Converting real life clinical data into analyzable format

Background

During the passage of time, revolutions in technology have continually increased the creation of information and its exchange. When humans advanced from spoken communication to written records they could steadily produce documents, papers, texts, books that some could read and learn from. Growth of writing were compilation of articles, tables and other records, which grew to become libraries. The ability to effortlessly widen accumulated data had to wait until the 15th century. Around 1439, Johannes Gutenberg developed the printing press, causing an astonishing growth in the sharing of information at an economical cost.

The 20th century generated a remarkable growth in the publication of scientific journals and monographs, most of which were not critically reviewed, as most physicians had no way to access to the existing medical information. Towards the late 20th century, the spread of computers and the internet providing immediate virtual access to diverse information has entirely changed the way knowledge is collected, stored, and circulated. The flow of information has been increasing at almost exponential levels. Today, data sets are measured in zettabytes (10^21 bytes). Cost-effectively collected and stored data allows researchers across the world to successfully advance understanding of science and medicine.

International Data Corporation (**IDC**) is one of the premier global providers of market intelligence, information technology, and a host of other areas. They predicted in a report released in Dec 2018, that the cumulative sum of the world’s data will grow from 33 zettabytes this year to a 175ZB by 2025, for a compounded annual growth rate of 61%. A zettabyte is a trillion gigabytes multiplied that by 175 times. This expansion of data has been seen in every industry, in every corner of the world. The unending increase in the quantity and flood of information denotes an important professional opportunity and a challenge at the same time for those in medicine and science.

As indicated by Toby Cosgrove MD, from Cleveland, medical information will soon double every 73 days, in year 2020, which used to take approximately 3.5 years in 2010. An estimated 800,000 papers are published yearly in 5,600 medical journals. It is projected that 12,000 new articles and 300 randomized controlled trials are added to Medline each week, and that new medical articles appear at a rate of one every 26 seconds. To be able to generate any kind of analysis and make accurate predictions, there is a need to access, connect various sources, collate, and consume all of the data. Data can be produced, obtained, and stored in a numerous number of structures.

Raw data, like unrefined gold buried deep in a mine is a precious resource. It is often:

* Inconsistent. It contains both relevant and irrelevant data
* Imprecise. It contains incorrectly entered information or missing values
* Repetitive. It contains duplicate data

Before anyone can benefit from raw data, it needs to be extracted, filtered through, understood, and transformed into something could be analyzed. Understanding the scope of data being analyzed and seeing the changes made to the data can accelerate the entire process of going from “information to building wisdom”.

Data access, extraction, cleansing, transforming, making it clean and consistent data are a few steps of data preparation or data wrangling. One of the surveys carried out by Forbes estimates that data cleaning accounts for up to 80% of the development time and cost in data warehousing projects. The subsequent sections will provide us with detailed information to go from an observation to information from a database point of view.

Material and Methods:

During an appointment at a hospital, diverse types of data are collected. Raw data are observations about individual patients created by the treating doctor at a hospital. These data may be in the form of measurements of patient’s characteristics such as age, gender, height, weight, blood pressure, heart rate, etc. Raw data may also include description of the medical history, physical exam information, clinical laboratory results (e.g., serum lipid values, hemoglobin levels), whole exome or genome sequences, imaging results (e.g., X-ray, magnetic resonance imaging [MRI]), procedure results (e.g., electroencephalogram [EKG], endoscopy), or self-reported data (e.g., symptoms, quality of life).

I would present a flow diagram of how the hospital visit unfolds creating data through patient and doctor interaction.

|  |  |  |
| --- | --- | --- |
| Step | Description | Pictorial representation |
| 1 | A patient visits a hospital | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9E73FB80.tmp |
| 2 | Medical data/ patient history is populated, either on paper or electronically | Electronic Health Record Stock Illustrations – 823 Electronic ... |
| 3 | Medical assessments are done if prescribed   1. Pathology reports 2. ECGs 3. CT scans 4. Ayurvedic tests 5. Panchakarma | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\840CC8A6.tmp  Heartbeat Line Icon. Heart Rhytm. ECG. Cardiogram Stock Vector ...  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\23EF0755.tmp  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\DD1FC92B.tmp  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\44C8A771.tmp |
| 4 | All the above combined together generates the Electronic health record / Electronic Medical record is generated  This can be called as “raw data” or “source data” | Clipart Electronic Health Record |
| 5 | The data from pathology labs, ECGs, CT scans come from multiple labs, hence there is a need of “data transfer” from different locations to the source database | Paper Sheets Having Indicating Arrows Line Icon Data Transform ... |
| 6 | As seen in step 3: Source data can come from many sources, hence there is a need to integrate data sources into 1 source called as any one of the following:   1. source database 2. data ware-house 3. data lake 4. data mart | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\ACE814D8.tmp  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\E79DFCBA.tmp |
| 7 | Once the source data is electronically made available, it has to be made accessible through a system | Monitor Screen Check Mark Symbol Padlock Data Access Icon — Stock ... |
| 8 | This point onwards, the data preparation steps begin   1. Data extraction from source to a staging area 2. Data is transformed as per the needs 3. Data is loaded into areas for the “end users” to use per needs | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\FC837A54.tmp |
| 9 | Data extraction from the source systems: Source to analysis area | Data Transfer Icon - Structured |
| 10 | Data is transformed by performing various steps:   1. Data filtering | Vector black filter data icon set. ... | Stock vector | Colourbox |
| 10 | 1. Data cleansing | The Dirty on Data Cleansing & Appending |
| 10 | 1. Data deduplication | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7DA8F17.tmp |
| 10 | 1. Data merging | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3628E2E0.tmp |
| 10 | 1. Data transposing | Questions from Alteryx Training | InterWorks |
| 10 | 1. Data mapping | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\1F9097E3.tmp |
| 10 | 1. Data outlier detection | Local outlier factor - Wikipedia |
| 10 | 1. Data imputation, as necessary | Chapter 4 Multiple Imputation | Book_MI.utf8.md |
| 10 | 1. Data aggregation | Download Free png Free Aggregation Icon 203770 | Download ... |
| 11 | Finally getting raw data to structured format.  Some of these steps are repeated on a periodic basis to ensure that the database is maintained at the expected levels of performance. | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\413B5782.tmp |
| 12 | Some of the data transformations are performed specific for each analysis so that the data is fit for purpose of that specific analysis |  |
| 12 | 1. Hospital management | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\62C2BC8D.tmp |
| 12 | 1. Individual researcher | Analytics - Free people icons |
| 12 | 1. Health authorities | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\553C02C8.tmp |
| 12 | 1. Data mining | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\69771C16.tmp |
| 12 | 1. Various publications | Modern Outline Style Data Analytics Icons Collection Stock ... |

The following visualization provides us with the Flow diagram from data source to final usage by various usage types.

Flow diagram from data source to final usage by various usage types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Sources | Staging area | Ware house | Data marts | Usage |
| Data access:  Monitor Screen Check Mark Symbol Padlock Data Access Icon — Stock ... | Data Transfer Icon - Structured Vector black filter data icon set. ... | Stock vector | Colourbox The Dirty on Data Cleansing & Appending | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\1F9097E3.tmp | Chapter 4 Multiple Imputation | Book_MI.utf8.md |  |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Operational system  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Coding dictionaries  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Clinical system  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\50D212FE.tmp  Flat files information | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Calculations and transformations | Data Warehouse Icons - Download Free Vector Icons | Noun Project  Curated and consistent data storage | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Operational data  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Pharmacy data  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Patient level data | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\62C2BC8D.tmp  Hospital management  Analytics - Free people icons  Researchers  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\553C02C8.tmp  Health authorities  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\69771C16.tmp  Data mining  Modern Outline Style Data Analytics Icons Collection Stock ...  Various documents |
|  | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7DA8F17.tmp C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3628E2E0.tmp  Questions from Alteryx Training | InterWorks | Local outlier factor - Wikipedia | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\413B5782.tmp |  |

Real actions adopted on acquiring data

The framework defined above has been followed to reach from the source data stage to analyzable data.

Data access

Since data is stored differently based on the type of data, different sets of tools are needed to connect to the respective data sources. E.g., structured data is stored in relational databases and uses SQL to query the data, unstructured data stored in Hadoop would use Hive, Spark or Pig, data extracts for file formats like CSV, TXT, JSON, XML, etc., and for other formats tools like Python and R are used.

The details for accessing the hospital management system are as follows: I was provided "read-only" access to the hospital database, so that there are no accidental updates to any records as well as no risk of changing any source information.

1. Install PostgresSQL locally on the system and then connect to the database as per details below.
2. Install Cygwin terminal locally on the system.
3. Login using the Cygwin terminal (the following command will prompt for password): psql -h 54.244.12.255 -p 5432 -d iaim -U iaim\_ro
4. Postgress Data Base details are as follows:
   * Hostname: 54.244.12.255
   * port: 5432
   * user: iaim\_ro
   * password: a1b2c3

This way, I was able to remotely access specific version of database without interfering in day-to-day hospital transactions. The data available in the SQL tables were used for the analysis. There were a lot of lab pathology reports uploaded into the database as pdf files have not been used to computational complexities.

Part 1

Details of the database

The database had approximately 200 tables. They covered various components of the hospital’s day-to-day functions right from the operational data to the patient level clinical information. The high level of classification of data types:

1. Operational tables:
   1. Hospital charges – IP, OP
   2. Operation theater charges
   3. Inventory of equipment
   4. Doctor charges, etc.
2. Reference dictionaries
   1. Disease codes
   2. Ayurvedic services
   3. Medication names
   4. Mast list of lab tests
   5. Names of city, state, Countries
3. Doctor details
   1. Doctor ID
   2. Relevant ward information
   3. Internal / Visiting / Part time / Full time
4. Patient information
   1. Patient details,
   2. Visit details
   3. Vital signs
   4. Registration details
   5. Discharge details
   6. Lab data details
   7. Diet details, etc.
5. Tables related to managing access levels and other IT related contexts

For this study, the following data was not used to avoid any controversies as well as to keep patient confidentiality:

1. Hospital monetary details
2. Doctor’s details
3. Patient details of sensitive nature – name, phone number, socio economic class, etc.

The next table presents inventory of tables (some tables have been omitted as per the above section). The tables names marked in yellow colors are used for the creation of the analysis ready datasets from the unending puzzle of all the tables.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| action\_rights | diet\_prescribed | hospital\_technical | package\_componentdetail | patient\_registration | section\_field\_options | store\_item\_batch\_details | test\_details |
| admission | discharge\_format\_detail | icu\_bed\_charges | package\_item\_charges | patient\_section\_details | service\_consumable\_usage | store\_item\_details | test\_org\_details |
| anesthesia\_type\_charges | doctor\_charges\_backup | ip\_bed\_details | package\_prescribed | patient\_section\_details\_orig | service\_documents | store\_item\_lot\_details | test\_results\_master |
| area\_master | doctor\_charges\_op\_backup | ip\_prescription | patient\_activities | patient\_section\_forms | service\_master\_charges | store\_patient\_indent\_details | test\_visit\_report\_signatures |
| bed\_details | doctor\_consultation | item\_supplier\_prefer\_supplier | patient\_consultation\_field\_values | patient\_section\_image\_details | service\_master\_charges\_backup | store\_patient\_indent\_main | test\_visit\_reports |
| bill | doctor\_consultation\_charge | manf\_master | patient\_demographics\_mod | patient\_section\_values | service\_org\_details | store\_po | tests\_conducted |
| bill\_activity\_charge | doctor\_medicine\_favourites | medicine\_dosage\_master | patient\_deposits | patient\_service\_prescriptions | services | store\_po\_main | tests\_prescribed |
| bill\_adjustment | doctor\_op\_consultation\_charge | medicine\_id\_health\_authority\_unique | patient\_deposits\_setoff\_adjustments | patient\_test\_prescriptions | services\_prescribed | store\_reagent\_usage\_details | theatre\_charges |
| bill\_charge | doctor\_org\_details | message\_recipient | patient\_details | ppfv\_form\_detail\_id | stk\_chkpt | store\_reagent\_usage\_main | diet\_charges |
| bill\_charge\_adjustment | dyna\_package\_category\_limits | mrd\_casefile\_attributes | patient\_details\_patient\_phone\_country\_code | preauth\_prescription | stock\_issue\_details | store\_reorder\_levels | url\_action\_rights |
| bill\_receipts | dyna\_package\_charges | mrd\_codes\_doctor\_master | patient\_discharge | preauth\_prescription\_activities | stock\_issue\_main | store\_retail\_customers | user\_services\_depts. |
| complaintslog | dyna\_package\_org\_details | mrd\_codes\_master | patient\_documents | prescribed\_medicines\_master | store\_adj\_details | store\_sales\_details | visit\_vitals |
| consultation\_charges | equipement\_charges | mrd\_diagnosis | patient\_general\_docs | progress\_notes | store\_adj\_main | store\_sales\_main | vital\_reading |
| consultation\_org\_details | estimate\_bill | mrd\_observations | patient\_hvf\_doc\_values | registration\_charges | store\_checkpoint\_details | store\_stock\_details | section\_field\_desc |
| deposit\_setoff\_total | estimate\_charge | operation\_charges | patient\_medicine\_prescriptions | sample\_collection | store\_estimate\_details | store\_transaction\_lot\_details | section\_master |
| diagnostic\_charges | favourite\_reports | operation\_org\_details | patient\_other\_medicine\_prescriptions | sch\_resource\_availability | store\_grn\_details | store\_transfer\_details | ha\_item\_code\_type |
| diagnostic\_charges\_backup | fixed\_asset\_master | other\_services\_prescribed | patient\_other\_prescriptions | sch\_resource\_availability\_details | store\_grn\_main | store\_transfer\_main | package\_charges |
| diagnostic\_reagent\_usage | follow\_up\_details | outsource\_sample\_details | patient\_packages | scheduler\_appointment\_items | store\_indent\_details | supp\_inv\_id | patient\_prescription |
| diagnostics | growth\_chart\_reference\_data | pack\_org\_details | patient\_pdf\_form\_doc\_values | scheduler\_appointments | store\_indent\_main | supplier\_master |  |

Understanding Source Databases

Comprehension of the database is more than knowing how it is built with tables, views, and relationships. In order to write meaningful queries one needs to understand how real world data was decoded and stored in the database.

**Tables:** When constructing queries it is important to understand a table’s purpose. Is the table used to organize patient visit data or a list of Ayurvedic prescribed medicines or other medical personnel related information etc.?

Before writing a query look over database’s table names. In many cases the names reveal the main topic of the tables. If looking for diagnosis data, then chances are the table will be named something akin to “Patient\_diagnosis.”

The tables listed as (1) PATIENT\_DETAILS should contain details about patients. (2) IP\_PRESCRIPTIONS table should have medications prescribed to patients who have been hospitalized for some reason. (3) STATE table looks like a reference table with names of states of India.

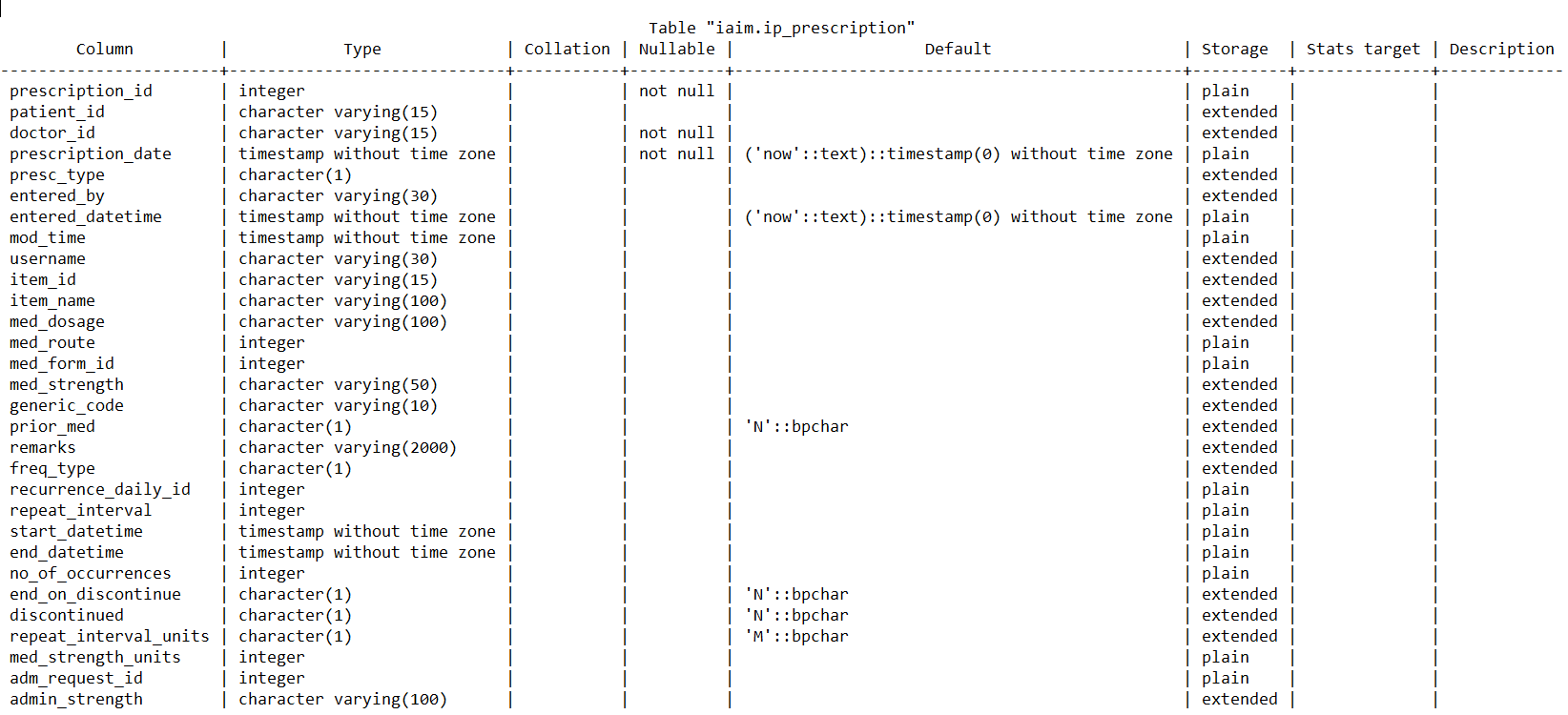
**Columns:** A table’s columns give a lot of information. It is important to have appropriate column names which explain the purpose of each column. Listed below are the column names in PATIENT\_DETAILS:

1. mr\_no
2. country
3. patient\_state
4. patient\_city
5. dateofbirth
6. patient\_gender
7. death\_date

The column names in this table are quite self-explanatory, with one exception; assumption that “mr\_no” is the unique patient identification variable.

The column’s data type also gives us the clues about what kind of data is expected in the column. E.g. IP\_PRESCRIPTION table has 30 columns or variables, see the screenshot below. The “type” column provides the details on the data type presented in the table.

* The table based on the name looks like related to the IP medication prescription
* Patient\_id and prescription\_id are key variables in the table
* Patient\_id: is the unique visits ID for each patient at each visit (mr\_no is the real patient ID, not present in the current table; this is a unique code which does not change)
* item\_id and item\_name are the columns which contain the medication name and ID
* All other variables explaining the course of the medication are present: start\_datetime and end\_datetime (start and end of medication), med\_route (route of administration), med\_dosage (doage), freq\_type (frequency)
* There a few variables containing system related information, these could be dropped from “staging area” as they are not useful for our intended analysis



Keys and Relationships: By inspecting the table names and columns some logical reasoning gets built, but how are they related to each other needs to be explored further? This is where it makes sense to review the relevant table’s primary keys to understand what values are used to identify the tables and to see if foreign key(s) from other tables can be used to make a relation. Typically, column names in across tables are named the same. If they are not the same and the same value is stored in multiple variables then it possesses additional challenges in establishing relationships between tables.

Part 2

Data extraction for downstream activities

Extraction is the process of retrieving data from a source system for further use in a data repository environment. After the extraction, data can be transformed and loaded into the data warehouse. This is the most challenging aspect of Extraction-Transformation-Load, as extracting data correctly will set the stage for how subsequent processes will go. Planning and creating the extraction process is one of the most time-consuming tasks in the whole process. The source systems might be very complex and therefore deciding which data needs to be extracted can be challenging. The data has to be extracted several times in a recurrent periodic way to ensure up-to-date data as compared to the warehouse.

There are several important considerations to be taken into account while designing extraction methods.

1. Full extraction: the data is completely extracted from source to repository every time, if the data size is very big then this step becomes very time consuming
2. Incremental extraction: at a particular point in time, only the data that has changed as well as updated since a well-defined event will be extracted, this is less time consuming compared to the first method. But technically this approach is more difficult as “time point”, “update”, “change” are difficult to define with complete accuracy across the whole organization.

An analyst or the team involved in this process must diagnose the data. Are the data responsive to the current analysis questions? What format are they in, and how much effort is required to put them into a format expected by downstream analysis tools? Are there data quality issues, such as missing data, inconsistent values, or unresolved duplicates? Next, the analyst must decide whether to continue working with the data, and, if so, the data must be transformed and cleaned into a usable state.

The extraction of the data allows subsequent analysis steps to be kept independent of the source data that is prone to changes, inconsistencies, duplications, etc. To optimize the overall performance of data preparation process, limit the fields needed for your analysis.

Data extracted from Hospital database

In our study, we had different versions of data, details in the table below. The analysis carried out on different versions of data provide numerically different answers, but the overall trends experienced tend to be in the same direction.

|  |  |  |  |
| --- | --- | --- | --- |
| Data version | Version 1 | Version 2 | Version 3 |
| Date time frame | From start of the hospital to Oct 2016 | From start of the hospital to Oct 2017 | PDF file version of data for 15 In Patients |
| Data domains | Lab  Vital signs  Diagnosis | All the available data in the hospital database | Specific patient visits case report forms |

Part 3

Data preparation

Reliable and reproducible data preparation facilitates well-organized analysis, minimizes mistakes and inaccuracies that occur to data through processing, and makes all processed data accessible to users. Below are the steps involved in data preparation

1. Merging, joining:

Combine/enrich relevant data from different datasets into a new dataset. Joining data is one of the most important functions of data transformation. A “join” is an operation that connects two or more database tables by their matching columns. This establishes a relationship between multiple tables, which merges table data together so a query can be made on the resultant data.

1. Appending

Combine two similar datasets into a larger dataset

1. Filtering

Rule-based reduction of a larger dataset into a smaller dataset. Data filtering includes techniques used to refine datasets. The goal of data filtering is to refine a data source to only what the user needs by eliminating repeated, irrelevant data. Data filters can be used like this to amend query results and data reports. Data filtering involves the selection of specific rows, columns, or fields to display from the dataset.

1. Deduping

Remove duplicates based on specific criteria as defined. Data deduplication is a data compression process to identify and remove repeated copies of information. Deduplication allows storage of one unique copy of data in data warehouse or database. This process examines incoming data and compares it to data that is already stored in the system. If the data is already there, deduplication algorithms delete the duplicate information while creating a reference to it.

1. Cleansing

Data cleansing involves deleting out-of-date, inaccurate, or incomplete information to increase the accuracy of data. The process might include parsing data to remove syntax errors, deleting record portions, and amending typos. It could also involve fixing duplication problems that result from merging multiple datasets. Data cleansing involves identifying incorrect data values and then either correcting them or rejecting them. They deal with INVALID values in single data elements or correlation across multiple data elements. This can be helpful in improving the accuracy of data.

Automated data cleansing programs can identify wrong values but generally cannot correct them. They can correct values only through synonym lists or correlation against tables showing valid combinations of values. Most of the time they identify a wrong value but cannot correct it. The only alternative is to change the value to unknown (NULL in most systems) or to reject the observation in which the bad value is contained. If it is rejected, they can either drop the target (creating a bigger problem) or manually investigate the value and reenter it into the target after correction.

Dropping observations has many problems. First, loss of some data. The correct data in other data elements of these observations may be more important to the target than by dropping the entire observation. Added problem is that it may create a structural problem relative to other observations in the same or other tables. Rejecting an observation may have the effect of causing many other observations to be rejected later, when referential constraints are enforced upon load.

1. Transforming

Convert missing values or derive a new column from existing column(s). Data transformation is the process of extracting good, reliable data from these sources. This involves converting data from one structure (or no structure) to another to integrate it with a data warehouse or with different applications. It allows to expose the information to advanced business intelligence tools to create performance reports and forecast future trends.

1. Aggregating

Roll up data to have data for analysis. Data Summarization is similar to data aggregation. It refers to the creation of different business metrics through the calculation of value totals. Data aggregation is a process that searches, gathers, summarizes and presents data in different reports.

1. Format revision

Format revisions fix problems that create trouble from variables having different data types. Some variables might be numeric, and others might be text. One data system could treat text vs. numeric information differently, so standardize the formats to integrate source data with the target data schema. This could involve the conversion of male/female, date/time, measurements, and other information into a consistent format. Field lengths can also be an issue—especially if the target schema has smaller character limits. In these cases, it may be necessary to standardize the length of fields by breaking up long serial numbers into their smaller parts and putting them into separate columns. Additionally, format revision could involve splitting up a comma-separated list of words or numbers into multiple columns.

1. Key Restructuring

When the tables in a data warehouse have keys with built-in meanings, serious problems can develop. For example, if a patient ID serves as a primary key, changing the patient ID format in the original data source means that the number would have to change everywhere it appears in the data system. That would cause a cascade of updates that over-burden or slow down the system. By drawing key connections from one table to another, key restructuring optimizes the data warehouse for speed and efficiency.

1. Bucketing/Binning

This transformation is used to change a numeric series into fixed, categorical ranges, say, from {2,5,8…} to {2-5, 6-9, 10-13…}. E.g., the seasonal fluctuations in diseases. Bucketing/binning allows isolation of noisy data. The focus away from short-term volatility provides a real representation of trends over time.

1. Z-Score Normalization and Max-Min Scaling

In scaling, data ranges are re-scaled. In z-score normalization, individual data features have zero-min and unit variance. Scaling is important because datasets often contain elements in varying units and ranges. This simple transformation allows for a compelling visual check as well.

1. Imputation

Missing values are one of the most common problems faced in data. The reason for the missing values might be human errors, disruptions in the data flow, privacy concerns, and so on. Whatever is the reason, missing values affect the analysis. Some algorithms drop the observations which have missing values. On the other hand, most of the algorithms do not accept datasets with missing values and gives an error.

1. Numerical Imputation

Imputation is a better option rather than dropping because it keeps the data size. However, there is an important selection of what and how to impute to the missing values. This “imputation” is a PhD topic in itself and too vast to explain in a short space. Imputations by statistics of central tendency (mean, median, min, max), regression methods, multiple imputations (same value imputed n numbers of times), chained imputations (imputations based on a certain sequence), etc. are a few methods available.

1. Handling Outliers

Before mentioning how outliers can be handled, I want to state that the best way to detect the outliers is to demonstrate the data visually. All other statistical methodologies are open to making mistakes, whereas visualizing the outliers gives a chance to take a decision with high precision. Outlier Detection with Standard Deviation and with percentiles is easily possible. As the observations are real data, whether to drop them or keep them is a dilemma faced by every researcher.

Part 4

What is this data used for?

It is very important to think through the data preparation stage about the data holistically. It is important to think about how people will use the data prepared. Understanding this context will help in determining which data set to use, how much data to bring into data wear house, and how to ultimately structure and shape the data. To get started, answer some basic questions:

1. Who is doing the analysis?

Consider the end users of the final data set. E.g., is the analyst the sole user who will access and understand all parts of the data for detailed analysis? On the other hand, will someone in a different role use the data set,? If it is the second option, then trim down the data set to only those measures. In this case, join the data and fact tables to get the information. Audience is critical while preparing data, similar to while creating a dashboard.

1. What type of questions need to be asked or answered?

In the data preparation process, it is important to understand how people will use the final data set—for complex analysis or for a quick summary. This detail influences the data preparation process significantly, determining both the amount of effort and detail. An analyst typically predicts the most common questions that people will ask of the data based on understanding of strategic business priorities, but there will likely be unanticipated questions that pop up. While preparing a data set, there is a balance between serving the immediate questions and allowing for further exploration. For example, someone may see a pharmacy sales trend during the last six months, but digging into a spike during a particular week requires deeper analysis and a daily granularity of the data.

Part 5

How to do it?

Data derivation, Data derivation involves the creation of special rules to “derive” the specific information wanted from the data source. Data derivation allows to create a set of transformation rules.

1. Prepare the way you think

Data preparation has a lot of different components, from restructuring to reformatting to cleaning, and should not be constrained by a specific order.

1. Compartmentalize each step

Creating new steps for a specific set of actions keeps your flow nice and tidy. Think of steps as folders in filing cabinet - organize files by their subject, making it easier to find. Similarly, the steps in the flow should group a set of changes that capture a particular task. Keep these actions in the same step, and add a descriptive name to help you understand the flow later on. This process builds in-built documentation sharing the flow with other analysts, it lets them find and reference the same actions, giving them a way to easily make any edits.

1. Modular approach to make maintenance easy

Document for reproducibility and collaboration, Staying organized throughout your preparation process is essential when you need to revisit and make a change to some step in the process. While there is a need to follow a specific set of instructions to clean the data, the data prep process will be a lot easier to edit and update.

Exploration of the database

As the aim of this study is to understand the every day ayurvedic clinical practice, we further explored the data and thought of building one of the many datasets – using clinical information of patients. This dataset has patient’s visit wise, longitudinal data from the very first day of hospital visit to the last day of hospital visit in period of 2011 until Oct 2017. The case report form at each visit captures disease and medication data, along with demographic, background data and a few more characteristics (outlined later in the document). This data creates documented complete picture of each patient from various parts of the database including (1) In patient visits, (2) Outpatient visits, (3) diseases reported as per Ayurvedic Classification dictionary, (4) Medication prescribed, (5) Ayurvedic services prescribed. These components of data are logically arranged in one dataset by using various data transformation steps. In addition, there are new variables derived to create necessary information for the potential analyses. Let us go through the challenges experienced to assemble the “reference dataset” from the source data and practical explanation of the “data preparation” steps.

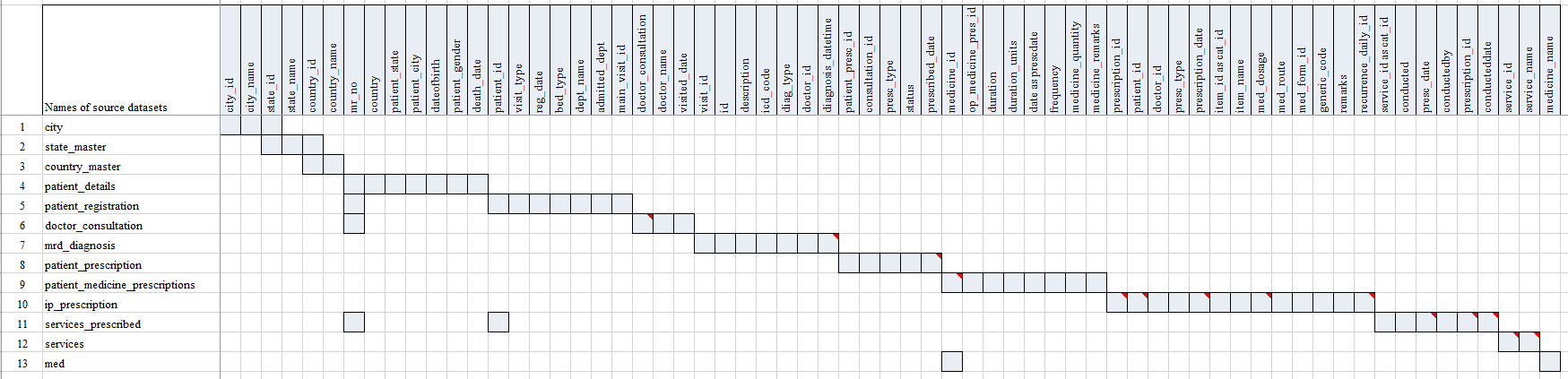
1. The database was manually explored using various SQL programming commands, the variables and observations were checked from numerous tables
2. Patient information and key variables needed to be understood: unique patient ID is MR\_NO, and unique ID for individual visit is PATIENT\_NO (many tables containing patients’ clinical information have this variable as the key variable)
3. Reference files needed to be used to reformat the coded variables
4. First section of the creation:
   1. Extract relevant data tables from the source database
   2. Transform the variables, join the tables based on logical link
   3. Create “staged data” or “snapshot of source data”
   4. Reference files (disease categories, Indian seasons) which are needed for calculations are developed using expert’s help
5. Second section of the program:
   1. Cleanse the tables
   2. Transform the tables for combining
   3. Join the tables on logical link
   4. Derive additional variables as necessary
   5. Filter the data using reference files created in the earlier section

In this process, we have used 13 source datasets (5 reference datasets and 8 patient level datasets) and ~65 variables to generate the necessary snapshot of the source data. These have been re-arranged into 6 datasets and ~40 variables. 3 additional reference files are used for further processing. 1 final dataset having ~30 variables from source and ~30 newly derived variables is built.

Flowcharts of the algorithms

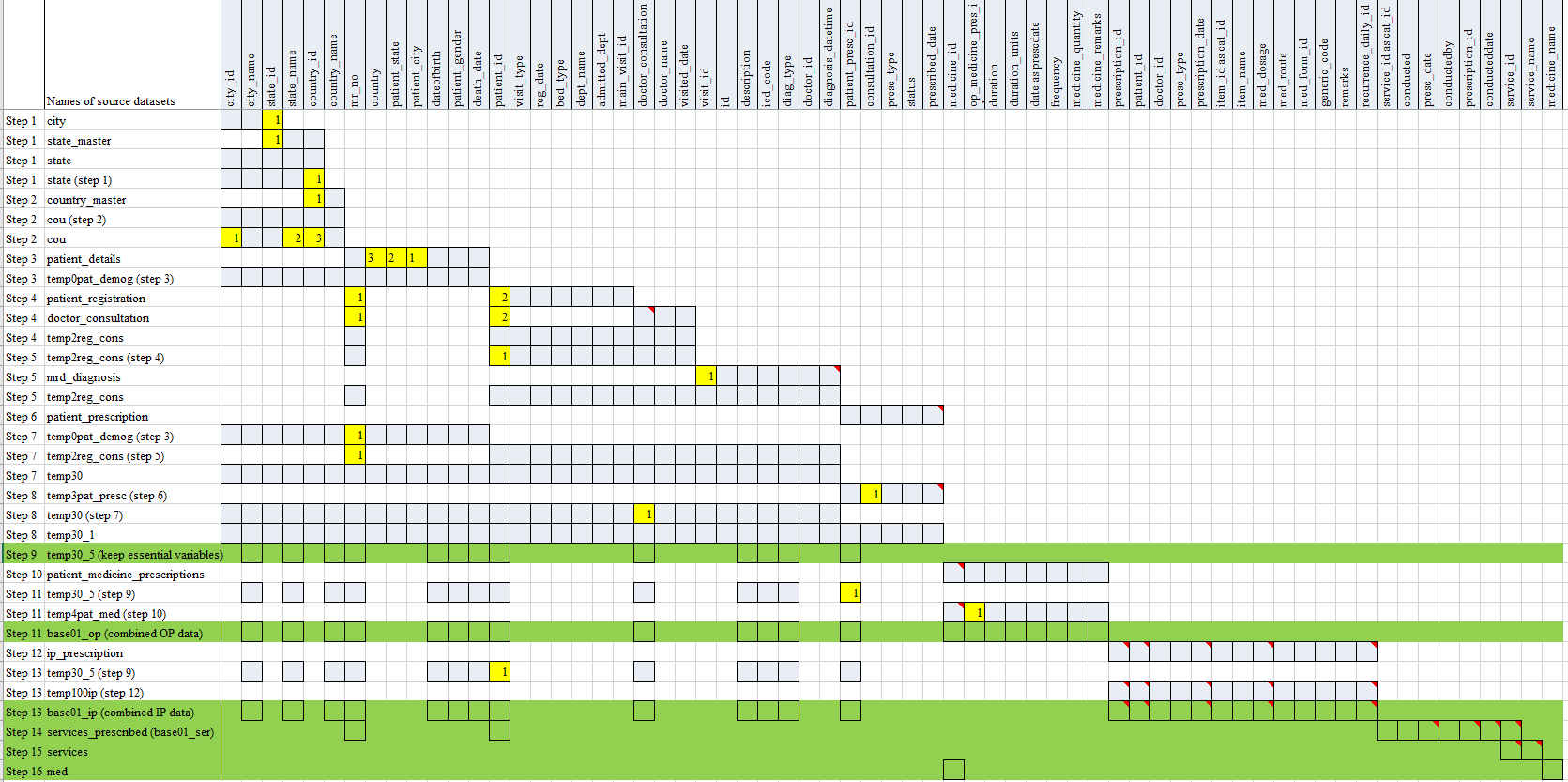
Methodology section 3a, 3b, 3c, 3d:

Picture option 3a:

Section 1: extraction from Source database (13 datasets, ~65 variables)

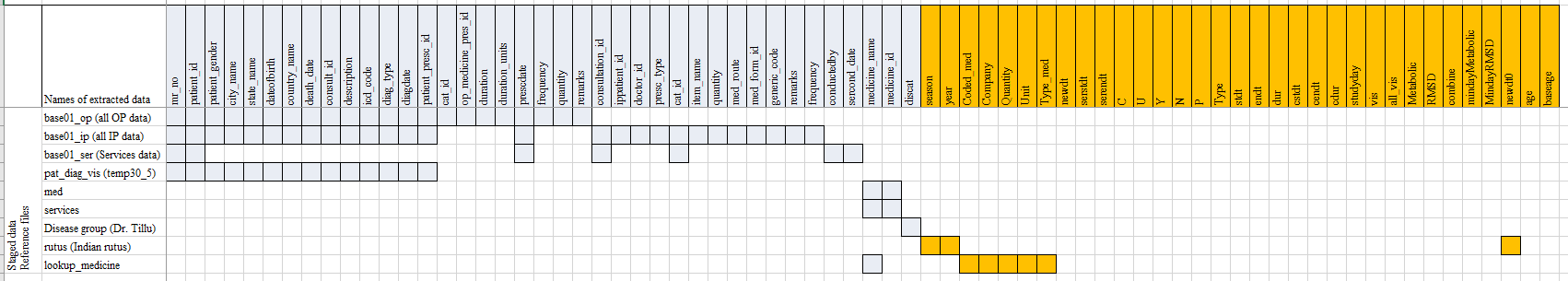
Picture option 3b:

Staged data converted into 6 datasets



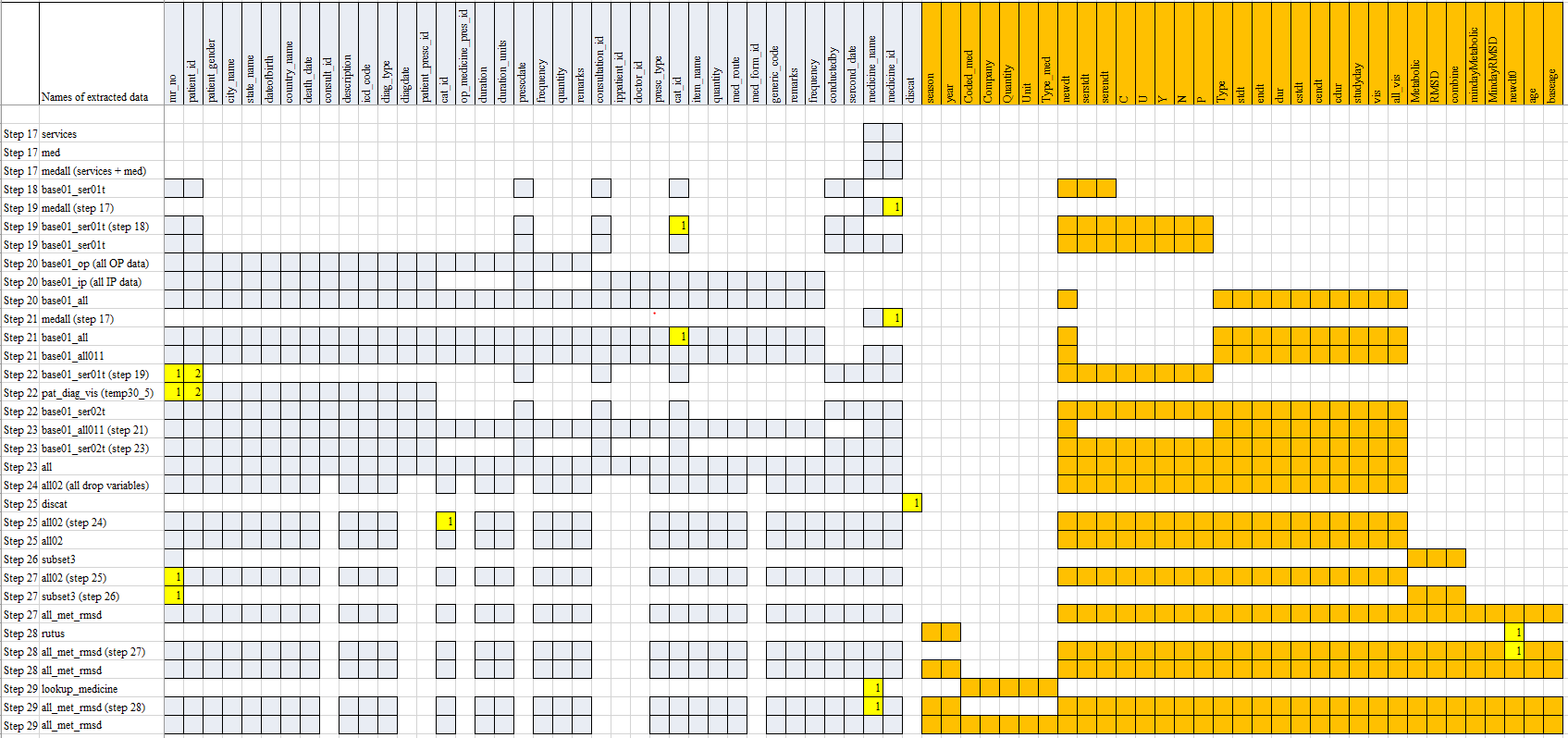
Picture option 3c:

Staged data (6 datasets, ~40 variables) + 3 reference files created based on inputs from experts



Picture option 3d:

1 Final dataset with ~30 source variables and ~30 new derived variables



Picture Option 4: (results section)

|  |  |  |  |
| --- | --- | --- | --- |
| Source data (SQL data file) | Staging data (csv files / R data files) | Data ware house (R data files) | Usage |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |  | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp  Longitudinal Patient data with disease, medication and Ayurvedic services information  ~30 variables from source  ~ 30 variables derived  ~50,000 patients  ~17,000+ patients: subsetted version for RMSD and Metabolic | Creation of additional analysis datasets  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |  |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | Actual analysis  Analytics - Free people icons |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | Learning from the existing database to be given back as learning  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp |  |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | Reference files for derivations and filtering of data | Clinical communication  C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\553C02C8.tmp |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp (txt file) |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp (txt file) |
| C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp | C:\Users\mahajvi1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9805807.tmp (txt file) |

Details of the reference dataset “01adsl\_met\_rmsd”

The subsequent table presents the details of the reference dataset

|  |  |  |
| --- | --- | --- |
| Variable name | Description | Derivation |
| mr\_no | Unique Patient ID | Source variable, no derivation needed  E.g. MR000001, MR040237, etc. |
| patient\_gender | Patient gender | Source variable, no derivation needed  E.g. M, F |
| patient\_id | Visit ID | Source variable, no derivation needed, the hospital database captures unique visit ID for each visit. |
| city\_name | City name | Source variable, no derivation needed |
| state\_name | State name | Source variable, no derivation needed |
| country\_name | Country name | Source variable, no derivation needed |
| dateofbirth | Date of birth | Source variable, no derivation needed, for some patients this is missing |
| newdt0 | Date of visit to hospital | Date of visit to hospital in numeric format  All the InPatient visits, OutPatient visits and Service related visits are combined from source datasets into dataset, unique visit and date combinations are created. |
| newdt | Date of visit to hospital | Character version of newdt0 |
| vis | Visit | 1. Based on all the InPatient visits, OutPatient visits and Service related visits unique visit numbers are created. 2. Visit numbers are numeric values from 1 to n, based on current version of data; a patient has maximum number 323 visits. |
| all\_vis | All visits | This variable contains maximum number of for each patient. all\_vis = max(vis) grouped by each mr\_no |
| all\_ip | All IP visits | This variable contains maximum number of for each patient for IP type of visits. all\_vis = max(vis) grouped by each mr\_no and visit type is IP. |
| all\_op | All OP visits | This variable contains maximum number of for each patient for OP type of visits. all\_vis = max(vis) grouped by each mr\_no and visit type is OP. |
| studyday | Study day | studyday = 1 when the visit minimum visit or first visit for a patient, else studyday is calculated as newdt0 – min(newdt0) + 1.  Studyday is never missing and never less than 0 for the dataset created. |
| age | Age of patient at that visit | If date of birth is non-missing for a patient, then age is calculated as round( (anydate(newdt) - anydate(dateofbirth) + 1)/365.25, digits = 0 ) |
| baseage | Age of patient at the first visit | Age at vis = 1 for each patient is stored as base age |
| death\_date | Date of death | Source variable, no derivation needed |
| cstdt | Min Start date | cstdt = min(newdt) |
| cendt | End date | cendt = max(newdt) |
| cdur | Total duration in days | cdur = max(newdt) - min(newdt) + 1 |
| stdt\_IP | Start date of IP visits | Minimum visit date for IP visits for each patient |
| endt\_IP | End date of IP visits | Maximum visit date for IP visits for each patient |
| dur\_IP | Duration of IP visits | dur\_IP = endt\_IP – stdt\_IP + 1 |
| stdt\_OP | Start date of OP visits | Minimum visit date for OP visits for each patient |
| endt\_OP | End date of OP visits | Maximum visit date for OP visits for each patient |
| dur\_OP | Duration of OP visits | dur\_OP = endt\_OP – stdt\_OP + 1 |
| serstdt | Service Start date | Minimum visit date for Service visits for each patient |
| serendt | Service End date | Maximum visit date for Service visits for each patient |
| Code | Code | Source variable, no derivation needed, ACD code |
| description | Description | Source variable, no derivation needed, description |
| Type | Type of visit | This variable identifies a visit either as IP or OP based on visit classification |
| diag\_type | Diagnosis type | Source variable, no derivation needed:  Primary or Secondary |
| year | Year | Year part of the newdt variable |
| season | Indian seasons | Derivation of Indian seasons based on the date variable for each visit:  # Add Indian rutus as new variables  # <https://www.drikpanchang.com/seasons/season-tropical-timings.html?geoname-id=1277333&year=2010>   * 01 Vasant Rutu * 02 Grishma Rutu * 03 Varsha Rutu * 04 Sharad Rutu * 05 Hemant Rutu * 06 Shishir Rutu |
| C, N, P, U, X, Y | Values related to Services offered to patients | Source variable, no derivation needed:   * C- Cancelled * U - Condn. Unnecessary * Y -Conducted * N - Not Conducted * P - Partially Conducted |
| presc\_type |  | Source variable, no derivation needed |
| medicine\_name | Medicine name | Source variable, no derivation needed  Prescribed medicine names follow a certain predefined naming convention. Medicine name + Quantity + Producer’s name are the details recorded for each prescribed medicine. |
| item\_name | Source value of medicine name | Source variable, no derivation needed |
| quantity | Quantity of prescribed medicine | Source variable, no derivation needed |
| med\_route | Route of administration of prescribed medicine | Source variable, no derivation needed |
| generic\_code |  | Source variable, no derivation needed |
| remarks | Notes provided by doctors for medicines | Source variable, no derivation needed |
| frequency | Frequency of prescribed medicine | Source variable, no derivation needed |
| duration | Duration of prescribed medicine | Source variable, no derivation needed |
| duration\_units | Unit for duration of prescribed medicine | Source variable, no derivation needed |
| Coded\_med | Only name of medicine | Derived from medicine\_name |
| Company | Name of the company producing the drug | Derived from medicine\_name |
| Quantity | Quantity of prescribed medicine | Derived from medicine\_name |
| Unit | Unit of prescribed medicine | Derived from medicine\_name |
| Type\_med | Type of medicine | Derived based on medicine\_name. Classified into different kinds of medicines, e.g.   * Ghritam * Kashayam * Asavam * Aristham * Bhasma * Abhyanga * Cream * Rasayanam * Tablet / Gulika / Vati * … |
| cat\_id |  |  |
| distype | Disease type | Disease type as OTHER, RMSD, Metabolic   1. If a disease code is present in Metabolic list then the value is Metabolic 2. If a disease code is present in RMSD list then the value is RMSD 3. Any other disease is classified as OTHER |
| Metabolic | Metabolic | If a patient has reported any Metabolic disease at least once then that patient is given value Metabolic = 1, else Metabolic =0 |
| RMSD | RMSD | If a patient has reported any RMSD disease at least once then that patient is given value Metabolic = 1, else Metabolic =0 |
| combine | Metabolic  RMSD  Both | 1. If a patient is classified only as Metabolic diseased patient then combine = 1, 2. If a patient is classified only as RMSD diseased patient then combine = 2, 3. If a patient is classified as Metabolic as well as RMSD diseased patient then combine = 99 |
| Minday Metabolic | First day on which reported metabolic disease | First day on which any metabolic disease has been reported by a patient. |
| Minday RMSD | First day on which reported RMSD disease | First day on which any RMSD disease has been reported by a patient. |