City, University of London

MSc in Data Science

Appendices to Project Report

2019

Appendices for Modelling Depression Recurrence Through Analysis of Electronic Health Records

Lesley Dwyer

Supervised by: Dr Cagatay Turkay

1st October 2019

Table of Contents

A. Appendix A – Project Proposal

INM373 Research Methods and Professional Issues: Research Project Proposal
Modelling Depression Recurrence through Analysis of Electronic Health Records

1 Introduction

The availability of electronic health records (EHRs) combined with machine learning can provide a means to better understanding illness, including depression. For patients diagnosed with depression, some will experience recurring episodes, or relapses, and others will not. I am proposing a project to investigate how machine learning can be used with anonymised EHRs to determine the likelihood that a patient will experience a recurrence of depression within a given time frame following remission of the illness.

1.1 Motivation

Previous research has used machine learning techniques to understand or predict mental illness. Machine learning has also been used to predict health outcomes using EHRs.
In one example, using anonymised EHRs, Hayes *et al*. (2012) examined associations between mortality and other factors, such as activities of daily living (ADL) impairment, social relationships, living conditions and occupational/recreational activities, for people with serious mental illness. Using a Cox regression model, they found that impairment of ADL was associated with increased mortality.
In another study, Pham *et al*. (2016) used EHRs and Long Short-Term Memory (LSTM), a recurrent neural network, to predict health trajectories. They assessed their model on a diabetes cohort and a mental health cohort.
Deep learning (DL) and LSTM were also discussed recently in Molecular Psychiatry in relation to EHRs and depression. Durstewitz *et al*. (2019) state that 'DL architectures—may prove particularly powerful in areas in which we already possess large and ever-growing data sets such as electronic health records (EHRs).' Also mentioned is a study using data from smartphones in which Suhara *et al*. (2017, as cited in Durstewitz *et al*., 2019) 'forecast severe depressive states based on individual histories of mood, behavioural logs, and sleep information using a LSTM architecture.'

The data captured in EHRs is sensitive, personal and protected, and it can therefore be difficult to access for research purposes. However, as mentioned by Perera *et al*. (2016), the South London and Maudsley NHS Foundation Trust (SLaM), a large mental healthcare provider in London, has developed the Clinical Records Interactive Search (CRIS) system designed for mental health research. It contains data from over 250,000 anonymised patient records from the trust's clinical systems capturing routine mental healthcare.
The **scope** of this project is to use machine learning to determine if a patient will experience a relapse of depression. I plan to utilise the CRIS data focusing only on patients diagnosed with depression. EHR data includes a series of patient events, so it can be viewed as a sequence prediction problem. After understanding and analysing the data, a sequential modelling algorithm will be selected, and I will build a model that produces results for patients unseen by the model. Variations of this model are also worth investigating, such as models for different patient groups or models based on limiting the data, if they prove to increase the model performance.

1.2 Research Question, Objectives and Beneficiaries

I am posing the following **research question** for this project:

- To what extent can machine learning be used with anonymised electronic health records to determine the likelihood a patient will experience a relapse/recurrence of depression within a given time frame following remission?

The **research objectives** to answer this question during the project are:

- To research sequence modelling algorithms and related literature
- To define attributes and thresholds from the source data to be used for analysis
- To cleanse and prepare the data for further analysis and use in modelling
- To perform exploratory data analysis to define hypotheses and data for modelling
- To build a machine learning model that determines if a patient will experience a relapse of depression
- To investigate whether building separate machine learning models for different patient groups will improve model performance
- To investigate whether placing limits on the data included in the model will improve model performance

The **outputs** of this project will be:

- A machine learning model that, for a new/unseen patient, classes that patient as ‘depression relapse’ or ‘no depression relapse’ within a defined timeframe. This may be in the form of multiple models if better model performance is achieved by building separate models for different patient groups.
- A final report including a literature review, data requirements, results of all analyses and any other project details

The anticipated **beneficiaries** of this work are clinicians working with patients who have been diagnosed with depression. I will be using anonymised data, but a modified version of this work using identifiers to identify the patient, could provide clinical decision support. It could indicate to a clinician whether a patient is more likely to experience another episode of depression in a given timeframe, e.g. in the next 12 months.

2 Critical Context

In addition to the research mentioned above, there have been several other studies using machine learning to investigate mental illness, and more specifically depression and depression relapse.

In 2014, Wang *et al*. did a study on predicting recurrence of major depression using data from the U.S. National Epidemiological Survey on Alcohol and Related Conditions (NESARC). The surveys were completed by patients who had Major Depressive Disorder (MDD) and had been in remission for at least two months. The researchers used a logistic regression model to predict the recurrence of depression.

A later study by Kessler *et al*. (2016) used machine learning to predict severity and persistence of MDD from baseline self-reports over a period of 10-12 years. The machine learning models used in this case were ensemble regression trees and penalized regression.

Two additional studies from Nie *et al*. (2016) and Sakurai *et al*. (2017) investigated prediction of depression relapse using the Sequenced Treatment Alternatives to Relieve Depression (STAR*D) clinical trial data. Nie *et al*. (2016) used a gradient boosting algorithm and a stochastic dual coordinate ascent algorithm, while Sakurai *et al*. (2017) used a Cox proportional hazard model.

In 2018, Dinga *et al*. used a penalized logistic regression machine learning model to predict the naturalistic course of depression. They incorporated 81 attributes from clinical, psychological, and biological data obtained from the Netherlands Study of Depression and Anxiety (NESDA). Patients with MDD or dysthymia were measured at the start of the study and again two years later, but no interventions were applied. The researchers were able to predict remission with 62% accuracy, and they found symptoms to be the most predictive factors.

All of these approaches used surveys or clinical trial data and not routine EHRs. Hayes *et al*. (2012) and Pham *et al*. (2016), as described in the previous section, considered machine learning approaches on EHRs, but not specifically on depression or depression relapse.

The research I plan to conduct would build on these studies. I would like to investigate the use of machine learning to predict depression relapse from anonymised EHRs. In addition, I want to consider different groups of patients and limiting the data as further modelling options.

3 Approaches

The methods I will use will be quantitative. At a high level, I will go through four phases: a research and setup phase, a define and analyse phase, a build and evaluate phase and a report phase. During these phases, I will conduct research on algorithms, complete a literature review, define data requirements, collect the data, prepare and analyse the data, build and evaluate machine learning models and produce my final report. I describe the details further in this section. I have also included a sub-section on ethical considerations for the project.

3.1 Research and Setup Phase

3.1.1 Literature Review focused on EHRs and Sequence Prediction

I plan to conduct a literature review at the start of my project to build on the literature I have found so far. The purpose will be to continue looking for examples of machine learning, sequential modelling or deep learning used to predict or understand depression patterns or relapse based on EHRs. I am also interested in how sequential prediction approaches have been used in healthcare, how the data was prepared, which algorithms were used and what were the outcomes. In addition to gaining a more complete understanding of the background and landscape of this research area, I want to ensure that my project is addressing a valid gap in the research. Suitable literature sources will be included in my final dissertation to provide context for my project.

3.1.2 Research and Prototype Algorithms

I will continue researching sequence prediction and behaviour prediction algorithms at the beginning of the project. Once a short list has been identified, I will build quick prototypes of these in Python and R to understand which libraries are used and any challenges to be aware of.

3.1.3 Setup System and Network Access

One challenge is that the CRIS data must be kept within the SLaM firewall, so I plan to access the systems and data onsite from a hot desk and remotely using a virtual private network (VPN). I will need to arrange for access to the SLaM network, the CRIS SQL database and the VPN. I plan to apply for this at the beginning of the project allowing plenty of time for this to be setup.

3.2 Define and Analyse Phase

3.2.1 Test Prototype Algorithms

One risk is that the computational power of the machines I'll be using, i.e. hot desks or virtual machines, won't be suitable for the algorithm chosen, especially if it's a deep learning algorithm. So, as part of this phase, I will train some sample data with the short list of algorithms to get an estimate of timings. This may mean some algorithms are ruled out.

3.2.2 Define CRIS Data Requirements

The CRIS data model is large and complex. It is also secondary-use data originating from EHRs not originally designed for research purposes. This presents some challenges with respect to this project.

During initial discussions with the CRIS team, one challenge identified was how to define remission and relapse. An idea proposed by the CRIS Academic Lead is to look at 'crisis events' to determine whether someone has relapsed. Using this approach, crisis and non-crisis events would need to be defined possibly based on the type of event. Another possibility is to look at symptoms or a symptom profile to determine if someone is in remission or experiencing a relapse. In addition, what timeframes are suitable for remission and relapse and how can the data be used to define this? How much event history is needed to produce enough sequences of remission and relapse to identify patterns in the data? Another challenge is that not everything entered into the system will have a date associated with it, so it may be difficult to associate some information with a timeframe.

I plan to liaise with the CRIS team early in the project to fully define these data requirements. This will likely involve a few meetings and some exploratory analysis and may be an iterative process to ensure suitable attributes and thresholds are used. I will also discuss approaches for the other modelling option planned for later in the project. This includes understanding which attributes could be useful for building models specific to different patient groups and how to limit the patient data for the model.

Although this may change, I envisage using the following CRIS attributes to define groups of patients for analysis: diagnosis, diagnosis dates, and admission or treatment start dates. I envisage using these attributes for analysis and to build the machine learning model(s): age, gender, ethnic group, risk assessment, level of face-to-face contact with clinical services, diagnosis, Health of the National Outcome Scale (HoNOS) scores, medication profile, depressive symptom profile, and corresponding dates for the variable types mentioned.

3.2.3 Collect CRIS Data

There are two options for collecting the CRIS data. One is to collect the data directly from the CRIS SQL database, and the other is to liaise with the CRIS team to define requirements and they will produce data extracts. The CRIS team supports other researchers, so an advantage to collecting my own data is that I would not be waiting behind other users' requests. I would also be able to adjust and re-run extracts quickly if I am managing this myself. However, the data model is complex, so I would need to spend more time up front learning it. The CRIS team is available for questions and they have provided a data dictionary. In addition, they sometimes offer training on how to use CRIS data with SQL, which I plan to attend if it is offered during the early weeks of my project. My plan is to collect the data myself, but if this becomes too cumbersome, I will request extracts from the CRIS team as early as possible.

3.2.4 Cleanse and Prepare Data

Once I have the data, I plan to cleanse and prepare it using Python and SQL. This will involve standard tasks like imputing missing values, but also things more specific to the project such as creating new features to represent remission and relapse and determining which patients should and shouldn't be included in further analysis. They may be excluded if they don't have enough data or perhaps they cannot be considered to be in remission based on the types of events they have.

During my initial meetings with the CRIS team, I will discuss data quality and availability as part of the data requirements. Where an attribute holds valuable information but is not well populated, I will work with them to find a valid alternative if there is one.

3.2.5 Exploratory Data Analysis

Once I have a clean dataset, I will do some exploratory data analysis in Python. The goal will be to gain a deeper understanding of the data, form initial hypotheses about how to model the data, and identify the data to be used in the model. Some initial questions I have of the data are mentioned below.

For instance, are there common patterns in the data when patients move between states of remission and relapse? Is it possible to identify two distinct groups of patients, one that stays in remission for longer periods of time and one that does not, in order to form two classes for training the model? If so, how do different features relate to relapse and remission for these groups?

This stage may be iterative in nature requiring me to collect and cleanse additional data.

3.2.6 Determine Sequential Modelling Algorithm

I have given some consideration to the types of machine learning algorithms that would be suitable for this type of problem. The research mentioned above used logistic regression, Cox regression, ensemble regression trees or penalized regression. Other techniques used were gradient boosting, stochastic dual coordinate ascent, and Cox proportional hazard. An algorithm I'm considering is LSTM used also by Pham *et al*. (2016) to model EHR data. Another algorithm I'm considering is a Hidden Markov Model (HMM). According to Elliot *et al*. (1995, cited in Bishop, 2006, p. 610), HMMs can be used for sequential data when the hidden variables are discrete.

Following the algorithm research, literature review, data analysis and further discussion with my supervisor, I will decide which algorithm to use for modelling.

3.3 Build and Evaluate Phase

3.3.1 Build, Train and Evaluate Model

First, I will shape the data in Python to get it in an appropriate format for the model. Then, I will build the machine learning model, and I may be able to extend one of the prototype models built earlier. I will define my training and evaluation methodology, and I will perform hyperparameter tuning to optimise the model.

My assumption at this stage is that it will be a sequence classification problem. For instance, based on being trained on a historical sequence of patient events, the model will output 'depression relapse' or 'no depression relapse' in the defined timeframe. Therefore, suitable evaluation methods would include classification accuracy, Area under the Receiver Operating Characteristic Curve (AUC), F-measure, precision, recall (sensitivity), specificity, and a confusion matrix.

3.3.2 Investigate Additional Model Options

Following the initial model build, I will consider two additional options for modelling the data to see if this can improve the model's performance. First, I will investigate building separate models for different patient groups, e.g. by age group. Second, I will investigate limiting the data in some way, e.g. only considering patients with a certain number of events. These models will be evaluated and compared to the one created previously.

3.4 Report Phase

3.4.1 Produce Draft Report

I plan to start drafting my report early in the project once I've completed the literature review. I want to ensure I capture as much detail as I go, so that nothing is left out when compiling the final report. I also assume there will be some downtime during the project while models are being trained or while waiting for feedback on questions, and during this time I plan to make progress on the report.

3.4.2 Finalise Report

After all the models have been built and evaluated, I will update the draft report with details about the models. I will also complete any other sections that need to be finalised. I plan to finish the report a few days early to leave some time at the end for contingency.

3.5 Ethics Considerations

Access to the CRIS system follows a strict security policy, including identification of someone to act as my CRIS Supervisor/Line Manager and approval by the CRIS Oversight Committee. I have made arrangements for a CRIS Supervisor. I have also obtained CRIS system approval and will abide by the security procedures in order to use the CRIS data. Ethics is considered as part of the Oversight Committee approval. Further details on this are described below and how it relates to the Ethics Review Form.

The Research Ethics page (City, University of London Department of Computer Science, 2019) states that 'Human participation covers direct data collection from people' including 'using personal identifiable data from individual records.'

Section A.1 questions 1.1 and 1.2 of the Ethics Review Form can be answered 'NO', as the CRIS system has already been given Ethics approval by the National Research Ethics Service (NRES) as mentioned on page 6 of their security model (South London and Maudsley NHS Foundation Trust, 2013). I have also confirmed this in writing with the CRIS Academic Lead. Sections A.2, A.3 and A.4 of the Ethics Review Form can be answered 'NO' because there is no direct human participation. Again, as stated on page 6 of their security model, CRIS is 'an anonymised data source for secondary analysis' (South London and Maudsley NHS Foundation Trust, 2013).

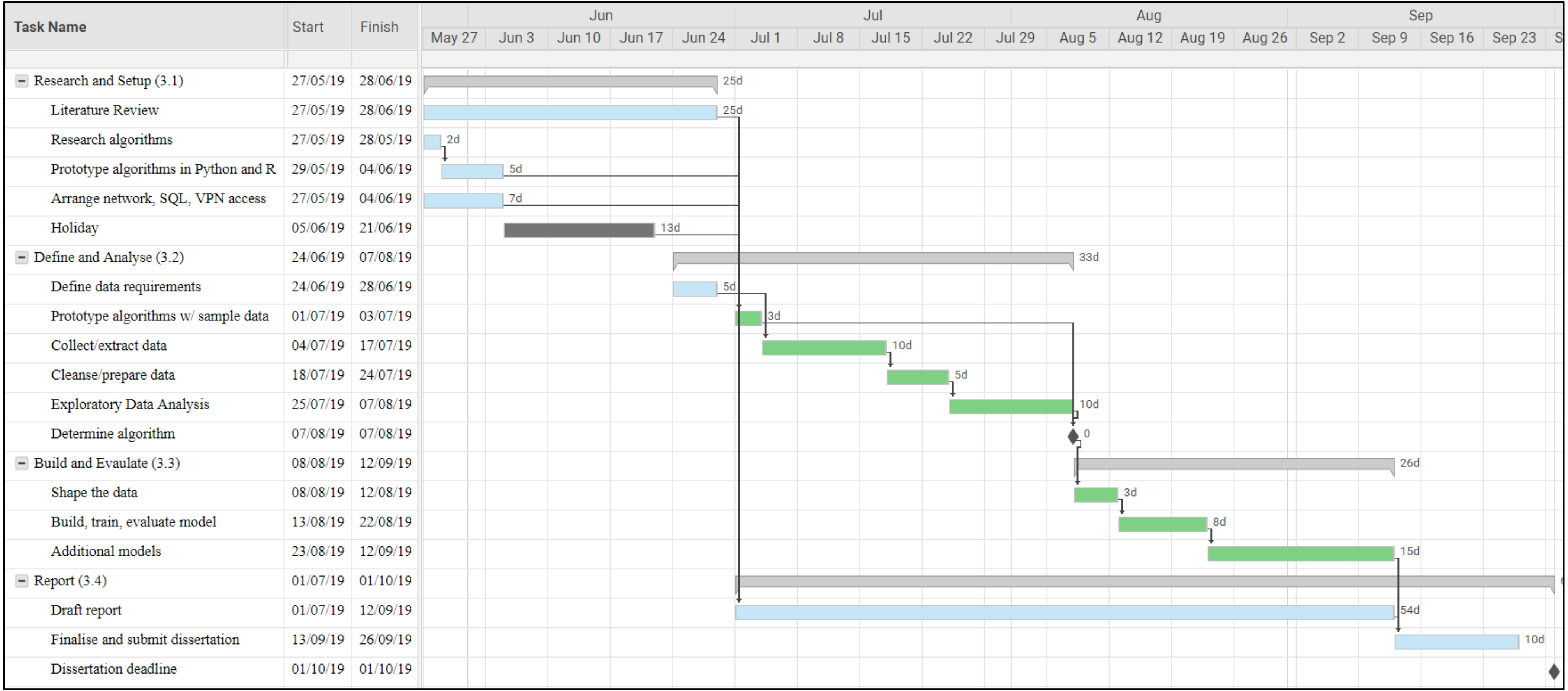
The City, University of London Department of Computer Science (2019) Ethics page states that 'Secondary analysis of anonymised data that is published or otherwise disseminated does not require ethical approval. Analysis of routinely collected anonymised data that does not contain personal information and through which individuals cannot be identified does not need approval either, as long as permission has been given by the data owner.'

Based on this information, my project will not require ethical approval.

I have also considered the sensitive nature of the subject matter of this project and my lack of clinical knowledge in this area. I will be liaising with the CRIS Academic Lead for questions requiring clinical expertise, and I will be transparent about assumptions made, potential biases and limitations of the study in my final report.

4 Work Plan

Below is a high-level work plan of the proposed research project. The iterative nature of the exploratory data analysis may cause additional data collection and cleansing, which is not reflected on the plan but is included in the time estimates. Meetings with my supervisor are not arranged yet, so they are not shown.



Task Name	Start	Finish
Research and Setup (3.1)	27/05/19	28/06/19
Literature Review	27/05/19	28/06/19
Research algorithms	27/05/19	28/05/19
Prototype algorithms in Python and R	29/05/19	04/06/19
Arrange network, SQL, VPN access	27/05/19	04/06/19
Holiday	05/06/19	21/06/19
Define and Analyse (3.2)	24/06/19	07/08/19
Define data requirements	24/06/19	28/06/19
Prototype algorithms w/ sample data	01/07/19	03/07/19
Collect/extract data	04/07/19	17/07/19
Cleanse/prepare data	18/07/19	24/07/19
Exploratory Data Analysis	25/07/19	07/08/19
Determine algorithm	07/08/19	07/08/19
Build and Evaulate (3.3)	08/08/19	12/09/19
Shape the data	08/08/19	12/08/19
Build, train, evaluate model	13/08/19	22/08/19
Additional models	23/08/19	12/09/19
Report (3.4)	01/07/19	01/10/19
Draft report	01/07/19	12/09/19
Finalise and submit dissertation	13/09/19	26/09/19
Dissertation deadline	01/10/19	01/10/19

🟩 Work done on SLaM network onsite (hot desk) or via VPN
🟦 Work not requiring SLaM network
⬛ Holiday
◆ Milestone

5 Risks

Below are the risks for this project.

Risk	Likelihood	Consequence	Impact	Mitigation
CRIS data model is difficult to understand	High – 3	Medium – 3	9	Work with the CRIS team to understand the data. If offered, attend the training on how to use CRIS via SQL. If it’s too complicated, get extracts built from the CRIS team instead of building them myself.
Difficulty meeting with CRIS Training Lead due to her supporting other MSc projects	Medium - 2	High - 4	8	Arrange meetings as early as possible. Come prepared to meetings with agenda and questions. Look for alternative resource(s) who can help with questions if needed.
Delays or difficulty accessing CRIS system or data; data must be accessed from within firewall (onsite or via VPN)	Medium - 2	High – 4	8	Project using CRIS data is already approved by the CRIS Oversight Committee. Apply early for NHS research passport (in progress). Arrange early for VPN access to work remotely. Define data requirements and extracts (if using) as early as possible.
Delays in accessing software	Medium - 2	High – 4	8	Python can be installed by me directly on hot desks/virtual machine. If using SQL database, apply for access early.
Difficulty meeting with CRIS Supervisor due to his busy schedule	Medium - 2	Medium - 3	6	Arrange meetings as early as possible. Come prepared to meetings with agenda and questions. Look for alternative resource(s) who can help with questions if needed.
Data quality is poor or not well-populated	Medium – 2	Medium – 3	6	Include time in the plan for data cleansing and preparation. Discuss data requirements with CRIS team early to identify best attributes to use.
Preferred software is not capable of handling (or is too onerous for) preferred methods	Medium – 2	Medium – 3	6	Research early on which algorithms are most suitable and if/how they can be implemented in Python. Also investigate alternative software, e.g. R, that could be used if needed.
Computational power of hot desks/virtual machines is lower than expected causing training of model to be time-consuming	Medium – 2	Medium – 3	6	Once the initial data set is produced, train sample models to get time estimates. Allow sufficient time in plan for training final model.
The model does not perform well	Medium - 2	Medium – 3	6	Try different subsets of data and different cohorts to see if results can be improved.
CRIS Supervisor no longer able to act in that capacity or leaves organisation	Low - 1	Very High - 5	5	Agree contingency plan, i.e. alternative supervisor, if he leaves or becomes unavailable to assist me.
CRIS Training Lead leaves organisation	Low - 1	High - 4	4	Agree contingency plan, i.e. alternative resource, if she leaves or becomes unavailable to assist me.
System failure, loss of work	Low – 1	High – 4	4	Backup work regularly.

6 References

Bishop, C. (2006) *Pattern Recognition and Machine Learning.* New York: Springer Science + Business Media, LLC.

City, University of London Department of Computer Science (2019) *Research Ethics*. Available at: https://www.city.ac.uk/department-computer-science/research-ethics. (Accessed: 26 March 2019).

Dinga, R., Marquand, A., Veltman, D., Beekman, A., Schoevers, R., van Hemert, A., Penninx, B., Schmaal, L. (2018) 'Predicting the naturalistic course of depression from a wide range of clinical, psychological, and biological data: a machine learning approach', *Translational Psychiatry*, 8. doi: 10.1038/s41398-018-0289-1.

Durstewitz, D., Koppe, G., Meyer-Lindenberg, A. (2019) 'Deep neural networks in psychiatry', *Molecular Psychiatry*, ISSN 1476-5578, doi: 10.1038/s41380-019-0365-9.

Hayes, R., Chang C., Fernandes, A., Begum, A., To, D., Broadbent, M., Hotopf, M., Stewart, R. (2012) 'Functional Status and All-Cause Mortality in Serious Mental Illness', *PLoS ONE* 7(9): e44613. Available at: https://doi.org/10.1371/journal.pone.0044613. (Accessed: 6 March 2019).

Kessler, R., van Loo, H., Wardenaar, K., Bossarte, R., Brenner, L., Cai, T., Ebert, D., Hwang, I., Li, J., de Jonge, P., Nierenberg, A., Petukhova, M., Rosellini, A., Sampson, N., Schoevers, R., Wilcox, M., Zaslavsky, A. (2016) 'Testing a machine-learning algorithm to predict the persistence and severity of major depressive disorder from baseline self-reports', *Molecular Psychiatry*, 21(10), pp. 1366–1371. doi: 10.1038/mp.2015.198.

Nie, Z., Gong, P. and Ye, J. (2016) 'Predict Risk of Relapse for Patients with Multiple Stages of Treatment of Depression', *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16,* San Francisco, CA, USA, 13-17 August. pp. 1795–1804. doi: 10.1145/2939672.2939870.

Perera, G., Broadbent, M., Callard, F., Chang, C., Downs, J., Dutta, R., Fernandes, A., Hayes, R., Henderson, M., Jackson, R., Jewell, A., Kadra, G., Little, R., Pritchard, M., Shetty, H., Tulloch, A., Stewart, R. (2016) 'Cohort profile of the South London and Maudsley NHS Foundation Trust Biomedical Research Centre (SLaM BRC) Case Register: current status and recent enhancement of an Electronic Mental Health Record-derived data resource.' *BMJ Open*, 6:e008721. doi:10.1136/bmjopen-2015-008721.

Pham, T., Tran, T., Phung, D., Venkatesh, S. (2017) 'Predicting healthcare trajectories from medical records: A deep learning approach', *Journal of Biomedical Informatics*, 69, pp. 218-229, Available at: https://doi.org/10.1016/j.jbi.2017.04.001 (Accessed: 26 March 2019).

Sakurai, H., Suzuki, T., Yoshimura, K., Mimura, M., Uchida, H. (2017) 'Predicting relapse with individual residual symptoms in major depressive disorder: a reanalysis of the STAR*D data', *Psychopharmacology*, 234(16), pp. 2453–2461. doi: 10.1007/s00213-017-4634-5.

South London and Maudsley NHS Foundation Trust (2013) *CRIS Security Model*. Available at: https://www.maudsleybrc.nihr.ac.uk/media/112184/cris-security-model.pdf. (Accessed: 26 March 2019).

South London and Maudsley NHS Foundation Trust (2019) *Clinical Record Interactive Search (CRIS)*. Available at: https://www.maudsleybrc.nihr.ac.uk/facilities/clinical-record-interactive-search-cris/ (Accessed: 4 January 2019).

Wang, J., Patten, S., Sareen, J., Bolton, J., Schmitz, N., MacQueen, G. (2014) 'Development and Validation of a Prediction Algorithm for use by Health Professionals in Prediction of Recurrence of Major Depression', *Depression and Anxiety*, 31(5), pp. 451–457. doi: 10.1002/da.22215.

Research Ethics Review Form: BSc, MSc and MA Projects

Computer Science Research Ethics Committee (CSREC)

http://www.city.ac.uk/department-computer-science/research-ethics

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with research ethics guidelines. In some cases, a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

In order to ensure that appropriate consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:

PART A: Ethics Checklist. All students must complete this part.
The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

PART B: Ethics Proportionate Review Form*.* Students who have answered "no" to all questions in A1, A2 and A3 and "yes" to question 4 in A4 in the ethics checklist must complete this part. The project supervisor has delegated authority to provide approval in such cases that are considered to involve MINIMAL risk.
The approval may be ***provisional*** – *identifying the planned research as* likely to involve MINIMAL RISK.
In such cases you must additionally seek ***full approval*** from the supervisor as the project progresses and details are established. ***Full approval*** must be acquired in writing, before beginning the planned research.

A.1 If you answer YES to any of the questions in this block, you must apply to an appropriate external ethics committee for approval and log this approval as an External Application through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
1.1	Does your research require approval from the National Research Ethics Service (NRES)? *e.g. because you are recruiting current NHS patients or staff?* *If you are unsure try - https://www.hra.nhs.uk/approvals-amendments/what-approvals-do-i-need/*	**NO**
1.2	Will you recruit participants who fall under the auspices of the Mental Capacity Act? *Such research needs to be approved by an external ethics committee such as NRES or the Social Care Research Ethics Committee - http://www.scie.org.uk/research/ethics-committee/*	**NO**
1.3	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners and those on probation? *Such research needs to be authorised by the ethics approval system of the National Offender Management Service.*	**NO**
A.2 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee, you must apply for approval from the Senate Research Ethics Committee (SREC) through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
2.1	Does your research involve participants who are unable to give informed consent? *For example, but not limited to, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf.*	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**

2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**
2.4	Does your project involve participants disclosing information about special category or sensitive subjects? *For example, but not limited to: racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings*	**NO**
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? *Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/*	**NO**
2.6	Does your research involve invasive or intrusive procedures? *These may include, but are not limited to, electrical stimulation, heat, cold or bruising.*	**NO**
2.7	Does your research involve animals?	**NO**
2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	**NO**
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the SREC, you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://ethics.city.ac.uk/ **Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee.**		*Delete as appropriate*
3.1	Does your research involve participants who are under the age of 18?	**NO**
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? *This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.*	**NO**
3.3	Are participants recruited because they are staff or students of City, University of London? *For example, students studying on a particular course or module.* *If yes, then approval is also required from the Head of Department or Programme Director.*	**NO**
3.4	Does your research involve intentional deception of participants?	**NO**
3.5	Does your research involve participants taking part without their informed consent?	**NO**
3.5	Is the risk posed to participants greater than that in normal working life?	**NO**
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	**NO**
A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK. **If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.**		*Delete as appropriate*

If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.		
4	Does your project involve human participants or their identifiable personal data? *For example, as interviewees, respondents to a survey or participants in testing.*	**NO**

B. Appendix B – Additional Ethics Forms

B.1 Ethics Part B Form

PART B: Ethics Proportionate Review Form

If you answered YES to question 4 and NO to all other questions in sections A1, A2 and A3 in PART A of this form, then you may use PART B of this form to submit an application for a proportionate ethics review of your project. Your project supervisor has delegated authority to review and approve this application under proportionate review. You must receive final approval from your supervisor in writing before beginning the planned research.

However, if you cannot provide all the required attachments (see B.3) with your project proposal (e.g. because you have not yet written the consent forms, interview schedules etc), the approval from your supervisor will be ***provisional***. You **must** submit the missing items to your supervisor for approval prior to commencing these parts of your project. Once again, you must receive written confirmation from your supervisor that any provisional approval has been superseded by with ***full approval*** of the planned activity as detailed in the full documents. **Failure to follow this procedure and demonstrate that final approval has been achieved may result in you failing the project module.**

Your supervisor may ask you to submit a full ethics application through Research Ethics Online, for instance if they are unable to approve your application, if the level of risks associated with your project change, or if you need an approval letter from the CSREC for an external organisation.

B.1 The following questions must be answered fully. **All grey instructions must be removed.**		*Delete as appropriate*
1.1.	Will you ensure that participants taking part in your project are fully informed about the purpose of the research?	**YES**
1.2	Will you ensure that participants taking part in your project are fully informed about the procedures affecting them or affecting any information collected about them, including information about how the data will be used, to whom it will be disclosed, and how long it will be kept?	**YES**
1.3	When people agree to participate in your project, will it be made clear to them that they may withdraw (i.e. not participate) at any time without any penalty?	**YES**
1.4	Will consent be obtained from the participants in your project? Consent from participants will be necessary if you plan to involve them in your project or if you plan to use identifiable personal data from existing records. "Identifiable personal data" means data relating to a living person who might be identifiable if the record includes their name, username, student id, DNA, fingerprint, address, etc. *If YES, you must attach drafts of the participant information sheet(s) and consent form(s) that you will use in section B.3 or, in the case of an existing dataset, provide details of how consent has been obtained.* *You must also retain the completed forms for subsequent inspection. Failure to provide the completed consent request forms will result in withdrawal of any earlier ethical approval of your project.*	**YES**
1.5	Have you made arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential?	**YES**

B.2 If the answer to the following question (B2) is YES, you must provide details		*Delete as appropriate*
2	Will the research be conducted in the participant's home or other non-University location? *If **YES**, you must provide details of how your safety will be ensured.*	**YES**

B.3 Attachments **ALL of the following documents MUST be provided to supervisors if applicable.** **All must be considered prior to final approval by supervisors.** **A written record of final approval must be provided and retained.**	***YES***	***NO***	***Not Applicable***
Details on how safety will be assured in any non-University location, including risk assessment if required (see B2)	**X**		
Details of arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential (see B1.5) *Any personal data must be acquired, stored and made accessible in ways that are GDPR compliant.*	**X**		
Full protocol for any workshops or interviews**			**X**
Participant information sheet(s)**	**X**		
Consent form(s)**	**X**		
Questionnaire(s)** *sharing a Qualtrics survey with your supervisor is recommended.*			**X**
Topic guide(s) for interviews and focus groups**			**X**
Permission from external organisations or Head of Department** *e.g. for recruitment of participants*			**X**

***If these items are not available at the time of submitting your project proposal, then **provisional approval** can still be given, under the condition that you must submit the final versions of all items to your supervisor for approval at a later date. **All** such items **must** be seen and approved by your supervisor before the activity for which they are needed begins. Written evidence of **final approval** of your planned activity must be acquired from your supervisor before you commence.*

Changes

If your plans change and any aspects of your research that are documented in the approval process change as a consequence, then any approval acquired is invalid. If issues addressed in Part A (the checklist) are affected, then you must complete the approval process again and establish the kind of approval that is required. If issues addressed in Part B are affected, then you must forward updated documentation to your supervisor and have received written confirmation of approval of the revised activity before proceeding.

Templates for Consent and Information

You must use the templates provided by the University as the basis for your participant information sheets and consent forms. You **must** adapt them according to the needs of your project before you submit them for consideration.

Participant Information Sheets, Consent Forms and Protocols must be consistent. Please ensure that this is the case prior to seeking approval. Failure to do so will slow down the approval process.

We strongly recommend using Qualtrics to produce digital information sheets and consent forms.

Further Information

http://www.city.ac.uk/department-computer-science/research-ethics

https://www.city.ac.uk/research/ethics/how-to-apply/participant-recruitment

https://www.city.ac.uk/research/ethics

B.2 Participant Information (Version 1.1, 11 July 2019)

PARTICIPANT INFORMATION

REC reference number, date and version of information sheet
Not applicable

Title of study
Modelling Depression Recurrence through Analysis of Electronic Health Records

Name of principal investigator/researcher
Lesley Dwyer

Invitation paragraph
We would like to invite you to take part in a research study. Before you decide whether you would like to take part it is important that you understand why the research is being done and what it would involve for you. Please take time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like more information. You will be given a copy of this information sheet to keep.

What is the purpose of the study?
The aim of this project is to investigate how machine learning can be used with anonymised electronic health records from the Clinical Records Interactive Search (CRIS) data to determine the likelihood that a patient will experience a recurrence of depression within a given time frame following remission of the illness. It will last approximately 3 months and is the final part of the Data Science MSc degree for the principal researcher.

Why have I been invited to take part?
You have been chosen to participate in this project based on your expertise of the subject matter and the data.

Do I have to take part?
Participation in the project is voluntary, and you can choose not to participate in part or all of the project. You can withdraw at any stage of the project without being penalised or disadvantaged in any way. It is up to you to decide whether or not to take part. If you do

decide to take part you will be asked to sign a consent form. If you decide to take part you are still free to withdraw at any time and without giving a reason.

What will happen if I take part?

- You will meet the researcher 1-2 times during the initial stages of the project to assist in defining the data requirements and then again 1-2 times to review the model results. The researcher may also contact you by email occasionally with follow-up questions or for clarification on something.
- Generally, meetings will last about an hour. These will be run as semi-structured interviews. The researcher will prepare an agenda with a set of questions to be covered, but there will also be the opportunity for a more general discussion about the topic(s) at hand.
- The research will take place at the Maudsley Hospital/King's College Denmark Hill campus.
- The study will run approximately 3 months, with the final report submitted on or before the 1st October 2019.

What are the possible disadvantages and risks of taking part?

The researcher does not foresee any disadvantages or risks of participating in the project.

Data privacy statement

City, University of London is the sponsor and the data controller of this study based in the United Kingdom. This means that we are responsible for looking after your information and using it properly. The legal basis under which your data will be processed is City's public task.

Your right to access, change or move your information are limited, as we need to manage your information in a specific way in order for the research to be reliable and accurate. To safeguard your rights, we will use the minimum personal-identifiable information possible (for further information please see https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/lawful-basis-for-processing/public-task/).

City will use your name and contact details to contact you about the research study as necessary. The only person at City who will have access to your identifiable information will be the principal researcher. City will keep identifiable information about you from this study for 10 years after the study has finished.

You can find out more about how City handles data by visiting https://www.city.ac.uk/about/governance/legal. If you are concerned about how we have processed your personal data, you can contact the Information Commissioner's Office (IOC) https://ico.org.uk/.

Will my taking part in the study be kept confidential?

- The principal researcher will have access to your name and contact details in order to conduct the research.
- Direct quotes will be anonymised and paraphrased in the final report.

- None of your personal information will be made public.
- Audio/video recording/photographs will not be used.
- Emails will be sent via City's email. Records will be kept by City for 10 years.

What will happen to the results?

Information provided by you may be included in the final dissertation, but your name will not be mentioned. If you wish to receive a copy of the research, please contact the principal researcher.

Who has reviewed the study?

This study has been approved by the research supervisor, Dr Cagatay Turkay, at City, University of London.

What if there is a problem?

If you have any problems, concerns or questions about this study, you should speak to the researcher or the researcher's supervisor. If you remain unhappy and wish to complain formally, you can do this through City's complaints procedure. To complain about the study, you need to phone 020 7040 3040. You can then ask to speak to the Secretary to Senate Research Ethics Committee and inform them that the name of the project is 'Modelling Depression Recurrence through Analysis of Electronic Health Records'.

You can also write to the Secretary at:
Anna Ramberg
Research Integrity Manager
City, University of London, Northampton Square
London, EC1V 0HB
Email: Anna.Ramberg.1@city.ac.uk

Insurance

City holds insurance policies which apply to this study. If you feel you have been harmed or injured by taking part in this study you may be eligible to claim compensation. This does not affect your legal rights to seek compensation. If you are harmed due to someone's negligence, then you may have grounds for legal action.

Further information and contact details

Principal Researcher:
Lesley Dwyer
Data Science MSc Student
City, University of London
Lesley.dwyer@city.ac.uk
079 4346 0585

Supervisor:
Dr Cagatay Turkay
Senior Lecturer
City, University of London
cagatay.turkay@city.ac.uk
020 7040 8415

Thank you for taking the time to read this information sheet.

B.3 Confidentiality Arrangement (Version 1.1, 11 July 2019)

CONFIDENTIALITY ARRANGEMENT

Name of principal investigator/researcher
Lesley Dwyer

REC reference number
Not Applicable

Title of study
Modelling Depression Recurrence through Analysis of Electronic Health Records

Please provide details of arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential (in reference to question B1.5 from Ethics Form, Part B).

- Lesley Dwyer, the principal researcher, will have access to participants' names and contact details in order to conduct the research.
- Direct quotes will be anonymised and paraphrased in the final report.
- None of the participants' personal information will be made public.
- Audio/video recording/photographs will not be used.
- Emails will be sent via City's email. Records will be kept by City for 10 years.
- This information will be provided to the participants on the Participant Information sheet, and participants will be asked to agree to this via an Informed Consent form.

B.4 Researcher Safety (Version 1.0, 2 July 2019)

RESEARCHER SAFETY

Name of principal investigator/researcher
Lesley Dwyer

REC reference number
Not Applicable

Title of study
Modelling Depression Recurrence through Analysis of Electronic Health Records

Where will the study be conducted?
The research will be conducted at The Maudsley Hospital/King's College Campus at Denmark Hill. More details about the location can be found here:
https://www.kcl.ac.uk/visit/denmark-hill-campus

How will the researcher's safety be ensured at this location (in reference to question B2 from Ethics Form, Part B)?

The site is a hospital and university campus with standard security measures in place, e.g. access to restricted areas by security pass, receptionist at the main entrance, etc. The meetings between the researcher and participants will take place at the participant's place of work, i.e. an office, desk, or meeting room onsite. Conducting the research at this location poses no further risk than if it were to be conducted onsite at City, University of London's main campus.

B.5 Informed Consent (Version 1.0, 2 July 2019)

INFORMED CONSENT

Name of principal investigator/researcher

Lesley Dwyer

REC reference number

Not Applicable

Title of study

Modelling Depression Recurrence through Analysis of Electronic Health Records

		Please tick or initial box
1	I confirm that I have read and understood the participant information Version 1.0 dated 2 July 2019 for the above study. I have had the opportunity to consider the information and ask questions which have been answered satisfactorily.	
2.	I understand that my participation is voluntary and that I am free to withdraw without giving a reason without being penalised or disadvantaged.	
3.	I agree to City recording and processing this information about me. I understand that this information will be used only for the purpose(s) explained in the participant information and my consent is conditional on City complying with its duties and obligations under the General Data Protection Regulation (GDPR).	
4.	I agree to take part in the above study.	

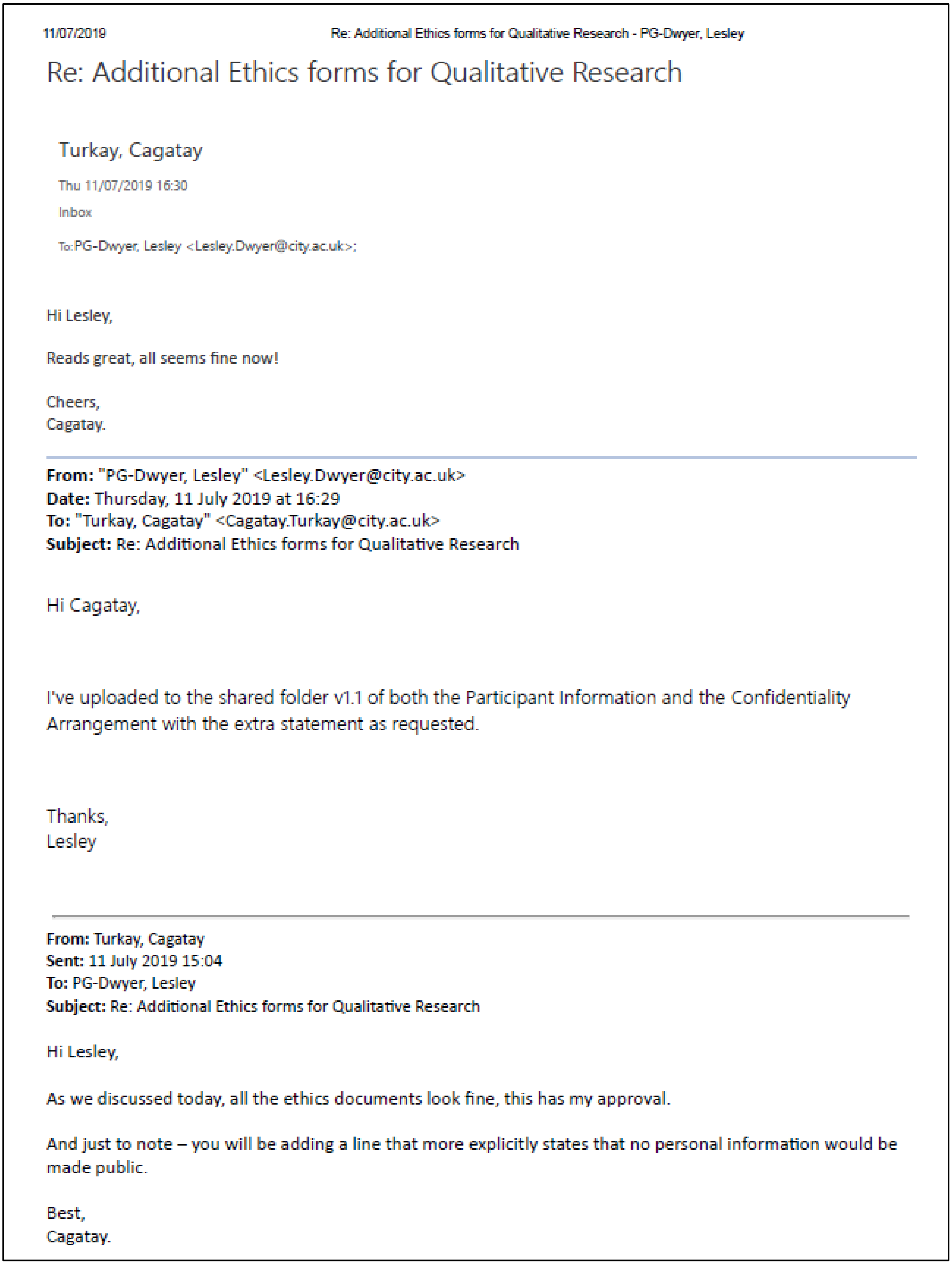
______________________ ______________________________ ______________

Name of Participant Signature Date

______________________ ______________________________ ______________

Name of Researcher Signature Date

B.6 Supervisor Approval Email



11/07/2019 Re: Additional Ethics forms for Qualitative Research - PG-Dwyer, Lesley

Re: Additional Ethics forms for Qualitative Research

Turkay, Cagatay

Thu 11/07/2019 16:30

Inbox

To:PG-Dwyer, Lesley <Lesley.Dwyer@city.ac.uk>;

Hi Lesley,

Reads great, all seems fine now!

Cheers,
Cagatay.

From: "PG-Dwyer, Lesley" <Lesley.Dwyer@city.ac.uk>
Date: Thursday, 11 July 2019 at 16:29
To: "Turkay, Cagatay" <Cagatay.Turkay@city.ac.uk>
Subject: Re: Additional Ethics forms for Qualitative Research

Hi Cagatay,

I've uploaded to the shared folder v1.1 of both the Participant Information and the Confidentiality Arrangement with the extra statement as requested.

Thanks,
Lesley

From: Turkay, Cagatay
Sent: 11 July 2019 15:04
To: PG-Dwyer, Lesley
Subject: Re: Additional Ethics forms for Qualitative Research

Hi Lesley,

As we discussed today, all the ethics documents look fine, this has my approval.

And just to note – you will be adding a line that more explicitly states that no personal information would be made public.

Best,
Cagatay.

C. Appendix C – Interview Questions and Notes

C.1 Interview with Subject Matter Expert (SME) 28th June, 2019

Questions:

1. How to define patients with depression (Primary Diag: F32, F33)?
    a. The ICD10 codes found in Primary Diag are the simplest way to extract this; in this case, F32 and F33 are the correct codes, but I may want to exclude some of the sub-codes with psychotic symptoms (see ICD10 website: https://icd.who.int/browse10/2016/en). Some patients may be missed if their diagnosis has only been recorded in text.
    b. Could also use text, but there is nothing to indicate this is the primary problem.
2. Which start date should be used to capture the initial set of patients (Referral Date, Accepted Date, Diagnosis Date, Combination of these)?
    a. This relates to an open question about how to define when someone is under SLaM's care. Should the overarching referral be used, as there may be a waiting period before the patient is actually seen, or the team/ward level? **Lesley to aim to speak to Informatician next week about this.**
    b. Not all referrals will be accepted.
3. Which end date should be used to capture the initial set of patients (Discharge Date)?
    a. Discharge date, and I plan to only consider patients where this is populated.
4. Is there a way to link a diagnosis to a referral?
    a. This will need to be manufactured.
5. How would I represent 'level of face-to-face contact' or 'level of service' (Number of face-to-face Events)?
    a. Events with 'Face to Face' but there will be different variations/spellings of this and also other values like 'Group' that may represent face-to-face.
    b. Can also look at 'Attended' for this; this also has variations like 'Attended-late', etc.
    c. Face to face events will probably be understated, as some will be marked as 'Other' or NULL.
6. How can I extract the 'symptom profile' (HoNOS score, PHQ9 score, NLP apps) including dates?
    a. HoNOS can be used although it's a fairly crude measure; most people will use HoNOS or HoNOS65+ (older patients).
    b. PHQ9 may be in text (Event Comments), so could possibly find related info with a keyword search.
    c. PHQ9 and GAD7 scores could possibly be pulled from IAPT services where patients are being treated by SLaM and IAPT at the same time; this can be linked on NHS ID. Although only a portion of patients will be using IAPT. **Lesley to talk to Clinician about possibly linking datasets.**
    d. NLP apps have symptoms entered and a date, but it would only be the date they were reported, not representative of the date range they were occurring. Also the number of times a symptom is mentioned/occurred may not be an indication of severity – it may indicate service use. **SME to send Lesley list of symptoms in NLP apps.**
7. Why would a patient have a re-referral without a previous discharge (e.g., 10393465)?
    a. This could be a data entry issue.
8. How should 2 diagnoses on same referral date be handled (e.g. 10396651)?

   a. Sometimes one diagnosis is entered and then later a different diagnosis is entered without removing the first. This may or may not indicate the first one is incorrect. This also could be a way of entering multiple diagnoses (comorbidities). There is variation in the way this data is entered.

9. How to identify DNAs (Events without 'Attended')?
   a. Events without 'Attended' and other variations of that.
   b. DNAs may be an indication they were too ill to attend.
10. How to identify comorbidities (additional Primary Diag's within same timeframe)?
    a. See # 8
    b. For non-mental health comorbidities, the diagnosis form is not as good.
11. Where would I find discharge reason (Discharge Method spell, Discharge Destination)? Clinician mentioned it was in text.
    a. Event text (Comments)
    b. Letter to patient or GP found in Attachments Text
    c. Lesley to extract unique list of reasons/destinations to review with Clinician; identify and possibly exclude any which indicate the patient is not in remission.
12. When extracting data from CRIS, is there a way to get it to show all data for repeated rows? E.g., if including event data, a row per event is created but then referral info may only show on 1 row
    a. Use the SQL database instead of the CRIS frontend
13. Field suggestions for Physical Health – a lot of them aren't populated – may be a question for Clinician
    a. Some are in NLP apps
       i. Blood pressure
       ii. Smoking
       iii. Diabetes
       iv. BMI
    b. Medication could be used to infer physical health, but it will not always be entered if it's not related to their treatment at SLaM.
    c. Other options which are not straightforward
       i. Link to HES (A&E attendance, etc.), requires additional project approvals
       ii. Algorithm which is used to indicate physical health; developer no longer works here, so it's unclear if this should/can be used. **Lesley to check with Clinician.**

C.2 Interview with Clinician 8th July, 2019

Agenda Topics/Questions

1. Review/confirm **output for analysis** is valid.
   a. Remove physical health. I've also decided to remove family/medical history as it's text based.
   b. Add marital status, area-level deprivation.
   c. Risk assessment (RA) has changed over the last few years, and the two versions are not directly comparable; would need to be a risk assessment showing something, i.e. suicidation and self-neglect. This can be extracted from both types of RA.
   d. HoNOS

i. Look at the most recent HoNOS at point of discharge, possibly the time since last HoNOS

ii. Could take all sub-scales (HoNOS, HoNOS 65+, etc.)

iii. Suggest binning the scores into 0 or 1 where a score of 2-4 is set to 1 and a score of 0-1 is set to 0.

e. Duration of an episode prior to discharge is unreliable before 2006.

f. Determine how to define medication profile. Could look at specific antidepressant names over last 12 months (before first discharge).

2. Should I assume the **sequence of patient events/records** is important when considering **relapse**?
   - a. Possibly, it could be.
3. Where is the best place to extract **symptom profile?** What is needed to do this?
   - a. NLP Apps?
      - i. In a separate database
      - ii. Create separate feature/variable for ‘depression symptoms in last 12 months’ (before first discharge date).
   - b. Link SLaM and IAPT to pull PHQ9 and GAD7 scores?
      - i. In a separate database; although there will not be much overlap.
4. Should **discharge** destination/DNAs/reason be used to exclude patients from **remission**?
   - a. Restrict ‘remission’ to discharges to GP and Home.
   - b. Don’t use Event non-attendance to define discharge/remission, but include it as a variable, i.e. number of events not attended in last 6 months (before first discharge date).
5. Where is the best place extract **physical health?** What is needed to do this?
   - a. Don’t extract this; will get some indication from HoNOS scale scores.
6. **Diagnostic categories** F32, F33, should I leave any sub-categories out, e.g. with psychotic symptoms?
   - a. Leave all sub-categories in to define patient group.
   - b. Create separate feature/variable for ‘psychotic symptoms in last 12 months’ (before first discharge date).
7. How to handle patients with **multiple Primary Diagnoses**?
   - a. Define the patient cohort first and use diagnosis from that.
   - b. Create separate features/variables for any history/mention of:
      - i. Manic – F30
      - ii. Bipolar – F31
      - iii. Organic – F0
      - iv. Substance – F1

v. Schizophrenia – F2

8. Which ways could the **data be segmented/restricted** to build specific models, e.g. by age group, limiting patients by number of events, etc.?
    a. Gender
    b. Possibly age group
    c. The fewer stratifications the better, as they can potentially lead to false positives.
9. **Face-to-face** contact values.
    a. Create a feature/variable for number of 'face to face' and 'attended' events in last 6 months (before first discharge).
    b. Inpatient events are not well-recorded. Create another feature/variable for number of days in hospital/inpatient in last 6 months (before first discharge).

C.3 Interview with SME 12th July, 2019

<u>**Questions**</u>:

1. What is the difference between Inpatient_episode and Ward_stay tables? I want to extract the number of days in hospital/inpatient in last 6 months.
    a. They should match. If you're just looking for the total number of days, inpatient_episode can be used. If you want to see the different wards they have visited, use ward_stay table.
2. Can patients come into the trust without a referral, i.e. without being in the Referral table?
    a. Yes, they may have events after an active referral has ended, e.g. a follow up or meeting with a social worker.
3. Should I exclude data before a certain time period, e.g. before 2007?
    a. Generally, 2007 is used as a starting point. Data before that was migrated. Consider that initial diagnoses in 2007 (or any date) may not actually be the first episode of depression. They may have happened outside of SLaM or data may not have been migrated. One idea is to look at a later year to start, e.g. 2010, to aim to capture all SLaM-entered first episodes.
4. Where can I find Area level deprivation field(s)?
    a. IMD tables for different years are in SQLCRIS_Common. There are scores and ranks for deprivation. There are LSOA and MSOA for census years 2001 and 2011. Join the patient address (must decide which address) to LSOA to deprivation score.
5. Which fields would fall into suicide ideation, self-harm in the Risk Assessment tables?
    a. The pre-2016 process was that a patient would fill in a short RA form and then if needed, they would fill out a full RA form. Now, the new process is that everyone fills

out a full (new version) RA. In the old form there are 4-5 questions together about suicide (e.g., suicide attempt, end life, suicide ideation, lethal) and one about self-harm. In the new form, there is one question available in the database about current or historical self-harm or suicide. If they answer yes to this question, they get asked more questions which are broken down between current and historical; however, the full list may only be available via XML, but I can ask if I need access to this.

6. How do I request access to NLP App symptom and medications from DBAs?
    a. Medication is in SQLCRIS_Common.
    b. Symptoms – CRIS team will provide this; I will send over my list of patient IDs, date range and symptoms.
7. Confirm list of depressive symptoms, anti-depressant medication.
    a. For anti-depressants, look in gazeteer_source_combined_final where the brf_code like ‘0403%’. Also, be careful about drugs that can act as anti-depressants but are normally anti-psychotics.

D. Appendix D – Data Specifications

D.1 Sequential Data Specification

D.1.1 Patients Initial Extract Specification

Data currency:	patients
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Patient	Identifier	EPR_Form	BrcId	links to all other tables	Patients_initial
Patient	Age	EPR_Form	cleaneddateofbirth	>=18 years old when referred	Patients_initial
Diagnosis	Initial Primary diagnosis	Diagnosis	Primary_Diag	F32 or F33	Patients_initial
Diagnosis	Initial Diagnosis date	Diagnosis	Diagnosis_date	minimum Diagnosis_date AND Diagnosis_date is not NULL	Patients_initial
Referral	Initial Spell number	Referral	Spell_number	links to other tables except EPR_Form	Patients_initial
Referral	Initial Accepted date	Referral	Accepted_date	Is not NULL	Patients_initial
Referral	Initial Discharge date	Referral	Discharge_date	Is not NULL	Patients_initial
Referral	Initial Discharge destination	Referral	Discharge_Destination_ID	GP or 'Home - No Follow Up Required'	Patients_initial
Patient	Gender	EPR_Form	Gender_ID		Patients_initial
Patient	Marital status	EPR_Form	Marital_Status_ID		Patients_initial
Patient	Ethnic group	EPR_Form	ethnicitycleaned		Patients_initial
Patient	Create date	EPR_Form	Create_Dttm		Patients_initial
Patient	Updated date	EPR_Form	Update_Dttm		Patients_initial
Referral	Initial Referral date	Referral	Referral_date		Patients_initial

Referral	Num Days in Initial Episode	Referral	Calculated	(Initial Discharge date) - (Initial Accepted date)	Patients_initial
Referral	Num Days to Initial Accepted	Referral	Calculated	(Initial Accepted date) - (Initial Referral date)	Patients_initial
Diagnosis, Referral	Num Days to Initial Diagnosis	Calculated	Calculated	(Initial Diagnosis date) - (Initial Accepted date)	Patients_initial
Referral	Diagnosis Num Days After Discharge	Referral	Calculated	(Initial Diagnosis date) - (Initial Discharge date)	Patients_initial

D.1.2 Patients Relapse Extract Specification

Data currency:	patients
Cohort definition:	adult patients with at least one subsequent depression referral following discharge from their initial depression referral.

Entity	**Attribute**	**Source table**	**Column name**	**Definition**	**Extract**
Patient	Identifier	EPR_Form	BrcId	links to all other tables	Patients_relapse
Diagnosis	Re-referral Primary diagnosis	Diagnosis	Primary_Diag	F32 or F33	Patients_relapse
Diagnosis	Re-referral Diagnosis date	Diagnosis	Diagnosis_date	> Initial Diagnosis date	Patients_relapse
Referral	Re-referral Accepted date	Referral	Accepted_date	> Initial Discharge date	Patients_relapse
Referral	Re-referral Spell Number	Referral	Spell_number	> Initial Spell number	Patients_relapse
Referral	Re-referral Discharge date	Referral	Discharge_date		Patients_relapse
Referral	Num Days in Remission	Referral	Calculated	(Re-referral Accepted date) - (Initial Discharge date)	Patients_relapse
Referral	Num Days in Relapse	Referral	Calculated	(Re-referral Discharge date) - (Re-referral Accepted date)	Patients_relapse
Referral, Diagnosis	Num Days between Diagnosis and Discharge	Referral, Diagnosis	Calculated	(Re-referral Discharge date) - (Re-referral Diagnosis date)	Patients_relapse

Referral	Relapse All	Referral	Calculated	1 or 0; 1 if: (Re-referral Accepted Date - Initial Discharge Date) <= 12M	Patients_relapse
Referral	Relapse 12M	Referral	Calculated	1 or 0; 1 if: (Re-referral Accepted Date - Initial Discharge Date) <= 12M	Patients_relapse
Referral	Relapse 24M	Referral	Calculated	1 or 0; 1 if: (Re-referral Accepted Date - Initial Discharge Date) <= 24M	Patients_relapse
Referral	Relapse 36M	Referral	Calculated	1 or 0; 1 if: (Re-referral Accepted Date - Initial Discharge Date) <= 36M	Patients_relapse

D.1.3 Patient Features Extracts Specification

Data currency:	patients
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
ONS_2015_IMD, Address	Area level deprivation	EPR_Form	Index_of_Multiple_Deprivation_Score	average of Index_of_Multiple_Deprivation_Score for LSOA11	AreaLevelDeprivation
Diagnosis, Referral	Psychotic symptoms last 12 months	Calculated	Calculated	1 or 0; 1 if Diagnosis.Primary_Diag in F32.3 or F33.3 AND Diagnosis.Diagnosis_date >= (Initial Discharge date - 12 months) AND Diagnosis.Diagnosis_date <= Initial Discharge date	PsychoticSymptoms

Old Risk Assessment	Suicide Ideation, Self-neglect	Risk_assessment	Calculated	1 or 0; 1 if any instance ever occurred where: Does_Patient_Have_History_Of_Suicide_Attempt_ID = 'Yes' OR If_So_Did_He_Use_Violent_Perceived_Lethal_Method_ID = 'Yes' OR Has_Patient_Made_Plan_To_End_Life_ID = 'Yes' OR Is_Patient_Expressing_Suicidal_Ideation_ID = 'Yes' OR Is_Patient_Expressing_Feelings_Of_Hopelessness_ID = 'Yes' OR Risk_Of_Deliberate_Self_Harm_ID = 'Yes' AND Assessed_Date <= Initial Discharge Date	RiskAssessmentOld
New Risk Assessment	Suicide Ideation, Self-neglect	RiskAssessmentTool	Calculated	1 or 0; 1 if any instance ever occurred where: Selfharmsuicide = 'Yes' AND DateOfAssessment <= Initial Discharge Date	RiskAssessmentNew
Event, Referral	Num face-to-face last 6 months	Calculated	Calculated	Count of rows where: Event.start_date >= (Initial Discharge date - 6 months) AND Event.start date <= Initial Discharge date AND Event.Event_Type_Of_Contact_ID in ('Face To Face', 'Face-to-face', 'Inpatient Event', 'Inpatient Shift') AND Event.Event_Outcome_ID in ('Attended', 'Attended on time/before HCP ready', 'Arrived late but was seen', 'Arrived late after HCP available but seen', 'Inpatient Event/Shift')	Face2FaceContact
Event, Referral	Num DNAs last 6 months	Calculated	Calculated	Count of rows where: Event.start_date >= (Initial Discharge date - 6 months) AND Event.start date <= Initial Discharge date AND Event.Event_Outcome_ID in ('DNA by Client', 'DNA', 'Did	Num_DNAs_6M

				not attend', 'DNA by Trust', 'DNA by Clinician')	
Inpatient Episode, Referral	Num days inpatient last 6 months	Calculated	Calculated	(Inpatient_episode.Discharge_date - Inpatient_episode.Admission_date) where: Inpatient_episode.Admission_date >= (Initial Discharge date - 6 months) AND Inpatient_episode.Discharge_date <= Initial Discharge date	InpatientDays
HoNOS, Referral	Most recent HoNOS	Calculated	Calculated	HoNOS closest to Initial Discharge Date AND Honos date is between initial accepted date and initial discharge date Consider: 1 or 0; 1 if: HoNOS score is 2-4 0 if: HoNOS score is 0-1 note: may want multiple features, 12 questions for each HoNOS scale score (5 Honos types)	Honos
Medication, NLP Medication	Medication last 12 months	Medication, tbl_medication_gazeteer_source_combined_final	Calculated	need to summarise (count, binary or one-hot encoding) note: 27 unique medications	Medication
NLP Symptom Apps	Symptoms last 12 months	Calculated	Calculated	CRIS Team to provide note: 21 unique depressive symptoms	
Diagnosis	History of manic	Diagnosis	Calculated	1 or 0; 1 if: Primary_Diag = F30 or Secondary_Diag_1 = F30 … Secondary_Diag_6 = F30 note: any history, do not limit to 2010+	History_manic
Diagnosis	History of bipolar	Diagnosis	Calculated	1 or 0; 1 if: Primary_Diag = F31 or Secondary_Diag_1 = F31 … Secondary_Diag_6 = F31 note: any history, do not limit to 2010+	History_Bipolar

Diagnosis	History of organic	Diagnosis	Calculated	2 or 0; 1 if: Primary_Diag = F0% or Secondary_Diag_1 = F0% … Secondary_Diag_6 = F0% note: any history, do not limit to 2010+	History_organic
Diagnosis	History of substance	Diagnosis	Calculated	3 or 0; 1 if: Primary_Diag = F1% or Secondary_Diag_1 = F1% … Secondary_Diag_6 = F1% note: any history, do not limit to 2010+	History_substance
Diagnosis	History of schizophrenia	Diagnosis	Calculated	4 or 0; 1 if: Primary_Diag = F2% or Secondary_Diag_1 = F2% … Secondary_Diag_6 = F2% note: any history, do not limit to 2010+	History_schizophrenia
Referral	Number of re-referrals	Referral	Spell_number	Count(Spell_number)	Rereferral_count
Referral	Discharge Destination	Referral	Discharge_Destination_ID		Discharge

D.2 Non-sequential Data Specification

D.2.1 Sequential Accepted events extract

Data currency:	referrals
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	**Attribute**	**Source table**	**Column name**	**Definition**	**Extract**
Referral	Patient ID	Referral	BrcId	Patient ID	Seq_Accepted
Referral	Date	Referral	Accepted_date	Accepted_date <= Initial_discharge_date AND Accepted_date is not NULL	Seq_Accepted
Referral	Event Type	Referral	N/A	Text: 'Accepted'	Seq_Accepted

D.2.2 Sequential Area Level Deprivation events extract

Data currency:	Area-level deprivation
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Address	Patient ID	Address	BrcId	Patient ID	Seq_AreaLevelDeprivation
Address	Date	Address	Start_date		Seq_AreaLevelDeprivation
IMD	IMD score	SQLCRIS_Common.ons_2015_imd	Index of multiple deprivation (IMD) score	Index of multiple deprivation (IMD) score	Seq_AreaLevelDeprivation
n/a	n/a	n/a	N/A	Text: 'Address_start'	Seq_AreaLevelDeprivation

D.2.3 Sequential Comorbidities events extract

Data currency:	diagnosis
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Diagnosis	Patient ID	Diagnosis	BrcId	Patient ID	Seq_Comorbidities
Diagnosis	Date	Diagnosis	Diagnosis_date	where: Primary_Diag in (F30, F31, F0, F1, F2) or Secondary_Diag_1 in (F30, F31, F0, F1, F2) … Secondary_Diag_6 in (F30, F31, F0, F1, F2) note: any history, do not limit to 2010+	Seq_Comorbidities
Diagnosis	Diagnosis	Diagnosis	Primary_Diag, Secondary_Diag_1, ... Secondary_Diag_6	where: Primary_Diag in (F30, F31, F0, F1, F2) or Secondary_Diag_1 in (F30, F31, F0, F1, F2) … Secondary_Diag_6 in (F30, F31, F0, F1, F2) note: any history, do not limit to 2010+	Seq_Comorbidities
Diagnosis	Event Type	Diagnosis	N/A	Text: 'Comorbidity'	Seq_Comorbidities

D.2.4 Sequential Diagnosis events extract

Data currency:	diagnosis (incl with Psychotic symptoms)
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Diagnosis	Patient ID	Diagnosis	BrcId	Patient ID	Seq_Diagnosis
Diagnosis	Date	Diagnosis	Diagnosis_date	Diagnosis_date >= Initial_accepted_date AND Diagnosis_date <= Initial_discharge_date AND Diagnosis_date is not NULL	Seq_Diagnosis
Diagnosis	Primary Diagnosis	Diagnosis	Primary_Diag	F32 or F33 (needs cleanup of values - some dupes with extra spaces)	Seq_Diagnosis
Diagnosis	Event Type	Diagnosis	N/A	Text: 'Diagnosis'	Seq_Diagnosis

D.2.5 Sequential Discharge events extract

Data currency:	referrals
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Referral	Patient ID	Referral	BrcId	Patient ID	Seq_Discharge
Referral	Date	Referral	Discharge_date	Discharge_date >= Initial_accepted_date AND Discharge_date <= Initial_discharge_date AND Discharge_date is not NULL	Seq_Discharge
Referral	Discharge destination	Referral	Discharge_destination_ID	GP or 'Home - No Follow Up Required'	Seq_Discharge
Referral	Event Type	Referral	N/A	Text: 'Discharge'	Seq_Discharge

D.2.6 Sequential DNAs Events events extract

Data currency:	Events
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Events	Patient ID	Events	BrcId	Patient ID	Seq_DNA

Events	Date	Events	start_date	Where: Event.start_date >= (Initial Accepted date) AND Event.start date <= Initial Discharge date AND Event.Event_Outcome_ID in ('DNA by Client', 'DNA', 'Did not attend', 'DNA by Trust', 'DNA by Clinician')	Seq_DNA
Events	Event Type	Events	N/A	Text: 'DNA'	Seq_DNA

D.2.7 Sequential Face to Face Events events extract

Data currency:	Events
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Events	Patient ID	Events	BrcId	Patient ID	Seq_F2F
Events	Date	Events	start_date	Where: Event.start_date >= (Initial Accepted Date) AND Event.start date <= Initial Discharge date AND Event.Event_Type_Of_Contact_ID in ('Face To Face', 'Face-to-face') AND Event.Event_Outcome_ID in ('Attended', 'Attended on time/before HCP ready', 'Arrived late but was seen', 'Arrived late after HCP available but seen')	Seq_F2F
Events	Event Type	Events	N/A		Seq_F2F

D.2.8 Sequential Honos events extracts

Data currency:	Honos (include Honos, Honos65, Honos_abi, Honos_Secure)
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Honos	Patient ID	Honos	BrcId	Patient ID	Seq_Honos
Honos	Date	Honos	Rating_Date	Rating_date >= Initial_accepted_date AND Rating_date <= Initial_discharge_date AND Rating_date is not NULL	Seq_Honos

Honos	Event Type	Honos	N/A	Text: 'Honos'	Seq_Honos
Honos	Honos_Scale 1	Honos	Agitated_Behaviour_Score_ID		Seq_Honos
Honos	Honos_Scale 2	Honos	Self_Injury_Score_ID		Seq_Honos
Honos	Honos_Scale 3	Honos	Problem_Drinking_Drugs_Score_ID		Seq_Honos
Honos	Honos_Scale 4	Honos	Cognitive_Problems_Score_ID		Seq_Honos
Honos	Honos_Scale 5	Honos	Physical_Illness_Score_ID		Seq_Honos
Honos	Honos_Scale 6	Honos	Hallucinations_Score_ID		Seq_Honos
Honos	Honos_Scale 7	Honos	Depressed_Mood_Score_ID		Seq_Honos
Honos	Honos_Scale 8_Type	Honos	Other_Mental_Problems_Type_ID		Seq_Honos
Honos	Honos_Scale 8	Honos	Other_Mental_Problems_Score_ID		Seq_Honos
Honos	Honos_Scale 9	Honos	Relationship_Problems_Score_ID		Seq_Honos
Honos	Honos_Scale 10	Honos	Daily_Living_Problems_Score_ID		Seq_Honos
Honos	Honos_Scale 11	Honos	Living_Conditions_Problems_Score_ID		Seq_Honos
Honos	Honos_Scale 12	Honos	Occupational_Problems_Score_ID		Seq_Honos
Honos	Honos_Total	Honos	Total		Seq_Honos
Honos	Honos_Adjusted_Total	Honos	Adjusted_Total		Seq_Honos

D.2.9 Sequential Inpatient events extracts

Data currency:	Inpatient episodes
Cohort definition:	adult patients with their earliest depression episode who have been discharged

Entity	**Attribute**	**Source table**	**Column name**	**Definition**	**Extract**
Inpatient Episodes	Patient ID	Inpatient_episode	BrcId	Patient ID	Seq_Inpatient
Inpatient Episodes	Date	Inpatient_episode	Admission_date	Where: Inpatient_episode.Admission_date >= (Initial Accepted date) AND Inpatient_episode.Discharge_date <= Initial Discharge date	Seq_Inpatient

Inpatient Episodes	Num_inpatient_days	Inpatient_episode	Calculated	(Inpatient_episode.Discharge_date - Inpatient_episode.Admission_date) Where: Inpatient_episode.Admission_date >= (Initial Accepted date) AND Inpatient_episode.Discharge_date <= Initial Discharge date	Seq_Inpatient
Inpatient Episodes	Event Type	Inpatient_episode	N/A	Text: 'Inpatient'	Seq_Inpatient

D.2.10 Sequential Medication events extracts

Data currency:	medication
Cohort definition	adult patients with their earliest depression episode who have been discharged

Entity	Attribute	Source table	Column name	Definition	Extract
Medication, NLP Medication	Patient ID	Medication, tbl_medication_gazeteer_source_combined_final	BrcId	Patient ID	Seq_Medication
Medication, NLP Medication	Date	Medication, tbl_medication_gazeteer_source_combined_final	Medication_Start_Date	Where: Medication_Start_date >= Initial_accepted_date AND Medication_Start_date <= Initial_discharge_date AND Medication_Start_date is not NULL	Seq_Medication
Medication, NLP Medication	Medication	Medication, tbl_medication_gazeteer_source_combined_final	gazetteer	Where: Medication_Start_date >= Initial_accepted_date AND Medication_Start_date <= Initial_discharge_date AND	Seq_Medication

				Medication_Start_date is not NULL	
Medication, NLP Medication	Event Type	Medication, tbl_medication_gazeteer_source_combined_final	N/A	Text: 'Anti-depressant Start'	Seq_Medication

E. Appendix E – SQL Queries

E.1 Area Level Deprivation

```
-- Area level deprivation extract

Select
patients_initial.BrcId, avg(imd.[Index of Multiple Deprivation (IMD) Score])
Avg_IMD_Score
from
(
	select -- patients_initial: adult patients with their earliest depression
diagnosis 2010 or later who have been discharged to Home or GP
	dp.BrcId
	, dp.Primary_diag as Initial_primary_diagnosis
	, dp.Diagnosis_date as Initial_diagnosis_date
	, dp.Spell_number as Initial_spell_number
	, r.Accepted_date as Initial_accepted_date
	, r.Discharge_Date as Initial_discharge_date
	, r.Discharge_Destination_ID as Initial_discharge_destination
	, r.Referral_Date as Initial_referral_date
	, datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
	, datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
	, datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
	, diagnosis_num_days_after_discharge = case
	when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
	datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
	else 0
	end
	, dp.Age
	, dp.Gender_ID
	, dp.Marital_Status_ID
	, dp.ethnicitycleaned
	, dp.patient_created_date
	, dp.patient_updated_date
	from
	(
		select -- patients with depression diagnosis
		epr.BrcId
		, CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
		, epr.cleaneddateofbirth
		, epr.Gender_ID
		, epr.Marital_Status_ID
		, epr.Create_Dttm as patient_created_date
		, epr.Updated_Dttm as patient_updated_date
		, epr.ethnicitycleaned
		, d.Primary_Diag
		, d.Diagnosis_Date
		, ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
		, d.Spell_Number
		from
		EPR_Form epr
		inner join
		Diagnosis d
		on epr.BrcId = d.BrcId
		where
		(d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
left outer join
Address a
on a.BrcId = patients_initial.BrcId
left outer join
SQLCRIS_Common.dbo.ons_2015_imd imd -- 2015 ONS data
on imd.[LSOA code (2011)] = a.LSOA11 -- 2011 Census
where (a.Start_Date <= patients_initial.Initial_discharge_date AND -- Address is not
effective after patient has been discharged
(a.End_Date >= patients_initial.Initial_accepted_date OR a.End_Date is null))
group by
patients_initial.BrcId
order by
patients_initial.BrcId
```

E.2 Demographics

```
-- Demographics extract

select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged to Home or GP and demographic details
dp.BrcId
, dp.Age
, dp.Gender_ID
, case when dp.Gender_ID in ('Female') then 1 else 0 end as Female
, case when dp.Gender_ID in ('Male') then 1 else 0 end as Male
, case when dp.Gender_ID in ('Other', 'Not Known', 'Not Specified') or dp.Gender_ID is
null then 1 else 0 end as OtherNotStated
, dp.Marital_Status_ID
, case when dp.Marital_Status_ID in ('Single') then 1 else 0 end as Single
, case when dp.Marital_Status_ID in ('Cohabiting') then 1 else 0 end as Cohabiting
, case when dp.Marital_Status_ID in ('Married/Civil Partner', 'Married') then 1 else 0
end as Married
, case when dp.Marital_Status_ID in ('Divorced/Civil Partnership Dissolved',
'Divorced') then 1 else 0 end as Divorced
, case when dp.Marital_Status_ID in ('Separated') then 1 else 0 end as Separated
, case when dp.Marital_Status_ID in ('Widowed/Surviving Civil Partner', 'Widowed')
then 1 else 0 end as Widowed
, case when dp.Marital_Status_ID in ('Not Known', 'Not Disclosed') or
dp.Marital_Status_ID is null then 1 else 0 end as OtherNotStated
, dp.ethnicitycleaned
, case when dp.ethnicitycleaned in ('British (A)') then 1 else 0 end as British
, case when dp.ethnicitycleaned in ('Any other ethnic group (S)', 'Not Stated (Z)') or
dp.ethnicitycleaned is null then 1 else 0 end as OtherNotStated
```

```
, case when dp.ethnicitycleaned in ('Any other white background (C)') then 1 else 0
end as OtherWhite
, case when dp.ethnicitycleaned in ('African (N)') then 1 else 0 end as African
, case when dp.ethnicitycleaned in ('Any other black background (P)') then 1 else 0
end as OtherBlack
, case when dp.ethnicitycleaned in ('Caribbean (M)') then 1 else 0 end as Caribbean
, case when dp.ethnicitycleaned in ('Any other Asian background (L)') then 1 else 0
end as OtherAsian
, case when dp.ethnicitycleaned in ('Irish (B)') then 1 else 0 end as Irish
, case when dp.ethnicitycleaned in ('Indian (H)') then 1 else 0 end as Indian
, case when dp.ethnicitycleaned in ('Pakistani (J)') then 1 else 0 end as Pakistani
, case when dp.ethnicitycleaned in ('White and black Caribbean (D)') then 1 else 0 end
as WhiteBlackCaribbean
, case when dp.ethnicitycleaned in ('Any other mixed background (G)') then 1 else 0
end as OtherMixed
, case when dp.ethnicitycleaned in ('Bangladeshi (K)') then 1 else 0 end as
Bangladeshi
, case when dp.ethnicitycleaned in ('Chinese (R)') then 1 else 0 end as Chinese
, case when dp.ethnicitycleaned in ('White and Black African (E)') then 1 else 0 end
as WhiteBlackAfrican
, case when dp.ethnicitycleaned in ('White and Asian (F)') then 1 else 0 end as
WhiteAsian
from
(
       select -- patients with depression diagnosis
       epr.BrcId
       , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as Age
       , epr.cleaneddateofbirth
       , epr.Gender_ID
       , epr.Marital_Status_ID
       , epr.Create_Dttm as patient_created_date
       , epr.Updated_Dttm as patient_updated_date
       , epr.ethnicitycleaned
       , d.Primary_Diag
       , d.Diagnosis_Date
       , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
       , d.Spell_Number
       from
       EPR_Form epr
       inner join
       Diagnosis d
       on epr.BrcId = d.BrcId
       where
       (d.Primary_Diag like '%F32%' OR -- depressive episode
       d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
       d.Diagnosis_Date is not null -- must have a diagnosis date
)
dp
inner join
Referral r
on
(r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
where
(r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
r.Accepted_date is not null AND -- must have been accepted to the trust
r.Discharge_Date is not null AND -- must have been discharged
r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
dp.Diagnosis_Date <= r.Discharge_Date AND
dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
order by r.BrcId
```

E.3 Discharge Destinations

```
-- Discharge Destinations extract

select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged and discharge destination
dp.BrcId,
r.Discharge_Date,
r.Discharge_Destination_ID
from
(
       select -- patients with depression diagnosis
       epr.BrcId
       , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as Age
       , epr.cleaneddateofbirth
       , epr.Gender_ID
       , epr.Marital_Status_ID
       , epr.Create_Dttm as patient_created_date
       , epr.Updated_Dttm as patient_updated_date
       , epr.ethnicitycleaned
       , d.Primary_Diag
       , d.Diagnosis_Date
       , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
       , d.Spell_Number
       from
       EPR_Form epr
       inner join
       Diagnosis d
       on epr.BrcId = d.BrcId
       where
       (d.Primary_Diag like '%F32%' OR -- depressive episode
       d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
       d.Diagnosis_Date is not null -- must have a diagnosis date
)
dp
inner join
Referral r
on
(r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
where
(r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
r.Accepted_date is not null AND -- must have been accepted to the trust
r.Discharge_Date is not null AND -- must have been discharged
dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
dp.Diagnosis_Date <= r.Discharge_Date AND
dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
order by r.BrcId
```

E.4 Face to Face Contact

```
-- Face to Face Contact extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP and count of face to face events last 6 months
patients_initial.BrcId,
count(e.CN_Doc_Id) as Num_F2F_events_6M
from
(
       select -- patients_initial
       dp.BrcId
```

```
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
```

```
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
left outer join
Event e
on
patients_initial.BrcId = e.BrcId
where
(e.start_date >= (Initial_discharge_date - 182.625) ) -- event occurs in the 6 months
before discharge
AND (e.Start_Date <= Initial_discharge_date )
AND e.Event_Type_Of_Contact_ID in ('Face To Face', 'Face-to-face')
AND e.Event_Outcome_ID in
('Attended', 'Attended on time/before HCP ready', 'Arrived late but was seen',
'Arrived late after HCP available but seen')
group by patients_initial.BrcId
order by count(e.CN_Doc_Id)
```

E.5 History of Bipolar

```
-- History of Bipolar extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with a history of bipolar
distinct d2.BrcId
, '1' as History_Bipolar
from
Diagnosis d2
inner join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
```

```
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
(d2.Primary_Diag like '%F31%' OR -- Bipolar
d2.Secondary_Diag_1 like '%F31%' OR
d2.Secondary_Diag_2 like '%F31%' OR
d2.Secondary_Diag_3 like '%F31%' OR
d2.Secondary_Diag_4 like '%F31%' OR
d2.Secondary_Diag_5 like '%F31%' OR
d2.Secondary_Diag_6 like '%F31%')
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
```

E.6 History of Manic

```
-- History of Manic extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with a history of manic
distinct d2.BrcId
, '1' as History_Manic
from
Diagnosis d2
inner join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case -- may remove those with diagnosis
after discharge
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in 
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
(d2.Primary_Diag like '%F30%' OR -- Manic
d2.Secondary_Diag_1 like '%F30%' OR
d2.Secondary_Diag_2 like '%F30%' OR
d2.Secondary_Diag_3 like '%F30%' OR
d2.Secondary_Diag_4 like '%F30%' OR
d2.Secondary_Diag_5 like '%F30%' OR
d2.Secondary_Diag_6 like '%F30%')
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
```

E.7 History of Organic

```
-- History of Organic extract

select -- adult patients with their earliest depression episode (since 2010) who have 
been discharged to Home or GP with a history of organic
distinct d2.BrcId
, '1' as History_Organic
from
Diagnosis d2
inner join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as 
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as 
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as 
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as 
Age
```

```
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
(d2.Primary_Diag like '%F0%' OR -- Organic
d2.Secondary_Diag_1 like '%F0%' OR
d2.Secondary_Diag_2 like '%F0%' OR
d2.Secondary_Diag_3 like '%F0%' OR
d2.Secondary_Diag_4 like '%F0%' OR
d2.Secondary_Diag_5 like '%F0%' OR
d2.Secondary_Diag_6 like '%F0%')
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
```

E.8 History of Schizophrenia

```
-- History of Schizophrenia extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with a history of Schizophrenia
distinct d2.BrcId
, '1' as History_Schizophrenia
from
Diagnosis d2
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
```

```
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
(d2.Primary_Diag like '%F2%' OR -- Schizophrenia
d2.Secondary_Diag_1 like '%F2%' OR
d2.Secondary_Diag_2 like '%F2%' OR
d2.Secondary_Diag_3 like '%F2%' OR
d2.Secondary_Diag_4 like '%F2%' OR
d2.Secondary_Diag_5 like '%F2%' OR
d2.Secondary_Diag_6 like '%F2%')
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
```

E.9 History of Substance

```
-- History of Substance extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with a history of substance
distinct d2.BrcId
, '1' as History_Substance
from
Diagnosis d2
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
```

```
                    , epr.Updated_Dttm as patient_updated_date
                    , epr.ethnicitycleaned
                    , d.Primary_Diag
                    , d.Diagnosis_Date
                    , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                    , d.Spell_Number
                    from
                    EPR_Form epr
                    inner join
                    Diagnosis d
                    on epr.BrcId = d.BrcId
                    where
                    (d.Primary_Diag like '%F32%' OR -- depressive episode
                    d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                    d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
(d2.Primary_Diag like '%F1%' OR -- Substance
d2.Secondary_Diag_1 like '%F1%' OR
d2.Secondary_Diag_2 like '%F1%' OR
d2.Secondary_Diag_3 like '%F1%' OR
d2.Secondary_Diag_4 like '%F1%' OR
d2.Secondary_Diag_5 like '%F1%' OR
d2.Secondary_Diag_6 like '%F1%')
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
```

E.10 HoNOS

```
-- HoNOS extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS scores closest to initial discharge
*
from
(
        select patients_initial.BrcId
        , h.Rating_Date as Honos_date
        , ROW_NUMBER() over (partition by h.BrcId order by h.Rating_Date desc) as
Honos_num --find last Honos
        , Agitated_Behaviour_Score_ID as Honos_Scale1
        , Self_Injury_Score_ID as Honos_Scale2
        , Problem_Drinking_Drugs_Score_ID as Honos_Scale3
        , Cognitive_Problems_Score_ID as Honos_Scale4
        , Physical_Illness_Score_ID as Honos_Scale5
        , Hallucinations_Score_ID as Honos_Scale6
        , Depressed_Mood_Score_ID as Honos_Scale7
        , Other_Mental_Problems_Type_ID as Honos_Scale8_Type
        , Other_Mental_Problems_Score_ID as Honos_Scale8
        , Relationship_Problems_Score_ID as Honos_Scale9
        , Daily_Living_Problems_Score_ID as Honos_Scale10
        , Living_Conditions_Problems_Score_ID as Honos_Scale11
        , Occupational_Problems_Score_ID as Honos_Scale12
        , Total as Honos_Total
        , Adjusted_Total as Honos_Adjusted_Total
        from
        HoNOS h
        inner join
        (
                select -- patients_initial
                dp.BrcId
                , dp.Primary_diag as Initial_primary_diagnosis
                , dp.Diagnosis_date as Initial_diagnosis_date
                , dp.Spell_number as Initial_spell_number
                , r.Accepted_date as Initial_accepted_date
                , r.Discharge_Date as Initial_discharge_date
                , r.Discharge_Destination_ID as Initial_discharge_destination
                , r.Referral_Date as Initial_referral_date
                , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
                , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
                , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
                , diagnosis_num_days_after_discharge = case
                when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
                datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
                else 0
                end
                , dp.Age
                , dp.Gender_ID
                , dp.Marital_Status_ID
                , dp.ethnicitycleaned
                , dp.patient_created_date
                , dp.patient_updated_date
                from
                (
                        select -- patients with depression diagnosis
                        epr.BrcId
                        , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS
Int) as Age
                        , epr.cleaneddateofbirth
                        , epr.Gender_ID
                        , epr.Marital_Status_ID
                        , epr.Create_Dttm as patient_created_date
                        , epr.Updated_Dttm as patient_updated_date
                        , epr.ethnicitycleaned
                        , d.Primary_Diag
                        , d.Diagnosis_Date
                        , ROW_NUMBER() over (partition by d.BrcId order by
d.Diagnosis_date) as Diagnosis_num
                        , d.Spell_Number
                        from
                        EPR_Form epr
                        inner join
                        Diagnosis d
                        on epr.BrcId = d.BrcId
                        where
                        (d.Primary_Diag like '%F32%' OR -- depressive episode
                        d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                        d.Diagnosis_Date is not null -- must have a diagnosis date
                 )
                 dp
                 inner join
                 Referral r
                 on
                 (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
                 where
                 (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
                 r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and
   later
                 r.Accepted_date is not null AND -- must have been accepted to the trust
                 r.Discharge_Date is not null AND -- must have been discharged
                 r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND
   -- in remission
                 dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral
   dates
                 dp.Diagnosis_Date <= r.Discharge_Date AND
                 dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
          )
          patients_initial
          on (h.BrcId = patients_initial.BrcId
          AND h.Spell_Number = patients_initial.Initial_spell_number)
          where h.Rating_Date >= patients_initial.Initial_accepted_date AND
          h.Rating_Date <= patients_initial.Initial_discharge_date
   )
   Honos
   where
   Honos_num = 1
```

E.11 HoNOS abi

```
-- HoNOS_abi extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS abi scores closest to initial discharge
*
from
(
        select patients_initial.BrcId
        , h.Rating_Date as Honos_date
        , ROW_NUMBER() over (partition by h.BrcId order by h.Rating_Date desc) as
Honos_num -- find last Honos
        , Agitated_Behaviour_Score_ID as Honos_Scale1
        , Self_Injury_Score_ID as Honos_Scale2
        , Problem_Drinking_Drugs_Score_ID as Honos_Scale3
        , Cognitive_Problems_Score_ID as Honos_Scale4
        , Physical_Illness_Score_ID as Honos_Scale5
        , Hallucinations_Score_ID as Honos_Scale6
        , Depressed_Mood_Score_ID as Honos_Scale7
        , Other_Mental_Problems_Type_ID as Honos_Scale8_Type
        , Other_Mental_Problems_Score_ID as Honos_Scale8
        , Relationship_Problems_Score_ID as Honos_Scale9
        , Daily_Living_Problems_Score_ID as Honos_Scale10
        , Living_Conditions_Problems_Score_ID as Honos_Scale11
        , Activities_Score_ID as Honos_Scale12
        , Total as Honos_Total
        , Adjusted_Total as Honos_Adjusted_Total
        from
        HoNOSabi h
        inner join
        (
               select -- patients_initial
               dp.BrcId
               , dp.Primary_diag as Initial_primary_diagnosis
               , dp.Diagnosis_date as Initial_diagnosis_date
               , dp.Spell_number as Initial_spell_number
               , r.Accepted_date as Initial_accepted_date
               , r.Discharge_Date as Initial_discharge_date
               , r.Discharge_Destination_ID as Initial_discharge_destination
               , r.Referral_Date as Initial_referral_date
               , datediff(day, r.Accepted_date, r.Discharge_Date) as
 num_days_in_initial_episode
               , datediff(day, r.Referral_Date, r.Accepted_date) as
 num_days_to_initial_accepted
               , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
 num_days_to_initial_diagnosis
               , diagnosis_num_days_after_discharge = case
               when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
               datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
               else 0
               end
               , dp.Age
               , dp.Gender_ID
               , dp.Marital_Status_ID
               , dp.ethnicitycleaned
               , dp.patient_created_date
               , dp.patient_updated_date
               from
               (
                      select -- patients with depression diagnosis
                      epr.BrcId
                      , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS
 Int) as Age
                      , epr.cleaneddateofbirth
                      , epr.Gender_ID
                      , epr.Marital_Status_ID
                      , epr.Create_Dttm as patient_created_date
                      , epr.Updated_Dttm as patient_updated_date
                      , epr.ethnicitycleaned
                      , d.Primary_Diag
                      , d.Diagnosis_Date
                      , ROW_NUMBER() over (partition by d.BrcId order by
 d.Diagnosis_date) as Diagnosis_num
                      , d.Spell_Number
                      from
                      EPR_Form epr
                      inner join
                      Diagnosis d
                      on epr.BrcId = d.BrcId
                      where
                      (d.Primary_Diag like '%F32%' OR -- depressive episode
                      d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                      d.Diagnosis_Date is not null -- must have a diagnosis date
               )
               dp
               inner join
               Referral r
               on
               (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
               where
               (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
               r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and
 later
```

```
                r.Accepted_date is not null AND -- must have been accepted to the trust
                r.Discharge_Date is not null AND -- must have been discharged
                r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND
-- in remission
                dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral
dates
                dp.Diagnosis_Date <= r.Discharge_Date AND
                dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
        )
        patients_initial
        on (h.BrcId = patients_initial.BrcId
        AND h.Spell_Number = patients_initial.Initial_spell_number)
        where h.Rating_Date >= patients_initial.Initial_accepted_date AND
        h.Rating_Date <= patients_initial.Initial_discharge_date
)
Honos
where
Honos_num = 1
```

E.12 HoNOS Secure

```
-- HoNOS Secure extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS Secure scores closest to initial discharge
*
from
(
        select patients_initial.BrcId
        , h.Rating_Date as Honos_date
        , ROW_NUMBER() over (partition by h.BrcId order by h.Rating_Date desc) as
Honos_num --find last Honos
        , Agitated_Behaviour_Score_ID as Honos_Scale1
        , Self_Injury_Score_ID as Honos_Scale2
        , Problem_Drinking_Drugs_Score_ID as Honos_Scale3
        , Cognitive_Problems_Score_ID as Honos_Scale4
        , Physical_Illness_Score_ID as Honos_Scale5
        , Hallucinations_Score_ID as Honos_Scale6
        , Depressed_Mood_Score_ID as Honos_Scale7
        , Other_Mental_Problems_Type_ID as Honos_Scale8_Type
        , Other_Mental_Problems_Score_ID as Honos_Scale8
        , Relationship_Problems_Score_ID as Honos_Scale9
        , Daily_Living_Problems_Score_ID as Honos_Scale10
        , Living_Conditions_Problems_Score_ID as Honos_Scale11
        , Occupational_Problems_Score_ID as Honos_Scale12
        , Total as Honos_Total
        , Adjusted_Total as Honos_Adjusted_Total
        from
        HoNOS_Secure h
        inner join
        (
                select -- patients_initial
                dp.BrcId
                , dp.Primary_diag as Initial_primary_diagnosis
                , dp.Diagnosis_date as Initial_diagnosis_date
                , dp.Spell_number as Initial_spell_number
                , r.Accepted_date as Initial_accepted_date
                , r.Discharge_Date as Initial_discharge_date
                , r.Discharge_Destination_ID as Initial_discharge_destination
                , r.Referral_Date as Initial_referral_date
                , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
                , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
```

```
                , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
                , diagnosis_num_days_after_discharge = case
                when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
                datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
                else 0
                end
                , dp.Age
                , dp.Gender_ID
                , dp.Marital_Status_ID
                , dp.ethnicitycleaned
                , dp.patient_created_date
                , dp.patient_updated_date
                from
                (
                        select -- patients with depression diagnosis
                        epr.BrcId
                        , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS
Int) as Age
                        , epr.cleaneddateofbirth
                        , epr.Gender_ID
                        , epr.Marital_Status_ID
                        , epr.Create_Dttm as patient_created_date
                        , epr.Updated_Dttm as patient_updated_date
                        , epr.ethnicitycleaned
                        , d.Primary_Diag
                        , d.Diagnosis_Date
                        , ROW_NUMBER() over (partition by d.BrcId order by
d.Diagnosis_date) as Diagnosis_num
                        , d.Spell_Number
                        from
                        EPR_Form epr
                        inner join
                        Diagnosis d
                        on epr.BrcId = d.BrcId
                        where
                        (d.Primary_Diag like '%F32%' OR -- depressive episode
                        d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                        d.Diagnosis_Date is not null -- must have a diagnosis date
                )
                dp
                inner join
                Referral r
                on
                (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
                where
                (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
                r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and
later
                r.Accepted_date is not null AND -- must have been accepted to the trust
                r.Discharge_Date is not null AND -- must have been discharged
                r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND
-- in remission
                dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral
dates
                dp.Diagnosis_Date <= r.Discharge_Date AND
                dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
        )
        patients_initial
        on (h.BrcId = patients_initial.BrcId
        AND h.Spell_Number = patients_initial.Initial_spell_number)
        where h.Rating_Date >= patients_initial.Initial_accepted_date AND
```

```
	h.Rating_Date <= patients_initial.Initial_discharge_date
)
Honos
where
Honos_num = 1
```

E.13 HoNOS 65+

```
-- HoNOS 65+ extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS 65+ scores closest to initial discharge
*
from
(
	select patients_initial.BrcId
	, h.Rating_Date as Honos_date
	, ROW_NUMBER() over (partition by h.BrcId order by h.Rating_Date desc) as
Honos_num --find last Honos
	, Agitated_Behaviour_Score_ID as Honos_Scale1
	, Self_Injury_Score_ID as Honos_Scale2
	, Problem_Drinking_Drugs_Score_ID as Honos_Scale3
	, Cognitive_Problems_Score_ID as Honos_Scale4
	, Physical_Illness_Score_ID as Honos_Scale5
	, Hallucinations_Score_ID as Honos_Scale6
	, Depressed_Mood_Score_ID as Honos_Scale7
	, Other_Mental_Problems_Type_ID as Honos_Scale8_Type
	, Other_Mental_Problems_Score_ID as Honos_Scale8
	, Relationship_Problems_Score_ID as Honos_Scale9
	, Daily_Living_Problems_Score_ID as Honos_Scale10
	, Living_Conditions_Problems_Score_ID as Honos_Scale11
	, Occupational_Problems_Score_ID as Honos_Scale12
	, Total as Honos_Total
	, Adjusted_Total as Honos_Adjusted_Total
	from
	HoNOS65 h
	inner join
	(
		select -- patients_initial
		dp.BrcId
		, dp.Primary_diag as Initial_primary_diagnosis
		, dp.Diagnosis_date as Initial_diagnosis_date
		, dp.Spell_number as Initial_spell_number
		, r.Accepted_date as Initial_accepted_date
		, r.Discharge_Date as Initial_discharge_date
		, r.Discharge_Destination_ID as Initial_discharge_destination
		, r.Referral_Date as Initial_referral_date
		, datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
		, datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
		, datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
		, diagnosis_num_days_after_discharge = case
		when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
		datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
		else 0
		end
		, dp.Age
		, dp.Gender_ID
		, dp.Marital_Status_ID
		, dp.ethnicitycleaned
		, dp.patient_created_date
		, dp.patient_updated_date
```

```
                from
                (
                        select -- patients with depression diagnosis
                        epr.BrcId
                        , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS
Int) as Age
                        , epr.cleaneddateofbirth
                        , epr.Gender_ID
                        , epr.Marital_Status_ID
                        , epr.Create_Dttm as patient_created_date
                        , epr.Updated_Dttm as patient_updated_date
                        , epr.ethnicitycleaned
                        , d.Primary_Diag
                        , d.Diagnosis_Date
                        , ROW_NUMBER() over (partition by d.BrcId order by
d.Diagnosis_date) as Diagnosis_num
                        , d.Spell_Number
                        from
                        EPR_Form epr
                        inner join
                        Diagnosis d
                        on epr.BrcId = d.BrcId
                        where
                        (d.Primary_Diag like '%F32%' OR -- depressive episode
                        d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                        d.Diagnosis_Date is not null -- must have a diagnosis date
                )
                dp
                inner join
                Referral r
                on
                (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
                where
                (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
                r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and
later
                r.Accepted_date is not null AND -- must have been accepted to the trust
                r.Discharge_Date is not null AND -- must have been discharged
                r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND
-- in remission
                dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral
dates
                dp.Diagnosis_Date <= r.Discharge_Date AND
                dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
        )
        patients_initial
        on (h.BrcId = patients_initial.BrcId
        AND h.Spell_Number = patients_initial.Initial_spell_number)
        where h.Rating_Date >= patients_initial.Initial_accepted_date AND
        h.Rating_Date <= patients_initial.Initial_discharge_date
)
Honos
where
Honos_num = 1
```

E.14 Inpatient Days

```
-- Inpatient Days extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with number Inpatient days last 6 months
distinct i.BrcId
, sum(datediff(day, i.Admission_Date, i.Discharge_date)) as Num_Inpatient_Days
from
```

```
Inpatient_episode i
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on
i.BrcId = patients_initial.BrcId
where
(i.Admission_Date >= (Initial_discharge_date - 182.625)) -- last 6 months
AND (i.Admission_Date <= Initial_discharge_date )
AND i.Discharge_Date is not null
AND i.Discharge_Date >= i.Admission_Date
group by i.BrcId
```

E.15 Medication

```
-- Medication extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with Medication last 12 months
distinct patients_initial.BrcId, m.Medicine_ID, m.Medication_Start_Date,
m.Medication_End_Date, med_list.gazetteer, med_list.core_term
, (case when lower(med_list.gazetteer) in ('agomelatine', 'valdoxan') then
'Agomelatine'
        when lower(med_list.gazetteer) in ('amitriptyline', 'amitriptyline
hydrochloride') then 'Amitriptyline'
        when lower(med_list.gazetteer) in ('citalopram', 'cipramil', 'citalopram
hydrobromide', 'citalopram hydrochloride') then 'Citalopram'
        when lower(med_list.gazetteer) in ('clomipramine', 'anafranil') then
'Clomipramine'
        when lower(med_list.gazetteer) in ('dosulepin', 'prothiaden', 'dothiepin') then
'Dosulepin'
        when lower(med_list.gazetteer) in ('sinepin', 'Doxepin') then 'Doxepin'
        when lower(med_list.gazetteer) in ('cymbalta', 'duloxetine', 'duciltia') then
'Duloxetine'
        when lower(med_list.gazetteer) in ('cipralex', 'escitalopram') then
'Escitalopram'
        when lower(med_list.gazetteer) in ('oxactin', 'fluoxetine', 'prozac', 'prozep',
'olena') then 'Fluoxetine'
        when lower(med_list.gazetteer) in ('faverin', 'fluvoxamine') then 'Fluvoxamine'
        when lower(med_list.gazetteer) in ('tofranil', 'imipramine') then 'Imipramine'
        when lower(med_list.gazetteer) in ('isocarboxazid') then 'Isocarboxazid'
        when lower(med_list.gazetteer) in ('lomont', 'lofepramine') then 'Lofepramine'
        when lower(med_list.gazetteer) in ('maprotiline') then 'Maprotiline'
        when lower(med_list.gazetteer) in ('mianserin') then 'Mianserin'
        when lower(med_list.gazetteer) in ('zispin', 'mirtazapine', 'mirtazepine') then
'Mirtazapine'
        when lower(med_list.gazetteer) in ('manerix', 'moclobemide') then 'Moclobemide'
        when lower(med_list.gazetteer) in ('nefazodone') then 'Nefazodone'
        when lower(med_list.gazetteer) in ('allegron', 'nortriptyline') then
'Nortriptyline'
        when lower(med_list.gazetteer) in ('oxitriptan') then 'Oxitriptan'
        when lower(med_list.gazetteer) in ('seroxat', 'paroxetine') then 'Paroxetine'
        when lower(med_list.gazetteer) in ('nardil', 'phenelzine') then 'Phenelzine'
        when lower(med_list.gazetteer) in ('edronax', 'reboxetine') then 'Reboxetine'
        when lower(med_list.gazetteer) in ('sertraline', 'lustral') then 'Sertraline'
```

```
       when lower(med_list.gazetteer) in ('parnate', 'tranylcypromine') then
'Tranylcypromine'
       when lower(med_list.gazetteer) in ('molipaxin', 'trazodone', 'trazadone') then
'Trazodone'
       when lower(med_list.gazetteer) in ('surmontil', 'trimipramine') then
'Trimipramine'
       when lower(med_list.gazetteer) in ('triptafen') then 'Triptafen'
       when lower(med_list.gazetteer) in ('tryptophan', 'pptimax') then 'Tryptophan'
       when lower(med_list.gazetteer) in ('depefex xl', 'depefex', 'efexor xl',
'effexor', 'efexor', 'venlafaxine', 'venlafaxine xl', 'foraven xl',
       'politid xl', 'politid', 'sunveniz xl', 'sunveniz', 'tonpular xl', 'tonpular',
'venadex xl', 'venaxx xl', 'venaxx', 'venlalic xl', 'venlalic',
       'viepax', 'alventa', 'amphero', 'rodomel', 'vencarm', 'venlablue', 'venladex',
'venlaneo', 'vensir', 'winfex') then 'Venlafaxine'
       when lower(med_list.gazetteer) in ('brintellix', 'vortioxetine') then
'Vortioxetine' end) as Antidepressant_generic

, (case when lower(med_list.gazetteer) in ('agomelatine', 'valdoxan') then 1 else 0
end) as Agomelatine
, (case when lower(med_list.gazetteer) in ('amitriptyline', 'amitriptyline
hydrochloride') then 1 else 0 end) as Amitriptyline
, (case when lower(med_list.gazetteer) in ('citalopram', 'cipramil', 'citalopram
hydrobromide', 'citalopram hydrochloride') then 1 else 0 end) as Citalopram
, (case when lower(med_list.gazetteer) in ('clomipramine', 'anafranil') then 1 else 0
end) as Clomipramine
, (case when lower(med_list.gazetteer) in ('dosulepin', 'prothiaden', 'dothiepin')
then 1 else 0 end) as Dosulepin
, (case when lower(med_list.gazetteer) in ('sinepin', 'Doxepin') then 1 else 0 end) as
Doxepin
, (case when lower(med_list.gazetteer) in ('cymbalta', 'duloxetine', 'duciltia') then
1 else 0 end) as Duloxetine
, (case when lower(med_list.gazetteer) in ('cipralex', 'escitalopram') then 1 else 0
end) as Escitalopram
, (case when lower(med_list.gazetteer) in ('oxactin', 'fluoxetine', 'prozac',
'prozep', 'olena') then 1 else 0 end) as Fluoxetine
, (case when lower(med_list.gazetteer) in ('faverin', 'fluvoxamine') then 1 else 0
end) as Fluvoxamine
, (case when lower(med_list.gazetteer) in ('tofranil', 'imipramine') then 1 else 0
end) as Imipramine
, (case when lower(med_list.gazetteer) in ('isocarboxazid') then 1 else 0 end) as
Isocarboxazid
, (case when lower(med_list.gazetteer) in ('lomont', 'lofepramine') then 1 else 0 end)
as Lofepramine
, (case when lower(med_list.gazetteer) in ('maprotiline') then 1 else 0 end) as
Maprotiline
, (case when lower(med_list.gazetteer) in ('mianserin') then 1 else 0 end) as
Mianserin
, (case when lower(med_list.gazetteer) in ('zispin', 'mirtazapine', 'mirtazepine')
then 1 else 0 end) as Mirtazapine
, (case when lower(med_list.gazetteer) in ('manerix', 'moclobemide') then 1 else 0
end) as Moclobemide
, (case when lower(med_list.gazetteer) in ('nefazodone') then 1 else 0 end) as
Nefazodone
, (case when lower(med_list.gazetteer) in ('allegron', 'nortriptyline') then 1 else 0
end) as Nortriptyline
, (case when lower(med_list.gazetteer) in ('oxitriptan') then 1 else 0 end) as
Oxitriptan
, (case when lower(med_list.gazetteer) in ('seroxat', 'paroxetine') then 1 else 0 end)
as Paroxetine
, (case when lower(med_list.gazetteer) in ('nardil', 'phenelzine') then 1 else 0 end)
as Phenelzine
```

```
, (case when lower(med_list.gazetteer) in ('edronax', 'reboxetine') then 1 else 0 end)
as Reboxetine
, (case when lower(med_list.gazetteer) in ('sertraline', 'lustral') then 1 else 0 end)
as Sertraline
, (case when lower(med_list.gazetteer) in ('parnate', 'tranylcypromine') then 1 else 0
end) as Tranylcypromine
, (case when lower(med_list.gazetteer) in ('molipaxin', 'trazodone', 'trazadone') then
1 else 0 end) as Trazodone
, (case when lower(med_list.gazetteer) in ('surmontil', 'trimipramine') then 1 else 0
end) as Trimipramine
, (case when lower(med_list.gazetteer) in ('triptafen') then 1 else 0 end) as
Triptafen
, (case when lower(med_list.gazetteer) in ('tryptophan', 'pptimax') then 1 else 0 end)
as Tryptophan
, (case when lower(med_list.gazetteer) in ('depefex xl', 'depefex', 'efexor xl',
'effexor', 'efexor', 'venlafaxine', 'venlafaxine xl', 'foraven xl',
       'politid xl', 'politid', 'sunveniz xl', 'sunveniz', 'tonpular xl', 'tonpular',
'venadex xl', 'venaxx xl', 'venaxx', 'venlalic xl', 'venlalic',
       'viepax', 'alventa', 'amphero', 'rodomel', 'vencarm', 'venlablue', 'venladex',
'venlaneo', 'vensir', 'winfex') then 1 else 0 end) as Venlafaxine
, (case when lower(med_list.gazetteer) in ('brintellix', 'vortioxetine') then 1 else 0
end) as Vortioxetine
from SQLCRIS_Common.dbo.tbl_medication_gazetteer_source_combined_final as med_list
inner join
SQLCRIS_Common.dbo.tbl_structured_meds_gazetteer_lookup as med_map
on med_list.gazetteer = med_map.gazetteer
inner join
Medication m
on m.Medicine_ID = med_map.Medicine_ID
right outer join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on patients_initial.BrcId = m.BrcId
where med_list.bnf_code like '0403%' --anti-depressants
AND (m.medication_start_date >= (Initial_discharge_date - 365.25) OR
m.medication_start_date is null) -- in the 12 months before discharge
AND (m.medication_start_date <= Initial_discharge_date OR m.medication_start_date is
null)
AND med_list.gazetteer <> 'Flupentixol' -- remove anti-psychotic
```

E.16 Medication (one hot encoded)

```
-- Medication (one hot encoded) extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with Medication last 12 months
distinct patients_initial.BrcId
, MAX(case when lower(med_list.gazetteer) in ('agomelatine', 'valdoxan') then 1 else 0
end) as Agomelatine
, MAX(case when lower(med_list.gazetteer) in ('amitriptyline', 'amitriptyline
hydrochloride') then 1 else 0 end) as Amitriptyline
, MAX(case when lower(med_list.gazetteer) in ('citalopram', 'cipramil', 'citalopram
hydrobromide', 'citalopram hydrochloride') then 1 else 0 end) as Citalopram
, MAX(case when lower(med_list.gazetteer) in ('clomipramine', 'anafranil') then 1 else
0 end) as Clomipramine
```

```
, MAX(case when lower(med_list.gazetteer) in ('dosulepin', 'prothiaden', 'dothiepin')
then 1 else 0 end) as Dosulepin
, MAX(case when lower(med_list.gazetteer) in ('sinepin', 'Doxepin') then 1 else 0 end)
as Doxepin
, MAX(case when lower(med_list.gazetteer) in ('cymbalta', 'duloxetine', 'duciltia')
then 1 else 0 end) as Duloxetine
, MAX(case when lower(med_list.gazetteer) in ('cipralex', 'escitalopram') then 1 else
0 end) as Escitalopram
, MAX(case when lower(med_list.gazetteer) in ('oxactin', 'fluoxetine', 'prozac',
'prozep', 'olena') then 1 else 0 end) as Fluoxetine
, MAX(case when lower(med_list.gazetteer) in ('faverin', 'fluvoxamine') then 1 else 0
end) as Fluvoxamine
, MAX(case when lower(med_list.gazetteer) in ('tofranil', 'imipramine') then 1 else 0
end) as Imipramine
, MAX(case when lower(med_list.gazetteer) in ('isocarboxazid') then 1 else 0 end) as
Isocarboxazid
, MAX(case when lower(med_list.gazetteer) in ('lomont', 'lofepramine') then 1 else 0
end) as Lofepramine
, MAX(case when lower(med_list.gazetteer) in ('maprotiline') then 1 else 0 end) as
Maprotiline
, MAX(case when lower(med_list.gazetteer) in ('mianserin') then 1 else 0 end) as
Mianserin
, MAX(case when lower(med_list.gazetteer) in ('zispin', 'mirtazapine', 'mirtazepine')
then 1 else 0 end) as Mirtazapine
, MAX(case when lower(med_list.gazetteer) in ('manerix', 'moclobemide') then 1 else 0
end) as Moclobemide
, MAX(case when lower(med_list.gazetteer) in ('nefazodone') then 1 else 0 end) as
Nefazodone
, MAX(case when lower(med_list.gazetteer) in ('allegron', 'nortriptyline') then 1 else
0 end) as Nortriptyline
, MAX(case when lower(med_list.gazetteer) in ('oxitriptan') then 1 else 0 end) as
Oxitriptan
, MAX(case when lower(med_list.gazetteer) in ('seroxat', 'paroxetine') then 1 else 0
end) as Paroxetine
, MAX(case when lower(med_list.gazetteer) in ('nardil', 'phenelzine') then 1 else 0
end) as Phenelzine
, MAX(case when lower(med_list.gazetteer) in ('edronax', 'reboxetine') then 1 else 0
end) as Reboxetine
, MAX(case when lower(med_list.gazetteer) in ('sertraline', 'lustral') then 1 else 0
end) as Sertraline
, MAX(case when lower(med_list.gazetteer) in ('parnate', 'tranylcypromine') then 1
else 0 end) as Tranylcypromine
, MAX(case when lower(med_list.gazetteer) in ('molipaxin', 'trazodone', 'trazadone')
then 1 else 0 end) as Trazodone
, MAX(case when lower(med_list.gazetteer) in ('surmontil', 'trimipramine') then 1 else
0 end) as Trimipramine
, MAX(case when lower(med_list.gazetteer) in ('triptafen') then 1 else 0 end) as
Triptafen
, MAX(case when lower(med_list.gazetteer) in ('tryptophan', 'pptimax') then 1 else 0
end) as Tryptophan
, MAX(case when lower(med_list.gazetteer) in ('depefex xl', 'depefex', 'efexor xl',
'effexor', 'efexor', 'venlafaxine', 'venlafaxine xl', 'foraven xl',
       'politid xl', 'politid', 'sunveniz xl', 'sunveniz', 'tonpular xl', 'tonpular',
'venadex xl', 'venaxx xl', 'venaxx', 'venlalic xl', 'venlalic',
       'viepax', 'alventa', 'amphero', 'rodomel', 'vencarm', 'venlablue', 'venladex',
'venlaneo', 'vensir', 'winfex') then 1 else 0 end) as Venlafaxine
, MAX(case when lower(med_list.gazetteer) in ('brintellix', 'vortioxetine') then 1
else 0 end) as Vortioxetine
from SQLCRIS_Common.dbo.tbl_medication_gazetteer_source_combined_final as med_list
inner join
SQLCRIS_Common.dbo.tbl_structured_meds_gazetteer_lookup as med_map
on med_list.gazetteer = med_map.gazetteer
```

```
inner join
Medication m
on m.Medicine_ID = med_map.Medicine_ID
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on patients_initial.BrcId = m.BrcId
where med_list.bnf_code like '0403%' --anti-depressants
AND (m.medication_start_date >= (Initial_discharge_date - 365.25) OR
m.medication_start_date is null) -- in the 12 months before discharge
AND (m.medication_start_date <= Initial_discharge_date OR m.medication_start_date is
null)
AND med_list.gazetteer <> 'Flupentixol' -- remove anti-psychotic
group by patients_initial.BrcId
```

E.17 Num DNAs Last 6 Months

```
-- Num DNAs Last 6 Months extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with number of Did Not Attend events last 6 months
patients_initial.BrcId, count(e.CN_Doc_Id) as Num_DNAs_6M
from
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
```

```
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
left outer join
Event e
on
patients_initial.BrcId = e.BrcId
where
(e.start_date >= (Initial_discharge_date - 182.625) ) -- event occurs in the 6 months
before discharge
AND (e.Start_Date <= Initial_discharge_date )
AND e.Event_Outcome_ID in
('DNA by Client', 'DNA', 'Did not attend', 'DNA by Trust', 'DNA by Clinician')
group by patients_initial.BrcId
order by count(e.CN_Doc_Id)
```

E.18 Patients Initial Episode

```
-- Patients Initial Episode extract

select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged to Home or GP
dp.BrcId
, dp.Primary_diag as Initial_primary_diagnosis
, LTRIM(RTRIM(SUBSTRING(REPLACE(dp.Primary_Diag,'-',''), 1, 5))) as
Trimmed_Primary_Diag
, (case when (LTRIM(RTRIM(SUBSTRING(REPLACE(dp.Primary_Diag,'-',''), 1, 4)))) = 'F33 '
then 1 else 0 end) as F33
```

```
, (case when (dp.Primary_Diag) like '%F33.0%' then 1 else 0 end) as F330
, (case when (dp.Primary_Diag) like '%F33.1%' then 1 else 0 end) as F331
, (case when (dp.Primary_Diag) like '%F33.2%' then 1 else 0 end) as F332
, (case when (dp.Primary_Diag) like '%F33.3%' then 1 else 0 end) as F333
, (case when (dp.Primary_Diag) like '%F33.4%' then 1 else 0 end) as F334
, (case when (dp.Primary_Diag) like '%F33.8%' then 1 else 0 end) as F338
, (case when (dp.Primary_Diag) like '%F33.9%' then 1 else 0 end) as F339
, (case when (LTRIM(RTRIM(SUBSTRING(REPLACE(dp.Primary_Diag,'-',''), 1, 4)))) = 'F32 '
then 1 else 0 end) as F32
, (case when (dp.Primary_Diag) like '%F32.0%' then 1 else 0 end) as F320
, (case when (dp.Primary_Diag) like '%F32.1%' then 1 else 0 end) as F321
, (case when (dp.Primary_Diag) like '%F32.2%' then 1 else 0 end) as F322
, (case when (dp.Primary_Diag) like '%F32.3%' then 1 else 0 end) as F323
, (case when (dp.Primary_Diag) like '%F32.8%' then 1 else 0 end) as F328
, (case when (dp.Primary_Diag) like '%F32.9%' then 1 else 0 end) as F329
, dp.Diagnosis_date as Initial_diagnosis_date
, dp.Spell_number as Initial_spell_number
, r.Accepted_date as Initial_accepted_date
, r.Discharge_Date as Initial_discharge_date
, r.Discharge_Destination_ID as Initial_discharge_destination
, (case when (r.Discharge_Destination_ID) = 'GP' then 1 else 0 end) as GP
, (case when (r.Discharge_Destination_ID) = 'Home - No Follow Up Required' then 1 else
0 end) as Home
, r.Referral_Date as Initial_referral_date
, datediff(day, r.Accepted_date, r.Discharge_Date) as num_days_in_initial_episode
, datediff(day, r.Referral_Date, r.Accepted_date) as num_days_to_initial_accepted
, datediff(day, r.Accepted_date, dp.Diagnosis_Date) as num_days_to_initial_diagnosis
, dp.Age
, dp.Gender_ID
, case when dp.Gender_ID in ('Female') then 1 else 0 end as Female
, case when dp.Gender_ID in ('Male') then 1 else 0 end as Male
, case when dp.Gender_ID in ('Other', 'Not Known', 'Not Specified') or dp.Gender_ID is
null then 1 else 0 end as OtherNotStated
, dp.Marital_Status_ID
, case when dp.Marital_Status_ID in ('Single') then 1 else 0 end as Single
, case when dp.Marital_Status_ID in ('Cohabiting') then 1 else 0 end as Cohabiting
, case when dp.Marital_Status_ID in ('Married/Civil Partner', 'Married') then 1 else 0
end as Married
, case when dp.Marital_Status_ID in ('Divorced/Civil Partnership Dissolved',
'Divorced') then 1 else 0 end as Divorced
, case when dp.Marital_Status_ID in ('Separated') then 1 else 0 end as Separated
, case when dp.Marital_Status_ID in ('Widowed/Surviving Civil Partner', 'Widowed')
then 1 else 0 end as Widowed
, case when dp.Marital_Status_ID in ('Not Known', 'Not Disclosed') or
dp.Marital_Status_ID is null then 1 else 0 end as OtherNotStated
, dp.ethnicitycleaned
, case when dp.ethnicitycleaned in ('British (A)') then 1 else 0 end as British
, case when dp.ethnicitycleaned in ('Any other ethnic group (S)', 'Not Stated (Z)') or
dp.ethnicitycleaned is null then 1 else 0 end as OtherNotStated
, case when dp.ethnicitycleaned in ('Any other white background (C)') then 1 else 0
end as OtherWhite
, case when dp.ethnicitycleaned in ('African (N)') then 1 else 0 end as African
, case when dp.ethnicitycleaned in ('Any other black background (P)') then 1 else 0
end as OtherBlack
, case when dp.ethnicitycleaned in ('Caribbean (M)') then 1 else 0 end as Caribbean
, case when dp.ethnicitycleaned in ('Any other Asian background (L)') then 1 else 0
end as OtherAsian
, case when dp.ethnicitycleaned in ('Irish (B)') then 1 else 0 end as Irish
, case when dp.ethnicitycleaned in ('Indian (H)') then 1 else 0 end as Indian
, case when dp.ethnicitycleaned in ('Pakistani (J)') then 1 else 0 end as Pakistani
, case when dp.ethnicitycleaned in ('White and black Caribbean (D)') then 1 else 0 end
as WhiteBlackCaribbean
```

```
, case when dp.ethnicitycleaned in ('Any other mixed background (G)') then 1 else 0
end as OtherMixed
, case when dp.ethnicitycleaned in ('Bangladeshi (K)') then 1 else 0 end as
Bangladeshi
, case when dp.ethnicitycleaned in ('Chinese (R)') then 1 else 0 end as Chinese
, case when dp.ethnicitycleaned in ('White and Black African (E)') then 1 else 0 end
as WhiteBlackAfrican
, case when dp.ethnicitycleaned in ('White and Asian (F)') then 1 else 0 end as
WhiteAsian
, dp.patient_created_date
, dp.patient_updated_date
from
(
       select -- patients with depression diagnosis
       epr.BrcId
       , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as Age
       , epr.cleaneddateofbirth
       , epr.Gender_ID
       , epr.Marital_Status_ID
       , epr.Create_Dttm as patient_created_date
       , epr.Updated_Dttm as patient_updated_date
       , epr.ethnicitycleaned
       , d.Primary_Diag
       , d.Diagnosis_Date
       , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
       , d.Spell_Number
       from
       EPR_Form epr
       inner join
       Diagnosis d
       on epr.BrcId = d.BrcId
       where
       (d.Primary_Diag like '%F32%' OR -- depressive episode
       d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
       d.Diagnosis_Date is not null -- must have a diagnosis date
)
dp
inner join
Referral r
on
(r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
where
(r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
r.Accepted_date is not null AND -- must have been accepted to the trust
r.Discharge_Date is not null AND -- must have been discharged
r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
dp.Diagnosis_Date <= r.Discharge_Date AND
dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
order by r.BrcId
```

E.19 Patients Relapse

```
-- Patients Relapse

select -- adult patients and their next depression referral following discharge from
their initial depression referral where initial referral is since 2010
*
from
(
select -- patient_relapse
```

```
distinct
patients_initial.BrcId
, patients_initial.Initial_discharge_date
, r2.Accepted_Date as Rereferral_accepted_date
, r2.Discharge_Date as Rereferral_discharge_date
, datediff(day, patients_initial.Initial_discharge_date, r2.Accepted_Date) as
Num_days_in_remission
, datediff(day, r2.Accepted_Date, r2.Discharge_Date) as Num_days_in_relapse
, 1 as relapse_All
, (case when datediff(day, patients_initial.Initial_discharge_date, r2.Accepted_Date)
<= 365.24 then 1 else 0 end) as relapse_in_12M
,  (case when datediff(day, patients_initial.Initial_discharge_date, r2.Accepted_Date)
<= 2*365.24 then 1 else 0 end) as relapse_in_24M
,  (case when datediff(day, patients_initial.Initial_discharge_date, r2.Accepted_Date)
<= 3*365.24 then 1 else 0 end) as relapse_in_36M
, ROW_NUMBER() over (partition by r2.BrcId order by r2.Spell_Number) as spell_num
from
(
              select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
inner join
Referral r2
on
r2.BrcId = patients_initial.BrcId
inner join
Diagnosis d2
on
(r2.BrcId = d2.BrcId AND r2.Spell_Number = d2.Spell_number)
where
r2.Accepted_date >= '01-jan-2010' AND -- only consider data 2010 and later
r2.Spell_Number > patients_initial.Initial_spell_number AND
(d2.Primary_Diag like '%F32%' OR -- depressive episode
d2.Primary_Diag like '%F33%') AND -- recurrent depressive episode
(d2.Diagnosis_Date > Initial_diagnosis_date) AND
(r2.Accepted_Date > Initial_discharge_date) AND
r2.Discharge_Date is not null AND
d2.Diagnosis_Date >= r2.Accepted_Date AND -- Diagnosis is within referral dates
d2.Diagnosis_Date <= r2.Discharge_Date
group by patients_initial.BrcId
, r2.Accepted_Date
, r2.Discharge_Date
, r2.Spell_Number
, patients_initial.Initial_discharge_date
, r2.BrcId
)
as relapse_All
where spell_num = 1 --only take first relapse episode
```

E.20 Psychotic Symptoms

```
-- Psychotic Symptoms extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with a history of Psychotic symptoms in last 12 months
distinct d2.BrcId,
'1' as Psychotic_symptoms_12M
from
Diagnosis d2
```

```
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
```

```
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
((d2.Primary_Diag like '%F32.3%' OR -- Severe with Psychotic symptoms
d2.Primary_Diag like '%F33.3%') OR -- Recurrant severe with Psychotic symptoms
(d2.Secondary_Diag_1 like '%F32.3%' OR
d2.Secondary_Diag_1 like '%F33.3%') OR
(d2.Secondary_Diag_2 like '%F32.3%' OR
d2.Secondary_Diag_2 like '%F33.3%') OR
(d2.Secondary_Diag_3 like '%F32.3%' OR
d2.Secondary_Diag_3 like '%F33.3%') OR
(d2.Secondary_Diag_4 like '%F32.3%' OR
d2.Secondary_Diag_4 like '%F33.3%') OR
(d2.Secondary_Diag_5 like '%F32.3%' OR
d2.Secondary_Diag_5 like '%F33.3%') OR
(d2.Secondary_Diag_6 like '%F32.3%' OR
d2.Secondary_Diag_6 like '%F33.3%'))
AND d2.Diagnosis_Date >= (patients_initial.Initial_discharge_date - 365.25) --
Psychotic symptoms last 12 months
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
```

E.21 Rereferral Count

```
-- Rereferral Count extract

select -- adult patients with at least one subsequent depression referral following
discharge from their initial depression referral where initial referral is since 2010
distinct
patients_initial.BrcId
, count(distinct r2.Spell_Number) as Count_Rereferral
from
(
              select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
```

```
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
inner join
Referral r2
on
r2.BrcId = patients_initial.BrcId
inner join
Diagnosis d2
on
(r2.BrcId = d2.BrcId AND r2.Spell_Number = d2.Spell_number)
where
r2.Accepted_date >= '01-jan-2010' AND -- only consider data 2010 and later
r2.Spell_Number > patients_initial.Initial_spell_number AND
(d2.Primary_Diag like '%F32%' OR -- depressive episode
d2.Primary_Diag like '%F33%') AND -- recurrent depressive episode
(d2.Diagnosis_Date > Initial_diagnosis_date) AND
(r2.Accepted_Date > Initial_discharge_date) AND
```

```
r2.Discharge_Date is not null AND
d2.Diagnosis_Date >= r2.Accepted_Date AND -- Diagnosis is within referral dates
d2.Diagnosis_Date <= r2.Discharge_Date
group by patients_initial.BrcId
```

E.22 Risk Assessment New

```
-- Rish Assessment New extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with suicide or self harm from Risk Assessment New form
distinct patients_initial.BrcId
, '1' as 'RA_New_SI_Selfharm'
from RiskAssessmentTool RA_new
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on
RA_new.BrcId = patients_initial.BrcId
where
RA_new.DateOfAssessment <= patients_initial.Initial_discharge_date AND -- RA before
initial discharge
SelfharmSuicide = 'Yes'
```

E.23 Risk Assessment Old

```
-- Risk Assessment Old extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with suicide ideation or self harm from Risk Assessment
Old form
distinct patients_initial.BrcId
, '1' as 'RA_Old_SI_Selfharm'
from Risk_assessment RA_old
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
```

```
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on
RA_old.BrcId = patients_initial.BrcId
where
RA_old.Assessed_Date <= patients_initial.Initial_discharge_date AND -- RA before
initial discharge
(Does_Patient_Have_History_Of_Suicide_Attempt_ID = 'Yes' OR
If_So_Did_He_Use_Violent_Perceived_Lethal_Method_ID = 'Yes' OR
Has_Patient_Made_Plan_To_End_Life_ID = 'Yes' OR
Is_Patient_Expressing_Suicidal_Ideation_ID = 'Yes' OR
Is_Patient_Expressing_Feelings_Of_Hopelessness_ID = 'Yes' OR
Risk_Of_Deliberate_Self_Harm_ID = 'Yes')
```

E.24 Sequential Accepted Events

```
-- Sequential Accepted events extract

select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged to Home or GP with Accepted date and event
dp.BrcId
, r.Accepted_date as 'Date'
, 'Accepted' as 'Event_type'
from
(
       select -- patients with depression diagnosis
       epr.BrcId
       , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as Age
       , epr.cleaneddateofbirth
       , epr.Gender_ID
       , epr.Marital_Status_ID
       , epr.Create_Dttm as patient_created_date
       , epr.Updated_Dttm as patient_updated_date
       , epr.ethnicitycleaned
       , d.Primary_Diag
       , d.Diagnosis_Date
       , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
       , d.Spell_Number
       from
       EPR_Form epr
       inner join
       Diagnosis d
       on epr.BrcId = d.BrcId
       where
       (d.Primary_Diag like '%F32%' OR -- depressive episode
       d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
       d.Diagnosis_Date is not null -- must have a diagnosis date
)
dp
inner join
Referral r
on
(r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
where
(r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
r.Accepted_date is not null AND -- must have been accepted to the trust
r.Discharge_Date is not null AND -- must have been discharged
r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
dp.Diagnosis_Date <= r.Discharge_Date AND
dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
order by r.BrcId
```

E.25 Sequential Area Level Deprivation Events

```
-- Area Level Deprivation events extract

Select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged to Home or GP with IMD score and Address start events
distinct patients_initial.BrcId
, a.Start_Date as 'Date'
, imd.[Index of Multiple Deprivation (IMD) Score] as 'IMD_score'
, 'Address_start' as 'Event_type'
from
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
```

```
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
left outer join
Address a
on a.BrcId = patients_initial.BrcId
left outer join
SQLCRIS_Common.dbo.ons_2015_imd imd -- 2015 ONS data
on imd.[LSOA code (2011)] = a.LSOA11 -- 2011 Census
where
(a.Start_Date <= (patients_initial.Initial_discharge_date) AND -- Address is not
effective after patient has been discharged
a.End_Date >= patients_initial.Initial_accepted_date OR a.End_Date is null)
AND imd.[Index of Multiple Deprivation (IMD) Score] is not null -- must have IMD score
group by
patients_initial.BrcId, imd.[Index of Multiple Deprivation (IMD) Score], a.Start_Date
order by
patients_initial.BrcId
```

E.26 Sequential Comorbidities Events

```
-- Sequential Comorbidities events extract

select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged to Home or GP with Comorbidity diagnosis dates and events
distinct d2.BrcId
, d2.Diagnosis_date as 'Date'
, (case when (d2.Primary_Diag like '%F30%' OR d2.Secondary_Diag_1 like '%F30%' OR
d2.Secondary_Diag_2 like '%F30%' --Manic
OR d2.Secondary_Diag_3 like  '%F30%') then 1 else 0 end) as F30
, (case when (d2.Primary_Diag like '%F31%' OR d2.Secondary_Diag_1 like '%F31%' OR
d2.Secondary_Diag_2 like '%F31%' -- Bipolar
OR d2.Secondary_Diag_3 like  '%F31%') then 1 else 0 end) as F31
, (case when (d2.Primary_Diag like '%F0%' OR d2.Secondary_Diag_1 like '%F0%' OR
d2.Secondary_Diag_2 like '%F0%' -- Organic
OR d2.Secondary_Diag_3 like  '%F0%') then 1 else 0 end) as F0
, (case when (d2.Primary_Diag like '%F1%' OR d2.Secondary_Diag_1 like '%F1%' OR
d2.Secondary_Diag_2 like '%F1%' -- Substance
OR d2.Secondary_Diag_3 like  '%F1%') then 1 else 0 end) as F1
, (case when (d2.Primary_Diag like '%F2%' OR d2.Secondary_Diag_1 like '%F2%' OR
d2.Secondary_Diag_2 like '%F2%' -- Schizophrenia
OR d2.Secondary_Diag_3 like  '%F2%') then 1 else 0 end) as F2
, 'Comorbidity' as 'Event_type'
, LTRIM(RTRIM(SUBSTRING(REPLACE(d2.Primary_Diag,'-',''), 1, 5))) as
Trimmed_Primary_Diag
, LTRIM(RTRIM(SUBSTRING(REPLACE(d2.Secondary_Diag_1,'-',''), 1, 5))) as
Trimmed_Secondary_Diag_1
, LTRIM(RTRIM(SUBSTRING(REPLACE(d2.Secondary_Diag_2,'-',''), 1, 5))) as
Trimmed_Secondary_Diag_2
, LTRIM(RTRIM(SUBSTRING(REPLACE(d2.Secondary_Diag_3,'-',''), 1, 5))) as
Trimmed_Secondary_Diag_3
, d2.Primary_Diag
, d2.Secondary_Diag_1
, d2.Secondary_Diag_2
, d2.Secondary_Diag_3
from
Diagnosis d2
inner join
```

```
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
(d2.Primary_Diag like '%F30%' OR
d2.Primary_Diag like '%F31%' OR
d2.Primary_Diag like '%F0%' OR
d2.Primary_Diag like '%F1%' OR
d2.Primary_Diag like '%F2%' OR
d2.Secondary_Diag_1 like '%F30%' OR
d2.Secondary_Diag_2 like '%F30%' OR
d2.Secondary_Diag_3 like '%F30%' OR
d2.Secondary_Diag_1 like '%F31%' OR
d2.Secondary_Diag_2 like '%F31%' OR
d2.Secondary_Diag_3 like '%F31%' OR
d2.Secondary_Diag_1 like '%F0%' OR
d2.Secondary_Diag_2 like '%F0%' OR
d2.Secondary_Diag_3 like '%F0%' OR
d2.Secondary_Diag_1 like '%F1%' OR
d2.Secondary_Diag_2 like '%F1%' OR
d2.Secondary_Diag_3 like '%F1%' OR
d2.Secondary_Diag_1 like '%F2%' OR
d2.Secondary_Diag_2 like '%F2%' OR
d2.Secondary_Diag_3 like '%F2%'
)
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
AND d2.Diagnosis_Date IS NOT NULL
```

E.27 Sequential Diagnosis Events

```
-- Sequential Diagnosis events extract

select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged to Home or GP with Depression diagnosis dates and events
distinct d2.BrcId
, d2.Diagnosis_date as 'Date'
, d2.Primary_Diag
, LTRIM(RTRIM(SUBSTRING(REPLACE(d2.Primary_Diag,'-',''), 1, 5))) as
Trimmed_Primary_Diag
, (case when (LTRIM(RTRIM(SUBSTRING(REPLACE(d2.Primary_Diag,'-',''), 1, 4)))) = 'F33 '
then 1 else 0 end) as F33
, (case when (d2.Primary_Diag) like '%F33.0%' then 1 else 0 end) as F330
, (case when (d2.Primary_Diag) like '%F33.1%' then 1 else 0 end) as F331
, (case when (d2.Primary_Diag) like '%F33.2%' then 1 else 0 end) as F332
, (case when (d2.Primary_Diag) like '%F33.3%' then 1 else 0 end) as F333
, (case when (d2.Primary_Diag) like '%F33.4%' then 1 else 0 end) as F334
, (case when (d2.Primary_Diag) like '%F33.8%' then 1 else 0 end) as F338
, (case when (d2.Primary_Diag) like '%F33.9%' then 1 else 0 end) as F339
, (case when (LTRIM(RTRIM(SUBSTRING(REPLACE(d2.Primary_Diag,'-',''), 1, 4)))) = 'F32 '
then 1 else 0 end) as F32
, (case when (d2.Primary_Diag) like '%F32.0%' then 1 else 0 end) as F320
, (case when (d2.Primary_Diag) like '%F32.1%' then 1 else 0 end) as F321
, (case when (d2.Primary_Diag) like '%F32.2%' then 1 else 0 end) as F322
, (case when (d2.Primary_Diag) like '%F32.3%' then 1 else 0 end) as F323
, (case when (d2.Primary_Diag) like '%F32.8%' then 1 else 0 end) as F328
, (case when (d2.Primary_Diag) like '%F32.9%' then 1 else 0 end) as F329
, 'Diagnosis' as 'Event_type'
from
Diagnosis d2
inner join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on d2.BrcId = patients_initial.BrcId
where
(d2.Primary_Diag like '%F32%' OR
d2.Primary_Diag like '%F33%')
AND d2.Diagnosis_Date <= patients_initial.Initial_discharge_date
AND d2.Diagnosis_Date >= patients_initial.Initial_accepted_date
AND d2.Diagnosis_Date IS NOT NULL
```

E.28 Sequential Discharge Events

```
-- Sequential Discharge events extract

select -- adult patients with their earliest depression diagnosis 2010 or later who
have been discharged to Home or GP with Discharge dates, type and events
dp.BrcId
, r.Discharge_date as 'Date'
, (case when (r.Discharge_Destination_ID) = 'GP' then 1 else 0 end) as GP
, (case when (r.Discharge_Destination_ID) = 'Home - No Follow Up Required' then 1 else
0 end) as Home
, 'Discharge' as 'Event_type'
from
(
        select -- patients with depression diagnosis
        epr.BrcId
        , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as Age
        , epr.cleaneddateofbirth
        , epr.Gender_ID
        , epr.Marital_Status_ID
        , epr.Create_Dttm as patient_created_date
        , epr.Updated_Dttm as patient_updated_date
        , epr.ethnicitycleaned
        , d.Primary_Diag
        , d.Diagnosis_Date
        , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
        , d.Spell_Number
        from
        EPR_Form epr
        inner join
        Diagnosis d
        on epr.BrcId = d.BrcId
        where
        (d.Primary_Diag like '%F32%' OR -- depressive episode
        d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
        d.Diagnosis_Date is not null -- must have a diagnosis date
)
dp
inner join
Referral r
on
```

```
(r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
where
(r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
r.Accepted_date is not null AND -- must have been accepted to the trust
r.Discharge_Date is not null AND -- must have been discharged
r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
dp.Diagnosis_Date <= r.Discharge_Date AND
dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
order by r.BrcId
```

E.29 Sequential DNAs Events

```
-- Sequential DNAs events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with Did Not Attend dates and events
patients_initial.BrcId
, e.Start_Date as 'Date'
, 'DNA' AS 'Event_type'
, 1 as 'DNA'
from
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
```

```
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
left outer join
Event e
on
patients_initial.BrcId = e.BrcId
where
(e.start_date >= (Initial_accepted_date) )
AND (e.Start_Date <= Initial_discharge_date )
AND e.Event_Outcome_ID in
('DNA by Client', 'DNA', 'Did not attend', 'DNA by Trust', 'DNA by Clinician')
```

E.30 Sequential Face to Face Events

```
-- Sequential Face to Face events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with Face-to-face dates and events
patients_initial.BrcId,
e.Start_Date as 'Date'
,'F2F_Event' as 'Event_type'
, 1 as 'F2F_Event'
from
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
```

```
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
              inner join
              Diagnosis d
              on epr.BrcId = d.BrcId
              where
              (d.Primary_Diag like '%F32%' OR -- depressive episode
              d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
              d.Diagnosis_Date is not null -- must have a diagnosis date
       )
       dp
       inner join
       Referral r
       on
       (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
       where
       (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
       r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
       r.Accepted_date is not null AND -- must have been accepted to the trust
       r.Discharge_Date is not null AND -- must have been discharged
       r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
       dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
       dp.Diagnosis_Date <= r.Discharge_Date AND
       dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
left outer join
```

```
Event e
on
patients_initial.BrcId = e.BrcId
where
(e.start_date >= (Initial_accepted_date) )
AND (e.Start_Date <= Initial_discharge_date )
AND e.Event_Type_Of_Contact_ID in ('Face To Face', 'Face-to-face')
AND e.Event_Outcome_ID in
('Attended', 'Attended on time/before HCP ready', 'Arrived late but was seen',
'Arrived late after HCP available but seen')
```

E.31 Sequential HoNOS Events

```
-- Sequential HoNOS events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS dates and events
patients_initial.BrcId
, h.Rating_Date as 'Date'
, 'Honos' as 'Event_type'
, Agitated_Behaviour_Score_ID as Honos_Scale1
, Self_Injury_Score_ID as Honos_Scale2
, Problem_Drinking_Drugs_Score_ID as Honos_Scale3
, Cognitive_Problems_Score_ID as Honos_Scale4
, Physical_Illness_Score_ID as Honos_Scale5
, Hallucinations_Score_ID as Honos_Scale6
, Depressed_Mood_Score_ID as Honos_Scale7
, Other_Mental_Problems_Type_ID as Honos_Scale8_Type
, Other_Mental_Problems_Score_ID as Honos_Scale8
, Relationship_Problems_Score_ID as Honos_Scale9
, Daily_Living_Problems_Score_ID as Honos_Scale10
, Living_Conditions_Problems_Score_ID as Honos_Scale11
, Occupational_Problems_Score_ID as Honos_Scale12
, Total as Honos_Total
, Adjusted_Total as Honos_Adjusted_Total
from
HoNOS h
inner join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
```

```
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on (h.BrcId = patients_initial.BrcId
AND h.Spell_Number = patients_initial.Initial_spell_number)
where h.Rating_Date >= patients_initial.Initial_accepted_date AND
h.Rating_Date <= patients_initial.Initial_discharge_date AND
h.Rating_Date IS NOT NULL
```

E.32 Sequential HoNOS abi Events

```
-- Sequential HoNOS abi events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS abi dates and events
patients_initial.BrcId
, h.Rating_Date as 'Date'
, 'Honos' as 'Event_type'
, Agitated_Behaviour_Score_ID as Honos_Scale1
, Self_Injury_Score_ID as Honos_Scale2
, Problem_Drinking_Drugs_Score_ID as Honos_Scale3
, Cognitive_Problems_Score_ID as Honos_Scale4
, Physical_Illness_Score_ID as Honos_Scale5
, Hallucinations_Score_ID as Honos_Scale6
, Depressed_Mood_Score_ID as Honos_Scale7
, Other_Mental_Problems_Type_ID as Honos_Scale8_Type
, Other_Mental_Problems_Score_ID as Honos_Scale8
, Relationship_Problems_Score_ID as Honos_Scale9
, Daily_Living_Problems_Score_ID as Honos_Scale10
, Living_Conditions_Problems_Score_ID as Honos_Scale11
, Activities_Score_ID as Honos_Scale12
, Total as Honos_Total
, Adjusted_Total as Honos_Adjusted_Total
from
HoNOSabi h
inner join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
       , dp.Marital_Status_ID
       , dp.ethnicitycleaned
       , dp.patient_created_date
       , dp.patient_updated_date
       from
       (
              select -- patients with depression diagnosis
              epr.BrcId
              , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
              , epr.cleaneddateofbirth
              , epr.Gender_ID
              , epr.Marital_Status_ID
              , epr.Create_Dttm as patient_created_date
              , epr.Updated_Dttm as patient_updated_date
              , epr.ethnicitycleaned
              , d.Primary_Diag
              , d.Diagnosis_Date
              , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
              , d.Spell_Number
              from
              EPR_Form epr
```

```
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on (h.BrcId = patients_initial.BrcId
AND h.Spell_Number = patients_initial.Initial_spell_number)
where h.Rating_Date >= patients_initial.Initial_accepted_date AND
h.Rating_Date <= patients_initial.Initial_discharge_date AND
h.Rating_Date IS NOT NULL
```

E.33 Sequential HoNOS Secure Events

```
-- Sequential HoNOS Secure events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS Secure dates and events
patients_initial.BrcId
, h.Rating_Date as 'Date'
, 'Honos' as 'Event_type'
, Agitated_Behaviour_Score_ID as Honos_Scale1
, Self_Injury_Score_ID as Honos_Scale2
, Problem_Drinking_Drugs_Score_ID as Honos_Scale3
, Cognitive_Problems_Score_ID as Honos_Scale4
, Physical_Illness_Score_ID as Honos_Scale5
, Hallucinations_Score_ID as Honos_Scale6
, Depressed_Mood_Score_ID as Honos_Scale7
, Other_Mental_Problems_Type_ID as Honos_Scale8_Type
, Other_Mental_Problems_Score_ID as Honos_Scale8
, Relationship_Problems_Score_ID as Honos_Scale9
, Daily_Living_Problems_Score_ID as Honos_Scale10
, Living_Conditions_Problems_Score_ID as Honos_Scale11
, Occupational_Problems_Score_ID as Honos_Scale12
, Total as Honos_Total
, Adjusted_Total as Honos_Adjusted_Total
from
HoNOS_Secure h
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
```

```
    dp.Diagnosis_Date <= r.Discharge_Date AND
    dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on (h.BrcId = patients_initial.BrcId
AND h.Spell_Number = patients_initial.Initial_spell_number)
where h.Rating_Date >= patients_initial.Initial_accepted_date AND
h.Rating_Date <= patients_initial.Initial_discharge_date AND
h.Rating_Date IS NOT NULL
```

E.34 Sequential HoNOS 65+ Events

```
-- Sequential HoNOS 65+ events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with HoNOS 65+ dates and events
patients_initial.BrcId
, h.Rating_Date as 'Date'
, 'Honos' as 'Event_type'
, Agitated_Behaviour_Score_ID as Honos_Scale1
, Self_Injury_Score_ID as Honos_Scale2
, Problem_Drinking_Drugs_Score_ID as Honos_Scale3
, Cognitive_Problems_Score_ID as Honos_Scale4
, Physical_Illness_Score_ID as Honos_Scale5
, Hallucinations_Score_ID as Honos_Scale6
, Depressed_Mood_Score_ID as Honos_Scale7
, Other_Mental_Problems_Type_ID as Honos_Scale8_Type
, Other_Mental_Problems_Score_ID as Honos_Scale8
, Relationship_Problems_Score_ID as Honos_Scale9
, Daily_Living_Problems_Score_ID as Honos_Scale10
, Living_Conditions_Problems_Score_ID as Honos_Scale11
, Occupational_Problems_Score_ID as Honos_Scale12
, Total as Honos_Total
, Adjusted_Total as Honos_Adjusted_Total
from
HoNOS65 h
inner join
(
    select -- patients_initial
    dp.BrcId
    , dp.Primary_diag as Initial_primary_diagnosis
    , dp.Diagnosis_date as Initial_diagnosis_date
    , dp.Spell_number as Initial_spell_number
    , r.Accepted_date as Initial_accepted_date
    , r.Discharge_Date as Initial_discharge_date
    , r.Discharge_Destination_ID as Initial_discharge_destination
    , r.Referral_Date as Initial_referral_date
    , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
    , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
    , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
    , diagnosis_num_days_after_discharge = case
    when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
    datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
    else 0
    end
    , dp.Age
    , dp.Gender_ID
    , dp.Marital_Status_ID
    , dp.ethnicitycleaned
    , dp.patient_created_date
```

```
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on (h.BrcId = patients_initial.BrcId
AND h.Spell_Number = patients_initial.Initial_spell_number)
where h.Rating_Date >= patients_initial.Initial_accepted_date AND
h.Rating_Date <= patients_initial.Initial_discharge_date AND
h.Rating_Date IS NOT NULL
```

E.35 Sequential Inpatient Events

```
-- Inpatient events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with Inpatient episode dates and events
i.BrcId
, i.Admission_Date as 'Date'
, datediff(day, i.Admission_Date, i.Discharge_date) as Num_Inpatient_Days
, 'Inpatient' as 'Event_type'
from
Inpatient_episode i
```

```
inner join
(
        select -- patients_initial
        dp.BrcId
        , dp.Primary_diag as Initial_primary_diagnosis
        , dp.Diagnosis_date as Initial_diagnosis_date
        , dp.Spell_number as Initial_spell_number
        , r.Accepted_date as Initial_accepted_date
        , r.Discharge_Date as Initial_discharge_date
        , r.Discharge_Destination_ID as Initial_discharge_destination
        , r.Referral_Date as Initial_referral_date
        , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
        , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
        , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
        , diagnosis_num_days_after_discharge = case
        when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
        datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
        else 0
        end
        , dp.Age
        , dp.Gender_ID
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on
i.BrcId = patients_initial.BrcId
where
(i.Admission_Date >= (Initial_accepted_date))
AND (i.Admission_Date <= Initial_discharge_date )
AND i.Discharge_Date is not null
AND i.Discharge_Date >= i.Admission_Date
```

E.36 Sequential Medication Events

```
-- Sequential Medication events extract

select -- adult patients with their earliest depression episode (since 2010) who have
been discharged to Home or GP with Medication dates and events
distinct patients_initial.BrcId
, m.Medication_Start_Date as 'Date'
, (case when lower(med_list.gazetteer) in ('agomelatine', 'valdoxan') then 1 else 0
end) as Agomelatine
, (case when lower(med_list.gazetteer) in ('amitriptyline', 'amitriptyline
hydrochloride') then 1 else 0 end) as Amitriptyline
, (case when lower(med_list.gazetteer) in ('citalopram', 'cipramil', 'citalopram
hydrobromide', 'citalopram hydrochloride') then 1 else 0 end) as Citalopram
, (case when lower(med_list.gazetteer) in ('clomipramine', 'anafranil') then 1 else 0
end) as Clomipramine
, (case when lower(med_list.gazetteer) in ('dosulepin', 'prothiaden', 'dothiepin')
then 1 else 0 end) as Dosulepin
, (case when lower(med_list.gazetteer) in ('sinepin', 'Doxepin') then 1 else 0 end) as
Doxepin
, (case when lower(med_list.gazetteer) in ('cymbalta', 'duloxetine', 'duciltia') then
1 else 0 end) as Duloxetine
, (case when lower(med_list.gazetteer) in ('cipralex', 'escitalopram') then 1 else 0
end) as Escitalopram
, (case when lower(med_list.gazetteer) in ('oxactin', 'fluoxetine', 'prozac',
'prozep', 'olena') then 1 else 0 end) as Fluoxetine
, (case when lower(med_list.gazetteer) in ('faverin', 'fluvoxamine') then 1 else 0
end) as Fluvoxamine
, (case when lower(med_list.gazetteer) in ('tofranil', 'imipramine') then 1 else 0
end) as Imipramine
, (case when lower(med_list.gazetteer) in ('isocarboxazid') then 1 else 0 end) as
Isocarboxazid
, (case when lower(med_list.gazetteer) in ('lomont', 'lofepramine') then 1 else 0 end)
as Lofepramine
, (case when lower(med_list.gazetteer) in ('maprotiline') then 1 else 0 end) as
Maprotiline
, (case when lower(med_list.gazetteer) in ('mianserin') then 1 else 0 end) as
Mianserin
, (case when lower(med_list.gazetteer) in ('zispin', 'mirtazapine', 'mirtazepine')
then 1 else 0 end) as Mirtazapine
, (case when lower(med_list.gazetteer) in ('manerix', 'moclobemide') then 1 else 0
end) as Moclobemide
, (case when lower(med_list.gazetteer) in ('nefazodone') then 1 else 0 end) as
Nefazodone
```

```
, (case when lower(med_list.gazetteer) in ('allegron', 'nortriptyline') then 1 else 0
end) as Nortriptyline
, (case when lower(med_list.gazetteer) in ('oxitriptan') then 1 else 0 end) as
Oxitriptan
, (case when lower(med_list.gazetteer) in ('seroxat', 'paroxetine') then 1 else 0 end)
as Paroxetine
, (case when lower(med_list.gazetteer) in ('nardil', 'phenelzine') then 1 else 0 end)
as Phenelzine
, (case when lower(med_list.gazetteer) in ('edronax', 'reboxetine') then 1 else 0 end)
as Reboxetine
, (case when lower(med_list.gazetteer) in ('sertraline', 'lustral') then 1 else 0 end)
as Sertraline
, (case when lower(med_list.gazetteer) in ('parnate', 'tranylcypromine') then 1 else 0
end) as Tranylcypromine
, (case when lower(med_list.gazetteer) in ('molipaxin', 'trazodone', 'trazadone') then
1 else 0 end) as Trazodone
, (case when lower(med_list.gazetteer) in ('surmontil', 'trimipramine') then 1 else 0
end) as Trimipramine
, (case when lower(med_list.gazetteer) in ('triptafen') then 1 else 0 end) as
Triptafen
, (case when lower(med_list.gazetteer) in ('tryptophan', 'pptimax') then 1 else 0 end)
as Tryptophan
, (case when lower(med_list.gazetteer) in ('depefex xl', 'depefex', 'efexor xl',
'effexor', 'efexor', 'venlafaxine', 'venlafaxine xl', 'foraven xl',
       'politid xl', 'politid', 'sunveniz xl', 'sunveniz', 'tonpular xl', 'tonpular',
'venadex xl', 'venaxx xl', 'venaxx', 'venlalic xl', 'venlalic',
       'viepax', 'alventa', 'amphero', 'rodomel', 'vencarm', 'venlablue', 'venladex',
'venlaneo', 'vensir', 'winfex') then 1 else 0 end) as Venlafaxine
, (case when lower(med_list.gazetteer) in ('brintellix', 'vortioxetine') then 1 else 0
end) as Vortioxetine
, 'Anti-depressant Start' as 'Event_type'
from SQLCRIS_Common.dbo.tbl_medication_gazetteer_source_combined_final as med_list
inner join
SQLCRIS_Common.dbo.tbl_structured_meds_gazetteer_lookup as med_map
on med_list.gazetteer = med_map.gazetteer
inner join
Medication m
on m.Medicine_ID = med_map.Medicine_ID
right outer join
(
       select -- patients_initial
       dp.BrcId
       , dp.Primary_diag as Initial_primary_diagnosis
       , dp.Diagnosis_date as Initial_diagnosis_date
       , dp.Spell_number as Initial_spell_number
       , r.Accepted_date as Initial_accepted_date
       , r.Discharge_Date as Initial_discharge_date
       , r.Discharge_Destination_ID as Initial_discharge_destination
       , r.Referral_Date as Initial_referral_date
       , datediff(day, r.Accepted_date, r.Discharge_Date) as
num_days_in_initial_episode
       , datediff(day, r.Referral_Date, r.Accepted_date) as
num_days_to_initial_accepted
       , datediff(day, r.Accepted_date, dp.Diagnosis_Date) as
num_days_to_initial_diagnosis
       , diagnosis_num_days_after_discharge = case
       when datediff(day, r.Discharge_Date, dp.Diagnosis_Date) > 0 then
       datediff(day, r.Discharge_Date, dp.Diagnosis_Date)
       else 0
       end
       , dp.Age
       , dp.Gender_ID
```

```
        , dp.Marital_Status_ID
        , dp.ethnicitycleaned
        , dp.patient_created_date
        , dp.patient_updated_date
        from
        (
                select -- patients with depression diagnosis
                epr.BrcId
                , CAST(datediff(day, epr.cleaneddateofbirth, getdate())/365.25 AS Int) as
Age
                , epr.cleaneddateofbirth
                , epr.Gender_ID
                , epr.Marital_Status_ID
                , epr.Create_Dttm as patient_created_date
                , epr.Updated_Dttm as patient_updated_date
                , epr.ethnicitycleaned
                , d.Primary_Diag
                , d.Diagnosis_Date
                , ROW_NUMBER() over (partition by d.BrcId order by d.Diagnosis_date) as
Diagnosis_num
                , d.Spell_Number
                from
                EPR_Form epr
                inner join
                Diagnosis d
                on epr.BrcId = d.BrcId
                where
                (d.Primary_Diag like '%F32%' OR -- depressive episode
                d.Primary_Diag like '%F33%') AND -- recurrent depressive episode
                d.Diagnosis_Date is not null -- must have a diagnosis date
        )
        dp
        inner join
        Referral r
        on
        (r.BrcId = dp.BrcId AND r.Spell_Number = dp.Spell_number)
        where
        (r.Accepted_Date - dp.cleaneddateofbirth) >= 18*365.25 AND -- adults only
        r.Accepted_date >= '01-jan-2010' AND -- only consider referrals 2010 and later
        r.Accepted_date is not null AND -- must have been accepted to the trust
        r.Discharge_Date is not null AND -- must have been discharged
        r.Discharge_Destination_ID in ('GP', 'Home - No Follow Up Required') AND -- in
remission
        dp.Diagnosis_Date >= r.Accepted_Date AND -- Diagnosis is within referral dates
        dp.Diagnosis_Date <= r.Discharge_Date AND
        dp.Diagnosis_num = 1 -- initial diagnosis on SLaM
)
patients_initial
on patients_initial.BrcId = m.BrcId
where med_list.bnf_code like '0403%' --anti-depressants
AND (m.medication_start_date >= (Initial_accepted_date) OR m.medication_start_date is
null)
AND (m.medication_start_date <= Initial_discharge_date OR m.medication_start_date is
null)
AND (med_list.gazetteer <> 'Flupentixol' AND
        med_list.gazetteer <> 'Flupenthixol' AND
        med_list.gazetteer <> 'Flupentixol Hydrochloride' AND
        med_list.gazetteer <> 'Fluanxol') -- remove anti-psychotic
```

F. Appendix F – Python (sample) and R Code

F.1 Python LSTM Prototype with Sample Data

This was used to estimate timings on hotdesks. It was run with 152 'patients' and 1,216 'events'.

The code was adapted from a blog post on Analytics Vidhya (2019) called 'A Hands-On Introduction to Time Series Classification (with Python Code)'.

In [1]:

```
# Import libraries
import numpy as np
import pandas as pd
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.preprocessing import sequence
from keras.optimizers import Adam
from keras.models import load_model
from keras.callbacks import ModelCheckpoint

# fix random seed for reproducibility
np.random.seed(7)
Using TensorFlow backend.
```

In [2]:

```
# Load data
df = pd.read_csv('Prototype_data.csv')
```

In [3]:

```
df.shape
```

Out[3]:

```
(1216, 89)
```

In [4]:

```
# Load target data
targets = pd.read_csv('Prototype_target.csv')
```

In [5]:

```
targets.shape
```

Out[5]:

```
(152, 1)
```

In [6]:

```
# Set sequence length
seq_len = 8
```

In [7]:

```
df = np.array(df)
targets = np.array(targets)
```

In [8]:

```
# Split train and test data
X_train = [df[i] for i in range(848)]
```

```
X_test = [df[i+848] for i in range(368)]
y_train = [targets[i] for i in range(106)]
y_test = [targets[i+106] for i in range(46)]
```

In [9]:

```
X_train = np.array(X_train)
X_test = np.array(X_test)
y_train = np.array(y_train)
y_test = np.array(y_test)
```

In [10]:

```
# Reshape data
X_train = np.reshape(X_train, (106, seq_len, X_train.shape[1]))
X_test = np.reshape(X_test, (46, seq_len, X_test.shape[1]))
```

In [11]:

```
X_train.shape
```

Out[11]:

```
(106, 8, 89)
```

In [35]:

```
# Create model with LSTM layer
model = Sequential()
model.add(LSTM(100, input_shape=(seq_len, 89)))
model.add(Dense(1, activation='sigmoid'))

# Loss function is binary_crossentropy as it's binary classification, ADAM
optimization algorithm used
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accur
acy'])
```

In [36]:

```
model.summary()
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 100)               76000
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 101
=================================================================
Total params: 76,101
Trainable params: 76,101
Non-trainable params: 0
_________________________________________________________________
```

In [48]:

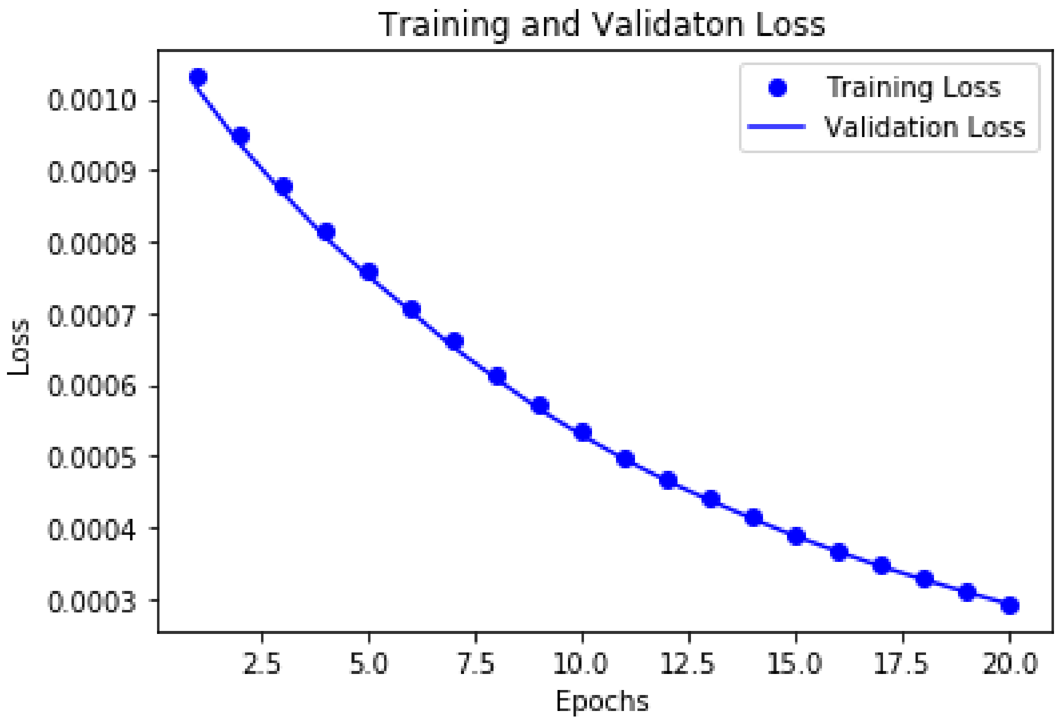
```
# Fit model - 200 epochs
history = model.fit(X_train, y_train, validation_split=0.2, epochs=20, batc
h_size=1)
Train on 84 samples, validate on 22 samples
Epoch 1/20
84/84 [==============================] - 1s 7ms/step - loss: 0.0010 - acc:
1.0000 - val_loss: 0.0010 - val_acc: 1.0000
Epoch 2/20
```

```
84/84 [==============================] - 1s 6ms/step - loss: 9.5151e-04 - a
cc: 1.0000 - val_loss: 9.3687e-04 - val_acc: 1.0000
Epoch 3/20
84/84 [==============================] - 1s 6ms/step - loss: 8.8006e-04 - a
cc: 1.0000 - val_loss: 8.6887e-04 - val_acc: 1.0000
Epoch 4/20
84/84 [==============================] - 1s 6ms/step - loss: 8.1702e-04 - a
cc: 1.0000 - val_loss: 8.0602e-04 - val_acc: 1.0000
Epoch 5/20
84/84 [==============================] - 1s 6ms/step - loss: 7.5937e-04 - a
cc: 1.0000 - val_loss: 7.5189e-04 - val_acc: 1.0000
Epoch 6/20
84/84 [==============================] - 1s 6ms/step - loss: 7.0801e-04 - a
cc: 1.0000 - val_loss: 7.0196e-04 - val_acc: 1.0000
Epoch 7/20
84/84 [==============================] - 1s 6ms/step - loss: 6.6048e-04 - a
cc: 1.0000 - val_loss: 6.5175e-04 - val_acc: 1.0000
Epoch 8/20
84/84 [==============================] - 1s 6ms/step - loss: 6.1425e-04 - a
cc: 1.0000 - val_loss: 6.0809e-04 - val_acc: 1.0000
Epoch 9/20
84/84 [==============================] - 1s 6ms/step - loss: 5.7210e-04 - a
cc: 1.0000 - val_loss: 5.6549e-04 - val_acc: 1.0000
Epoch 10/20
84/84 [==============================] - 1s 6ms/step - loss: 5.3368e-04 - a
cc: 1.0000 - val_loss: 5.2894e-04 - val_acc: 1.0000
Epoch 11/20
84/84 [==============================] - 1s 6ms/step - loss: 4.9949e-04 - a
cc: 1.0000 - val_loss: 4.9503e-04 - val_acc: 1.0000
Epoch 12/20
84/84 [==============================] - 1s 6ms/step - loss: 4.6859e-04 - a
cc: 1.0000 - val_loss: 4.6479e-04 - val_acc: 1.0000
Epoch 13/20
84/84 [==============================] - 1s 6ms/step - loss: 4.4006e-04 - a
cc: 1.0000 - val_loss: 4.3748e-04 - val_acc: 1.0000
Epoch 14/20
84/84 [==============================] - 1s 6ms/step - loss: 4.1402e-04 - a
cc: 1.0000 - val_loss: 4.1186e-04 - val_acc: 1.0000
Epoch 15/20
84/84 [==============================] - 1s 6ms/step - loss: 3.9000e-04 - a
cc: 1.0000 - val_loss: 3.8734e-04 - val_acc: 1.0000
Epoch 16/20
84/84 [==============================] - 1s 7ms/step - loss: 3.6752e-04 - a
cc: 1.0000 - val_loss: 3.6582e-04 - val_acc: 1.0000
Epoch 17/20
84/84 [==============================] - 1s 6ms/step - loss: 3.4697e-04 - a
cc: 1.0000 - val_loss: 3.4510e-04 - val_acc: 1.0000
Epoch 18/20
```

```
84/84 [==============================] - 1s 6ms/step - loss: 3.2768e-04 - a
cc: 1.0000 - val_loss: 3.2668e-04 - val_acc: 1.0000
Epoch 19/20
84/84 [==============================] - 1s 6ms/step - loss: 3.0998e-04 - a
cc: 1.0000 - val_loss: 3.0914e-04 - val_acc: 1.0000
Epoch 20/20
84/84 [==============================] - 1s 6ms/step - loss: 2.9358e-04 - a
cc: 1.0000 - val_loss: 2.9242e-04 - val_acc: 1.0000
```

In [49]:

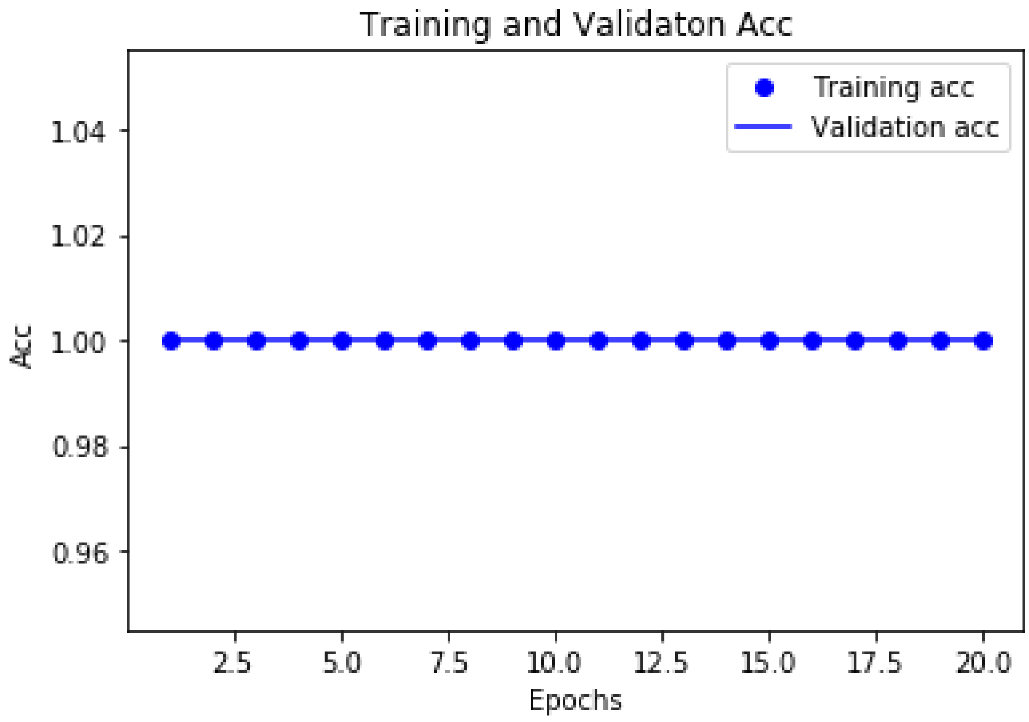
```python
# Plot training and validation loss
# Code provided by Chollet (2018, p. 74)

import matplotlib.pyplot as plt

history_dict = history.history
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, 'bo', label='Training Loss')
plt.plot(epochs, val_loss_values, 'b', label='Validation Loss')
plt.title('Training and Validaton Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```



In [50]:

```python
# Plot training and validation accuracy
# Code provided by Chollet (2018, p. 75)
```

```
plt.clf()
acc = history_dict['acc']
val_acc = history_dict['val_acc']
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and Validaton Acc')
plt.xlabel('Epochs')
plt.ylabel('Acc')
plt.legend()
plt.show()
```



In [51]:

```
# View the probability produced of each test record
# Code provided by Jason Brownlee (2019) in 'How to Calculate Precision, Recall, F1, and More for Deep Learning Models’

yhat_prob = model.predict(X_test)
```

In [52]:

```
# Predict crisp classes for test set
# Code provided by Jason Brownlee (2019) in 'How to Calculate Precision, Recall, F1, and More for Deep Learning Models’

yhat_classes = model.predict_classes(X_test)
```

In [53]:

```
# Calculate additional metrics
# Code provided by Jason Brownlee (2019) in 'How to Calculate Precision, Recall, F1, and More for Deep Learning Models’

from sklearn import metrics
accuracy = metrics.accuracy_score(y_test, yhat_classes)
```

```
print('Accuracy: %f' % accuracy)
# precision tp / (tp + fp)
precision = metrics.precision_score(y_test, yhat_classes)
print('Precision: %f' % precision)
# recall: tp / (tp + fn)
recall = metrics.recall_score(y_test, yhat_classes)
print('Recall: %f' % recall)
# f1: 2 tp / (2 tp + fp + fn)
f1 = metrics.f1_score(y_test, yhat_classes)
print('F1 score: %f' % f1)
Accuracy: 1.000000
Precision: 1.000000
Recall: 1.000000
F1 score: 1.000000
```

F.2 R RNN Prototype with Sample Data

```
# Code provided by Mic (2016) in a blog post called ‘Plain vanilla recurrent neural networks in R: waves prediction’

# Clear workspace
rm(list=ls())

# Set seed for reproducibility purposes
set.seed(10)

# Generate some test data
X <- matrix(rbinom(150000 * 90, 1, 0.5), ncol = 90, nrow = 150000)
Y <- matrix(1, ncol = 1, nrow = 150000)

# Reshape data
dim(X) <- c(15000, 10, 90)
dim(Y) <- c(15000, 10, 1)

# Split data into training and testing sets
train <- 1:10000
test <- 10001:15000

# Train model. Keep out the last two sequences.
model <- trainr(Y = Y[train,,],
                X = X[train,,],
                learningrate = 0.05,
                hidden_dim = 100,
                numepochs = 20)

# Predicted values
Yp <- predictr(model, X)

# Plot predicted vs actual. Training set + testing set
```

```
plot(as.vector(t(Y)), col = 'red', type = 'l', main = "Actual vs predicted", ylab = "Y,Yp")
lines(as.vector(t(Yp)), type = 'l', col = 'blue')
legend("topright", c("Predicted", "Real"), col = c("blue","red"), lty = c(1,1), lwd = c(1,1))

# Plot predicted vs actual. Testing set only.
plot(as.vector(t(Y[test,])), col = 'red', type='l', main = "Actual vs predicted: testing set", ylab = "Y,Yp")
lines(as.vector(t(Yp[test,])), type = 'l', col = 'blue')
legend("topright", c("Predicted", "Real"), col = c("blue","red"), lty = c(1,1), lwd = c(1,1))
```