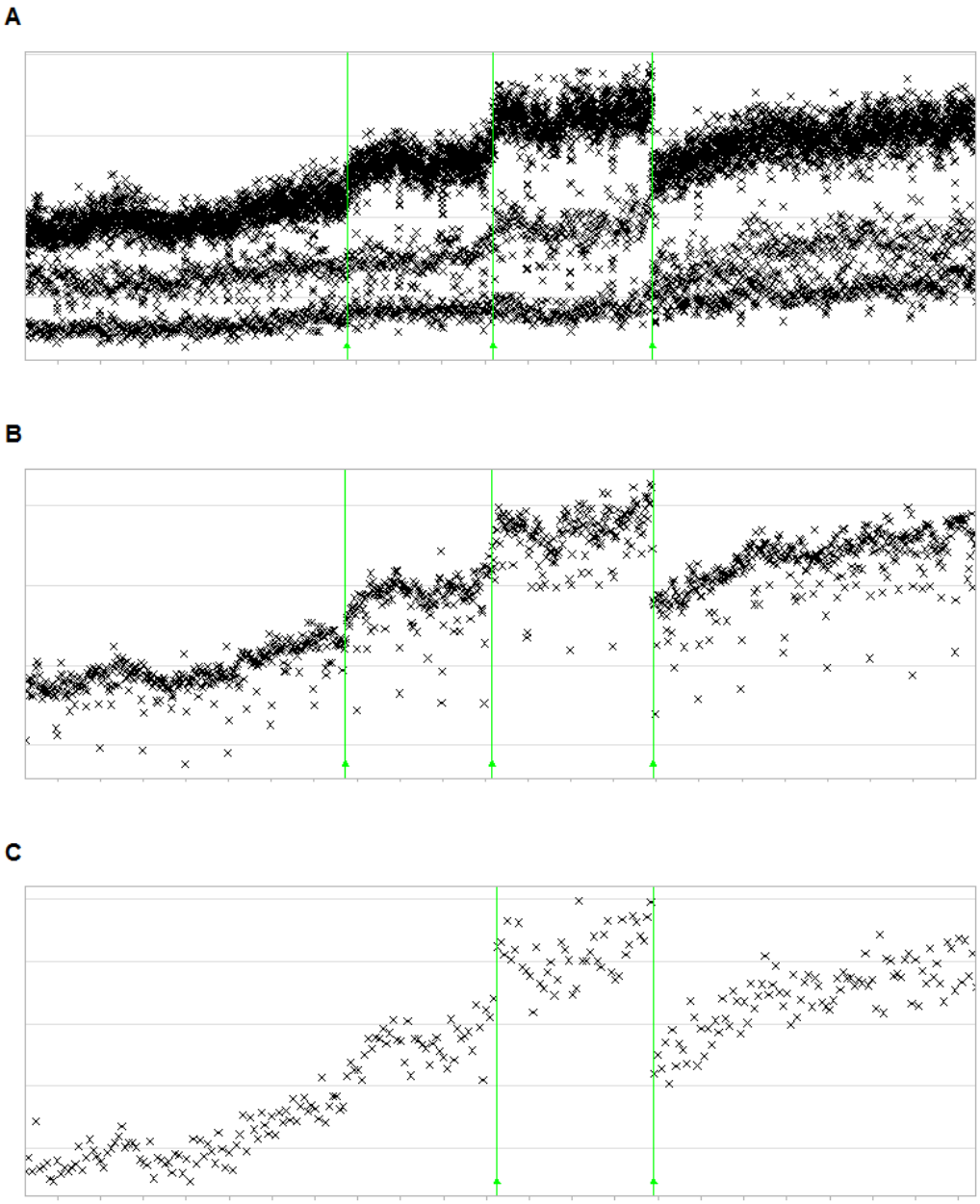


Figure 5-5. Example of time series created when aggregating the same data by A. day, B. week, and C. month. Green lines with triangles are labels drawn by the volunteers

The final performance of each method on the reserved test set are shown in Table 5-11. The performance achieved at the different granularity levels was similar to that achieved at the overall level, at typically no higher than around 50% for both sensitivity and PPV. All but one method (cpt.np) slightly favoured monthly series, and performance was usually weakest on daily series, but it was not dramatically different between the different granularities.

Table 5-11. Final performance of each automated method on the reserved test set of 1234 images, by aggregation granularity.
These images contained a total of 2788 crowd-sourced consensus labels for change points

Granularity	MCC	Sensitivity	Specificity	PPV	NPV
cpt.meanvar					
Day	0.363	51.9% (48.2, 55.6)	100% (100, 100)	25.5% (23.3, 27.8)	100% (100, 100)
Week	0.399	37.8% (34.8, 40.9)	99.9% (99.9, 99.9)	42.3% (39.1, 45.6)	99.9% (99.9, 99.9)
Month	0.414	43.7% (40.8, 46.6)	99.5% (99.5, 99.5)	40.1% (37.4, 42.9)	99.6% (99.5, 99.6)
cpt.var					
Day	0.290	35.9% (32.4, 39.5)	100% (100, 100)	23.5% (21.1, 26.1)	100% (100, 100)
Week	0.320	35.6% (32.7, 38.7)	99.8% (99.8, 99.8)	28.9% (26.4, 31.6)	99.9% (99.9, 99.9)
Month	0.351	41.5% (38.6, 44.4)	99.3% (99.2, 99.3)	30.6% (28.3, 33)	99.6% (99.5, 99.6)
cpt.np					
Day	0.461	54.0% (50.4, 57.7)	100% (100, 100)	39.4% (36.3, 42.5)	100% (100, 100)
Week	0.463	58.2% (55.1, 61.3)	99.8% (99.8, 99.8)	37.1% (34.7, 39.5)	99.9% (99.9, 99.9)
Month	0.396	54.1% (51.1, 57.0)	99.0% (99.0, 99.1)	29.9% (27.9, 31.9)	99.6% (99.6, 99.7)
DeCAFS					
Day	0.052	78.3% (75.1, 81.2)	94.0% (94.0, 94.0)	0.383% (0.353, 0.417)	100% (100, 100)
Week	0.108	95.8% (94.4, 96.9)	87.3% (87.3, 87.4)	1.41% (1.32, 1.5)	100% (100, 100)
Month	0.213	86.8% (84.6, 88.6)	89.8% (89.6, 89.9)	6.05% (5.69, 6.43)	99.9% (99.9, 99.9)
OcpTsSdEwma					
Day	0.323	17.2% (14.6, 20.1)	100% (100, 100)	60.8% (53.9, 67.3)	100% (100, 100)
Week	0.476	41.8% (38.7, 44.9)	99.9% (99.9, 99.9)	54.3% (50.7, 57.8)	99.9% (99.9, 99.9)
Month	0.572	48.3% (45.3, 51.2)	99.8% (99.8, 99.9)	68.3% (65.0, 71.5)	99.6% (99.6, 99.6)
OcpPewma					
Day	0.172	27.8% (24.6, 31.2)	99.9% (99.9, 99.9)	10.7% (9.39, 12.2)	100% (100, 100)
Week	0.242	20.7% (18.3, 23.4)	99.9% (99.9, 99.9)	28.6% (25.4, 32)	99.9% (99.8, 99.9)
Month	0.490	51.1% (48.1, 54.0)	99.6% (99.5, 99.6)	47.8% (45, 50.7)	99.6% (99.6, 99.7)

5.4.5.2 Performance of automated methods by aggregation function

A grid search was conducted to identify the best automated method and parameters for each aggregation function, based on MCC. For methods where the discrepancy analysis had identified false positives due to smooth changes in level, the method was also run on the first-differenced values. The optimal methods, parameters, and transformations based on the tuning set are shown in Appendix C.2.2.

The final performance of the optimal method per aggregation function when run on the reserved test set is shown in Table 5-12. There were only two aggregation functions that achieved a sensitivity and PPV both over 50%. One was the number of nonconformant values (nonconformant_n) using the cpt.np method, with a sensitivity of 72.2% (49.1, 87.5), specificity of 100% (100, 100), PPV of 81.3% (57.0, 93.4), and NPV of 100% (100, 100), based on 13 true positives, 60,771 true negatives, 3 false positives, and 5 false negatives. The other was the maximum string length (maxlength) using the cpt.meanvar method, with a sensitivity of 88.1% (75.0, 94.8), specificity of 99.9% (99.9, 100), PPV of 61.7% (49.0, 72.9), and NPV of 100% (100, 100), based on 37 true positives, 38,067 true negatives, 23 false positives, and 5 false negatives.

The optimal method for each individual aggregation function varied widely, with all six of the automated methods being optimal for at least one aggregation function. The methods were more effective in identifying change points in the raw time series values than in the first-differenced time series.

Given the smaller sample size in the test set, there were two aggregation functions that did not contain any change points identified by crowd-sourced visual inspection (the percentage of nonconformant values (nonconformant_perc) and the percentage of duplicated rows (nonzero_perc)) and so performance of the automated methods could not be assessed on them.

Table 5-12. Final performance of the optimal automated method and transformation on the reserved test set of 1234 images, by aggregation function. These images contained a total of 2788 crowd-sourced consensus labels for change points

Aggregation function	No. of time series	No. of crowd-sourced change points	Matthews Correlation Coefficient	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value	Optimal method	Optimal transformation
nonconformant_n	34	18	0.766	72.2% (49.1, 87.5)	100% (100, 100)	81.2% (57.0, 93.4)	100% (100, 100)	cpt.np	None
maxlength	19	42	0.737	88.1% (75, 94.8)	99.9% (99.9, 100)	61.7% (49.0, 72.9)	100% (100, 100)	cpt.meanvar	None
midnight_perc	19	63	0.648	47.6% (35.8, 59.7)	100% (100, 100)	88.2% (73.4, 95.3)	99.9% (99.9, 99.9)	OcpTsSdEwma	None
distinct	76	232	0.508	58.2% (51.8, 64.4)	99.9% (99.9, 99.9)	44.6% (39.1, 50.2)	99.9% (99.9, 100)	cpt.np	None
meanlength	18	53	0.503	64.2% (50.7, 75.7)	99.9% (99.9, 99.9)	39.5% (29.9, 50.1)	100% (99.9, 100)	DeCAFS	None
missing_perc	141	257	0.488	42.4% (36.5, 48.5)	100% (100, 100)	56.2% (49.2, 63.0)	100% (100, 100)	OcpTsSdEwma	None
subcat_perc	190	300	0.485	39.3% (34.0, 45.0)	100% (100, 100)	59.9% (52.9, 66.5)	100% (100, 100)	OcpTsSdEwma	None
sum	6	4	0.474	75.0% (30.1, 95.4)	99.9% (99.9, 100)	30.0% (10.8, 60.3)	100% (99.9, 100)	OcpPewma	None
n	230	768	0.473	75.1% (72, 78.1)	99.8% (99.8, 99.8)	29.9% (27.9, 32.0)	100% (100, 100)	cpt.meanvar	None

Aggregation function	No. of time series	No. of crowd-sourced change points	Matthews Correlation Coefficient	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value	Optimal method	Optimal transformation
subcat_n	199	276	0.473	43.1% (37.4, 49.0)	100% (100, 100)	52.0% (45.5, 58.4)	100% (100, 100)	OcpTsSdEwma	None
missing_n	162	425	0.455	34.6% (30.2, 39.2)	100% (100, 100)	60.0% (53.8, 65.9)	99.9% (99.9, 99.9)	OcpTsSdEwma	None
midnight_n	20	96	0.445	37.5% (28.5, 47.5)	99.9% (99.9, 100)	52.9% (41.2, 64.3)	99.9% (99.9, 99.9)	OcpTsSdEwma	None
minlength	18	101	0.372	33.7% (25.2, 43.3)	99.9% (99.9, 99.9)	41.5% (31.4, 52.3)	99.9% (99.8, 99.9)	cpt.meanvar	None
median	5	25	0.357	16.0% (6.4, 34.7)	100% (100, 100)	80.0% (37.6, 96.4)	99.9% (99.8, 99.9)	OcpTsSdEwma	None
max	20	58	0.297	56.9% (44.1, 68.8)	99.3% (99.2, 99.4)	15.8% (11.5, 21.3)	99.9% (99.9, 99.9)	cpt.np	First-difference
min	32	55	0.288	29.1% (18.8, 42.1)	99.9% (99.9, 99.9)	28.6% (18.4, 41.5)	99.9% (99.9, 99.9)	OcpPewma	None
mean	5	15	0.258	13.3% (3.74, 37.9)	100% (99.9, 100)	50.0% (15, 85)	99.9% (99.8, 99.9)	cpt.meanvar	First-difference
nonconformant_perc	35	0	*					cpt.var	None
nonzero_perc	5	0	*					OcpTsSdEwma	None

* there were no change points identified by crowd-sourced visual inspection for the images in the reserved test set corresponding to these aggregation functions, so final performance of the optimal automated methods could not be assessed

5.5 Discussion

None of the identified R packages were designed to detect all 5 types of change points that would be of interest from a data quality perspective, namely sudden changes in level, variability, trend, presence/absence of data points, or irregular outliers. Of the 10 automated methods that were shortlisted (to cover the range of change point types and a range of detection methods), none accurately detected the change points that were identified by crowd-sourced visual inspection. Most of the methods generated large numbers of false positive calls, and applying a first difference transformation to the time series did not improve their performance.

The aggregation functions that were the "easiest" to detect change points in were the number of nonconformant values (nonconformant_n) and the maximum string length (maxlength). This was understandable for the 'maxlength' aggregation function, since the maximum string length of a uniqueidentifier field produces time series that typically demonstrate a piecewise constant mean, which is ideal for change point detection. For the 'nonconformant_n' aggregation function, the values were very low in general, with far fewer change points present per image than for other aggregation functions, and so were perhaps easier to identify due to their rarity. The "hardest" aggregation functions to detect change points in were the minimum, maximum, and mean value (min, max, mean). Notably, these all apply to numeric and datetime fields only, though it is not immediately obvious why they perform particularly poorly with the methods investigated.

Arguably, the type of change point that is hardest to detect is a sudden change in trend. The three methods designed to detect a change in trend either failed due to errors or were prohibitively slow, and even if this were not the case, they could not be restricted to only detect sudden trend changes and so would likely still have made false positive calls in time series containing an underlying smooth trend.

Time series that contained NAs were excluded from this analysis, since most automated methods do not accept time series containing NAs. NAs are created when there are no records in the timepoint, and manifest as changes in presence/absence of data points. While this means that this type of change point was not assessed in this study, when considering a final application where all change points across a dataset will be examined together by a researcher, any presence/absence-related change points should still be picked up by the number of values present (n) aggregation function and so flagging them in other aggregation functions as well might be considered unnecessary duplication. In this case, simply omitting them when they occur in a time series might be the most appropriate action.

Change point detection methods are typically designed to detect a specific type of change point, such as a change in level or a change in variability, and are typically developed and assessed using simulated data, which is much better behaved than real world EHR data. It is perhaps not surprising then, that overall performance was poor for a collection of time series that exhibit multiple different types of change points over a background of gradual trends and changes.

In an ideal world, a single change point detection method would be able to identify all types of temporal artefacts, though until then, it may be beneficial for developers to focus on specific subgroups of time series (such as particular data field types or aggregation functions) rather than trying to cover all scenarios. In this case, it might be worth focussing attention initially on aggregation functions that apply to all types of data fields e.g. the number of values present (n) or the number of missing values (missing_n), since even a perfect method for an uncommon aggregation function, e.g. the maximum string length of a uniqueidentifier (maxlength), might not be that useful in practice.

Findings from this chapter demonstrate that we are still a long way from being able to fully automate the detection of change points in EHRs, but this is an interesting and wide open field for further methods research. This is particularly true given the increasing drive to use

EHR data for machine learning and exploratory data mining[99], where the analysts will likely be a further step removed from the data, and more naturally inclined towards automation.

For an automated method to be trusted, it needs to be shown to be effective on data from the particular field in which it is to be applied. Having a large open-source collection of labelled real-world time series such as that produced during the crowd-sourcing Zooniverse project, will facilitate the future development and validation of change point detection methods that can work in a generalised setting like data quality assessment. To this end, these labelled time series have been made publicly available to download from https://doi.org/10.5281/zenodo.7331161.

5.6 Conclusion

Automated change point detection methods currently available in R are not effective at identifying change points relating to data quality in EHRs. However, the newly-available set of labelled time series produced from this study will facilitate future development and validation of such methods. A substantial amount of further work is still needed before automation of change point detection becomes viable, so visual inspection will continue to be the most appropriate method for the short to medium term.

CHAPTER 6: *daiquiri*: Data quality reporting for temporal datasets – a new R package

6.1 Chapter summary

The initial data analysis phase of a research study is when researchers should be screening their raw data for potential data quality issues, but there is currently very little practical guidance and few tools to assist researchers in doing this efficiently and transparently.

Change points occur very frequently in EHRs, and unfortunately, automatic change point detection methods currently available in R do not appear to be very accurate when compared to human visual inspection.

Nevertheless, the task of identifying change points in EHRs can still be made easier and more efficient for researchers, by automating the creation of appropriate time series, and presenting them in a way that enables quick visual assessment.

This chapter describes a novel open-source R package called *daiquiri*, developed as part of this doctoral thesis, which generates user-friendly reports that enable quick visual review of temporal shifts in record-level data. It works on EHR datasets containing any number and type of data fields, and requires minimal data preparation or specialist knowledge. Time series plots showing aggregated values are automatically created for each data field depending on its contents (e.g. min/max/mean values for numeric data, no. of distinct values for categorical data), as well as overviews for missing values, non-conformant values, and duplicated rows. The resulting reports can be easily shared either with co-authors or as a supplementary file to a publication, thereby increasing study reproducibility and transparency.

6.2 Implementation and architecture

6.2.1 Technologies

R is a free software environment for statistical computing and graphics. It runs on a wide variety of operating systems, and is very popular in the research community. The base R installation can be extended by installing R packages created by the R community. The default place to access R packages is CRAN (The Comprehensive R Archive Network), where members of the public can contribute packages provided they pass a range of checks.

6.2.1.1 Dependencies

daiquiri builds on a small number of existing R packages that are well-established and popular in their fields. This means that their functionality has been extensively tested and they are likely to remain available and well-maintained into the future. Most notable among these are:

- *readr* – for reading tabular data from delimited files and validating them against the expected datatypes in each column
- *data.table* – for fast processing of large-scale tabular data
- *ggplot2* – for generating attractive plots
- *reactable* – for creating attractive html tables that can be sorted and filtered interactively
- *rmarkdown* – converts template text files into html reports that can be viewed in a browser, with many built-in features such as tabbed frames

6.2.1.2 Source control

All source code is stored under ‘git’ version control, meaning that a complete history of code changes is available and earlier versions can easily be re-accessed if necessary.

6.2.1.3 Unit testing

Unit tests have been written for all major functions, to ensure that the code does what it is designed to do. Each unit test tests one thing (hence the name), and in combination they cover all the functionality of the software.

6.2.1.4 Continuous integration

Every time a code change is 'pushed' to the main git repository, a set of 'continuous integration' actions are initiated automatically. The most important of these is 'R CMD check', which runs a comprehensive set of checks for R packages, including re-running all unit tests to check that nothing has inadvertently been broken, and checks that the package can be run successfully on a range of operating systems. Other 'continuous integration' actions include calculating how much of the code is covered by unit tests, and building the documentation website.

6.2.2 Software peer review

The package has successfully completed peer review for code quality (https://github.com/ropensci/software-review/issues/535) and been accepted into the rOpenSci (https://ropensci.org) suite of R packages for open and reproducible research. Inclusion in the rOpenSci organisation means that it will receive ongoing maintenance support from the rOpenSci community as well as be promoted via their blog and social media, aiding its discoverability.

6.2.3 Availability

The package has been available to download and install from CRAN since 11th November 2022. A documentation website (https://ropensci.github.io/daiquiri/) describes the package and how to use it without needing to install it first.

The source code has also been made publicly available as a code repository on GitHub (https://github.com/ropensci/daiquiri), with an open source GPL-3 licence, meaning that the code is free to reuse and modify provided that any new software that uses it can only be

released under the same (or a less permissive) open source licence. Making the code public also assists with sustainability as it means that bug fixes and new features can be made by anyone, removing reliance on the original developer or funder to continue maintaining it.

6.2.4 Usability

Design choices were made to make it as easy as possible for researchers to use the package, with minimal need for data preparation or specialist knowledge. Record-level EHR data can be passed in directly, without needing to conform to any specific data model, and aggregation functions pre-defined to cover the most common checks for temporal data quality. Default values are provided for all function parameters where possible. Five different researchers who work with EHRs tested the package on a range of platforms (Windows, Macintosh, Ubuntu) and R versions (3.5.1, 3.6.3, 4.0.3, 4.1.0, 4.1.2) and reviewed the help documentation for clarity and completeness. They also reviewed the format and content of the reports created by the package for correctness and usefulness. Any bugs or usage difficulties identified by the researchers were fixed, and any feature requests were recorded for future consideration.

With regard to speed, on a Windows 10 laptop with a 4-core i7 processor and 16GB of memory, a 40MB file containing 25 columns and 200,000 rows took under three minutes to process, and a 600MB file containing 18 columns and 3.6 million rows took under seven minutes to process.

6.2.5 Configurability

6.2.5.1 Choice of data field types

The main thing the user needs to do is to specify what type of data is expected in each field (column), e.g. a date, a number, a nominal category, etc. The package uses this information to calculate the number of values that are not of the expected type, and to decide which aggregation functions to use for the time series.

One column only is chosen to be the ‘timepoint’ field (though the package could be run again using a different column as the timepoint field if needed). This column is used as the independent time variable on the x-axis of each time series plot.

The list of possible field types for the columns is below. Different time series are generated depending on the type of field. All types will have time series generated for the number of values present, as well as the number and percentage of missing values.

- ft_timepoint() - identifies the data field which should be used as the independent time variable (and will form the x-axis of any time series plots). There should be one and only one of these specified. The number of values present for this field is equivalent to the number of records in the data set (after duplicates have been removed). By default, additional time series are created for the number and percentage of values which do not contain a time portion, though this can be switched off if the ‘includes_time’ parameter is set to FALSE (e.g. if you already know in advance that there are no time portions in any of the values). Time series are not created for missing values in the timepoint field as they cannot be assigned to a date. Instead, the total number of missing values is included in a summary table.
- ft_uniqueidentifier() - identifies data fields which contain a (usually computer-generated) identifier for an entity, e.g. a patient. It does not need to be unique within the dataset. Values are treated as character strings, with additional time series created for minimum, maximum, and mean length of the string.
- ft_categorical() - identifies data fields which should be treated as categorical. Values are treated as character strings. Additional time series are created for the number of distinct values, and if the ‘aggregate_by_each_category’ parameter is set to TRUE, further time series are created for the number and percentage of values within each distinct subcategory value.
- ft_numeric() - identifies data fields which contain numeric values that should be treated as numbers (rather than as codes). Additional time series are created for the

minimum, maximum, mean, and median value, and the number and percentage of non-conformant values.

- ft_datetime() - identifies data fields which contain date (and optionally time) values that should be treated as continuous. Additional time series are created for the minimum, maximum, and mean value, and the number and percentage of non-conformant values. By default, time series are also created for the number and percentage of values which do not contain a time portion, though this can be switched off if the 'includes_time' parameter is set to FALSE (e.g. if the user already knows in advance that there are no time portions in any of the values).
- ft_freetext() - identifies data fields which contain free text values. Only the number of values present and number/percentage missing will be evaluated.
- ft_simple() - identifies data fields where the user only wants the number of values present and number/percentage missing to be evaluated (but which are not necessarily free text).
- ft_ignore() - identifies data fields which should be ignored. These are not loaded.

Lastly, a number of time series are generated for the dataset as a whole, namely:

- The number and percentage presence of duplicate records (i.e. where the entire row is the same as another row in the dataset)
- The total number of values present across all data fields
- The total number and percentage of missing and of non-conformant values (where relevant) across all data fields

6.2.5.2 Choice of aggregation functions

While in Chapter 4 it was shown that some aggregation functions did not add much further value in identifying change points in the eight data extracts studied, this cannot be assumed to be true for all data sources or study designs, so all the aggregation functions previously investigated are included in the package. There are two aggregation functions that can lead

to a large number of additional time series, namely the number and the percentage of values within each subcategory of a categorical field (subcat_n/subcat_perc). These are therefore not generated by default, but an option has been made available to turn them on, which can reduce the total number of time series considerably if there are many subcategories. Aggregation functions that only add a single additional time series (such as 'sum' for duplicate records) are always included as there will not be any noticeable time gain in excluding them.

6.2.5.3 Choice of aggregation granularity

Daily aggregations provide the most detailed time series so this level is most suitable to be used as the default. However, it may be unhelpfully detailed if the main analysis is conducted at a much coarser level of granularity or if the data is sparse, so different levels of granularity have been allowed to be specified depending on the user's requirement, namely daily, weekly, monthly, quarterly or yearly.

6.2.6 Data quality reports

A data quality report is generated in html format, and can be viewed in a browser as well as be shared as an attachment. In the interests of reproducibility, the top of the report contains metadata stating the date the report was created and which version of the package and of R it was created with. The rest of the report comprises of three sections:

Source data – This tab contains four child tabs which provide: an overview of the data contained in each data field, a list of data fields which were not imported, a complete list of validation warnings (such as any values identified as non-conformant), and an overall summary of the data imported. See Figure 6-1 for screenshots.

A

Source data | Aggregated data | Individual data fields

Data fields imported | Data fields ignored | Validation warnings | Summary

Field name	Field type	Total values	Missing values	Min value	Max value	Validation warnings	Column position
Prescription ID	uniqueidentifier	9163	0 (0%)	10000	9999	0	1
Prescription Date	timepoint	9163	0 (0%)	2021-01-01	2021-12-31 23:45:00	2	2
AdmissionDate	datetime	5161	4002 (44%)	2020-07-01	2021-12-31	1	3
Drug	freetext	9163	0 (0%)	Abacavir + lamiVUDine	Voriconazole	0	4
Dose	numeric	9154	9 (0.1%)	0.2	7e+05	5	5
DoseUnit	categorical	9134	29 (0.3%)	g	unit(s)	0	6
PatientID	ignore	NA	NA	NA	NA	NA	7
Location	categorical	9163	0 (0%)	SITE1	SITE2	0	8

B

Source data | Aggregated data | Individual data fields

Data fields imported | Data fields ignored | Validation warnings | Summary

Field name	Details	Instances
PrescriptionDate	Missing or invalid value in Timepoint field	2
AdmissionDate	expected valid date, but got '2021-06-31'	1
Dose	expected no trailing characters, but got '1.5g'	1
Dose	expected no trailing characters, but got '4.5 grams'	1
Dose	expected a double, but got 'See Instructions'	2
Dose	expected no trailing characters, but got '80/400 mg'	1

C

Source data | Aggregated data | Individual data fields

Data fields imported | Data fields ignored | Validation warnings | Summary

Columns in source	8
Columns imported	7
Column used for timepoint	PrescriptionDate
Min timepoint value	2021-01-01
Max timepoint value	2021-12-31 23:45:00
Rows in source	9168
Duplicate rows	3
Rows missing timepoint values	2
Rows imported	9163
Strings interpreted as missing values	"","NULL"
Total validation warnings	8

Figure 6-1. Screenshots of the 'Source data' tab from a daiquiri report.
A. Overview of the contents of each data field imported. B. List of warnings encountered when validating the data against the expected field types. C. Summary of the data imported

Aggregated data – This tab contains five child tabs which display plots showing the number of values present per data field over time, number of missing and number of non-conformant values over time, as well as the number of duplicate records over time. Data fields and timepoints where these values are not applicable are coloured in grey. See Figure 6-2. The final child tab shows a brief summary of the aggregated data.

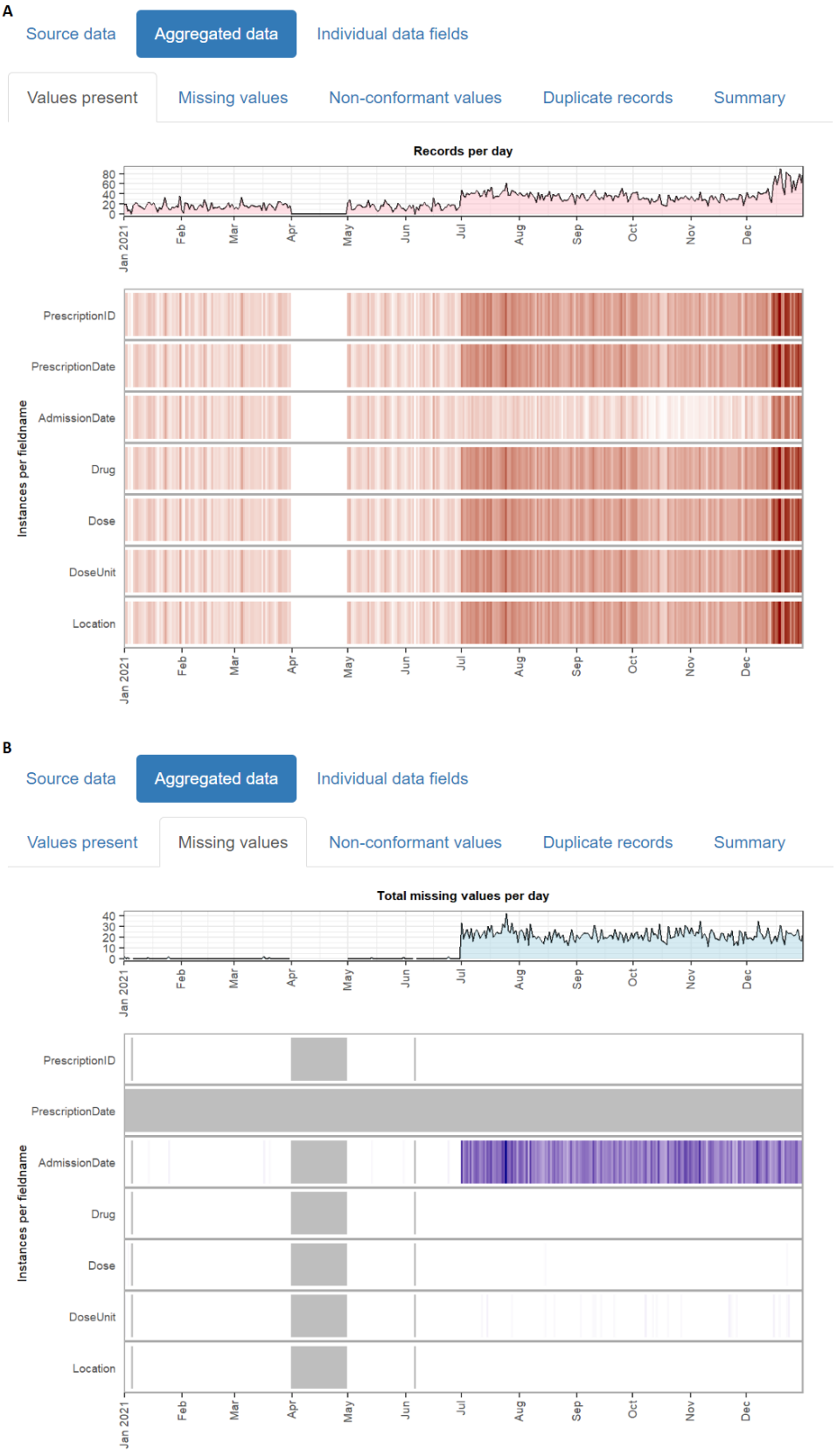


Figure 6-2. Screenshots of the 'Aggregated data' tab from a daiquiri report.
A. Visualisation of the number of values present in each data field over time. B. Visualisation of the number of missing values in each data field over time. Data fields and timepoints where these values are not applicable are coloured in grey.

Individual data fields – This tab contains two levels of child tabs. The first level consists of one tab per data field in the dataset. Within each data field tab, there is a further set of child tabs displaying a plot for each time series created, which varies according to the type of data field. See Figure 6-3 for an example.

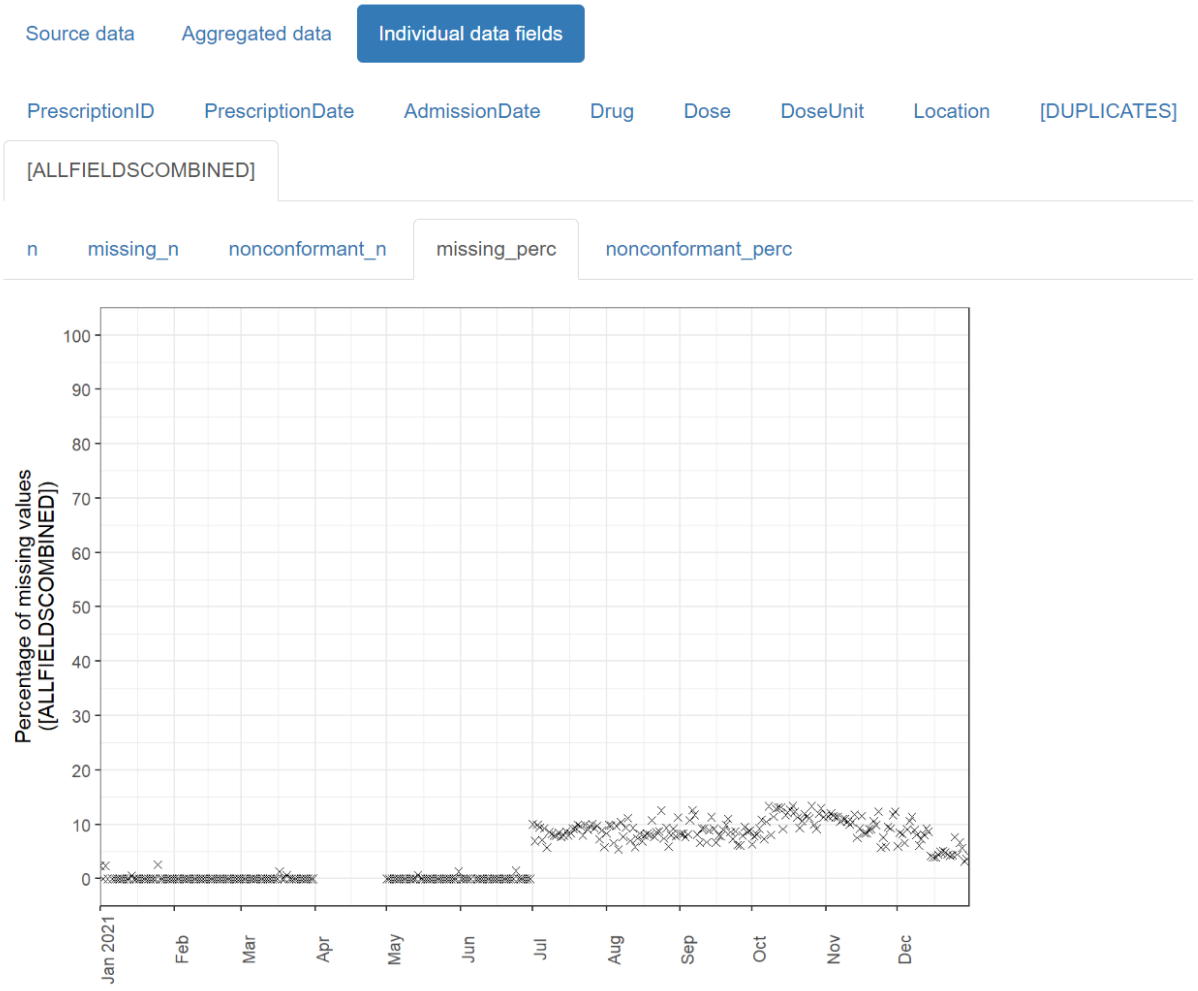


Figure 6-3. Screenshot of Individual data fields tab. This time series represents the percentage of missing values across all data fields in the dataset combined.

In addition to the report, the user can optionally choose to save a log of the processing steps that took place to create the report. This can be useful for tracing any errors or slow-running steps.

6.3 Comparison to existing tools

There were two open-source tools found during the literature review that investigated data quality in EHRs: Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems (ACHILLES)[26], and Data Quality Explorer[29].

ACHILLES produces descriptive statistics and interactive visualisations for observational health databases that conform to the Observational Medical Outcomes Partnership Common Data Model (OMOP CDM)[100]. In relation to temporal data quality, it plots time series of numbers of records relating to predefined concepts such as visit occurrence, as well as plotting prevalence of different conditions, see Figure 6-4. However, it does not appear to facilitate data quality assessments of numeric or datetime data values. Importantly, it only works on OMOP CDM databases and so will not be useable by the average EHR researcher who works with bespoke extracts of data.

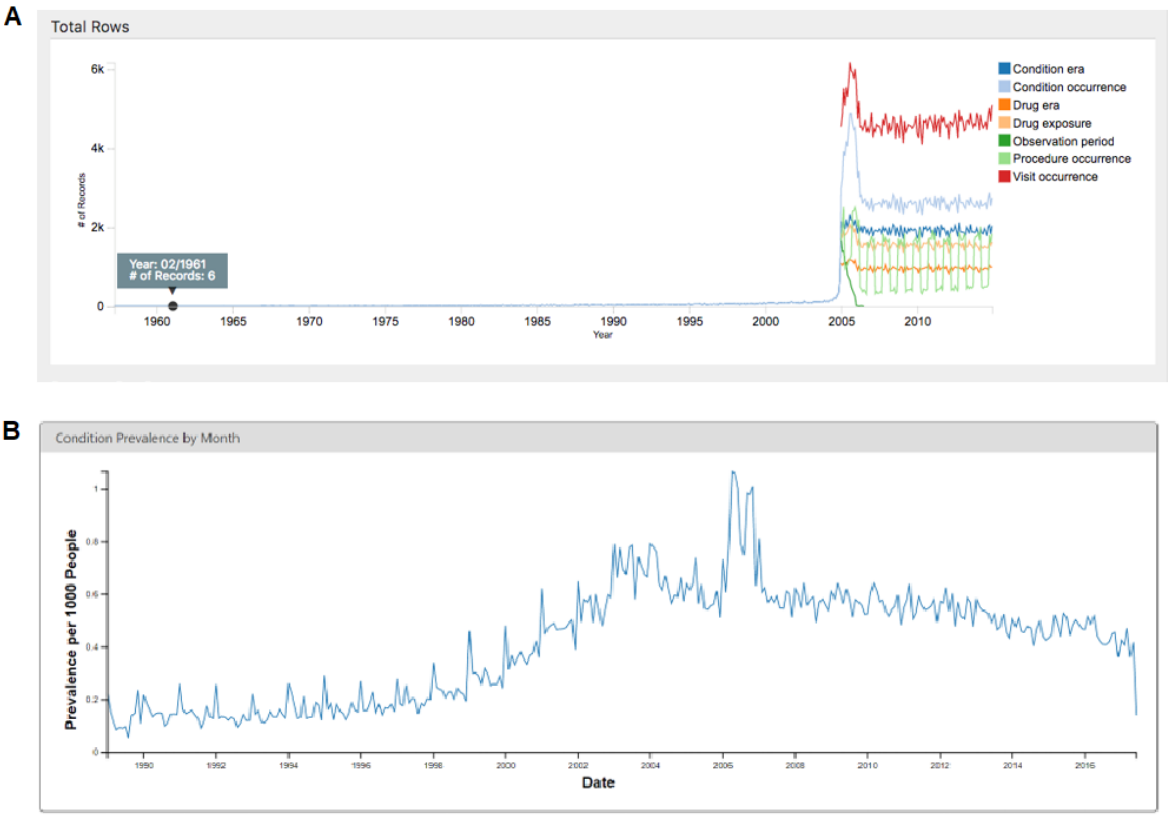


Figure 6-4. Examples of time series plots from ACHILLES software tool.
A. Data density plot showing how the bulk of the data starts in 2005. B. Monthly rate of diabetes codes in the database

Data Quality Explorer (DQ^e) is a suite of interactive, database-agnostic tools for assessing data quality, so far with 2 components: completeness (DQ^e-c)[28], and variability across site location and time (DQ^e-v)[29]. DQ^e-c produces an HTML report that contains summary statistics (frequencies of rows, unique values, missingness, and percent missingness for each column and table) and simple graphs of counts and proportions of missingness. However, no functionality that could be used to investigate temporal change points was apparent. DQ^e-v produces interactive box and scatter plots (see Figure 6-5), as well as a regression-based analysis (for smoothing and outlier identification) for variability in prevalence of a condition of interest. Similarly to *daiquiri* it works on delimited text files rather than on a database, however the data extracts need to be prepared in advance to a pre-specified format, with the frequencies of the condition(s) of interest as well as the denominators pre-calculated. This means it still requires some manual effort to create extracts in the appropriate format as well as to decide what conditions to check. Similarly to ACHILLES, it does not appear to facilitate data quality assessments of numeric or datetime data values.

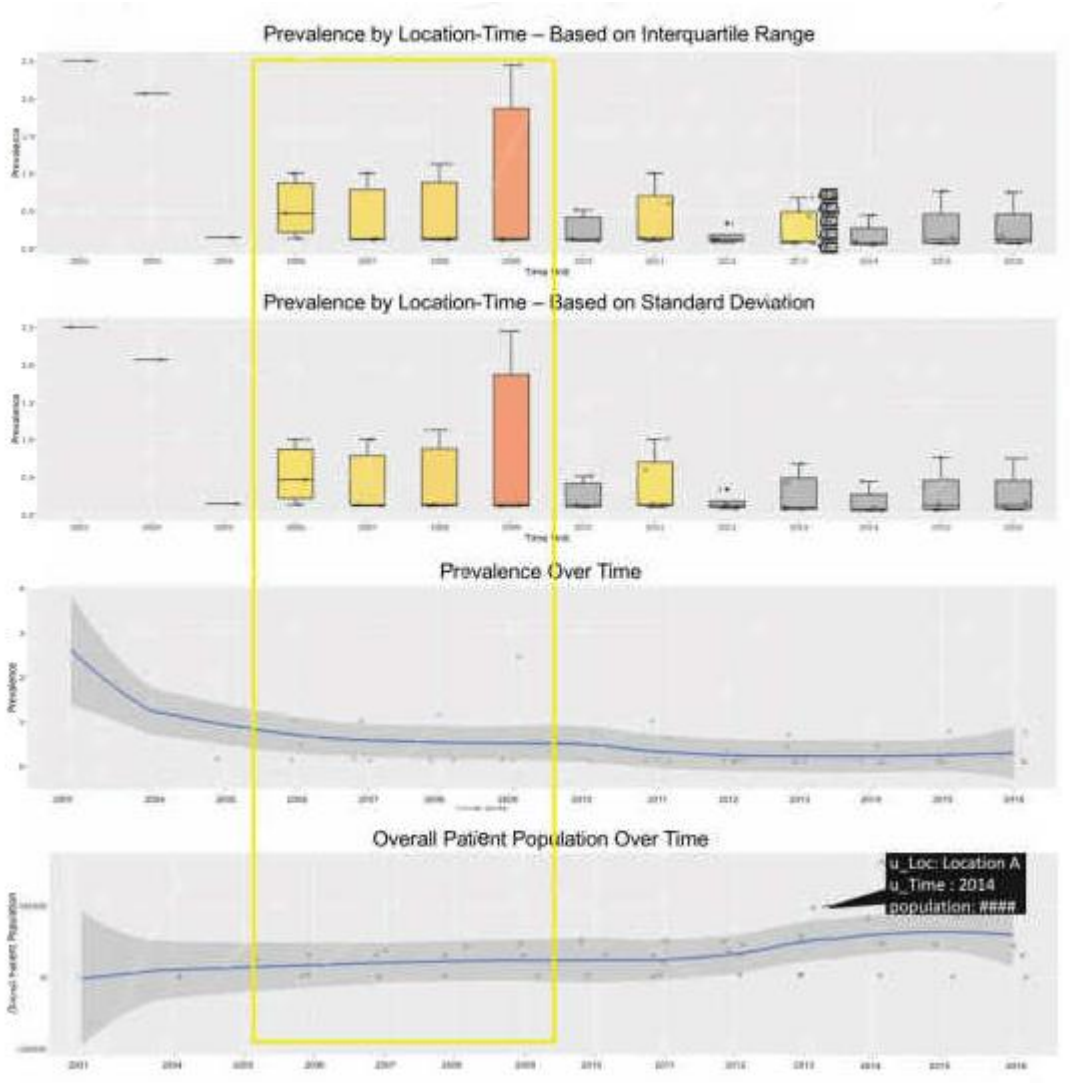


Figure 6-5. Examples of plots from DQ^e-v software tool showing variability in the number of doses per visit over time.
The yellow box highlights a period with relative high variability

As of 8th March 2022, there were 8 R packages on CRAN with the term “data quality” in their title. Three of these (*biogeo*, *fdq* and *RawHummus*) were domain-specific (for biodiversity species occurrence patterns, forest databases, and liquid chromatography–mass spectrometry instrument logs respectively) and not appropriate for use with EHR data. One package, *StatMeasures*[101], provided a range of simple functions that can be run on any generic dataset, with the data quality-focussed functions essentially reporting summary statistics (including missingness, no. of distinct values, min/max/mean depending on the datatype of each column in the data). It has since been removed from the CRAN repository

and does not appear to be actively maintained. The remaining four packages were all focussed on different aspects of EHR data.

daqapo[102] (Data Quality Assessment for Process-Oriented Data) works on activity log (one row per event, with at least two timestamps (for start and end date)) or event log (one row per event, with single timestamp for the event) data, with additional columns specifying the case identifier (e.g. hospital spell number) and the type of activity (e.g. patient registration, clinical exam). The user chooses which functions they want to run, e.g. detect_missing_values() returns records with missing values, either overall, grouped across a factor (categorical) column, or just for a specific column, and the results are output to the console and/or a data frame for later re-use. The closest piece of functionality that was found that could be used to investigate temporal data quality was the detect_inactive_periods() function, where the user specifies a threshold (in minutes) of the maximum amount of time they believe it is reasonable for there to be no events, and the function returns the points in the log where there was no activity for more than that amount of time. The package appears designed to be used interactively, and no options were apparent for creating a report from the results. Furthermore, it is designed specifically for use on process-oriented data and did not appear suitable for use on typical (data-oriented) EHR datasets.

dataquieR[103] (Data Quality in Epidemiological Research) works on generic data-oriented datasets alongside a metadata file which describes the columns in the dataset, including specifying the allowable values/ranges, codes for missing values, and expected probability distribution (where applicable). A range of different data quality functions can be run on the data (overall or stratified by a chosen categorical column), such as amount of missingness, or deviations from specified limits or values. Tables and plots are generated either as required or compiled in an html report containing multiple menu items. The closest piece of functionality that was found that could be used to investigate temporal data quality were the com_unit_missingness() and com_segment_missingness() functions when stratified by

month (see Figure 6-6), though given the tabular format this may not work well at finer-grained granularities.

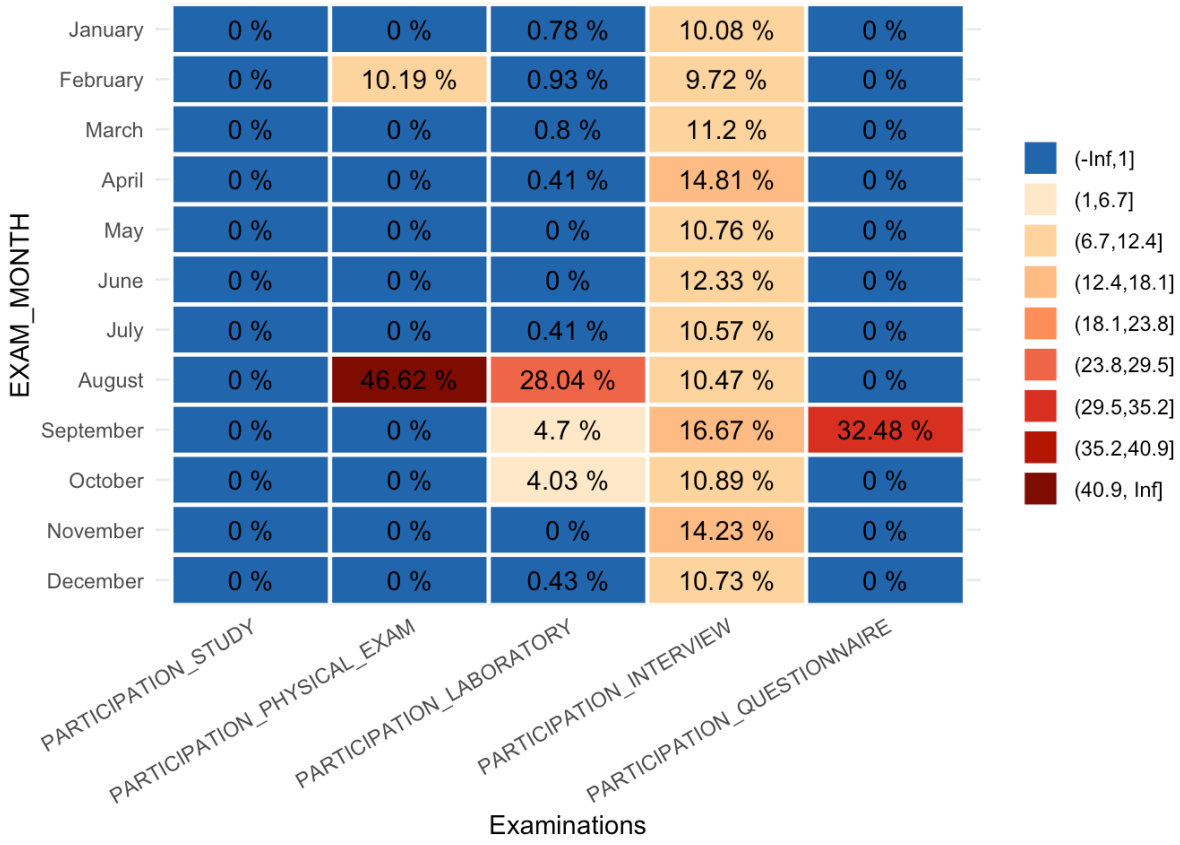


Figure 6-6. Example plot from the dataquieR package showing missingness over time

DQAstats[104] and its related graphical interface package *DQAgui*[105] provide descriptive (univariate) analyses of categorical and continuous variables of a source database and optionally compares it to a target database based on the distinct values and valid values. It appears to be designed primarily for use in institutional Extract Transform Load processes, where new data is regularly added to an existing database and thus where a set of standardised validation checks that can be repeated easily is beneficial. It checks value conformance against constraints specified in a metadata repository, and performs a small number of “atemporal plausibility” checks (such as checking for female sex if a certain examination code is used). It also checks for duplicate records. The tool provides one main function, dqa(), to create a comprehensive PDF document, which presents all statistics and results of the data quality assessment. Results are presented as tables containing summary

statistics for each column in the dataset. No functions that could be used to investigate temporal change points were apparent.

The EHR-related R packages were all first released recently (post-2020), showing that software support for data quality checking is starting to become an area of interest and activity.

In addition to the R packages described above, there were 8 packages on CRAN which contained the related term “data cleaning” in their title. “Data cleaning” is a slightly different concept to “data quality”, in that its focus is on removing or amending erroneous values, rather than on understanding their context and potential impact. For example, the *datacleanr*[106] package deploys a *Shiny* app (i.e. an interactive web-based application) to help users inspect their data for anomalies and to label and remove them in a reproducible way.

In summary, there are a small number of other software applications currently available that address data quality, but the functionality available for identifying change points was typically limited to the prevalence of different medical concepts over time. Furthermore, visualisations often treated time categorically (across separate box plots or in a table, as opposed to as time series), which is not suitable for fine-grained granularities.

6.4 Potential future enhancements

The *daiquiri* package is currently still in its infancy and there are a number of different ways that it could be enhanced to improve its functionality.

Regarding the available aggregation functions, it may be useful to allow users to configure which of the existing aggregation functions they want to apply to their data. For example, a user may not want to view plots of non-conformant data values if they happen to know that their data has already been pre-screened for non-conformance. Or a user may want the option of only producing time series for a predefined subset of aggregation functions (such

as the minimal set proposed in Chapter 4) if they know they have an exceptionally large number of data fields or if their time is limited. Further, it may be beneficial if users were able to define additional aggregation functions that they want applied to their data.

Regarding the identification of change points, it may be desirable for the plots to be interactive in terms of being able to zoom in or to change the scale, such as is made possible with the *plotly*[107] package. However, this would result in increased file size for the reports, and it is arguable that adding interactivity would make any interpretations less reproducible. The ultimate goal would be for prospective change points to be automatically identified and highlighted, though this will depend on when automated methods become accurate enough to do so.

Another area of functionality that would be useful would be a way for the user to annotate or record any change points that they have identified visually, alongside the report itself. The line drawing tool employed in the Zooniverse project to label the change points could potentially be used as a template for this.

One of the benefits of the GitHub platform is that it includes an area where users can report issues or request new features, and as more people start to use it, their feedback will inform which direction future developments should take.

6.5 Conclusion

A new and novel open-source R package called *daiquiri* was developed as part of this doctoral project, which takes a generic temporal dataset and automatically generates a wide range of time series ready for quick visual inspection. This greatly reduces the effort required for researchers to check for temporal data quality issues in their data, and thus should enable them to conduct this task more frequently and thoroughly. The resulting html reports are reproducible and shareable, and can thus contribute to forming a transparent record of the entire analysis process.

CHAPTER 7: Summary and conclusions

7.1 Chapter summary

This thesis had 4 main objectives:

1. Summarise the published literature regarding the assessment of data quality in EHRs for research, in particular the detection of sudden, unexpected temporal change points.
2. Increase our understanding of the prevalence of temporal change points in real-world EHR datasets.
3. Assess a range of methods for identifying temporal change points in real-world EHR datasets, both visually and by automation.
4. Develop open-source tools for improving the efficiency, transparency, and reproducibility of identifying temporal data quality issues in EHRs for research, in particular by using automation where possible.

This chapter summarises the main findings of the thesis in relation to these 4 objectives, discusses its major strengths and limitations, as well as its implications for future research.

7.2 Current literature relating to the assessment of data quality in EHRs for research

There was very little in the scientific literature up to mid-2019 relating to the assessment of data quality in large medical datasets. Of the six frameworks and four guidelines found which discussed intrinsic data quality in EHRs (Completeness, Conformance, and Plausibility), only four suggested specific checks to conduct within all the three categories. The most common methods suggested for assessing data quality were calculating simple summary statistics and visual inspection of graphs.

Focussing on the identification of temporal artefacts (or 'change points') in EHRs, there were particular gaps in the understanding of their prevalence as well as in efficient methods for identifying them either visually or automatically. Only two articles were found which described their prevalence, and these were limited to biomedical test results and to the coded values used in categorical data fields. Only one article employed a method which appeared promising as a way to automate their identification, though this was applied to biomedical test results rather than to EHR data in general. No articles were found which assessed the relative performance of different aggregation methods for their effectiveness for visual inspection of change points.

The lack of attention to issues of data quality management in the literature suggests that many if not most researchers who use EHR data will be unaware of, or do not appreciate the importance of, these potential biases in their datasets. This means that there is a risk of large numbers of studies where no data quality checks have been made for temporal artefacts, potentially leading to invalid results, inferences and conclusions. This is particularly important for studies where the key exposures of interest relate to calendar-date based interventions such as changes in policy.

There is insufficient reporting of the initial data analysis (IDA) stage of observational studies in medical journals[108], and data quality checks are an important aspect of IDA. There have been many technological advances in recent years, such as electronic articles which allow for inclusion of supplementary files, free platforms such as Zenodo (https://zenodo.org) and GitHub (https://github.com) which offer long-term public hosting of code and other files, and notebook tools (where code is interspersed within a human readable document) such as Jupyter (https://jupyter.org) and R Markdown (https://rmarkdown.rstudio.com) which assist in the sharing and understanding of code. As the awareness of open and reproducible research practices increases, this lack of reporting will become increasingly difficult to justify.

7.3 Understanding the prevalence of temporal change points in real-world EHR datasets

A global citizen-science project called “Health Record Hiccups” was created to collect “gold standard” labels for temporal change points (i.e. sudden changes in level, variability, trend, presence/absence of data points, or irregular outliers), using crowd-sourced visual inspection. Approximately 8000 time series plots displaying aggregated values from eight different data extracts representing patient, clinician, and laboratory activity at a large UK hospital group, were inspected for abrupt change points by over 2000 volunteers, and consensus labels generated using density based clustering with noise. The change points identified by crowd-sourced visual inspection were similarly accurate to change points identified by an experienced data scientist.

Change points were detected in all eight data extracts examined, in every data field and in almost every year of data that each extract covered. This was far higher than was initially expected. The only two studies known to have reported the prevalence of change points in EHRs also found a high number, although not as high as in this study, namely 49 changes in level and 39 changes in discretisation over 192 different laboratory tests across 17 years from a Paris hospital, and 9 changes in coding over 46 different categorical variables over 3 years at a Spanish hospital. This suggests that change points occur very frequently in EHRs.

Given the high probability of change points occurring in any data extract taken from EHRs for research, it is all the more important that data quality checks are conducted as a matter of routine at the IDA stage of studies. In order to encourage this, the culture around methods reporting needs to be changed to emphasise the importance of IDA. This could be done by extending existing reporting guidelines such as RECORD, and promoting open science practices via increased training in up-to-date methods and tools. In addition to targeting individual researchers, institutions could be incentivised to promote an open research culture. In the UK, the Research Excellence Framework is the system for assessing the

quality of research in higher education institutions, with outputs such as journal articles assessed against three criteria: originality, significance, and rigour. It would perhaps not be a bad idea to include transparency as a future criterion as well.

7.4 Assessing methods for identifying temporal change points in real-world EHR datasets

A range of methods were investigated for their utility in identifying temporal change points both visually and automatically.

7.4.1 Visual identification of change points

The time series plots which were visually inspected for change points by the citizen-science volunteers were generated by applying a variety of different aggregation functions (e.g. number of missing values or number of distinct values) and aggregation granularities (i.e. per day, week, or month) to each data field and data extract.

Every single aggregation function and aggregation granularity revealed at least one change point, though more change points were identified using a daily granularity than a weekly or monthly granularity, and functions measuring frequencies were generally more effective than functions measuring proportions. However, if these aggregation methods were used alone, many change points would have been missed.

If too many images are expected to be generated and inspected by researchers, there is a risk they will feel overwhelmed with the task and discouraged from performing it. Software tools that are user-friendly and that do the “grunt work” of generating and presenting time series images can streamline the task to the extent where it is possible to inspect hundreds of images with little apparent effort. This was found to be the case with the Zooniverse project, where the user interface enabled dozens of images to be inspected (by a single person) in a matter of minutes. Therefore, with the development of appropriate tools, the reality of the visual inspection task may be far less onerous than imagined.

Finally, it is still incumbent upon researchers to decide what the most appropriate checks are based on their particular study. Not all change points are equal, so the goal is not necessarily to find as many change points as possible, but rather in deciding which aggregation functions will provide the best chance of finding any “important” change points. While what is important will vary from one study to another, missingness is likely to be universally relevant and therefore should always be included.

7.4.2 Automated identification of change points

A range of 10 different change point detection methods from the R CRAN library were tested against the “gold standard” labels produced by crowd-sourced visual inspection across 5526 different time series. None of these automated methods were effective at identifying the locations of the change points.

The main weakness of the automated methods was that they tended to generate large numbers of false positive calls, presumably since real-world EHR data does not conform to their model assumptions. There are a number of different avenues that could be explored to develop more effective automated methods. To improve the performance of methods that assume piecewise constant segments, further investigations could be conducted to try to find suitable transformations to remove gradual trends and heteroskedasticity. However, the best performing method in this study (albeit by a small margin) did not assume piecewise constant segments but rather was an online method designed to adapt to gradual changes in mean and variability in a time series, and so that may be a better starting point. Another approach altogether might be to use pattern recognition methods to try to mimic what the human eye and brain do so well naturally. Ultimately, validation of automated methods against real-world data is essential for them to become trusted and therefore useful. The large labelled dataset created during this project will assist greatly with this.

There is a long way to go before automated change points detection methods become viable for assessing data quality, so visual inspection will continue to be the most appropriate method for the short to medium term.

7.5 Developing open-source tools for identifying temporal data quality issues

A novel open-source R package called *daiquiri* was created as part of this thesis, which assists researchers in checking for temporal data quality issues in their data by automatically generating a wide range of time series ready for quick visual inspection. The package works on generic temporal datasets containing columns of any data type, and generates time series plots showing aggregated values for each column depending on its contents (e.g. min/max/mean values for numeric data, no. of distinct values for categorical data), as well as overviews for missing values, non-conformant values, and duplicated rows.

Furthermore, the reports that are generated are easily shareable as attachments or as supplementary data files, which increases the reproducibility and transparency of this part of the initial data analysis process.

In comparison with other open-source tools identified in this study, *daiquiri* is the only one which facilitates checking for temporal change points using a variety of different aggregation functions and on different types of data fields. It also has the least set-up cost, since it doesn't require datasets to conform to pre-specified data models, or to be hosted on specific platforms, or to be supplied with detailed metadata specifications. This makes it much easier and quicker for the average researcher to use, which should lead to increased uptake.

Nonetheless, researchers can only use tools that they are aware of, and so promotion of the package is also required. Inclusion in the rOpenSci suite of packages will assist greatly with this, giving users the reassuring knowledge that it has passed software peer review, as well as providing access to rOpenSci's established community. Presentation at R conferences and R user groups will increase its exposure further. Since being made public earlier this year, the package's GitHub repository has received 18 "stars" from users around the world, from a range of fields including finance, geology, and health informatics. In the first five days following its release to CRAN, the package was downloaded 85 times by users of RStudio

(the most popular integrated development environment for R users). It has also been used by colleagues in the UK Health Security Agency and Office for National Statistics.

7.6 Strengths and limitations

This thesis makes a number of novel and substantial contributions to the much overlooked field of temporal data quality assessment of EHRs.

By systematically and thoroughly inspecting the data from a range of different EHR extracts, it has created strong evidence of the widespread prevalence of temporal change points in EHR data. This provides justification for the need to raise awareness amongst researchers of the need to conduct temporal data quality assessments on their data prior to performing their main analyses.

Additionally, by creating an easy-to-use open-source tool in the popular statistical language R, it enables the *immediate* improvement of both the conduct and transparency of the initial data analysis stage of research studies, by greatly reducing the time and effort required of researchers to check for and report on any temporal data quality issues in their data. This will result in more reliable and better quality health research, as well as increased trust in the scientific process.

Furthermore, by employing citizen-science crowd-sourcing, it has generated a large collection of real-world time series with “gold standard” labels for the locations of change points. During the course of this study, only three publicly-available time series datasets that contain real-world data with change points labelled by (expert) humans were identified. These are the Yahoo S5 dataset[109] which contains 67 real-world time series from traffic to Yahoo services, the Numenta Anomaly Benchmark[110] which contains 47 real-world time series from a variety of sources, and the Turing Change Point Dataset[111] which contains 37 time series from a range of different scientific fields. In comparison, this collection of 5526 time series provides a vastly larger sample against which to conduct benchmarking of

automated change point detection methods, which will in turn lead to much greater confidence in any results.

One limitation is that EHR data from only one hospital Trust was used, which may lead to questions about generalisability. However, firstly, the Trust includes multiple different geographically distinct hospitals (one in Banbury, the remainder in Oxford) and multiple different specialties and hence systems and cultures within them. Secondly, an attempt was made to mitigate this by covering a range of hospital activities, namely inpatient, outpatient and emergency department attendances, antibiotic prescribing by clinicians, as well as four different types of laboratory tests and results. All organisations undergo changes in systems and processes over time, so there is no reason to believe that these types of changes would not also manifest in other EHRs as well.

Another limitation is that only change detection methods that were written in the R language were considered for testing against the crowd-sourced labels, and it is possible that automated methods have been written in other languages such as Python. However, given that there have not been any large real-world collections of labelled time series available for testing and improving such methods before now, it is unlikely to expect that they would perform any better than the methods written in R.

7.7 Directions for future research

There are a number of areas which would benefit from further research. The most obvious one is the development of better automated change point detection methods, so that the work can be delegated to tireless computers. These methods would need to be able to work on real-world time series containing large amounts of background noise, and ideally filter out any gradual changes in level and variability without requiring any pre-processing by the user. Initial focus on time series representing simple counts would be the most widely applicable and useful.

The ultimate goal would be an automated system which could detect all types of abrupt change points to a high degree of accuracy, but which would also allow humans to visually confirm or reject the change points identified in order to maintain confidence in the system. Further information such as a “confidence score” or an “effect size” per change point would also be valuable to researchers in deciding which change points to prioritise their time on.

Another important question is what researchers should do when they find change points in their data. Some changes may be able to be ignored, but others may require the re-basing of values to ensure consistency either side of the change point(s). For example, a change in the percentage of missing values in a particular data field may not be important if the missingness occurs completely at random. However, a sudden change in the mean of a data field used as a variable in a longitudinal study is likely to require adjustment. Simulation studies could potentially be conducted to understand how different sizes and locations of change points can affect statistical model results.

A further area of research interest would be in investigating if it is possible to work out what may have caused a particular change point, and whether or not particular types of events manifest in recognisable ways. For example, it may be possible to obtain some clues from triangulation, e.g. if there is a sudden reduction in the length of the values in a uniqueidentifier field this would suggest that a fundamental change had occurred in the computerised system generating the identifiers, or if there was a sudden increase in the number of records at the same time as an increase in the number of distinct values in, say, a field representing a physical location, this may suggest that a new site (and associated records) had been added to the data source. Ultimately though, it will likely be impossible to confirm the exact causes without access to external sources of information.

7.8 Conclusion

This thesis has shown that temporal change points occur very frequently in EHR datasets, and so awareness of this fact should be raised among researchers who use this type of

data. Thorough assessments for temporal data quality should be conducted and reported, in order to reduce the risk of creating biased study results, and to improve reproducibility and transparency of studies.

Automated change point detection methods currently available in R are not effective at identifying these types of change points, so visual inspection will continue to be the most appropriate method for the short to medium term.

The process of visually detecting and reporting temporal data quality issues can still be made more efficient by using open-source tools such as the new *daiquiri* package in R.

References

1. Evans RS. Electronic Health Records: Then, Now, and in the Future. *Yearb Med Inform* 2016;Suppl 1:S48-61. doi: 10.15265/IYS-2016-s006 [published Online First: 2016/05/21]

2. Warren LR, Clarke J, Arora S, et al. Improving data sharing between acute hospitals in England: an overview of health record system distribution and retrospective observational analysis of inter-hospital transitions of care. *BMJ Open* 2019;9

3. NHS Digital. Hospital Episode Statistics (HES) 2022 [Available from: https://digital.nhs.uk/data-and-information/data-tools-and-services/data-services/hospital-episode-statistics accessed 28 June 2022.

4. Kahn MG, Callahan TJ, Barnard J, et al. A Harmonized Data Quality Assessment Terminology and Framework for the Secondary Use of Electronic Health Record Data. *EGEMS (Wash DC)* 2016;4(1):1244. doi: 10.13063/2327-9214.1244

5. von Elm E, Altman DG, Egger M, et al. Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) statement: guidelines for reporting observational studies. *BMJ* 2007;335(7624):806-8. doi: 10.1136/bmj.39335.541782.AD

6. Benchimol EI, Smeeth L, Guttmann A, et al. The REporting of studies Conducted using Observational Routinely-collected health Data (RECORD) statement. *PLoS Med* 2015;12(10):e1001885. doi: 10.1371/journal.pmed.1001885

7. Sauerbrei W, Abrahamowicz M, Altman DG, et al. STRengthening analytical thinking for observational studies: the STRATOS initiative. *Stat Med* 2014;33(30):5413-32. doi: 10.1002/sim.6265

8. Huebner M, Vach W, le Cessie S. A systematic approach to initial data analysis is good research practice. *J Thorac Cardiovasc Surg* 2016;151(1):25-7. doi: 10.1016/j.jtcvs.2015.09.085

9. Johnson SG, Speedie S, Simon G, et al. A Data Quality Ontology for the Secondary Use of EHR Data. *AMIA Annual Symposium proceedings AMIA Symposium* 2015;2015:1937-46. [published Online First: 2015/01/01]

10. Kahn MG, Raebel MA, Glanz JM, et al. A pragmatic framework for single-site and multisite data quality assessment in electronic health record-based clinical research. *Med Care* 2012;50 Suppl:S21-9. doi: 10.1097/MLR.0b013e318257dd67

11. Weiskopf NG, Weng C. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *J Am Med Inform Assoc* 2013;20(1):144-51. doi: 10.1136/amiajnl-2011-000681

12. Smith M, Lix LM, Azimaee M, et al. Assessing the quality of administrative data for research: a framework from the Manitoba Centre for Health Policy. *J Am Med Inform Assoc* 2017 doi: 10.1093/jamia/ocx078

13. Liaw ST, Rahimi A, Ray P, et al. Towards an ontology for data quality in integrated chronic disease management: a realist review of the literature. *Int J Med Inform* 2013;82(1):10-24. doi: 10.1016/j.ijmedinf.2012.10.001

14. Rajan NS, Gouripeddi R, Mo P, et al. Towards a content agnostic computable knowledge repository for data quality assessment. *Computer Methods and Programs in Biomedicine* 2019;177:193-201.

15. Saez C, Martinez-Miranda J, Robles M, et al. Organizing data quality assessment of shifting biomedical data. *Stud Health Technol Inform* 2012;180:721-5.

16. Kahn MG, Brown JS, Chun AT, et al. Transparent reporting of data quality in distributed data networks. *EGEMS (Wash DC)* 2015;3(1):1052. doi: 10.13063/2327-9214.1052

17. McIntosh LD, Juehne A, Vitale CRH, et al. Repeat: a framework to assess empirical reproducibility in biomedical research. *BMC Med Res Methodol* 2017;17(1):143. doi: 10.1186/s12874-017-0377-6

18. Weiskopf NG, Bakken S, Hripcsak G, et al. A Data Quality Assessment Guideline for Electronic Health Record Data Reuse. *EGEMS (Wash DC)* 2017;5(1):14. doi: 10.5334/egems.218

19. Huebner M, le Cessie S, Schmidt CO, et al. A Contemporary Conceptual Framework for Initial Data Analysis. *Observational Studies* 2018;4:171-92.

20. World Health Organization. Data quality review: a toolkit for facility data quality assessment. Geneva, 2017.

21. Sheikhtaheri A, Nahvijou A, Sedighi Z, et al. Development of a tool for comprehensive evaluation of population-based cancer registries. *Int J Med Inform* 2018;117:26-32. doi: 10.1016/j.ijmedinf.2018.06.006

22. Pukkala E. Nordic biological specimen bank cohorts as basis for studies of cancer causes and control: quality control tools for study cohorts with more than two million sample donors and 130,000 prospective cancers. *Methods Mol Biol* 2011;675:61-112. doi: 10.1007/978-1-59745-423-0_3

23. Messenger JC, Ho KK, Young CH, et al. The National Cardiovascular Data Registry (NCDR) Data Quality Brief: the NCDR Data Quality Program in 2012. *J Am Coll Cardiol* 2012;60(16):1484-8. doi: 10.1016/j.jacc.2012.07.020

24. Couchoud C, Lassalle M, Cornet R, et al. Renal replacement therapy registries--time for a structured data quality evaluation programme. *Nephrol Dial Transplant* 2013;28(9):2215-20. doi: 10.1093/ndt/gft004

25. Kodra Y, Posada de la Paz M, Coi A, et al. Data Quality in Rare Diseases Registries. *Adv Exp Med Biol* 2017;1031:149-64. doi: 10.1007/978-3-319-67144-4_8

26. Observational Health Data Sciences and Informatics. ACHILLES for data characterization [Available from: https://www.ohdsi.org/analytic-tools/achilles-for-data-characterization/ accessed 28/11/2019.

27. Huser V, DeFalco FJ, Schuemie M, et al. Multisite Evaluation of a Data Quality Tool for Patient-Level Clinical Data Sets. *EGEMS (Wash DC)* 2016;4(1):1239. doi: 10.13063/2327-9214.1239 [published Online First: 2017/02/06]

28. Estiri H, Stephens KA, Klann JG, et al. Exploring completeness in clinical data research networks with DQe-c. *J Am Med Inform Assoc* 2018;25(1):17-24. doi: 10.1093/jamia/ocx109 [published Online First: 2017/10/27]

29. Estiri H, Stephens K. DQ(e)-v: A Database-Agnostic Framework for Exploring Variability in Electronic Health Record Data Across Time and Site Location. *EGEMS (Wash DC)* 2017;5(1):3. doi: 10.13063/2327-9214.1277 [published Online First: 2018/06/23]

30. Hoeven LRV, Bruijne MC, Kemper PF, et al. Validation of multisource electronic health record data: an application to blood transfusion data. *BMC Med Inform Decis Mak* 2017;17(1):107. doi: 10.1186/s12911-017-0504-7

31. Johnson SG, Speedie S, Simon G, et al. Application of An Ontology for Characterizing Data Quality For a Secondary Use of EHR Data. *Appl Clin Inform* 2016;7(1):69-88. doi: 10.4338/ACI-2015-08-RA-0107 [published Online First: 2016/04/16]

32. Newton-Dame R, McVeigh KH, Schreibstein L, et al. Design of the New York City Macroscope: Innovations in Population Health Surveillance Using Electronic Health

Records. *EGEMS (Wash DC)* 2016;4(1):1265. doi: 10.13063/2327-9214.1265 [published Online First: 2017/02/06]

33. Taggart J, Liaw ST, Yu H. Structured data quality reports to improve EHR data quality. *Int J Med Inform* 2015;84(12):1094-8. doi: 10.1016/j.ijmedinf.2015.09.008 [published Online First: 2015/10/21]
34. Dugas M, Kuhn K, Kaiser N, et al. XML-based visualization of design and completeness in medical databases. *Med Inform Internet Med* 2001;26(4):237-50. doi: 10.1080/1463923011008137 [published Online First: 2002/01/11]
35. Sengupta S, Bachman D, Laws R, et al. Data Quality Assessment and Multi-Organizational Reporting: Tools to Enhance Network Knowledge. *EGEMS (Wash DC)* 2019;7(1):8. doi: 10.5334/egems.280 [published Online First: 2019/04/12]
36. Knowlton J, Belnap T, Patelesio B, et al. A Framework for Aligning Data from Multiple Institutions to Conduct Meaningful Analytics. *EGEMS (Wash DC)* 2017;5(3):2. doi: 10.5334/egems.195 [published Online First: 2018/06/09]
37. Health and Social Care Information Centre. Data Quality Checks Performed on SUS and HES Data: NHS Digital, 2016.
38. Walker KL, Kirillova O, Gillespie SE, et al. Using the CER Hub to ensure data quality in a multi-institution smoking cessation study. *J Am Med Inform Assoc* 2014;21(6):1129-35. doi: 10.1136/amiajnl-2013-002629 [published Online First: 2014/07/06]
39. Canadian Institute for Health Information. CIHI's Information Quality Framework: Canadian Institute for Health Information, 2017.
40. Lee K, Weiskopf N, Pathak J. A Framework for Data Quality Assessment in Clinical Research Datasets. *AMIA Annual Symposium proceedings AMIA Symposium* 2017;2017:1080-89.
41. Ouedraogo M, Kurji J, Abebe L, et al. A quality assessment of Health Management Information System (HMIS) data for maternal and child health in Jimma Zone, Ethiopia. *PLoS One* 2019;14(3):e0213600. doi: 10.1371/journal.pone.0213600 [published Online First: 2019/03/12]
42. Rogers JR, Callahan TJ, Kang T, et al. A Data Element-Function Conceptual Model for Data Quality Checks. *EGEMS (Wash DC)* 2019;7(1):17. doi: 10.5334/egems.289 [published Online First: 2019/05/09]
43. Wanyonyi KL, Radford DR, Gallagher JE. Electronic primary dental care records in research: A case study of validation and quality assurance strategies. *Int J Med Inform* 2019;127:88-94. doi: 10.1016/j.ijmedinf.2019.04.007 [published Online First: 2019/05/28]
44. Kurniati AP, Rojas E, Hogg D, et al. The assessment of data quality issues for process mining in healthcare using Medical Information Mart for Intensive Care III, a freely available e-health record database. *Health Informatics J* 2019;25(4):1878-93. doi: 10.1177/1460458218810760 [published Online First: 2018/11/30]
45. Tran DT, Havard A, Jorm LR. Data cleaning and management protocols for linked perinatal research data: a good practice example from the Smoking MUMS (Maternal Use of Medications and Safety) Study. *BMC Med Res Methodol* 2017;17(1):97. doi: 10.1186/s12874-017-0385-6
46. Dziadkowiec O, Callahan T, Ozkaynak M, et al. Using a Data Quality Framework to Clean Data Extracted from the Electronic Health Record: A Case Study. *EGEMS (Wash DC)* 2016;4(1):1201. doi: 10.13063/2327-9214.1201 [published Online First: 2016/07/19]

47. Chan KS, Fowles JB, Weiner JP. Review: electronic health records and the reliability and validity of quality measures: a review of the literature. *Med Care Res Rev* 2010;67(5):503-27. doi: 10.1177/1077558709359007

48. Jones JA, Farnell B. Missing and incomplete data reduces the value of general practice electronic medical records as data sources in research. *Aust J Prim Health* 2007;13(1):74-80.

49. Sayer GP, McGeechan K, Kemp A, et al. The General Practice Research Network: the capabilities of an electronic patient management system for longitudinal patient data. *Pharmacoepidemiol Drug Saf* 2003;12(6):483-9. doi: 10.1002/pds.834 [published Online First: 2003/09/30]

50. Callahan TJ, Bauck AE, Bertoch D, et al. A Comparison of Data Quality Assessment Checks in Six Data Sharing Networks. *EGEMS (Wash DC)* 2017;5(1):8. doi: 10.5334/egems.223 [published Online First: 2018/06/09]

51. Khare R, Utidjian L, Ruth BJ, et al. A longitudinal analysis of data quality in a large pediatric data research network. *J Am Med Inform Assoc* 2017;24(6):1072-79. doi: 10.1093/jamia/ocx033 [published Online First: 2017/04/12]

52. Chen H, Hailey D, Wang N, et al. A review of data quality assessment methods for public health information systems. *Int J Environ Res Public Health* 2014;11(5):5170-207. doi: 10.3390/ijerph110505170

53. Liaw ST, Taggart J, Dennis S, et al. Data quality and fitness for purpose of routinely collected data--a general practice case study from an electronic practice-based research network (ePBRN). *AMIA Annual Symposium proceedings AMIA Symposium* 2011;2011:785-94. [published Online First: 2011/12/24]

54. Kim HS, Lee SH, Kim TM, et al. Developing a multi-center clinical data mart of ACEI and ARB for real-world evidence (RWE). *Clin Hypertens* 2018;24:18. doi: 10.1186/s40885-018-0103-7 [published Online First: 2018/12/20]

55. Goldberg SI, Shubina M, Niemierko A, et al. A Weighty Problem: Identification, Characteristics and Risk Factors for Errors in EMR Data. *AMIA Annual Symposium proceedings AMIA Symposium* 2010;2010:251-5. [published Online First: 2011/02/25]

56. van Vlymen J, de Lusignan S, Hague N, et al. Ensuring the Quality of Aggregated General Practice Data: Lessons from the Primary Care Data Quality Programme (PCDQ). *Stud Health Technol Inform* 2005;116:1010-5. [published Online First: 2005/09/15]

57. Berndt DJ, Fisher JW, Hevner AR, et al. Healthcare data warehousing and quality assurance. *Computer* 2001;34(12):56-65.

58. Hodgson S, Beale L, Parslow RC, et al. Creating a national register of childhood type 1 diabetes using routinely collected hospital data. *Pediatr Diabetes* 2012;13(3):235-43. doi: 10.1111/j.1399-5448.2011.00815.x [published Online First: 2011/10/25]

59. Qualls LG, Phillips TA, Hammill BG, et al. Evaluating Foundational Data Quality in the National Patient-Centered Clinical Research Network (PCORnet(R)). *EGEMS (Wash DC)* 2018;6(1):3. doi: 10.5334/egems.199 [published Online First: 2018/06/09]

60. Cox DR, Donnelly CA. Principles of applied statistics. Cambridge, UK ; New York: Cambridge University Press 2011.

61. Han J, Kamber M, Pei J. Data Mining Concepts and Techniques. Third edition. ed. Burlington: Elsevier Science,, 2011:1 online resource (744 pages).

62. Petrie A, Sabin C. Medical statistics at a glance. 2nd ed. Malden, Mass.: Blackwell 2005.

63. Haynes K, Bilker WB, Tenhave TR, et al. Temporal and within practice variability in the health improvement network. *Pharmacoepidemiol Drug Saf* 2011;20(9):948-55. doi: 10.1002/pds.2191 [published Online First: 2011/07/15]

64. Bailie R, Bailie J, Chakraborty A, et al. Consistency of denominator data in electronic health records in Australian primary healthcare services: enhancing data quality. *Aust J Prim Health* 2015;21(4):450-9. doi: 10.1071/PY14071 [published Online First: 2014/10/28]

65. Ta CN, Weng C. Detecting Systemic Data Quality Issues in Electronic Health Records. *Stud Health Technol Inform* 2019;264:383-87. doi: 10.3233/shti190248 [published Online First: 2019/08/24]

66. Garcia-de-Leon-Chocano R, Munoz-Soler V, Saez C, et al. Construction of quality-assured infant feeding process of care data repositories: Construction of the perinatal repository (Part 2). *Comput Biol Med* 2016;71:214-22. doi: 10.1016/j.compbiomed.2016.01.007 [published Online First: 2016/03/08]

67. Perez-Benito FJ, Saez C, Conejero JA, et al. Temporal variability analysis reveals biases in electronic health records due to hospital process reengineering interventions over seven years. *PLoS One* 2019;14(8):e0220369. doi: 10.1371/journal.pone.0220369 [published Online First: 2019/08/08]

68. Looten V, Kong Win Chang L, Neuraz A, et al. What can millions of laboratory test results tell us about the temporal aspect of data quality? Study of data spanning 17 years in a clinical data warehouse. *Comput Methods Programs Biomed* 2018:104825. doi: 10.1016/j.cmpb.2018.12.030 [published Online First: 2019/01/08]

69. Amin SG. Control charts 101: a guide to health care applications. *Qual Manag Health Care* 2001;9(3):1-27. doi: 10.1097/00019514-200109030-00003 [published Online First: 2001/05/25]

70. Saez C, Zurriaga O, Perez-Panades J, et al. Applying probabilistic temporal and multisite data quality control methods to a public health mortality registry in Spain: a systematic approach to quality control of repositories. *J Am Med Inform Assoc* 2016;23(6):1085-95. doi: 10.1093/jamia/ocw010 [published Online First: 2016/04/24]

71. Pollock AM, Benster R, Vickers N. Why did treatment rates for colorectal cancer in south east England fall between 1982 and 1988? The effect of case ascertainment and registration bias. *J Public Health Med* 1995;17(4):419-28.

72. Ali M, Tins J, Burckhardt BB, et al. Fit-for-Purpose Quality Control System in Continuous Bioanalysis During Long-Term Pediatric Studies. *AAPS J* 2019;21(6):104. doi: 10.1208/s12248-019-0375-1 [published Online First: 2019/09/06]

73. Yasaka T, Nakano K, Nakano M, et al. The long-term quality control and clinical data evaluation of individuals in AMHTS. *Med Inform (Lond)* 1979;4(3):173-85. doi: 10.3109/14639237909010912 [published Online First: 1979/07/01]

74. Giunta D, Fuentes N, Pazo V, et al. Creation of a hyponatremia registry supported by an industry-derived quality control methodology. *Appl Clin Inform* 2011;2(1):86-93. doi: 10.4338/ACI-2010-07-RA-0040 [published Online First: 2011/01/01]

75. Bhattacharya AA, Umar N, Audu A, et al. Quality of routine facility data for monitoring priority maternal and newborn indicators in DHIS2: A case study from Gombe State, Nigeria. *PLoS One* 2019;14(1):e0211265. doi: 10.1371/journal.pone.0211265 [published Online First: 2019/01/27]

76. Saez C, Garcia-Gomez JM. Kinematics of Big Biomedical Data to characterize temporal variability and seasonality of data repositories: Functional Data Analysis of data temporal evolution over non-parametric statistical manifolds. *Int J Med Inform*

2018;119:109-24. doi: 10.1016/j.ijmedinf.2018.09.015 [published Online First: 2018/10/22]

77. Arksey H, O'Malley L. Scoping studies: towards a methodological framework. *International Journal of Social Research Methodology* 2005;8(1):19-32. doi: 10.1080/1364557032000119616

78. Munn Z, Peters MDJ, Stern C, et al. Systematic review or scoping review? Guidance for authors when choosing between a systematic or scoping review approach. *BMC Medical Research Methodology* 2018;18(1):143. doi: 10.1186/s12874-018-0611-x

79. Hemkens LG, Benchimol EI, Langan SM, et al. The reporting of studies using routinely collected health data was often insufficient. *J Clin Epidemiol* 2016;79:104-11. doi: 10.1016/j.jclinepi.2016.06.005

80. Ester M, Kriegel H-P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Portland, Oregon: AAAI Press, 1996:226–31.

81. Flahault A, Cadilhac M, Thomas G. Sample size calculation should be performed for design accuracy in diagnostic test studies. *J Clin Epidemiol* 2005;58(8):859-62. doi: 10.1016/j.jclinepi.2004.12.009 [published Online First: 2005/07/16]

82. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 2020;21(1):6. doi: 10.1186/s12864-019-6413-7 [published Online First: 2020/01/04]

83. Fowler PW, Wright C, Spiers H, et al. A crowd of BashTheBug volunteers reproducibly and accurately measure the minimum inhibitory concentrations of 13 antitubercular drugs from photographs of 96-well broth microdilution plates. *Elife* 2022;11 doi: 10.7554/eLife.75046 [published Online First: 2022/05/20]

84. Open Science Collaboration. PSYCHOLOGY. Estimating the reproducibility of psychological science. *Science* 2015;349(6251):aac4716. doi: 10.1126/science.aac4716 [published Online First: 2015/09/01]

85. Baker M. 1,500 scientists lift the lid on reproducibility. *Nature* 2016;533(7604):452-4. doi: 10.1038/533452a [published Online First: 2016/05/27]

86. Nosek BA, Alter G, Banks GC, et al. SCIENTIFIC STANDARDS. Promoting an open research culture. *Science* 2015;348(6242):1422-5. doi: 10.1126/science.aab2374 [published Online First: 2015/06/27]

87. The Comprehensive R Archive Network [Available from: https://cran.r-project.org/ accessed 1/3/2019.

88. Verbesselt J, Hyndman R, Newnham G, et al. Detecting Trend and Seasonal Changes in Satellite Image Time Series. *Remote Sensing of Environment* 2010;114(1):106-15. doi: 10.1016/j.rse.2009.08.014

89. Bai J, Perron P. Critical values for multiple structural change tests. *Econometrics Journal* 2003;6:72-78.

90. Killick R, Eckley I. changepoint: An R Package for Changepoint Analysis. *Journal of Statistical Software* 2014;58(3):1-19.

91. Haynes K, Fearnhead P, Eckley I. A computationally efficient nonparametric approach for changepoint detection. *Statistics and Computing* 2017;27(5):1293-305.

92. Romano G, Rigaill G, Runge V, et al. Detecting Abrupt Changes in the Presence of Local Fluctuations and Autocorrelated Noise. *Journal of the American Statistical Association* 2021 doi: 10.1080/01621459.2021.1909598

93. Ding J, Xiang Y, Shen L, et al. Multiple Change Point Analysis: Fast Implementation and Strong Consistency. *IEEE Transactions on Signal Processing* 2017;65(17):4495-510. doi: 10.1109/TSP.2017.2711558

94. Raza H, Prasad G, Li Y. EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments. *Pattern Recognition* 2015;48(3):659-69.

95. Carter KM, Streilein WW. Probabilistic reasoning for streaming anomaly detection. *IEEE Statistical Signal Processing Workshop (SSP)* 2012:377-80. doi: 10.1109/SSP.2012.6319708

96. Zeileis A, Leisch F, Hornik K, et al. strucchange: An R Package for Testing for Structural Change in Linear Regression Models. *Journal of Statistical Software* 2002;7(2):1-38. doi: 10.18637/jss.v007.i02

97. Zeileis A, Kleiber C, Krämer W, et al. Testing and Dating of Structural Changes in Practice. *Computational Statistics & Data Analysis* 2003;44:109-23.

98. Maeng H, Fryzlewicz P. Detecting linear trend changes in data sequences. *preprint* 2021

99. Jensen PB, Jensen LJ, Brunak S. Mining electronic health records: towards better research applications and clinical care. *Nat Rev Genet* 2012;13(6):395-405. doi: 10.1038/nrg3208

100. Observational Health Data Sciences and Informatics. OMOP Common Data Model [Available from: https://www.ohdsi.org/data-standardization/the-common-data-model/.

101. Jain A. StatMeasures: Easy Data Manipulation, Data Quality and Statistical Checks. R package version 1.0. 2015. https://CRAN.R-project.org/package=StatMeasures.

102. Martin N. daqapo: Data Quality Assessment for Process-Oriented Data. R package version 0.3.2. 2022. https://CRAN.R-project.org/package=daqapo

103. Richter A, Schmidt CO, Struckmann S. dataquieR: Data Quality in Epidemiological Research. R package version 1.0.9. 2021. https://dataquality.ship-med.uni-greifswald.de/.

104. Kapsner LA, Mang JM, Mate S, et al. Linking a Consortium-Wide Data Quality Assessment Tool with the MIRACUM Metadata Repository. *Applied Clinical Informatics* 2021;12(4):826-35. doi: 10.1055/s-0041-1733847

105. Kapsner LA, Mang JM. DQAgui: Graphical User Interface for Data Quality Assessment. R package version 0.1.9. 2022. https://CRAN.R-project.org/package=DQAgui.

106. Hurley A. datacleanr: Interactive and Reproducible Data Cleaning. R package version 1.0.3. 2021. https://CRAN.R-project.org/package=datacleanr.

107. Sievert C. Interactive Web-Based Data Visualization with R, plotly, and shiny: Chapman and Hall/CRC 2020.

108. Huebner M, Vach W, le Cessie S, et al. Hidden analyses: a review of reporting practice and recommendations for more transparent reporting of initial data analyses. *BMC Med Res Methodol* 2020;20(1):61. doi: 10.1186/s12874-020-00942-y [published Online First: 2020/03/15]

109. Yahoo Research. S5 - A Labeled Anomaly Detection Dataset Webscope2015 [Available from: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70 accessed 12 July 2022.

110. Ahmad S, Lavin A, Purdy S, et al. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 2017;262:134-47. doi: 10.1016/j.neucom.2017.04.070

111. van den Burg GJJ, Williams CKI. An Evaluation of Change Point Detection Algorithms. *arXiv preprint* 2020 doi: 10.48550/ARXIV.2003.06222

Appendix A: Chapter 2

A.1 Synonyms for umbrella search terms used

All terms *within* each column were combined with OR operators, before constructing the overall search formula, e.g. for (“Data quality”) AND (“EHRs”) AND (“Research”). Synonyms were constructed in collaboration with a librarian.

Data quality	**EHRs**	**Research**	**Temporal**
data quality	EHR$	research	temporal
data cleaning	electronic health record$	secondary use	time-series
data accuracy	electronic patient record$	observational stud$	secular
quality control	routinely collected data$	epidemiolog$	trend$
initial data analysis	administrative data$	secondary data use	time series
data screening	electronic medical record$		longitudinal
information quality	EMR$		long term
	data warehouse		over time
	clinical research data$		
	biomedical data$		
	secondary data$		
	clinical data$		
	registry		
	registries		
	information system$		

Appendix B: Chapter 3

B.1 List of data fields included

There were 8 data extracts included in total. One data field from each data extract was chosen to be the ‘timepoint’ field, and this was used to represent the date of the record. In the IORD database, patient identifiers are replaced with a ClusterID, and each data extract includes this pseudonym as well as the Sex, BirthMonth and DeathDate derived during the anonymisation process.

Data extract	Data field	Data field type
antibiotics	PrescriptionID	uniqueidentifier
antibiotics	PrescriptionDate	timepoint
antibiotics	PrescriptionStatus	categorical
antibiotics	PrescriptionType	categorical
antibiotics	Drug	categorical
antibiotics	Formulation	categorical
antibiotics	Dose	numeric
antibiotics	DoseUnit	categorical
antibiotics	Frequency	categorical
antibiotics	Route	categorical
antibiotics	Indication	freetext
antibiotics	ScheduledStartDateTime	datetime
antibiotics	ScheduledStopDateTime	datetime
antibiotics	DurationEnteredByPrescriber	freetext
antibiotics	ReviewDateTime	datetime
antibiotics	FirstAdministrationDateTime	datetime
antibiotics	LastAdministrationDateTime	datetime
antibiotics	NumberOfDosesAdministered	numeric
antibiotics	SpecialInstructions	freetext
antibiotics	Weight	numeric
antibiotics	Clusterid	uniqueidentifier
antibiotics	Sex	categorical
antibiotics	BirthMonth	datetime
antibiotics	Deathdate	datetime

Data extract	Data field	Data field type
antibiotics	AntibioticsSource	categorical
inpat_episode	EpisodeID	uniqueidentifier
inpat_episode	SpellID	uniqueidentifier
inpat_episode	EpisodeNumber	numeric
inpat_episode	EpisodeStartDate	datetime
inpat_episode	EpisodeEndDate	datetime
inpat_episode	DominantEpisode	categorical
inpat_episode	AdmissionDate	datetime
inpat_episode	AdmissionMethodCode	categorical
inpat_episode	AdmissionSourceCode	categorical
inpat_episode	DischargeDate	timepoint
inpat_episode	DischargeMethodCode	categorical
inpat_episode	DischargeDestinationCode	categorical
inpat_episode	IntendedManagementCode	categorical
inpat_episode	PatientClassificationCode	categorical
inpat_episode	ConsultantCodeAnon	uniqueidentifier
inpat_episode	ConsultantMainSpecialtyCode	categorical
inpat_episode	TreatmentFunctionCode	categorical
inpat_episode	LocalSubSpecialtyCode	categorical
inpat_episode	SiteCode	categorical
inpat_episode	EthnicGroupCode	categorical
inpat_episode	AdministrativeCategoryCode	categorical
inpat_episode	GPPracticeCode	categorical
inpat_episode	ReferringOrganisationCode	categorical
inpat_episode	CarerSupportCode	categorical
inpat_episode	OperationStatusCode	categorical
inpat_episode	NeonatalLevelOfCareCode	categorical
inpat_episode	FirstWard	categorical
inpat_episode	LastWard	categorical
inpat_episode	PrimaryDiagCode	freetext
inpat_episode	FirstProcCode	freetext
inpat_episode	FirstProcDate	datetime
inpat_episode	CDSSubmissionStatus	freetext
inpat_episode	ClusterID	uniqueidentifier

Data extract	Data field	Data field type
inpat_episode	Sex	categorical
inpat_episode	BirthMonth	datetime
inpat_episode	Deathdate	datetime
inpat_episode	PostcodeStub	categorical
inpat_episode	IMDScore	numeric
inpat_episode	LocalAuthority	categorical
outpat_episode	EpisodeID	uniqueidentifier
outpat_episode	SiteCode	categorical
outpat_episode	GPPracticeCode	categorical
outpat_episode	ReferringOrganisationCode	categorical
outpat_episode	EthnicGroupCode	categorical
outpat_episode	AdministrativeCategoryCode	categorical
outpat_episode	ConsultantMainSpecialtyCode	categorical
outpat_episode	TreatmentFunctionCode	categorical
outpat_episode	LocalSubSpecialtyCode	categorical
outpat_episode	ConsultantCodeAnon	uniqueidentifier
outpat_episode	CarerSupportCode	categorical
outpat_episode	OperationStatusCode	categorical
outpat_episode	PrimaryDiagCode	freetext
outpat_episode	FirstProcCode	freetext
outpat_episode	FirstProcDate	datetime
outpat_episode	ClinicCode	categorical
outpat_episode	AppointmentDate	timepoint
outpat_episode	FirstAttendanceCode	categorical
outpat_episode	AttendanceStatusCode	categorical
outpat_episode	AttendanceOutcomeCode	categorical
outpat_episode	ReferralSourceCode	categorical
outpat_episode	ServiceTypeCode	categorical
outpat_episode	PriorityTypeCode	categorical
outpat_episode	LastDNAorCancelledDate	datetime
outpat_episode	MedicalStaffType	categorical
outpat_episode	CDSSubmissionStatus	freetext
outpat_episode	ClusterID	uniqueidentifier
outpat_episode	Sex	categorical

Data extract	Data field	Data field type
outpat_episode	BirthMonth	datetime
outpat_episode	Deathdate	datetime
outpat_episode	PostcodeStub	categorical
outpat_episode	IMDScore	numeric
outpat_episode	LocalAuthority	categorical
ed_attendance	AttendanceID	uniqueidentifier
ed_attendance	SiteCode	categorical
ed_attendance	ArrivalDateTime	datetime
ed_attendance	AssessmentDateTime	datetime
ed_attendance	TreatmentDateTime	datetime
ed_attendance	ConclusionDateTime	datetime
ed_attendance	DepartureDateTime	timepoint
ed_attendance	ArrivalModeCode	categorical
ed_attendance	AttendanceCategoryCode	categorical
ed_attendance	AttendanceDisposalCode	categorical
ed_attendance	IncidentLocationTypeCode	categorical
ed_attendance	PatientGroupCode	categorical
ed_attendance	ReferralSourceCode	categorical
ed_attendance	GPPracticeCode	categorical
ed_attendance	EthnicGroupCode	categorical
ed_attendance	PrimaryDiagCodeFull	categorical
ed_attendance	PrimaryInvestCode	categorical
ed_attendance	PrimaryTreatCodeFull	categorical
ed_attendance	CDSSubmissionStatus	freetext
ed_attendance	ClusterID	uniqueidentifier
ed_attendance	Sex	categorical
ed_attendance	BirthMonth	datetime
ed_attendance	Deathdate	datetime
ed_attendance	PostcodeStub	categorical
ed_attendance	IMDScore	numeric
ed_attendance	LocalAuthority	categorical
bchem_creatinine	CollectionID	uniqueidentifier
bchem_creatinine	CollectionDateTime	timepoint
bchem_creatinine	ReceiveDateTime	datetime

Data extract	Data field	Data field type
bchem_creatinine	Location	categorical
bchem_creatinine	ElecReqID	uniqueidentifier
bchem_creatinine	LabNo	uniqueidentifier
bchem_creatinine	SetID	uniqueidentifier
bchem_creatinine	SetName	categorical
bchem_creatinine	Comment	freetext
bchem_creatinine	AuthorisationDate	datetime
bchem_creatinine	FinalDate	datetime
bchem_creatinine	DateTimeComment	freetext
bchem_creatinine	TestID	uniqueidentifier
bchem_creatinine	TestName	categorical
bchem_creatinine	RefRange	categorical
bchem_creatinine	Units	categorical
bchem_creatinine	Value	numeric
bchem_creatinine	LIMSSource	categorical
bchem_creatinine	ClusterID	uniqueidentifier
bchem_creatinine	Sex	categorical
bchem_creatinine	BirthMonth	datetime
bchem_creatinine	Deathdate	datetime
haem_neutrophils	CollectionID	uniqueidentifier
haem_neutrophils	CollectionDateTime	timepoint
haem_neutrophils	ReceiveDateTime	datetime
haem_neutrophils	Location	categorical
haem_neutrophils	ElecReqID	uniqueidentifier
haem_neutrophils	LabNo	uniqueidentifier
haem_neutrophils	SetID	uniqueidentifier
haem_neutrophils	SetName	categorical
haem_neutrophils	Comment	freetext
haem_neutrophils	AuthorisationDate	datetime
haem_neutrophils	FinalDate	datetime
haem_neutrophils	DateTimeComment	freetext
haem_neutrophils	TestID	uniqueidentifier
haem_neutrophils	TestName	categorical
haem_neutrophils	RefRange	categorical

Data extract	Data field	Data field type
haem_neutrophils	Units	categorical
haem_neutrophils	Value	numeric
haem_neutrophils	LIMSSource	categorical
haem_neutrophils	ClusterID	uniqueidentifier
haem_neutrophils	Sex	categorical
haem_neutrophils	BirthMonth	datetime
haem_neutrophils	Deathdate	datetime
micro_blc	CollectionDateTime	timepoint
micro_blc	AccessionNumber	uniqueidentifier
micro_blc	BatTestCode	categorical
micro_blc	BatTestName	categorical
micro_blc	TestCode	categorical
micro_blc	TestName	categorical
micro_blc	SpecimenCode	categorical
micro_blc	SpecimenFull	freetext
micro_blc	RequestCode	categorical
micro_blc	RequestFull	freetext
micro_blc	ReceiveDateTime	datetime
micro_blc	ResultDateTime	datetime
micro_blc	ResultMethod	categorical
micro_blc	Result	freetext
micro_blc	ResultFull	freetext
micro_blc	ResultFurtherInfo	freetext
micro_blc	PatientLastKnownLocation	categorical
micro_blc	CreditDateTime	datetime
micro_blc	FinalDateTime	datetime
micro_blc	SpecimenCollectionLocation	categorical
micro_blc	BugCode	categorical
micro_blc	BugCodeFull	categorical
micro_blc	BugName	categorical
micro_blc	BugResult	freetext
micro_blc	SusceptabilityBatch	categorical
micro_blc	SusceptabilityMethod	categorical
micro_blc	DrugCode	categorical

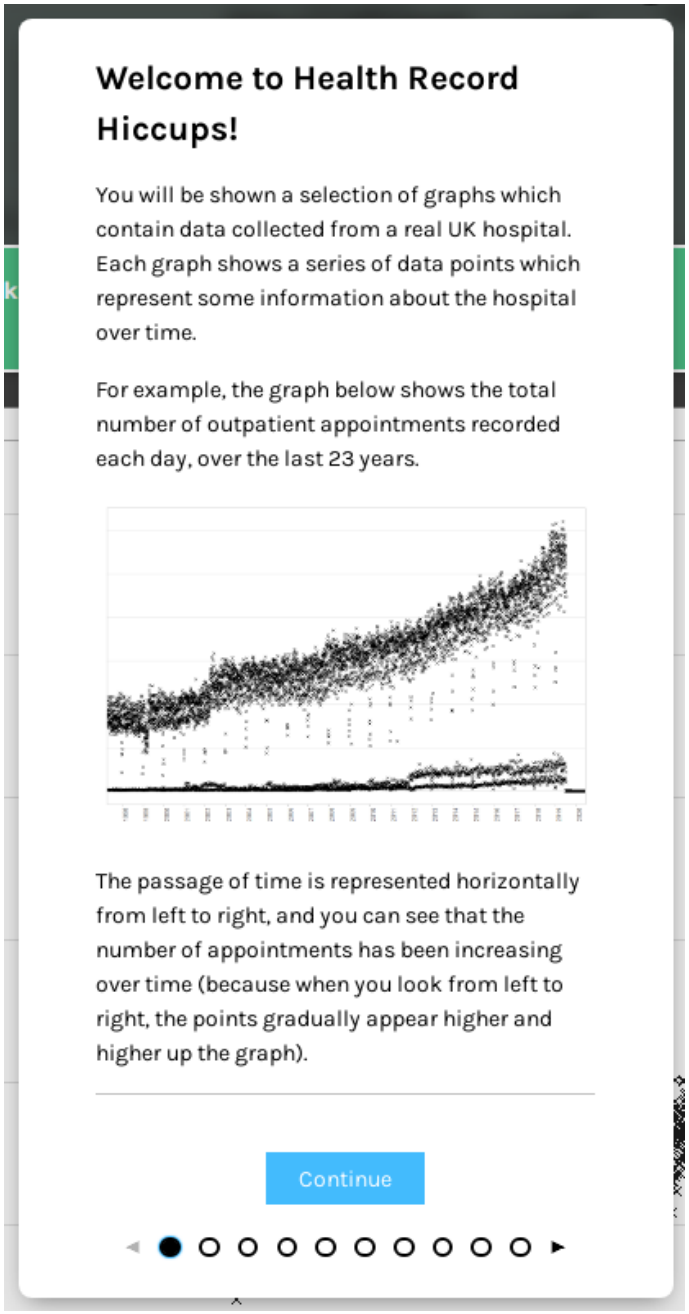
Data extract	Data field	Data field type
micro_blc	DrugName	categorical
micro_blc	DrugResult	freetext
micro_blc	MIC	freetext
micro_blc	ClusterID	uniqueidentifier
micro_blc	Sex	categorical
micro_blc	BirthMonth	datetime
micro_blc	Deathdate	datetime
micro_blc	MicroSource	categorical
micro_ecol	CollectionDateTime	timepoint
micro_ecol	AccessionNumber	uniqueidentifier
micro_ecol	BatTestCode	categorical
micro_ecol	BatTestName	categorical
micro_ecol	TestCode	categorical
micro_ecol	TestName	categorical
micro_ecol	SpecimenCode	categorical
micro_ecol	SpecimenFull	freetext
micro_ecol	RequestCode	categorical
micro_ecol	RequestFull	freetext
micro_ecol	ReceiveDateTime	datetime
micro_ecol	ResultDateTime	datetime
micro_ecol	ResultMethod	categorical
micro_ecol	Result	freetext
micro_ecol	ResultFull	freetext
micro_ecol	ResultFurtherInfo	freetext
micro_ecol	PatientLastKnownLocation	categorical
micro_ecol	CreditDateTime	datetime
micro_ecol	FinalDateTime	datetime
micro_ecol	SpecimenCollectionLocation	categorical
micro_ecol	BugCode	categorical
micro_ecol	BugCodeFull	categorical
micro_ecol	BugName	categorical
micro_ecol	BugResult	freetext
micro_ecol	SusceptabilityBatch	categorical
micro_ecol	SusceptabilityMethod	categorical

Data extract	Data field	Data field type
micro_ecol	DrugCode	categorical
micro_ecol	DrugName	categorical
micro_ecol	DrugResult	freetext
micro_ecol	MIC	freetext
micro_ecol	ClusterID	uniqueidentifier
micro_ecol	Sex	categorical
micro_ecol	BirthMonth	datetime
micro_ecol	Deathdate	datetime
micro_ecol	MicroSource	categorical

B.2 Full instructions for Zooniverse project

Volunteers were presented with a 10-step tutorial before they were shown any real images for classification. This tutorial could be re-accessed at any time. In addition, a field guide (which contained additional examples) and a help page were also available at any time.

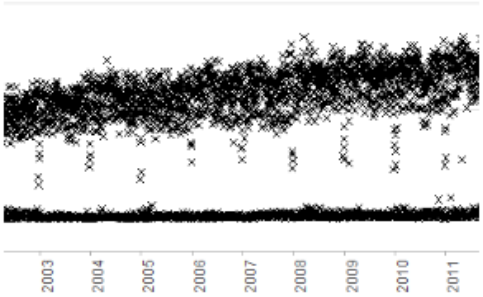
Tutorial



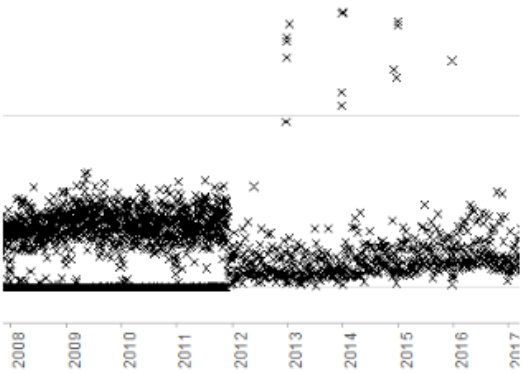
What you are looking for

Any data that represents real life will naturally change over time, as the world it represents inevitably changes.

Where changes are "real", we would expect them to appear smoothly or perhaps to repeat regularly. For example:



However, in practice we often also see **sudden**, **unnatural-looking** changes, which raise suspicions about the consistency of the data, and it is these that we want you to identify. For example, see how the data in the graph below looks very different before versus after 2012:

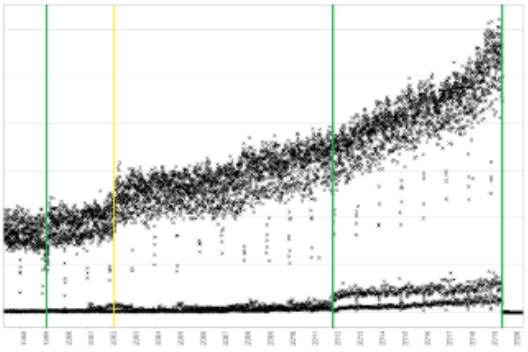


Continue

What you need to do

Your task is to look for and mark on the image wherever you see a ***sudden*** change in the way the data points appear over time.

You do this by adding one or more vertical lines to the image - green if it is a clear change, or yellow if you are unsure.



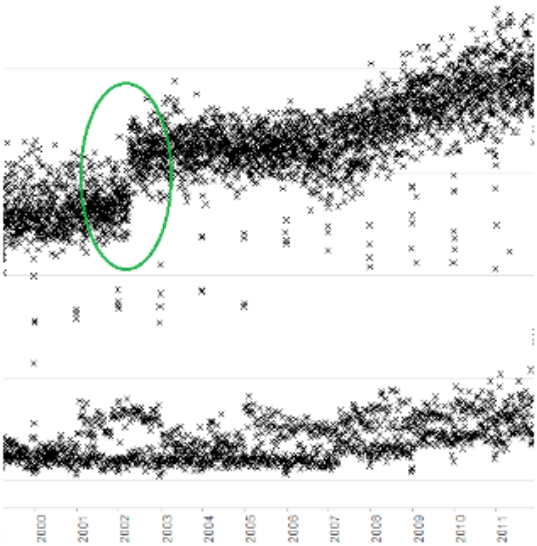
There are different ways in which the data can change suddenly, and we will describe these in detail next. You don't need to remember the names, you just need to be able to spot them when they occur. You can also find them in the field guide at any time.

Continue

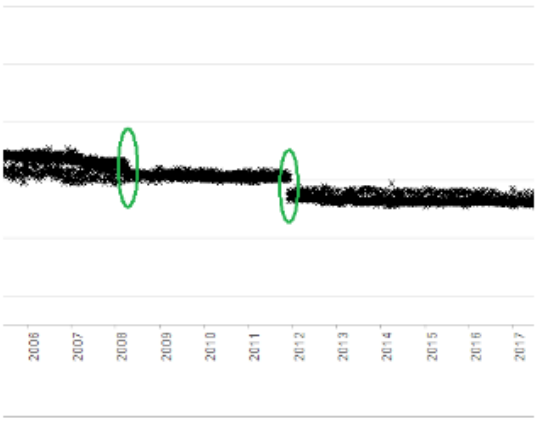
Sudden step change

A **step** change is a sudden shift up or down, like a step on a staircase.

For example:



Or:



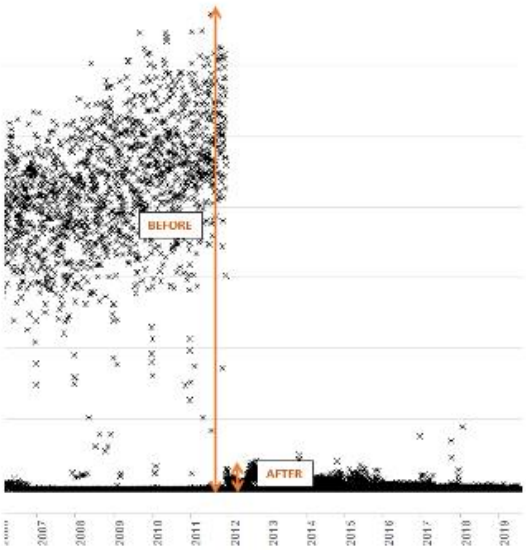
Continue

◂ ○ ○ ○ ● ○ ○ ○ ○ ○ ○ ▸

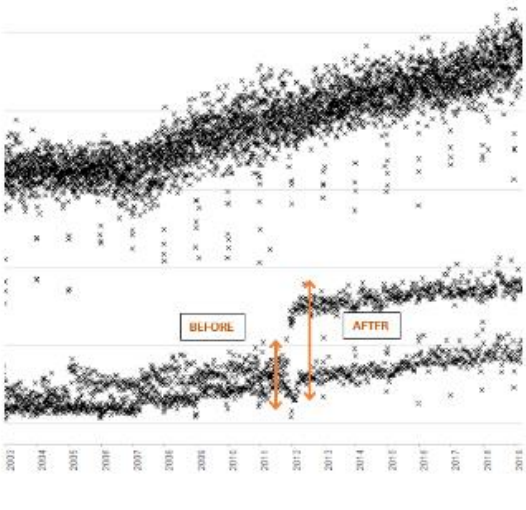
Sudden change in (vertical) variability

Sudden changes in **variability** are where the "noisiness" or "spread" of the data changes suddenly, in the vertical (up and down) direction.

The change in spread could be across the *whole data range*:



Or it could be a change in spread *within* the data range:



Continue

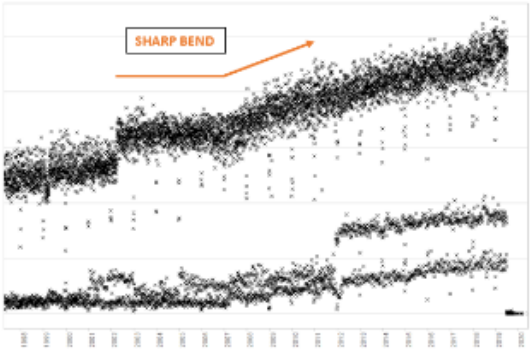
◂ ○ ○ ○ ○ ● ○ ○ ○ ○ ○ ▸

Sudden change in trend

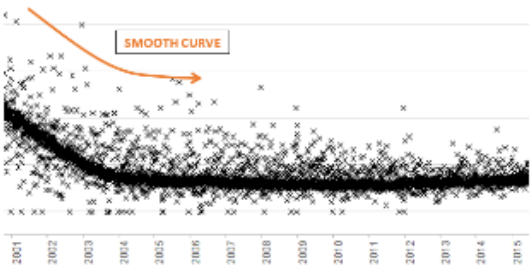
A sudden change in **trend** is where the data changes direction abruptly, like the sharp bend of an elbow or a door hinge. Remember, ***smooth changes should be ignored.***

(It is not always clear whether a trend change is sudden or not, so you can choose to mark the change with a yellow line if you are unsure.)

For example, here is a sudden trend change:



Whereas the smooth change below should ***not*** be marked:



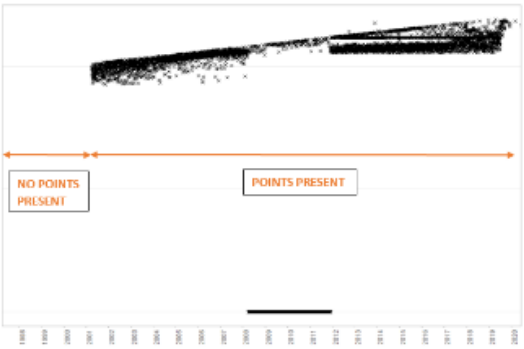
Continue

◂ ○ ○ ○ ○ ○ ● ○ ○ ○ ○ ▸

Sudden change in (horizontal) presence/absence

A sudden change in **presence/absence** is where, when looking horizontally (left and right), there are suddenly no points at all for the time period (or almost no points).

For example:



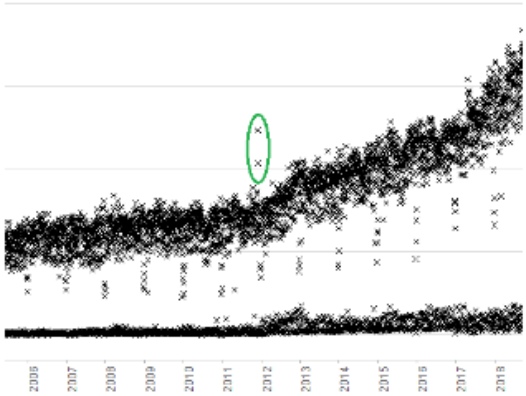
Continue

Single or small cluster of outliers

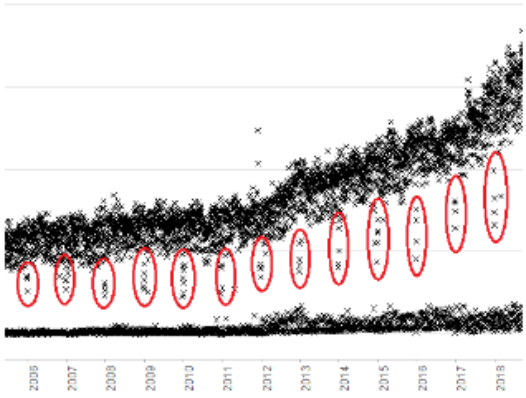
Outliers are a single point or a small cluster of points which look like they don't belong.

However, you should ***ignore outliers that appear to repeat regularly and predictably***.

For example, here is one outlier that doesn't follow a regular pattern, so this one should be marked:



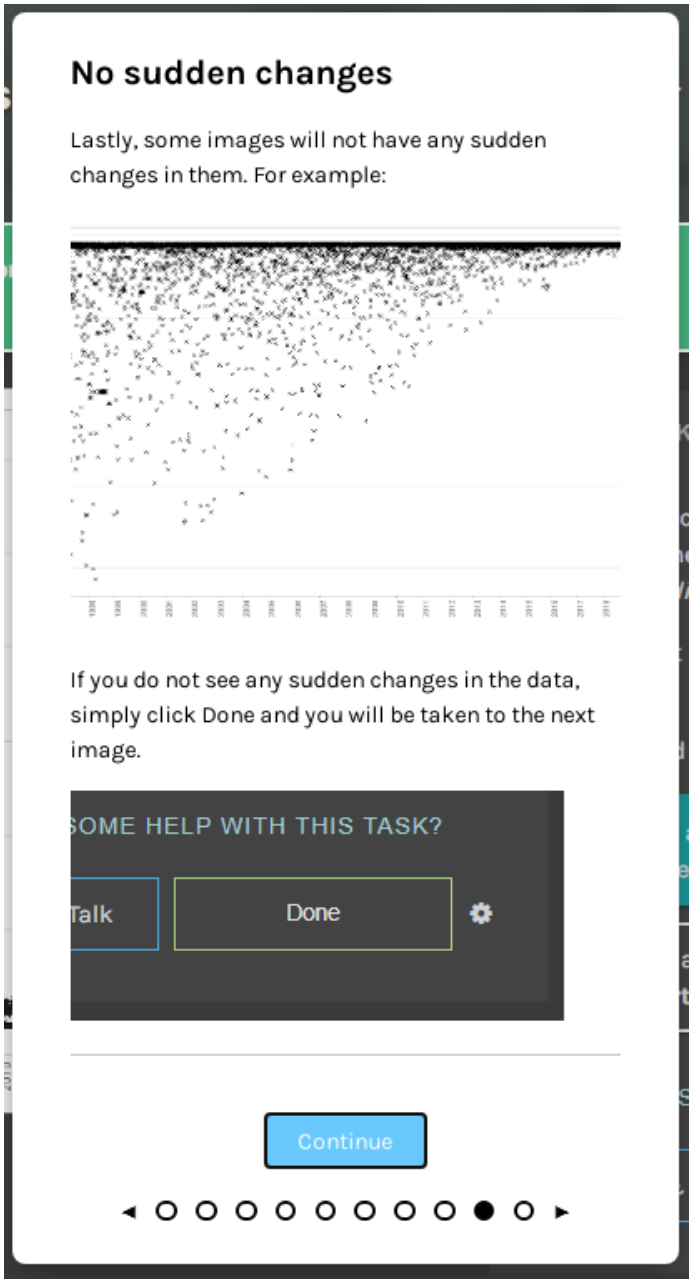
Whereas the ones that appear every year around December should be ignored since they are predictable:

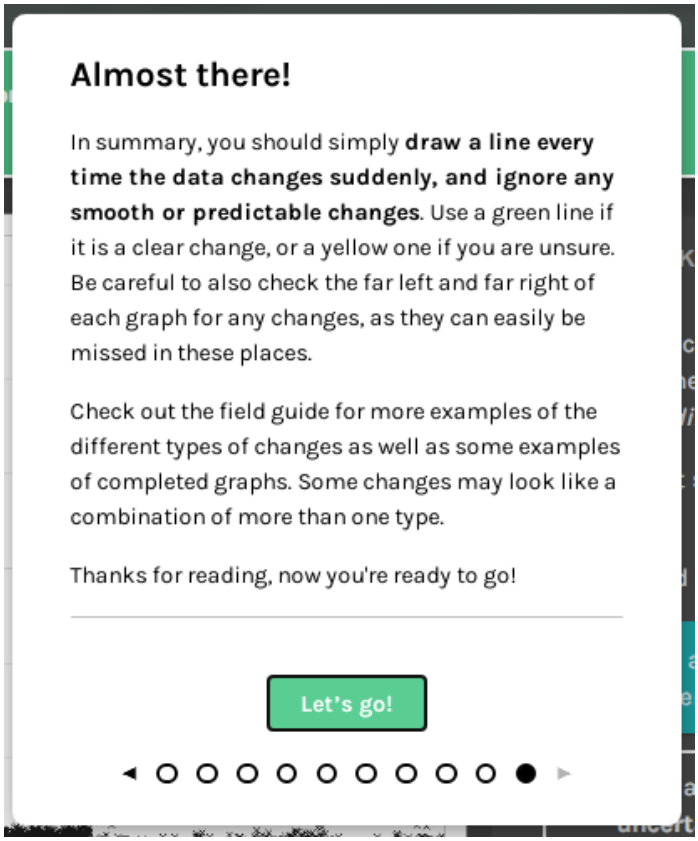


Outliers should be marked with a line just like other changes.

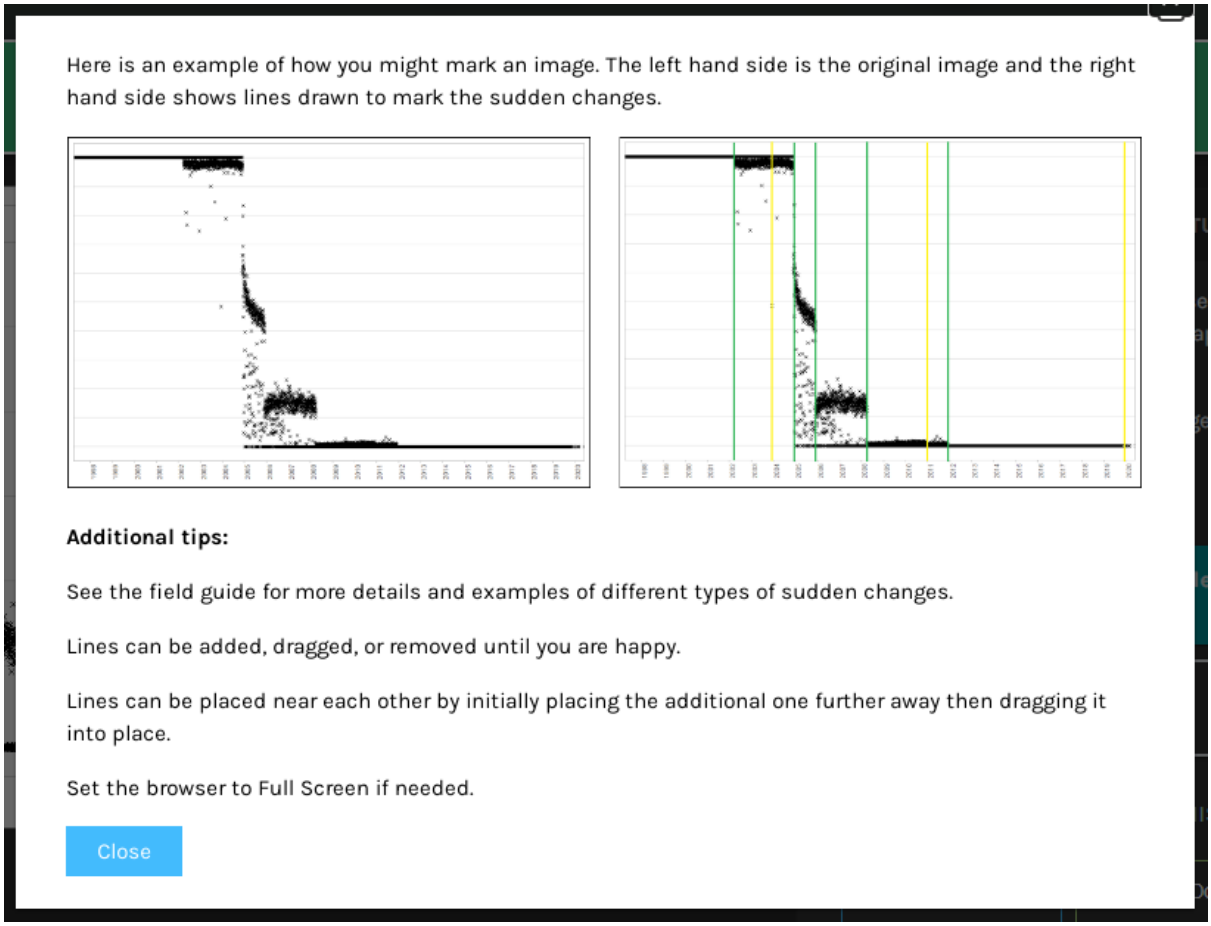
Continue

◄ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ►


No sudden changes
Lastly, some images will not have any sudden changes in them. For example:
If you do not see any sudden changes in the data, simply click Done and you will be taken to the next image.
SOME HELP WITH THIS TASK?
Talk
Done
Continue


Almost there!
In summary, you should simply draw a line every time the data changes suddenly, and ignore any smooth or predictable changes. Use a green line if it is a clear change, or a yellow one if you are unsure. Be careful to also check the far left and far right of each graph for any changes, as they can easily be missed in these places.
Check out the field guide for more examples of the different types of changes as well as some examples of completed graphs. Some changes may look like a combination of more than one type.
Thanks for reading, now you're ready to go!
Let's go!

Help page



B.3 Sample size calculation

The number of cases required (q_β) was calculated using the following process as described by Flahault et al.[81], using the parameter values $\pi = 0.9$, α = 0.025, γ = 0.7:

Let x be an observation from the binomial distribution with parameters n and π. An estimate $\hat{\pi}_L$ of the 1 – α lower confidence limit for π is the solution to

$$\sum_{i=x}^{n} \binom{n}{i} \hat{\pi}_L^i (1-\hat{\pi}_L)^i = \alpha.$$

Because $\hat{\pi}_L$ is increasing in x, the α-quantile, γ, of the distribution of $\hat{\pi}_L$ is the solution to

$$\sum_{i = q_\beta}^{n} \binom{n}{i} \gamma^i (1-\gamma)^i = \alpha,$$

Where q_β is a β-quantile of the binomial distribution with parameters *n* and *π*.

To obtain the size of the sample (*n*) required from the population in order to get the correct number of cases, the smallest *n* was calculated such that

$$\sum_{x = N_{cases}}^{n} \binom{n}{x} Prev^x (1-Prev)^{n-x} \geq 0.95,$$

with N_{cases} = q_β from above, and *Prev* (prevalence) = 0.15.

Appendix C: Chapter 5

C.1 Change point detection packages considered

Package Name and Title	Type(s) of change	Data model(s)	Segmentation method(s)	Online/ Offline
bfast - Breaks For Additive Season and Trend (BFAST)	Trend (continuous & discontinuous) Seasonality	Piecewise linear + piecewise seasonality + Gaussian errors	Iterative decomposition + Bai & Perron (2003)	Both
BreakPoints - Identify Breakpoints in Series of Data	Level	Piecewise constant, Gaussian, gamma, nonparametric	Unnamed	Offline
bwd - Backward Procedure for Change-Point Detection	Level	Gaussian, nonparametric	BWD	Offline
changepoint - Methods for Changepoint Detection	Level Variability	Gaussian, plus Poisson, gamma for meanvar	PELT (Pruned Exact Linear Time)/Binary Segmentation/Segment Neighbourhood	Offline
changepoint.np - Methods for Nonparametric Changepoint Detection	Level Variability	Nonparametric	ED-PELT	Offline
changepointsVar - Change-Points Detections for Changes in Variance	Variability	Gaussian	Based on cumSeg	Offline

Package Name and Title	Type(s) of change	Data model(s)	Segmentation method(s)	Online/ Offline
cpm - Sequential and Batch Change Detection Using Parametric and Nonparametric Methods	Level Variability General distribution changes	iid Gaussian, Bernoulli, glr, exponential, nonparametric	Sequential monitoring. Have to set a startup period (recommended >=20) where no CPs will be detected, and believe this applies after each CP is found	Online
cpss - Change-Point Detection by Sample-Splitting Methods	Level Variability	Gaussian, glm, binomial, multinomial, Poisson, exponential, geometric, Dirichlet, gamma, beta, chsq, invgauss	PELT/WBS/SN/BS+	Offline
cumSeg - Change Point Detection in Genomic Sequences	Level	Nonparametric (only assumes zero-mean noise)	Cusum + lars	Offline
DeCAFS - Detecting Changes in Autocorrelated and Fluctuating Signals	Level	Random walk + AR(1)	Extension of Fpop	Offline
ecp - Non-Parametric Multiple Change-Point Analysis of Multivariate Data	Any distributional change	Nonparametric	Agglomerative, divisive, dynamic programming and pruning	Offline

Package Name and Title	Type(s) of change	Data model(s)	Segmentation method(s)	Online/ Offline
FDRSeg - FDR-Control in Multiscale Change-Point Segmentation	Level	Gaussian	FDRSeg (FDR = false discovery rate)	Offline
ffstream - Forgetting Factor Methods for Change Detection in Streaming Data	Level	Nonparametric	Sequential monitoring. Have to set a burn-in period (default is 50) where no CPs will be detected, and this applies after each CP is found	Offline
fpop - Segmentation using Optimal Partitioning and Function Pruning	Level	Unclear, mentions exponential family	fpop	Offline
GNSSseg - Homogenization of GNSS Series	Level	Decomposition, gaussian errors	DP+	Offline
IDetect - Isolate-Detect Methodology for Multiple Change-Point Detection	Level Trend (continuous)	Gaussian, heavy tail	Isolate-Detect	Offline
jointseg - Joint Segmentation of Multivariate (Copy Number) Signals	Level	Gaussian?	RBS/GFLars/DP then pruned	Offline
mosum - Moving Sum Based Procedures for Changes in the Mean	Level	Nonparametric	Control chart	Offline

Package Name and Title	Type(s) of change	Data model(s)	Segmentation method(s)	Online/ Offline
ocp - Bayesian Online Changepoint Detection	Level?	Gaussian, Poisson	Iterative run length	Online
offlineChange - Detect Multiple Change Points from Time Series	Level	Piecewise AR	Multiwindow	Offline
otsad - Online Time Series Anomaly Detectors	Abrupt level (including outliers)	Nonparametric	Prediction-based/window-based	Online
stepR - Multiscale Change-Point Inference	Level	Gaussian+	SMUCE/H-SMUCE	Offline
strucchangeRcpp - Testing, Monitoring, and Dating Structural Changes: C++ Version	Level Trend	Linear regression, iid piecewise Gaussian errors	Bai & Perron (2003)	Offline
trendsegmentR - Linear Trend Segmentation and Point Anomaly Detection	Trend (continuous and discontinuous) Outlier	Piecewise linear, iid Gaussian errors	4-step wavelet-transform based	Offline
TSMCP - Fast Two Stage Multiple Change Point Detection	AR model (piecewise stationary)	AR	SMSBE-VSAR (simultaneous multiple structural break estimation and variable selection)	Offline

C.2 Best performing parameters by aggregation method, based on the tuning set

C.2.1 Best performing parameters for each automated method by aggregation granularity, based on the raw values in the tuning set

Granularity	MCC	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value	Parameter value
Cpt.meanvar						
Day	0.288	0.490	0.999	0.170	1.000	90
Week	0.447	0.437	0.999	0.459	0.999	91
Month	0.442	0.458	0.995	0.435	0.996	28
Cpt.var						
Day	0.266	0.337	1.000	0.210	1.000	20
Week	0.327	0.366	0.998	0.295	0.999	7
Month	0.337	0.385	0.993	0.306	0.995	2
Cpt.np						
Day	0.460	0.523	1.000	0.404	1.000	16
Week	0.476	0.604	0.998	0.377	0.999	7
Month	0.390	0.518	0.991	0.303	0.996	3
DeCAFS						
Day	0.048	0.765	0.930	0.003	1.000	5
Week	0.109	0.957	0.879	0.014	1.000	1
Month	0.220	0.863	0.901	0.064	0.999	2
OcpTsSdEwma						
Day	0.291	0.167	1.000	0.505	1.000	threshold = 0.05, m = 9, l = 7
Week	0.510	0.463	0.999	0.562	0.999	threshold = 0.07, m = 7, l = 3
Month	0.545	0.455	0.998	0.661	0.996	threshold = 0.15, m = 6, l = 3

Granularity	MCC	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value	Parameter value
OcpPewma						
Day	0.143	0.274	0.999	0.075	1.000	alpha0 = 0.9, beta = 0, l = 11
Week	0.272	0.250	0.999	0.298	0.999	alpha0 = 0.9, beta = 0, l = 12
Month	0.469	0.474	0.996	0.473	0.996	alpha0 = 0.8, beta = 0, l = 4

C.2.2 Best performing parameters and transformations for each automated method by aggregation function, based on the tuning set

Aggregation function	No. of time series	No. of crowd-sourced change points	MCC	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value	Method	Transformation	Parameter value
maxlength	30	85	0.802	0.953	1.000	0.675	1.000	cpt.meanvar	None	41
nonconformant_n	97	47	0.771	0.702	1.000	0.846	1.000	cpt.np	None	3
nonzero_perc	12	4	0.750	0.750	1.000	0.750	1.000	OcpTsSdEwma	None	threshold = 0.01, m = 6, l = 6
sum	11	30	0.708	0.633	1.000	0.792	1.000	OcpPewma	None	alpha0 = 0.9, beta = 0, l = 7
meanlength	31	165	0.656	0.588	0.999	0.735	0.999	DeCAFS	None	20
missing_perc	379	682	0.580	0.526	1.000	0.640	1.000	OcpTsSdEwma	None	threshold = 0.15, m = 7, l = 3
midnight_perc	34	71	0.570	0.423	1.000	0.769	0.999	OcpTsSdEwma	None	threshold = 0.02, m = 7, l = 4

Aggregation function	No. of time series	No. of crowd-sourced change points	MCC	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value	Method	Transformation	Parameter value
midnight_n	48	174	0.557	0.511	0.999	0.610	0.999	OcpTsSdEwma	None	threshold = 0.15, m = 7, l = 2
distinct	189	675	0.535	0.566	0.999	0.508	0.999	cpt.np	None	5
subcat_perc	466	687	0.534	0.457	1.000	0.625	1.000	OcpTsSdEwma	None	threshold = 0.15, m = 7, l = 4
minlength	31	162	0.503	0.457	0.999	0.556	0.999	cpt.meanvar	None	100
subcat_n	457	700	0.500	0.454	1.000	0.551	1.000	OcpTsSdEwma	None	threshold = 0.07, m = 7, l = 3
mean	13	45	0.487	0.333	1.000	0.714	0.999	cpt.meanvar	First-difference	30
n	505	1647	0.485	0.704	0.999	0.335	1.000	cpt.meanvar	None	10
missing_n	358	1104	0.483	0.332	1.000	0.702	0.999	OcpTsSdEwma	None	threshold = 0.13, m = 7, l = 3
nonconformant_perc	96	6	0.445	0.833	1.000	0.238	1.000	cpt.var	None	18
median	13	48	0.385	0.208	1.000	0.714	0.998	OcpTsSdEwma	None	threshold = 0.11, m = 10, l = 4

Aggregation function	No. of time series	No. of crowd-sourced change points	MCC	Sensitivity	Specificity	Positive Predictive Value	Negative Predictive Value	Method	Transformation	Parameter value
max	66	174	0.345	0.724	0.995	0.166	1.000	cpt.np	First-difference	2
min	54	101	0.273	0.238	1.000	0.316	0.999	OcpPewma	None	alpha0 = 0.9, beta = 0, l = 15

missing_n = number of missing values, missing_perc = percentage of missing values, nonconformant_n = number of non-conformant values, nonconformant_perc = percentage of non-conformant values, sum = sum of duplicate records removed, nonzero_perc = percentage of records which had been duplicated, n = number of non-missing values, min = minimum value, max = maximum value, mean = mean value, median = median value, midnight_n = number of values with no time element, midnight_perc = percentage of values with no time element, minlength = minimum string length, maxlength = maximum string length, meanlength = mean string length, distinct = number of distinct values, subcat_n = number of non-missing values within each subcategory, subcat_perc = percentage of non-missing values within each subcategory