Document Information

Analyzed document	Vinay_Thesis Working-Jan2023.pdf (D156288495)
Submitted	1/19/2023 5:33:00 AM
Submitted by	Giridharan R
Submitter email	registrar@tdu.edu.in
Similarity	3%
Analysis address	registrar.tdu@analysis.urkund.com

Sources included in the report

W	URL: https://pharmaphorum.com/views-and-analysis/a_history_of_the_pharmaceutical_industry/ Fetched: 1/19/2023 5:33:00 AM	1
W	URL: https://cioms.ch/#:~:text=The%20Council%20for%20International%20Organizations,WHO%20and%20UNES... Fetched: 1/19/2023 5:33:00 AM	1
W	URL: https://www.xplenty.com/blog/data-transformation-explained/ Fetched: 1/19/2023 5:34:00 AM	9
W	URL: https://www.ironsidegroup.com/2019/07/16/data-preparation-business-analytics/ Fetched: 1/19/2023 5:34:00 AM	1
W	URL: https://www.cdisc.org/ Fetched: 1/19/2023 5:34:00 AM	2
W	URL: https://www.transceleratebiopharmainc.com/ Fetched: 1/19/2023 5:34:00 AM	1
W	URL: https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R Fetched: 1/19/2023 5:33:00 AM	27
W	URL: http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges Fetched: 1/19/2023 5:33:00 AM	7
W	URL: https://www.wma.net/policies-post/wma-declaration-of-helsinki-ethical-principles-for-medical-r... Fetched: 1/19/2023 5:33:00 AM	1
W	URL: https://www.postgresql.org/ Fetched: 1/19/2023 5:34:00 AM	1

Entire Document

1 | P a g e Analysis of hospital based Ayurvedic clinical practice to gain Real World data knowledge Vinay Mahajan PhD thesis Guides: Dr. Ashwini Godbole, Dr. Girish Tillu, Dr. Ashwini Mathur

9 | P a g e Acronyms ACD: Ayurvedic Classification Dictionary A-HMIS: Ayurveda Hospital Management Information System AYUSH: Ayurveda, Yoga and Naturopathy, Unani, Siddha and Homeopathy CCRAS: Central Council for Research in Ayurvedic Sciences CDISC: Clinical Data Interchange Standards Consortium COPD: Chronic Obstructive Pulmonary Disease COSTART: Coding Symbols for Thesaurus of Adverse Reaction Terms CTC: Clinical Toxicity Grade DARWIN EU: Data Analysis and Real-World Interrogation Network EHR: Electronic Health Record EMA: European Medicinal Agency EMR: Electronic Medical Record EU: European Union GCP: Good Clinical Practice GCRP: Good Clinical Research Practice GDP: Gross Domestic Product HM: Hospital Management I-AIM: Institute of Ayurveda and Integrative medicines ICD: International Classification of Diseases ICH: International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human Use IDC: International Data Corporation IDC: International Data Corporation IP visit: In-Patient visit ISO: International Organization for Standardization LOINC: Logical Observation Identifiers Names and Codes MedDRA: Medical Dictionary for Regulatory Activities NCI: National Cancer Institute, USA NEJM: New England Journal of Medicine OP visit: Out-Patient visit PMDA: Pharmaceuticals and Medical Devices Agency (Health Authority Japan) RCT: Randomized Controlled Trial RMSD: Rheumatic and Musculoskeletal disease RWD: Real World Data RWE: Real World Evidence SQL: Structured Query Language TDU: The University of Trans-Disciplinary Health Sciences & Technology THERAN: THE Research Application Nexus TKDL: Traditional Knowledge Digital Library UHC: Universal Health Coverage US FDA: US Food and Drug Administration

10 | P a g e WHO: World Health Organization WHO-ART: World Health Organization Adverse Reactions Terminology WHO-DDE: World Health Organization Drug Dictionary Enhanced

11 | P a g e Preface Various electronic equipment like computers, mobile devices, wearables, and other sensors collect and store huge amounts of health-related data. This explosion of data carries potential to better design and conduct clinical studies to answer questions previously thought infeasible. Advancement of cutting-edge analytical capabilities is allowing researchers to analyze and comprehend this data at greater depths, permitting medical product development and approval at an accelerated speed [1]. Real world data (RWD) is the information relating to patient health status and/or the delivery of health care routinely collected from a variety of sources like epidemiological studies, clinical practice, already published articles to answer questions previously thought infeasible. Approval of Ibrance by US FDA for male breast cancer, a drug already approved for females and French health authorities allowing a Real World Evidence (RWE) study of 600+ patients, over a period of 18 months, for a conditional re-imbursement scheme in COPD, are a couple of recent examples of approvals using RWD data. A study carried out by Clarivate Analytics, USA, reports 27 (non-exhaustive list), &gt;5% of all approved drugs, examples of drug approvals by US FDA, EMA, Japan's PMDA and Health Canada, across broad spectrum of medicines between years 1998 and 2019 using RWD from Electronic Health Records and registries. These data were used either as primary data, when non-comparative data were available to demonstrate tolerability and efficacy, or as a supportive data when validating findings. This provides increasing usage of "naturally reported data" in drug approvals in modern biomedicine. These examples provide evidence of novel use of data, which may have otherwise gone unused. The power available to society would have never been unearthed if not for this way of use of RWD [2]. Is Ayurvedic area dealing with the same type of challenge of not realizing the potential of available data? Just to give a glimpse of enormity of data: more than 10 crore number of patients have been reported on AYUSH website (As of May 2020). More than 140+ countries have population of less than 10 crores [3]. It is safe to assume that the conceptual developments in Ayurvedic knowledge base have taken place through everyday observations and basic laws of nature. These fundamentals have been adjusted to the relevant times as per the passage of time based on observations and experiences, where there are no artificial restrictions on usage of medicines, duration of treatment or type of patients to treat, which is next to impossible in a protocol driven clinical trial setting [4] [5]. Taking inspiration from respected Prof Patwardhan's quote, "Charaka would not have ignored modern technologies if they had been available during his time" [6], this study attempts to discover hidden wealth of Ayurveda related information in EHRs created at TDU hospital using modern methods of data sciences and statistical programming. Since 2011 to October 2017, the hospital database contained data for approximately 51,000 patients, more than 1,50,000 visits, close to 900 disease types and more than 3,000 variations of medical procedures [7]. The proposed study "Analysis of hospital based Ayurvedic clinical practice to gain Real World data

12 | P a g e knowledge" targets the methodological and learning framework as well as creation of many tools based on free softwares for various stakeholders in following categories: • Hospital managements, clinicians, and patients • Universities and learning institutes – clinical communication, researchers to build vital evidence-base • Policy makers – AYUSH and relevant ministries • Healthcare providers - Ayurveda Healthcare systems, General healthcare systems

13 | P a g e 1 Introduction 1.1 Origins of Pharmaceutical Industry

52% **MATCHING BLOCK 1/51** W

The origins of the pharmaceutical industry go back to the apothecaries and pharmacies that gave traditional therapies going back to the Middle Ages.

The modern medicines regulation began only after revolutionary development in the 19 th century life sciences, in chemistry, physiology and pharmacology, which put a robust foundation for the modern drug research and development [8]. In year 1947, one of the first international standards was established due to the ethical and medical misconduct of the Nazis in World War II. The Nuremberg Code created 10 standards that include voluntary informed consent, risk to subject must be weighed against benefits, actions that injure human subjects must be avoided, and the subject has the right to end the experiment at any time [9]. Unfortunate events like deaths due to diethylene glycol poisoning in the US in 1930s, the thalidomide disaster in late 1950s, catalyzed the development of medicines regulation. By the year 1964 World Medical Assembly developed the Declaration of Helsinki. It became a key international document describing the ethical principles that underlie GCP. These principles provide guidance in medical research involving human subjects. The document has been amended on a regular basis, with its most recent update done in October 2013, in Brazil [10]. The World Health Organization (WHO) and

55% **MATCHING BLOCK 2/51** W

the United Nations Educational, Scientific and Cultural Organization (UNESCO) in 1949 jointly established The Council for International Organizations of Medical Sciences (

CIOMS) [11]. The use of statistics as a scientific field to support R&D of new medicines grew multifold since the Kefauver-Harris Amendments (1962) and continues to grow [12]. These clearly stated that the Food and Drug Administration (FDA) would require "substantial evidence" of the impact of a drug in a clinical trial setting and proof of safety will not be sufficient for new drug approvals as had been required in the initial regulatory requirements. It issued a document titled "Proposed International Guidelines for Biomedical Research Involving Human Subjects" to help apply the principles of the Declaration of Helsinki and the Nuremberg Code, in 1982. These guidelines were revised in 1992, which resulted in the "International Ethical Guidelines for Biomedical Research Involving Human Subjects." Due to the increasing global footprint of clinical research, the International Conference on Harmonisation (ICH) was established in 1990 [13]. In 2016, the name was changed to International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH). ICH created guidelines based on 4 pillars of quality, safety, efficacy, and regulatory obligations to protect public health. These guidelines get updated on a regular basis [13]. These efforts have shown that, in the USA and all over world, since 1970s, the value of medicine has been clearly demonstrated. Human population have exhibited a longer life expectancy, a lower infant mortality rate, and the higher quality of life. 1.2 Elaboration of Clinical Trials: Origin of RCT Blinded Trials Randomized Controlled Trial (RCT) is a classical research design of randomly allocating participants to one or the other treatments under investigation. Randomization is the fundamental characteristic of an RCT, and it describes the random distribution of participants to the study arms. RCTs aim towards supporting a conclusion that the difference in the outcomes among participants in study arms was exclusively caused by the intervention, as randomization
14 | P a g e equalizes the study group in all other factors. Thus, RCTs set the standard of excellence in health sciences research. Although, randomization reduces the bias of participant assignment to the intervention and control group, it does not rule out the chance of bias, for example, due to investigator/caregiver or the patients themselves or bias generated during subjective adjudication of an outcome variable. However, RCTs do help in reliable description of causality between the intervention and outcome. Blinding as a procedure, assists in monitoring and controlling several types of biases that might unintentionally creep into the study. Two major biases, namely performance bias and ascertainment bias can be controlled using blinding. Blinding deals with four groups of people associated with a trial: the study participants, the investigator(s), the outcome assessor(s), and the data analyst(s). Depending upon the groups of people blinded, trials are categorized as open label, single blinded, double blinded, triple blinded, and quadruple blinded trials [14]. Physicians and clinical researchers have remained confident that RCTs deliver the most rigorous test of preventive, diagnostic, and therapeutic interventions. They are universally denoted as the "gold standard" of experimental medical analysis, as an undisputable starting point in preventive, diagnostic or therapeutic evaluation. An article written by Alvan Feinstein and Ralph Horwitz in The New England Journal of Medicine (NEJM) in December 1982, is the first instance of RCTs being referred to as "gold standard" [15]. 1982 appears to be late date for the first usage of the "gold standard" terminology. The authors say that they were eager to be proven wrong on this 1982 date. But despite massive searches across many textbooks, conference materials, journals, and archival collections have not thrown up any earlier date [15]. While probing details regarding RCTs on the ClinTrial.gov website the following data was seen on 24 th April 2020 [16], Link. Total trials in the ClinTrial.gov database was 3,36,905. The estimated % of RCTs ranged between 3% and 20%. Remaining 80% to 97% of the registered trials are not RCTs (Figure 1-1). As of April 2020, Clintrial.gov database had 19% RCTs of all the studies registered in the database. This provides us with enough evidence that a lot of research work is carried outside of the RCT framework. Figure 1-1: RCTs on ClinTrials.gov

15 | P a g e This is a screenshot taken from the ClinTrial.gov website showing number of Randomized Clinical Trials. Estimates of RCT range between 3% considering “Randomized control trial (1,862)” and “randomized clinical trial (6,234)” and 20%: considering “Randomized control trial (1,862)”, “Randomized (56,945)” and “randomized clinical trial (6,234)” 1.3 Modern Hospitals, Everyday Clinical Practice and Healthcare Environment Health information provided by hospitals is vital for public health administrators. Such data, accumulated from all the relevant hospitals, are essential in formulating health policy planning for the district / state / country, by matching this data with population statistics and other demographic data, and the health resources of the district / state / country. This medical information, containing not only data about the patient's illness but also the success or failure of therapy, could be a precious source of clinical training for medical students, nurses, and related health care work force. This could also be basis of new medicines and therapies. The development of new therapeutic drugs is built upon careful observations of experienced physicians, and the comparison and analyses of administered drugs, gathered from the patient's data. This in turn leads to continuous medical progress in uncovering new entity of a disease or a new method of diagnosis [17]. One trend all modern hospitals have in common is that the amount of processed information generated per patient is constantly increasing e.g., Electronic Medical Record (EMR), patient 16 | P a g e charts, CT scans, X-rays, ECGs, pathology reports, wearable devices, sensor data, etc. When such information is properly compiled and aggregated, it could provide data for efficient hospital administration. How each patient is treated can yield important medical data, while the number of patients handled and treated, grouped by sex, age, and diagnosis, provides basic information that could be useful for administrators [18]. This information is processed manually or partially automatically in many instances. This inefficient method results in delays in reaching crucial treatment decisions, for instance when test results fail to get back to the physicians in time, or in delayed action by administrations when a prompt response might be needed. 1.4 Real world evidence and observational studies Real-world data are data captured in an observational manner, in a natural setting. Ordinary clinical practice, producing a never-ending flow of results from everyday practice, can be viewed as infinite sequence of unsystematic observational studies. There must be rational use of this outcome data from patients collected from everyday clinical practice as it is far more representative of the general population and medical practice than those data coming from formal clinical trials [19]. Practice-based medicine is an important way to advance science [20]. For this to happen, the treating doctor should be a “clinician researcher investigator”. Doctors even in their busy practice, can see patterns, then formulate a research hypothesis, and then proceed to test it. Hence, a doctor can be good at good clinical practice (GCP) and good clinical research practice (GCRP). One can support the other and vice versa, and this way medicine advances [21]. Investigation of literature across time points from year 2000 to year 2020 shows that well designed observational studies produced treatment effects comparable to RCTs: Year 2000: 99 reports across 5 clinical topics covering observational studies provided comparable results like those of the randomized, controlled trials. There was no systematic over estimation of treatment effects seen [22]. Year 2007: RCTs provides one kind of knowledge but could prevent us from understanding other properties of a drug. When observational epidemiological studies are carried out correctly, they can be both more conceptually complex and more powerful than an average RCT, especially in assessing drug safety. Both kinds of research if done with rigor and with scientific humility, can be supplementary / complementary to each other [23]. Year 2014: There was little evidence found for significant effect estimate differences between observational studies and RCTs regardless of specific observational study design, heterogeneity, or inclusion of studies of pharmacological interventions. The results highlight that level of heterogeneity in meta-analysis of RCTs or observational studies is an important factor [24]. Year 2020: Cumulative evidence suggests that appropriately conducted RWD studies have the potential to support regulatory decisions in the absence of RCT data. Further work may be needed to better show the settings in which RWD analyses can robustly and consistently match

17 | P a g e the results of RCTs and the situations in which they cannot match. After careful consideration of the potential for bias, regulators can then determine when they would accept RWD in place of an RCT [25]. The data from RWE studies and Observational studies have a wealth of information which if processed accurately and summarized in a structured manner can lead us to numerous assertions and confirmations. There have been many attempts to generate clinical evidence from primary health care by systematic utilization of patient records. But this has not been easily possible due to the deficient clinical data, in-accurate input “garbage in” leading to “garbage out”, insufficient follow-up, and very few fully completed case records with risk factors, co-morbidities, etc. Both clinical practice and research based on the principles of primary care could contribute substantially as follows: Clinical practice (1) uses medical knowledge and supports confirmation of research findings, (2) focuses on individual patients but still generates data, (3) has a short action span and immediate reward, (4) it regards authority, follows custom, earns income, and encourages research. Clinical research complimentarily (1) creates knowledge by focusing on future clinical practice, (2) focuses on groups of patients to generate data, (3) has a long action span and delayed reward of discovery, (4) questions authority, challenges custom, earns reputation, and enriches practice. 1.5 EHR across the globe In early 2000s countries like Canada, UK, New Zealand, Estonia, etc. started collecting clinical practice data at a national level. The key challenges experienced while implementing Electronic Medical Records (EMR) faced by them were: infrastructure creation, policy & regulations, standards & interoperability, and research, development & education [26]. Resource shortages, amplified by various socio-economic problems pose substantial difficulties for development of workable healthcare solutions which are global in nature. Healthcare development is one of most important aspects for the progress of both social and economic development of the world. More than 1/4 th of the world’s population has unmet healthcare needs. On World Health Day 2018, WHO introduced the concept of Universal Health Coverage (UHC): healthcare for everyone and everywhere. However, this has not been completely effective due to the financial limitations within systems and factors like aged-population-related chronic diseases influence the availability, accessibility, and quality of care [27]. However, EMR data has been widely used for analysis and many papers have been published. These have generated supportive data for a variety of clinical outcomes, evaluation methods, and implementation of new technology or interventions along with awareness of unintended consequences and thus supporting the clinical practice decisions and aiding to improve the healthcare process or clinical outcomes for the patients. 1.6 Role of Statistics, Analyst, Programmer Statistics emerged as an extended stream from mathematics, operational research, and economics. Application of statistics in various fields of research like genomics, epidemiology,

18 | P a g e nutrition, biological science, biomedical research connoted the word “biostatistics”. Apart from biological sciences, statistics is applied in variety of other fields like market research, insurance, trades and stocks, banking etc. The 1990s presented explosive increase in applications of computationally demanding methods. These methods were naturally based on statistical principles, due to the emphasis on exploratory nature of the problem and the importance of data. Statistical modeling has become more complex due to the volume of data, computational requirements, and varied data sources. These developments led to creation of new roles like statistical programmer - statistical analyst, clinical programmer - clinical analyst [28]. Clinical data analysts (or clinical informatics analyst) are healthcare professionals responsible for confirming the validity of scientific experiments and data gathered. They apply their knowledge of data acquisition, data management, data analysis, and data interpretation to healthcare data, providing actionable insights that doctors, clinical scientists, and others can use. They may be responsible for automating internal and external reports, creating executive-level dashboards, and presenting information to help various stakeholders understand the operational impact of the data. Data analysts ensure that processes and protocols associated with clinical research are followed, thus improving overall care [29], [30], [31]. They provide data insights that drive clinical process improvement, such as reducing readmissions and hospital-acquired conditions [32]. Clinical data analysts generate and provide the results of clinical business intelligence to management and all stakeholders. They coordinate with other researchers (e.g., clinical strategy, clinical operations) to determine the questions to be answered as well as the appropriate measures that should be taken to ensure data analysis proves to be useful. These roles combine strengths from multiple areas to build a powerful storyline on six dimensions of quality care: safe, effective, patient centered, timely, efficient, and equitable [33]. 1.7 Traditional Chinese Medicine history and philosophy Traditional Chinese Medicine (TCM) has been in practice for more 3000 thousand years [34] through Shang Dynasty (1600-1046 BC), Han Dynasty (206 BC–220 AD), and Zhou dynasty (1100-221 BC) of China [35]. Classic of Changes (Yi Jing) and Classic of poetry (Shi Jing) are considered the oldest medical writings available for TCM [34]. Subsequent foundation of TCM are based on the four classics across different era are Inner Canon of the Yellow Emperor (Huang Di Nei Jing, ~26 BCE), Yellow Emperor’s Canon of Eighty-One Difficult Issues (Nan Jing, ~106 CE), Treatise on Cold Damage Disorders (Shang Han Lun, ~206 CE), and Shennong’s Materia Medica (Shen Nong Ben Cao Jing, ~220 CE) [34]. Acupuncture, herbal medicine, diet, movement, and manual therapy are considered the five main branches of TCM [36]. TCM is based on the Five Elements theory, Wood, Fire, Earth, Metal, and Water which describes the human body. A blend of Chinese philosophy, culture, ritual, and medical practices, TCM and herbal medicine demand a broad understanding which is not limited to science [37]. Zheng (syndrome), a basic unit in TCM, decides the therapeutic methods. The process of how to get the outcome is called differentiation of Zheng, which is based on the physiology and pathology of TCM helps in determining the course of treatment [36]. This narrative practice, either in the form of classics or passed down orally, retains the medical knowledge of ancient Chinese people. Until the 20 th century, most Chinese could only use TCM as a medical service.

19 | P a g e After the modern medicine breakthroughs TCM has been going re-assessment efforts based on modern scientific methods [34], [38], [39], [40], [41], [42]. One such example is of Artemisinin and its derivative dihydroartemisinin which was referred to in a medical classic by Ge Hong (284–346 CE). This drug was tested by Tu and her colleagues using modern methods. It has saved millions of lives threatened by malaria throughout the world [43], [44]. For this phenomenal work, Tu was awarded the 2011 Lasker Award for clinical research and the 2015 Nobel Prize in Physiology or Medicine [45], [46]. Researchers have found use of “case record” in advancing traditional knowledge through the history of TCM in various phases of maturity. First phase defined by the creation and development of the discipline of TCM medical records; second phase covers the standardization of writing structure of TCM medical records; third phase outlined by books written by veteran TCM doctors on recording and studying TCM medical records. Most recent stages of development account for the explosion of TCM case reports published in journals; the establishment of TCM medical records databases and application platforms integrating computer programs and artificial intelligence; and the last but not the least, many reporting guidelines have been developed in order to improve the reporting quality of case report in TCM [35]. A short review of the available databases: Post year 2000, many researchers have developed many web-based freely available databases covering topics like molecular properties, substructures, TCM ingredients, and TCM classification, based on intended drug actions. TCM Database@Taiwan database contains more than 20,000 pure compounds isolated from 453 TCM ingredients [47]. SymMap is a symptom mapping database integrating TCM with modern medicine in common aspects at both the phenotypic and molecular levels [48]. TCMSP: traditional Chinese medicine systems pharmacology database and analysis platform. It consists of all the 499 Chinese herbs registered in the Chinese pharmacopoeia with 29,384 ingredients, 3,311 targets and 837 associated diseases. It contains tools for visualization and analysis of TCM results on the network level for chemists, biologists, and pharmacologists [49]. TCMIO: Traditional Chinese Medicine on Immuno-Oncology database covering the mechanisms of TCM in cancer immunity and TCM- inspired identification of novel drug leads for cancer immunotherapy [50]. TCMID: traditional Chinese medicine integrative database for herb molecular mechanism analysis [51]. YaTCM: Yet another Traditional Chinese Medicine Database for Drug Discovery [52]. A short review of the clinical trials and registries for TCM: There are not many registries for TCM yet, but some are getting developed. The China Stroke Registry for Patients with Traditional Chinese Medicine (CASES-TCM) study is a prospective, multicenter, observational disease registry aiming to register 20,000 hospitalized patients [53]. The China Amyotrophic Lateral Sclerosis Registry of Patients with Traditional Chinese Medicine (CARE-TCM) provides an opportunity to better understand which TCM interventions patients with ALS are receiving, what the characteristics of patients with ALS are, and how these interventions impact clinical measures. This registry was initiated in March 2021. This study

20 | P a g e includes a voluntary nationwide registry. Detailed data collection will be performed every 3 months for 5 years. Baseline characteristics and 5-year survival will be collected [54]. A review of TCM trials on Clintrials.gov till Sept 2015 shows 1270 TCM trials listed on the website. Most of these trials were carried out either in China or in the USA. More than 50% of the trials were for acupuncture and 35% for herbal medicines. 55.7% that were studies with less than 100 enrolled subjects. 8.7% of completed studies had reported results of trials. Even though there are issues, it is good to see these studies getting registered on Clintrials.gov [55]. Text data extraction using deep learning models has been tried out in TCM on literature, but online search did not filter up many such examples on patient data [56]. My search could not filter Real world evidence studies within TCM area. Since early 2000s there have been some efforts put in at the national level on outcomes research and big data. In year 2010 officially RWE concept was rolled out for TCM, and 2016 onwards this has been spread across the country. Chinese Drug administration has started using RWE in pre-approval process for Supporting evidence for investigational new drug. Chinese national authority NMPA has issued a guidance for “Clinical development of traditional Chinese medicine hospital preparations” under the overall guidance for “Key Considerations in Using Real-World Evidence to Support Drug Development” Center for Drug Evaluation, NMPA May, 2019 [57], [58]. This data shows that within TCM area there is an opportunity to carry out work based on naturally reported patient level data. 1.8 The Indian context India has a mixed system of healthcare consisting of government hospitals, private hospitals, family doctors and private medical practices. We see this trend reflected in the actual health seeking behavior of communities where people tend to combine medicine systems like Allopathy, Ayurveda, Siddha, Sowa Rigpa, Unani, Homeopathy and Yoga depending on the nature of the disease [59]. India, having one of the largest rural populations in the world, it becomes imperative to be both inclusive and optimal in the use of available resources from any stream of healthcare. Along with global healthcare challenges, Indian healthcare faces specific issues like inadequate healthcare resources, insufficient funding, poor healthcare infrastructure and rural–urban disparity. The national healthcare budget is a lowly single digit percentage of the Gross Domestic Product (GDP). The number of doctors, nurses and health workers are small in comparison to high population in certain regions. The availability and utilization of technology is also at the lower end of the usage [60]. Integrating modern medicine and AYUSH systems, acronym for Ayurveda, Yoga and Naturopathy, Unani, Siddha and Homeopathy, folk medicines as well as technological advances into the India healthcare system can generate new opportunities for healthcare. These efforts will be noteworthy where healthcare systems have limitations in terms of infrastructure, expertise, and human resources. This integration will not only allow utilization of the positives of each of the medical streams but will generate new data-based evidence due to integration possible using technology. Ongoing monitoring and updating patient’s data together with consistent updates to physicians can help in identifying disease and treatment patterns. Through this analysis,

21 | P a g e authorities or the government can employ specific precautions, or even start necessary facilities to avoid healthcare issues before the situation gets out of hand [61]. The application of such assessed and analyzed data are substantial as the information brings out more and more knowledge about an area or a person. In under-resourced countries such as India, the resultant benefits are expanded further, suffering is reduced, hospital resources are saved and socio-economic improvements that lift a nation’s wellbeing are recognized. Moreover, India could achieve improved healthcare delivery, care audit, epidemiological study and quick response to epidemics and bring economic benefits to individuals by reducing the healthcare cost. 1.9 National level efforts AYUSH This section covers efforts put in by the Government of India at the National level. The Ministry of AYUSH is a part of Government of India to promote and expand use of AYUSH systems of health care and medicine in India. To administer the different medicine systems encompassed by the Ministry of AYUSH, it has five research councils or departments, affiliated courses, and affiliated national institutes [3]. One of AYUSH affiliates, CCRAS has developed National AYUSH Morbidities and Standardized Terminologies E-Portal NAMASTE portal. It provides information about Standardized terminologies and Morbidity codes along with dedicated data entry module for updating morbidity statistics in consolidated form as well as on real time basis [62]. To publicize the merits of AYUSH systems across the globe, a web-based portal for Research publications was launched in 2011, which is being maintained by NIIMH Hyderabad [63]. Ayurveda Hospital Management Information System (A-HMIS) is a complete IT platform for all functions of health care delivery systems and patient care in AYUSH centers. THERAN (THE Research Application Nexus): HMIS is developed by Siddha Central Research Institute, Chennai under Central Council for Research in Siddha. These are 2 additional examples of ongoing work under Ministry of AYUSH [64]. Ministry of AYUSH has created AYUSH GRID to cohesively implement projects under Digital India Movement. It is an amalgamation of IT projects planned for advancement of AYUSH pan India [65]. The dashboard available on AYUSH homepage (accessed on 25 th May 2020), reveals the following facts: the screenshot shows almost 10 crore patients treated at some or the other point (Figure 1-2 , Figure 1-3). There are approximately 140+ countries with less than 10 crore population, which provides a perspective on the size of data available at the AYUSH level. It remains to be seen how to convert “data into information, information into knowledge and knowledge into wisdom” [66]. Figure 1-2: Screenshot of a dashboard from AYUSH website

22 | P a g e Dashboard from AYUSH website covering: AYUSH institutes, Infrastructure projects, National medicinal plants board, Health infrastructure, Education and communication, Research, Practitioners and patients, Drug Industry, Budget, and Schemes Figure 1-3: Ayurvedic practitioners and patients from AYUSH dashboard Screenshot from AYUSH website covering: Approximately 4.5 lakhs Ayurvedic practitioners and more than 10 crore treated patients. Data regarding different medical systems within AYUSH has been presented. Ayurvedic practitioners and patients treated by ayurvedic interventions are considerably more than by any other traditional medical practice.

23 | P a g e 1.10 Science of ayurveda There are some models proposed by various authors to handle complex and tricky situations arising in defining and understanding the action of mechanism of Ayurvedic intervention. As described by Dr. Girish Tillu in his talk at TDU, huge observational data for Ayurvedic medicines covers large number of books and manuscripts, 57 authentic books (Drug and cosmetic act 1940, page 47) [67], more than 4500 diseases including subtypes and conditions (Ayusoft database) [68], more than 81,000 formulations (TKDL database) [69], more than 4,00,000 Practitioners in India [66], infinite documents, references, experiential data, living tradition and knowledge in public domain [7]. Dravyaguna (Pharmacology), Bhaisajya Kalpana (Pharmaceutics), Nidana (Diagnosis) and Chikitsa (Management principles). This data points to a validated knowledge base. Dr. D. B. Vaidya has explained the concept of reverse pharmacology to understand the action mechanism of Ayurvedic intervention. He says that there is huge amount of observational data available, showing relatively low side effects that have been reported. This existing data should be used as basis to carry out large interventional Ayurvedic trials to assess safety, efficacy, and pharmacokinetic information. This approach will be economical and could be less time consuming compared to the sequential drug development or the hierarchical model used in western medicine. He further talks about integrating meticulously documented experiential and experimental observations [70]. Prof R. H. Singh has opined that lab-based research experiments within Ayurvedic area during the last 50 years have not been rewarding. On the other hand, literary experiments to make a few of the classical Ayurvedic texts accessible to masses have been extremely useful. This situation warrants newer strategies of scientific research without compromising on the fundamental principles of Ayurveda [4]. Prof. Bhushan Patwardhan writes that there are substantial similarities between the traditional systems like Ayurveda and modern medicines. Ayurveda emphasizes on health promotion, disease prevention, early diagnosis, and personalized treatment. The modern medicine system approach uses predictive, preventive, and personalized medicine (PPPM). In case of Ayurveda, the evidence can be drawn from two main sources: (1) Evidence based on historical and classical nature of clinical practice supported by credible and accepted documentation. (2) Evidence based on ongoing scientific research to support various theories, medicines and procedures used in Ayurvedic medicine [71] [72]. Dr. Ram Manohar has expressed that Ayurveda is based on 5000 years of clinical practice. Hence, practice-based clinical trials should complement natural ways to gain insights [73]. Dr. Baghel's interpretation is that one should think of Ayurveda being in the developmental phase like any other medical systems. Like many other scholars he thinks that Ayurveda is a pure science based on logical explanation, which is called Darshana. Ongoing research in Ayurveda should impact academics, pharmacy, and practice in a profound way to convert data into information, information into knowledge and knowledge into wisdom [74].

24 | P a g e As per Prof. Darshan Shankar's analysis as of 2015, at the national level, Ayurveda receives a meagre 3% of the Central health Budget and at the State level, making it difficult to fund any meaningful research projects. Despite Ayurveda's strengths, it has some limitations in current scenario. To advance the science there is a need to embrace tools of information technology to organize its vast multifaceted data, in searchable formats. Meticulously documented clinical experiences interpreted through Ayurveda-biology will expand rejuvenation of healthcare in India [59]. All the thought leaders cited here point to the strengths of Ayurveda as well as the immediate needs. They have pointed out that the research must be of high quality, and it must be impactful. They have indicated the need for experimental as well as experiential research. They have already provided a few new solutions and have urged to the research community to find new ways of tackling problems [75]. 1.11 Potential opportunities for Real World Data analysis within Ayurveda The western medicines are developed using a method called as hierarchical method where it tries answering questions with limited scope e.g., what is the efficacy of a particular drug, what is the safety profile of a drug? This method assumes a step wise approach and deals with the problem in successively conducted clinical trials of various types in a specific sequence. The pharmacology of the molecule is ascertained first at the very beginning. These studies are followed by cohort studies including open-label randomized studies but in general the clinical trial testing usually concludes with a blinded, randomized controlled trial (RCT). As already pointed out, RCTs are the "gold standard" of evidence generation as they offer most internal validity and minimal bias [76]. These studies could be complemented by using case studies, case series. This "one step at a time" approach has worked very well in the western medicine framework. Ayurveda has been practiced for more than a millennium and is widely accepted in India as a worthy medical system. Over the last few decades' people across the world have gained knowledge and realized the importance of the age-old medical system and are constantly driven towards it. Although, having been in practice for ages Ayurveda still does not enjoy the recognition which the Western medicine does. Hence there needs to be a structured approach towards making this possible. The untapped potential of Ayurveda needs to be scientifically communicated globally for a wider reach for it to be utilized as a public health tool for promotion of health and prevention of diseases [76]. Ayurvedic vaidya usually use paper-based case report to record a patient's Ayurvedic parameters along with other details of medical consultation. These are typically not exchanged with other vaidyas. There is a huge amount of data available on paper and if digitized could be a big revolutionary step. Increased use and interoperability with electronic medical records of digital Ayurvedic patient management systems are required. Based on a report published by AYUSH [66], there are 4.5 lakh registered Ayurvedic practitioners. Even if 5% of doctors start using EMRs, i.e., 22,500 doctors and if data for 2 new patients is entered every day (~225 working

25 | P a g e days) for the whole year, 50 lakh unique patients' data can be generated in a single year. Currently, this gold mine of data has not been built yet. 1.12 What this study aims to contribute to This study aims to contribute to interests of multiple stakeholders involved in clinical research. Based on arguments given above about availability of observational data, availability of technology, the Indian context and the specific case about Ayurveda and need to understand the science behind it, the study would highlight multiple use cases. The study would highlight tools and information generated through these tools and this could be interchangeably used by interested stakeholders including: 1.12.1 Hospital management To keep any hospital functioning smoothly, operational insights from routine hospital data are important to improve management and efficiency of day-to-day activities. How many patients are present in the database? What are the characteristics of these patients? What is the gender distribution and age group distribution? Which countries, states, cities do they come from? How many times do they visit the hospital? What is the number of In-Patients & Out-Patients? What kind of assessments are done at each visit? What is the duration of visits for a patient? Which diseases are getting treated? Are the patients benefiting? How do you measure benefit? Regular analysis of data would give insights which will allow for operational proficiencies, cost savings and eventually profitability. Finally, patients' satisfaction would improve if a hospital functions efficiently. 1.12.2 Clinicians or treating doctors Let us understand what kind of benefits the clinicians can have from this study. Traditionally many clinicians in their own practice use paper CRF to capture patient data. The hospital has electronic data capture system, which is essentially doing the same job, but data capture is electronically done. Are they ready to adopt to a new way of working? Does onboarding training at hospital cover this part of the job? Do they see this as an additional burden or an integral part of day-to-day work? How can experience of clinicians using the Health Information system be viewed? Do doctors like data entry part of job? Do the doctors have time for real data entry while consulting patients? Answers to some of these questions can be generated indirectly by understanding the quality of the database and data contents. Does hospital management take any feedback from the clinicians who are the primary "end users" of the data capture system? This timely feedback loop should provide great inputs into evolution of the system both operationally as well as scientifically. The electronic data capture provides unique opportunities to clinicians such as they have access to the data from other practicing clinicians. This gives indirect learning opportunities of understanding treatment protocols, treatment variations employed, rare diseases treated at the hospital. Retrospective analysis of disease and treatment should provide ideas about disease variations, appropriateness of documentation, disease – disease combinations, disease – treatment combinations. These documented combinations could be clinically meaningful, could be season wise, gender wise, age wise varying. Retrospective analysis of treatments should

26 | P a g e provide tendencies of treatment prescription such as use of classical treatments, herbo mineral treatments. Data review and analysis tools developed for this thesis can enhance patient and doctor interactions. 1.12.3 Universities and students Teaching material for students: Over the years, teaching methodology has not transformed even though there are quite a lot of advances in modern methodology. Can learning objectives of different kinds be tackled by making insights generated from this data available to the students. For example, can complex disease-treatment relationships be made easily visually available. Can interesting ways of explaining text and advice from Ayurveda which have contemporary relevance be created? For example, occurrence of certain diseases in certain geographic areas or season. Very few large-scale studies like this have taken place in Ayurveda, so this study with large amounts of observational data can provide exploratory opportunities to describe findings, textually as well as diagrammatically. Tools developed here can be used as a supplementary material for any MD / PhD student. Scientific literature generation by researchers: Most hospitals in India or any part of the world mainly focus on treatment and not on research publications. Can a research team be put together for medical communication who publish papers as their primary job? If there is no known profile of patients visiting an Ayurvedic hospital and if this data can be generated and represented in the right form, it will provide novel information. How can we measure the strengths and weaknesses of an Ayurvedic practice? How should researchers evaluate changes in results of a practice over time? Is it possible to build new hypothesis? Is prescribed treatment truly personalized? Is it possible to trace back the treatment regimen followed and compare it to classical fundamentals? Is there a way to compare the demographics and patient characteristics from a Ayurvedic hospital against a mainstream western hospital? Which are rare diseases identified in the database? Can a clinically meaningful document be written, like a case series about this rare disease? How can anyone use these data as "secondary use"? Can this data be used by insurance companies? Do the approved labels of medicines and prescriptions in the database match each other? Metal based formulations are questioned by non Ayurvedic community, what insights can be drawn about the rasa-aushadhis? Which are these medicines? For what diseases are they given and for what duration? Before providing the metal-based treatment and after providing the metal- based treatment, is there any difference in duration seen in treatments? What is the percentage of patients that are prescribed these medicines and what is the percentage of duration of all the duration of treatment given to these patients? 1.12.4 Policy makers – AYUSH and relevant ministries, insurance sector: Policy makers and insurance companies can use insights from this data to decide on which treatments to cover for insurance. Government agencies can use this data to promote Ayurveda as a medical system throughout the world. They can make policies similar to policies that govern western medicine.

27 | P a g e 1.13 Introduction to real life data As presented above, the potential benefits of gathering and analyzing such data are huge. A quick history of how this ability to collect data has come about is given below. This history also highlights the challenges of collecting and analyzing data in general on top of challenges associated with clinical research and Ayurveda specifically. With the passage of time, revolutions in technology have continually increased the creation of information and its exchange. With the advancement for communication from spoken to written, it became simpler to create texts, books thereby documentation; thus aiding transfer of knowledge from one person to another as well as from generation to generation without losing any data in translation. With increased and improved writing, compilation of articles, tables and records, there came a time where storing them became important, thus came in the libraries. The ability to effortlessly widen accumulated data had to wait until the 15 th century. Around 1439, Johannes Gutenberg developed the printing press, causing an astonishing growth in the sharing of information at an economical cost. The 20 th 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 20 th 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 [77]. 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 issued in Dec 2018 that the world's cumulative data will grow from 33 zettabytes 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 growth of data has been seen in every industry, in every corner of the world. The relentless increase in the quantity and flood of information denotes an important professional opportunity, but a challenge simultaneously for those in medicine and science [78]. As indicated by Toby Cosgrove MD, from Cleveland, medical information doubles every 73 days, in the year 2020, as compared to approximately 3.5 years in 2010. An estimated 8,00,000 papers were published in 5,600 medical journals every year. It is projected that 12,000 new articles and 300 randomized controlled trials will be added to Medline each week, and that new medical articles will appear at a rate of one every 26 seconds [79]. 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 the data. Data is being produced, obtained, and stored in numerous number of structures [80]. 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,
28 | P a g e 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, ECGs, questionnaire data, or self-reported data (e.g., symptoms, quality of life). Raw data, unprocessed source

100% MATCHING BLOCK 3/51 W

data, like unrefined gold buried deep in a mine is a precious resource.

It

71% MATCHING BLOCK 5/51 W

is often: (1) Inconsistent, containing both relevant and irrelevant data, (2) Imprecise, containing incorrectly entered information or missing values, (3) Repetitive, containing duplicate data.

To utilize the raw data to its fullest potential, it needs to be extracted, filtered through, understood, and transformed into analyzable format. 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. 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" [81] [82] [83] [84]. 1.14 Introduction to clinical data understanding The health care industry uses either a paper-based record keeping method and/or electronic health record (EHR) system to manage patient data. More and more organizations are using electronic data capture, but the practicing doctors in individual clinics may still be documenting observations on paper. The EHR has become an integral part of medical care, which transforms health care service quality and improves physicians' satisfaction and facilitates patients' decision. Accurate information from EHR enables physicians' decision making and measures clinical validity, which in turn upgrades the quality of patient care. This functionality is crucial during diagnosis and therapy, which benefits medical and legal practices too [80]. Health authorities and top-level journals require the data to be submitted along with research papers. The analyzable data set, is the result of many decisions made by varied people, as explained above. The errors, flaws, or biases in the processing of source data, will not necessarily be identified in the analyzable dataset. After the electronic data entry, new variables are generated to support further analysis. The final cleaned analyzable datasets consist of various components such as participant characteristics and primary outcome, pre-specified secondary and tertiary outcomes, adverse event data and exploratory data [85]. Physicians and other scientists are getting better at producing data. But we must become proficient—with or without the help of technology—at mining and managing the data in ways that will allow us to use it to maximum effect [86]. The full analyzable dataset is generally the most useful set of data to share, with large and likely important benefits to science and society. Secondly, the full analyzable dataset provides scientific validity to the outcome and ensures replication and repeatability. Further, meta-analysis increases the statistical power of detecting effects and maximizes the value of the outcome in the clinical knowledge base. Finally, analyzable data allows for further scientific discovery through additional secondary analyses, as well as the conduct of exploratory research to generate hypotheses for additional studies [87].

29 | P a g e 1.15 Introduction to study of demographics and patient characteristics We will proceed with understanding the data contents and see if any of the questions raised earlier can be solved. Demography is the study of the population. It explains the composition, the distribution and the data trends seen in the population. Roles and functions for demography studies can be broadly defined as, (1) population projections, (2) inputs into government budget, (3) evidence-based policy, and (4) communication of vital statistics [88]. There is very little data on the profile of patients accessing traditional systems of medicine. A comparative study of profile of patients using an Ayurveda clinic and modern medicinal clinic will help in understanding of utilization of services and preference for health seeking behaviour. 1.16 Introduction to study of diagnostics and interventions Diagnosis is one of the most important aspects in the process of treatment of the disease or condition. It is a patient-centered, cyclic process of gathering information, analyzing information, determining the health condition, and defining the type of intervention and continuously monitoring the progress till the desired state of functions/doshas is arrived at. Ayurvedic treatment involves removal of the causative factors. It assists in getting the functions/doshas into balance. The success of a treatment is possible only by timely and accurate diagnosis, a tailor-made intervention accompanied by an effective collaboration of the physician and the patient [89]. The term comorbidity refers to the coexistence of multiple diseases in relation to a primary disease in a patient. Patients report multiple diseases during their visits to the hospital. Some of these reported disorders are expected and some are unexpected. There are known as well as unknown disease combinations present due to biological linkages. Clinical and epidemiological studies indicate that disease comorbidities have a great impact on health status, selection of appropriate treatments and health system costs. Understanding comorbidities and their etiology is key to identify new preventive and therapeutic strategies. Finally, all these steps (1) accessing and understanding real life data, (2) converting that data into analyzable format, (3) understanding demographics and patient characteristics, and (4) understanding diagnostics and interventions should help in building transdisciplinary evidence to increase the scientific understanding outside of the community, then increase the confidence and thereby widening the user base. 1.17 Structure of the thesis document The thesis is structured as follows: Chapter 2 covers study design and numerous methods applied for data analysis. In section 2.7, table (Table 2-2) presents a consolidated view of various analysis, context about different analysis and their possible relationships with each other. Detailed observations of these experiments are documented in chapter 3 Results. Challenges with respect to overall system architecture and data collection are documented in chapter 3. Chapter 4 provides some solutions to these challenges. Supplementary interpretations are further illustrated in chapter 4. Chapter 5 covers several conclusions drawn based on the analysis and key take away message. Chapter 6 appendix provides detailed material generated for this PhD work. Appendix section 6.2 provides details about the master dataset developed for whole of this study. Appendix section 6.3 provides information regarding the source database. Appendix section 6.4

30 | P a g e provides a table to track all analysis presented in the thesis. Appendix section 6.5 provides R, SQL, and D3js programs developed to generate different datasets and analysis. Chapter 7 covers the references. All the tables and figures are cross referenced and clicking on the table number or figure number in the document takes the reader to that table or figure. Column "Figure number and analysis name (linked to the actual figure)" in Table 2-2: Proposed methods and analysis use cases allows a reader to go the actual link of specific analysis. "Link to analysis" part under any figure or table allows a reader to go to actual link of specific analysis.

31 | P a g e 2 Methods 2.1 Study design This was a retrospective study of Electronic Health Records (EHR) at TDU. Electronic Health Records of patients from 2011 to 2017 are used. It contained data for more than 51,000 patients, more than 1,50,000 visits, more than 900 variations of disease types, more than 3,000 variations of medical procedures. The study was approved by the authorities of IAIM and TDU (refer Appendix section 6.1). We explored "naturally reported data" for getting insights into demographics, health-seeking behaviors, and other health parameters. Sensitive information related to patients and doctors was not extracted to maintain confidentiality. Data was analyzed through SQL [90] and R programs [91], python [92], Java [93], D3js [94] and tableau [95] software. A high-level pictorial representation of the technical study is displayed below (Figure 2-1). Figure 2-1: Pictorial representation of analysis 2.2 Data analysis design The data analysis was represented using many different methods such as: (1) tabular representation using frequency counts [96], (2) descriptive summary statistics [96], (3) data representation on world / country map [97], (4) boxplot representation [98], (5) barplot [99] and (6) dotplot representation [99], (7) radar plot representation [100], (8) individual patient level data listings – line by line data representation, (9) various types of bubble plots [99], [101], (10) circular data representation [102], (11) collapsible tree diagram [103], (12) treemap / mosaic plot [104], (13) butterfly plot [105], (14) area plot [99], (15) calendar plot [99]. After converting the source data into analyzable format, the next logical step is to generate various descriptive Statistics. This assessment is covered by frequencies, measures of central tendency (also called averages), and measures of variability. Frequency statistics means to count the number of times that each variable occurs. E.g., number of males and females, number of diseases reported, number of treatments prescribed, to name a few. This calculation is displayed by both the absolute or actual number and relative or percentage totals. Measures of central tendency provide a number that represents the entire data for a particular variable, such as mean, median. Measures of variability indicate the degree to which scores differ around the average. It is essential for any end user to have a sound understanding of study data. Initial statistics for all studies should include descriptive statistics [96]. Use R program to generate tabular or graphical Use R program to create analysis data tables Use source tables from the SQL server Use SQL queries to combine necessary Use Tableau to generate interactive visual analysis

32 | P a g e For many centuries humans have used maps for various reasons. Maps are used to display geographically linked data providing a clearer and more intuitive visualizations. Maps allow to see the distribution of data in each area going from district, state, country. Technology has enabled creation of interactive maps. These allow zooming in and out, panning around, identifying certain features, querying data by topic(s) or specific indicator(s), producing reports and visualizing information in the map [97]. Boxplots were invented in the 1970s by American statistician John Wilder Tukey. They are also called as a box and whisker plot. Boxplots display five-number summary of data - the minimum, first quartile, median, third quartile, and maximum. Sometimes, the mean is also indicated by a dot or a cross on the box plot [98]. William Playfair is considered to have designed the bar plot. A bar plot is a graph that represents the category of data with rectangular bars. The length and height are proportionate to the values which they represent. The bar plots can be drawn either horizontally or vertically. One of the axes of the plot shows specific categories being compared. The other axis denotes the measured values corresponding to those categories [99]. A dot plot can be used for any graph that is translating data in a dot or small circle. A dot plot is also known as a strip plot. It is a simple form of data visualization consisting of data points plotted as dots on a graph with an x- and y-axis. These types of charts are used to graphically depict certain data trends or groupings [99]. A radar plot is a 2-dimensional representation of multivariate data as a polygon. Each variable included in the plot is represented as an axis. All axes have the same origin. The position and angle of axes are usually not informative, so any variable can be represented in any order. If different axes represent months or seasons, then the order of representation would be important as there is a certain meaning to ordering. The bars on each axis represented by the variable are called radii. A radar plot looks like an irregular polygon arranged on top of each other, all with the same center. These are used in many fields such as medicines, sports, education, and different businesses. Radar plots are also called spider plots, cobweb plots, star plots, polar plots to name a few [100]. A bubble plot is a plot that displays three dimensions of data. Each entity with its triplet variable 1, variable 2, variable 3 of associated data. 2 of the 3 variables are used on the x and y axis and a numeric variable determining bubble size is used to determine the size [99], [101]. An area plot is a variation of line plot. In this plot data points are connected by a continuous line and the area below the lines is filled with colors or textures. This plot compares two or more quantities with an area chart. Area graphs can be effective for showing the fluctuations of various data series over time [99]. A butterfly plot is a comparative bar plot or a histogram that displays the distribution of a variable for two subpopulations. A butterfly plot could be displayed either vertically or horizontally [105].

33 | P a g e A calendar plot is a visualization used to show activity over the course of a long span of time, such as months or years. They are used to illustrate how some variable alters depending on the day of the week, or how it changes over time [99]. The treemap functions as a visualization composed of nested rectangles. This was developed by Prof. Ben Shneiderman, from the University of Maryland. The rectangles represent certain categories within a selected dimension and are ordered in a hierarchy, or "tree." Quantities and patterns can be compared and displayed in a limited chart space. Treemaps represent part to whole relationships [95], [104]. Tabular listings of data are generated at individual patient level. These are equivalent to displaying source data. They are the basis for all tables, listings, and figures (TLFs). To make the listings readable different components are data are shown in different listings. Patient profile listing covers all data for a patient in one consolidated report. This is used in data review and clinical review. These various analyses enable data to be reported in different levels of details. Most of these representations are interactive, end user can perform filtering tasks while using the visualizations. Tableau's drill down facility provides additional ways of analyzing the data. Tooltip functionality allows extra dimension to provide more details [95]. 2.3 Converting real life clinical data into analyzable format 2.3.1 Data access Patient data was stored in the hospital database. "Read only" access was provided to the hospital database, to avoid any accidental updates to the records, thus preventing the risk of source data change or loss. The details for accessing the hospital management system are as follows: 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 xx.yy.zz.ww -p ABCD -d iaim -U iaim_ro 4. Postgress Data Base details are as follows: • Hostname: xx.yy.zz.ww • port: ABCD • user: iaim_ro • password: efghijk The real login and IP details are not presented to keep the confidentiality of secure access.

34 | P a g e An independent (not interfering in the day-to-day transactions of the hospital), remote access for the specific version of the database was established. 2.3.2 Data preparation It is very important to think through the data preparation stage about the data holistically. Audience is critical while preparing data. It is important to assess, who will use the data and where and when and for what purpose. The answer to these questions determines how the data should be processed. Data preparation has a lot of different components, from restructuring to reformatting to cleaning, and should not be constrained by a specific order. This data is stored in a central place called as a data warehouse. It is a central repository of data within for an organization. Data flows into a data warehouse from various systems. This data is used to make operational, business, and scientific decisions [106]. This detail influences the data preparation process significantly, determining both the amount of effort and detail [82]. The following steps were followed in data preparation (Figure 2-2). The relevant details can be found in individual R, SQL, python programs (refer Appendix 6.5): • Merging, joining: Combine relevant data from different datasets into a new dataset.

43% **MATCHING BLOCK 4/51** W

A "join" is an operation that connects two or more datasets by their matching columns. This establishes a relationship between multiple datasets, which merges data together so a query can be made on the combined data [84]. •

Appending: Combine two or more

83% **MATCHING BLOCK 12/51** W

similar datasets into a single dataset [84]. • Filtering: Rule-based reduction of a larger dataset into a smaller dataset.

91% **MATCHING BLOCK 6/51** W

The goal of data filtering is to refine a data source to only what the user needs.

96% **MATCHING BLOCK 7/51** W

Data filtering involves the selection of specific rows, columns, or fields to display from the dataset [82] [84]. •

Deduping: Remove duplicates based on a defined criterion.

80% **MATCHING BLOCK 8/51** W

Data deduplication is a data compression process to identify and remove repeated copies of information.

Deduplication allows storage of

59%	MATCHING BLOCK 9/51	W

one unique copy of data in the database. This process allows for examining incoming data and compares it to data that is already stored in the system [82] [84]. •

Transforming:

89%	MATCHING BLOCK 10/51	W

This involves converting data from one structure (or no structure) to another to integrate it with a data warehouse or with different applications [82] [84] [106]. •

Format revision: Format revisions fix problems of different data types. Similar data captured in different formats creates problem for analysis. E.g., one dataset may capture treatment information as a coded numeric variable whereas another dataset may capture the same treatment information as a text. This inconsistency results in misrepresentation as well as sometimes loss of information. Along with the data format it is important to ensure the variables have appropriate lengths so that no data is truncated. Standardizing the data formats and lengths ensures correct data joins and appends. This could involve the conversion of male - female, units from one unit to another, datetime, phone number list, etc. to name a few into a consistent format. Format adjustment could involve dividing a comma-separated list into multiple columns [84].
35 | P a g e • Categorical variable creation: This transformation is

81%	MATCHING BLOCK 11/51	W

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 using Indian seasons, RMSD and Metabolic disease group creation, treatment groupings into kashaya, asava, arka, etc. [84].
36 | P a g e Figure 2-2: Flow diagram from data source to final usage by various usage types Data Sources Staging area Ware house Data marts Usage Data access: Various deliverables Operational system Coding dictionaries Clinical system Flat files information Calculations and transformations Curated and consistent data storage Operational data Pharmacy data Patient level data Hospital management Researchers Health authorities Data mining The source data from operational system, clinical system, and coding dictionaries is combined logically. Along with the source information, there are a few additional files created using subject matter expertise. These are logically added to the other source datasets. These give rise to intermediate datasets created for calculations and transformations. Some of these are temporary datasets and some of these are stored in a staging area used by analysts. Staging area is an area dedicated to individual project.
37 | P a g e Final output datasets are stored in a central place called as a data warehouse. It is a central repository of data within for an organization. These datasets are used to make operational, business, and scientific decisions by various stakeholders by converting them into interactive analysis, dashboards, formal project reports.
38 | P a g e 2.3.3 Data derivation 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 In- Patient visits, Out-Patient visits, Diseases reported as per Ayurvedic Classification dictionary, Medication prescribed and Ayurvedic services prescribed. These components of data were logically arranged in one dataset by using various data transformation steps. In addition, there were new variables derived to create necessary information for the potential analyses. Some of the challenges experienced to assemble the “reference dataset” from the source data and practical explanation of the “data preparation” steps taken are given below. 1. The database was manually explored using various SQL programming commands to check variables and observations from numerous tables (Figure 2-3) 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: • Extract relevant data tables from the source database (Figure 2-4) • Transform the variables, join the tables based on logical link • Create “staged data” (Figure 2-5) which can be also called as snapshot of data • Reference files (disease categories, Indian seasons) which were needed for calculations were developed using expert’s help (Figure 2-6) 5. Second section of the program: • Cleanse the tables • Transform the tables for combining • Join the tables using logical link • Derive additional variables as necessary • Filter the data using reference files created in the earlier section 6. In this process, we used 13 source datasets (5 reference datasets and 8 patient level datasets) and 72 variables to generate the necessary snapshot of the source data. These were re- arranged into 6 datasets and 40 variables. 3 additional reference files were used for further processing. 1 final dataset having 39 variables from source and 33 newly derived variables is built. (Figure 2-7). All these steps were covered in 50 stages of programming.
39 | P a g e 7. The entire workflow is pictorially depicted in (Figure 2-8). 8. Information about the final dataset is detailed in (Table 6-1).

40 | P a g e Figure 2-3: A glimpse of data tables used to store source data from the database action_rights diet_prescribed hospital_technical package_componentdetail patient_registration section_field_options store_item_batch_det ails test_details admission discharge_format_detail icu_bed_charges package_item_charges patient_section_details service_consumable_usa ge store_item_details test_org_details anesthesia_type_char ges doctor_charges_backup ip_bed_details package_prescribed patient_section_details_o rig service_documents store_item_lot_details test_results_master area_master doctor_charges_op_bac kup ip_prescription patient_activities patient_section_forms service_master_charges store_patient_indent_ details test_visit_report_sig natures bed_details doctor_consultation item_supplier_prefer_supplie r patient_consultation_field_v alues patient_section_image_d etails service_master_charges_ backup store_patient_indent_ main test_visit_reports Bill doctor_consultation_cha rge manf_master patient_demographics_mod patient_section_values service_org_details store_po tests_conducted bill_activity_charge doctor_medicine_favour ites medicine_dosage_master patient_deposits patient_service_prescripti ons services store_po_main tests_prescribed bill_adjustment doctor_op_consultation _charge medicine_id_health_authorit y_unique patient_deposits_setoff_adju stments 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_receipts dyna_package_charges mrd_codes_doctor_master patient_discharge preauth_prescription_acti vities stock_issue_main store_retail_customer s user_services_depts. complaintslog dyna_package_org_deta ils mrd_codes_master patient_documents prescribed_medicines_m aster 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_det ails 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_prescriptio ns sample_collection store_estimate_details store_transaction_lot_ details section_master diagnostic_charges favourite_reports operation_org_details patient_other_medicine_pres criptions 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_u sage follow_up_details outsource_sample_details patient_packages scheduler_appointment_it ems store_indent_details supp_inv_id patient_prescription diagnostics growth_chart_reference _data pack_org_details patient_pdf_form_doc_value s scheduler_appointments store_indent_main supplier_master This table presents a glimpse of inventory of data tables in database. The cells marked in yellow are used for the generation of the analysis ready datasets.

41 | P a g e Figure 2-4: Extraction of relevant data from Source database Each row in the above figure is one source dataset. Each column represents a variable. The gray-coloured cell denotes the presence of the variable in the dataset. There are 13 datasets, and 72 variables represented in the above table used to derive analysis datasets

42 | P a g e Figure 2-5: Staged data converted into 6 datasets The source datasets have been merged step by step using the variables marked in yellow colour. The above picture shows 16 steps taken to generate 6 datasets, marked in Green for subsequent processing. The variables marked in numbered yellow squares are the logical links between the datasets and are used for data preparation steps.

43 | P a g e Figure 2-6: Staged data This picture lists 6 datasets created by earlier processing + 3 reference files provided by the experts. Using the source variables (gray-coloured columns), additional variables marked in Orange are created. User defined files: Disease group file: this file was created by Dr. Girish Tillu outlining the disease codes for Metabolic and Rheumatic and Musculoskeletal disease (RMSD) areas. Rutus: the calendar months are transformed into Indian rutus, https://www.drikpanchang.com/seasons/season-tropical- timings.html?geoname-id=1277333&year=2010, lookup_medicine file: this file was created by Dr. Prasan Shankar classifying medicines into groups of medicines such as: Ghritam, Kashayam, Asavam, Aristham, Bhasma, Abhyanga, Cream, Rasayanam, Tablet / Gulika / Vati, etc..

44 | P a g e Figure 2-7: Final dataset with 39 source variables and 33 new derived variables The above figure provides a step-by-step flow of creating the final dataset. The final dataset named “all_met_rmsd” is created through the above complex processing, which will form the basis of many analyses explained later in the thesis. This process is followed for every analysis carried out. The variables marked in numbered yellow squares are the logical links between the datasets and are used for data preparation steps.

45 | P a g e Figure 2-8: Data flow from source data to interpretable results Source data (SQL data file) Staging data (csv files / R data files) Data ware house (R data files) Usage city 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 state_master country_master base01_op (all OP data) patient_details base01_ip (all IP data) Actual analysis patient_registration base_01_ser (Services data) doctor_consultation pat_diag_vis (temp30_5) mrd_diagnosis med Learning from the existing database to be given back as learning patient_prescription services patient_medicine_prescriptions ip_prescription Reference files for derivations and filtering of data Clinical communication services_prescribed (Disease group txt file) services (Indian rutus txt file) med (Medicine type txt file)

46 | P a g e SQL data: source data captured in the database, staging data: logically combined intermediate datasets from source data by software development, Reference files: files needed for deriving certain information which is not present in the source database, Longitudinal data: dataset having 1 record per patient, per visit, per disease and per treatment, the dataset is filtered for the Metabolic and Rheumatic and Musculoskeletal disease (RMSD) patients, analysis is carried out using such derived dataset(s).

47 | P a g e 2.4 Clinical data understanding 2.4.1 Broad checks on the datasets As a part of clinical understanding, structural and contents checks were performed for completeness, correctness, to identify duplication, to name a few across 90+ datasets and 500+ variables. Some of these were programmatic checks and some were manual checks to make the data available for exploration and "analysis ready". Unique values for each variable were checked to understand the value level detail for consistency and variations. The following data and contents review was done for vital sign dataset, lab measurement dataset, treatment dataset, as well as review of clinically important variables. After reviewing the source datasets for clinical understanding, derived datasets were also reviewed. 2.4.2 Contents checks 500+ variables were captured across many datasets for each visit and each patient (Table 6-3) were classified and mapped into the following categories, (1) Ayurvedic data, (2) Background, (3) Disease, (4) Doctor's Notes, (5) Food / Exercise, (6) Hospital Visit, (7) Lab report, (8) Measurement, and (9) Treatment / Procedure. If there was any non-missing data present in a particular variable then a pseudo value "Yes" was assigned, if the data was missing then a pseudo "Blank or No" value was assigned for the purpose of analysis. This data was presented as a listing for each patient for each visit (day) by the categorization presented above. If the data was available, it was presented as a color-coded bar. If the data was missing, then it was presented as a white blank space (Figure 3-2). 2.4.3 Visit pattern analysis Frequency counts of 4 parameters, (1) new Out-Patients added on that day, (2) total number of patients visiting on that day, (3) total number of In-Patient visits on that day, and (4) total number of Out-Patient visits on that day were calculated for each day to understand the patient flow to hospital from year 2011 to 2016. The calculated information was represented on a calendar. 2.4.4 Patient disease and treatment journey view Patient profile report generation module was also checked to understand the contents. Two longitudinal interactive views were created to display individual patient data. first version of patient profile contains the following information (Figure 3-4): Patient ID (mr_no), gender, study day, In-Patient visits are displayed in blue colour and Out-Patient visits are displayed in Orange colour. The tooltip of the interactive display holds information about the following data points not displayed on the page: (1) Study day, (2) Total duration of hospital visits, (3) Disease description variable accompanying ACD codes, (4) Medicine provided at that visit, (5) Minday Metabolic: First day on which any metabolic disease has been reported by patient, (6) Minday RMSD: First day on which any RMSD disease has been reported by patient. Second version of patient profile contains the following information (Figure 3-5): Patient ID (mr_no), gender, base age, category, Code, description, study day. The Diseases were displayed

48 | P a g e in blue coloured bars and treatments prescribed were marked in orange coloured bars. Disease duration and treatment duration bars were created as follows: Duration between minimum and maximum reported date for a disease as well as prescribed treatment was calculated, this duration was displayed on the visualization. The tooltip contains information about the following data points not displayed on the page: (1) Daystt: Start of event in days, (2) Disdur: Duration of event in days, (3) Disstt: Start date of event, (4) Diend: End date of event. 2.5 Studying demographics and patient specific factors Analysis datasets created in the earlier sections (converting clinical data into analyzable format and Clinical data understanding) are used to generate necessary analysis. If the existing variables were sufficient to produce the results, then these were used as is. In case additional information was need then that was derived as appropriate. Reports using tableau software were created. Multiple types of data visualizations were used so that data was represented appropriately. Preliminary analysis was carried out by Dr. Girish Tillu and he found that the database contains a lot of patients in Metabolic area and (Rheumatic and Musculoskeletal disease) RMSD area [7]. 10 Metabolic and 97 RMSD disease codes were identified (Table 6-2). The analysis was split into 2 major sections in this thesis. Reports were created for the complete dataset and additional reports were created on a subset of patients' metabolic and RMSD disease areas. Following interactive reports were created and were analyzed for the complete set of patients to gain insights into patient demographic and patient specific factors: (1) A tabular summary of total number of patients treated (Figure 3-6), (2) Patient analysis by country – a Country-wise visualization on the world map (Figure 3-7), (3) Age distribution by country and gender – 2 boxplot representations (Figure 3-8), (4) A tabular summary of Blood group distribution be gender (Figure 3-9), (5) A boxplot representation of analysis of number of visits and types of visit (IP / OP) (Figure 3-10), (6) Number of diseases reported by gender – a descriptive summary statistics table (Figure 3-11) Subsequent reports were created for metabolic and RMSD patients: (7) A bubble plot data tabulation for patients reporting RMSD and Metabolic diseases (Figure 3-12), (8) Disease distribution by Age and gender – a boxplot representation (Figure 3-13), (9) A tabular representation of Patient visit duration for Disease categories by Gender (Figure 3-14), using the following logic: The duration between the first visit and the last visit for each patient has been calculated and categorized as follows: &lt;= 1 day, &lt;= 1 month, &lt;= 2 months, &lt;= 3 months, &lt;= 6 months, &lt;= 1 year, &lt;=2 years, &lt;= 3 years, &lt;= 4 years and &lt;= 5 years. In this analysis patients were counted multiple times as per available data for each time period. A patient visiting for more than 5 years was counted in all categories. If a patient discontinued in the 4 th month then that patient was counted in Day 1, &lt;=1 month, &lt;=2 months, &lt;=3 months categories. The colour gradient moves from Red to Green denoting low to high number of patients in each category. (10) Seasonal Variations within Metabolic and RMSD disease areas by Indian rutus (Vasant, Grishma, Varsha, Sharad, Hemant, and Shishir) [107] and gender (Figure 3-15). Pre and Post Disease Classification Analysis was carried out for Metabolic and RMSD disease areas to

49 | P a g e understand the disease trajectories [108] (Figure 3-16) The underlying data was generated from every day medical practice at the hospital. Hence the diseases were reported almost at random. The following analysis used first occurrence of any disease as day 1 for an individual patient. Using this as a reference day “before period” and “after period” was derived. “Before period” provides significant amount of “baseline data”, “after period” provides specific insights into what would happen after the onset of the reference disease. The following algorithm was used to create the underlying data for analysis: 1. Each of the 107 diseases (10 Metabolic and 97 RMSD) was considered as a reference disease. 2. Day 1 was calculated as the reference day 1 for individual patient for each disease. 3. Other diseases for the same patient were arranged either before or after compared to this reference disease. 4. Duration was calculated before and after day 1, which is the reference day. This calculation provided the background view as well as future view. 5. This referencing allowed for more informative background disease as well as background medicine information. The duration was split into the following time points: Table 2-1: Visit window table for Pre and post analysis Before After Day 1 as reference Before 1 month Within 1 month Before 2 months Within 2 months Before 3 to 6 months Within 3 to 6 months Before 7 to 12 months Within 7 to 12 months Before 2nd year Within 2nd year Before 3rd year Within 3rd year Before 4th year Within 4th year Before 5 year Within 5 year 50 | P a g e 2.6 Studying diagnostics and interventions Diagnostics and interventions were studied using disease - disease, disease - treatment combinations / co-occurrences by using various methods. (1) Ayurvedic Classification of Disease (ACD) and International Classification of Diseases (ICD) [109] mapping exercise was carried out to understand the underlying disease burden (Figure 3-17, Figure 3-18, Figure 3-19 , Figure 3-20), (2) Summary table for disease by Prakriti and gender was created (Figure 3-21), (3) Co-morbidity analysis was carried out using 3 different approaches (Figure 3-22, Figure 3-23, Figure 3-24, Figure 3-25, and, Figure 3-26), (4) Treatment and disease analysis at individual patient level was carried out to understand treatment protocol (Figure 3-27, Figure 3-28, Figure 3-29), (5) Area graph representation of diseases was created to show variations related day-to- day, seasons, gender, and diseases (Figure 3-30). (6) Mosaic plot displays were put together for disease and treatment combinations (Figure 3-31, Figure 3-33, Figure 3-34). (7) Cross tabulation of prescribed treatments and disease group by gender was generated, a couple examples for specific disease conditions or specific treatment were shown to provide the utility of this analysis (Figure 3-35). (8) Treatment regimen using bhasma is very specific to Ayurveda, an analysis was carried out to understand the duration of treatment pre and post usage of bhasma (Figure 3-36). Additional Disease – treatment analysis with pre and post visit window approach was performed as described: (9) Circular view representation was created (Figure 3-37 , Figure 3-38), (10) Distance metrics analysis for disease trajectories and medicine trajectories were created (Figure 3-39 , Figure 3-40), and (11) Multi-dimensional data representation using radar plot displays were created (Figure 3-41), (12) Dynamic bubble plot visualization was created (Figure 3-42). International Classification of Diseases (ICD) codes were used in patient paperwork, including hospital records, medical charts, visit summaries, and bills. These codes guarantee that a patient obtains right treatment and were charged appropriately for any medical services. An attempt was made to map Ayurvedic Classification Dictionary (ACD) codes with ICD codes by manually comparing the ACD dictionary. Summary tables and boxplots summarizing the ICD classes with the frequency of patient’s visits and classifying by gender along with their duration of visit were created. Bar graph representation for disease by Prakriti and gender was created. Co-morbidity analysis was carried out by using 3 different approaches: First approach produced a bubble plot, boxplots, summary statistics tables and frequency tables for number of other diseases reported along with the primary disease. Second approach recreated the same analysis by visit window of each month. Third approach created disease trajectories are visually displayed in a form of collapsible tree (Figure 3-22, Figure 3-23, Figure 3-24, Figure 3-25, and, Figure 3-26). Algorithm for first approach: (1) A unique combination of Patient ID, gender and reported disease at any given time point was created. (2) Subsequently, a dataset having combination of diseases for an individual patient was created. E.g., if a patient had reported 5 unique diseases,

51 | P a g e then all the combinations of these 5 diseases were created i.e. 5 C 2 combinations were created i.e. 10 combinations. (3) The resulting data had the following structure: Patient ID, Disease1, Disease2, and Gender. (4) Frequency count of distinct patients was calculated for each Disease1, Disease2 combination and gender. (5) Using this data following analysis was carried out: Summary statistics of age group for each disease by gender; Boxplot of age group for each disease by gender; Bubble plot for each disease where the bubble size was determined by the count of unique patient IDs. For each disease number of other unique diseases by gender were reported. Tooltip on the bubble plot provides information about count of distinct number of patients, and summary statistics for age group. The dashboard is controlled by a "Primary Code" or a reference disease and relevant data is displayed on the page. Other bubbles in the bubble plot, display the diseases reported by this subset of patients at any point in time (these could be clinically related or unrelated or could have occurred before or after the occurrence of reference disease). The tooltip shows minimum, median, and maximum age and distinct counts of patients. A table on the left side shows number of other diseases experienced by the patient (Figure 3-22, Figure 3-23, Figure 3-24). Algorithm for second approach: Same calculations for first algorithm were followed to create co- morbidities, but now in addition the time factor of month was added to get insights into seasonal variation and bubble plots by gender and month using a reference disease were created (Figure 3-25). Algorithm for third approach: (1) Diseases experienced by each patient were sorted by date and only the first instance of a disease was retained. This chronological list of diseases is labelled as "disease trajectory". Similar analysis when carried out for medicines, the trajectory created is labelled as "medicine trajectory" (2) For each disease trajectory the frequency counts were created and were displayed as a collapsible tree. (3) The tree has filled blue dots which open additional branches, white filled blue dots are the end of the branch, (N=xx) at each of the branches display number of patients reporting that disease trajectory. Disease trajectories were created using R programming. Final output was stored in Json file. Json file was used as the input to the D3js Java programming. Index.html file was hosted on the Github page to create the interactive page https://coursephd.github.io (Figure 3-26). Treatment and disease analysis at individual patient level was carried out. (1) When a disease was reported for the first time then that was counted as "first time disease reported", any subsequent repetition was counted as "Repeat". (2) When a treatment was prescribed for the very first time then that was counted "first time treatment prescribed", any subsequent repetition was counted as "Repeat". (3) These two calculations were repeated throughout the complete duration for each patient. Area graph representation of diseases was created to show variations related to day-to-day changes, seasonal changes, by gender, and diseases. This analysis provides frequency count of patients for each disease by month and by gender. (1) Unique combinations of patients, diseases by date and gender were created, (2) Frequency counts of females were displayed in blue color and counts for males were displayed in orange color. (3) this visual opens for each day, by
52 | P a g e clicking on "+" sign on the x-axis, providing monthly to weekly to daily view without having to go through multiple visualizations. A mosaic view of disease and intervention was created to explore the following: (1) Total number of interventions prescribed during one disease. (2) To check if there were possible relationships between different diseases and interventions considering multiple diseases reported and multiple interventions prescribed, layers of visualizations were created in the following manner: (1) TreeMapDisMed-Parameter sheet is used to filter a particular disease, which is displayed as a green colored box, (2) smaller boxes inside each disease display one intervention each, (3) Medicine-count sheet provides information on total number of different interventions prescribed for the selected disease, (4) Medicine-list sheet shows a detailed list of interventions with the total patients prescribed with them as well as total patients suffering from the disease (Figure 3-31). TreeMapDisMed-Parameter sheet can be used to filter a particular intervention and the whole analysis could be performed from an intervention's perspective. (1) TreeMapDisMed-Parameter sheet is used to filter a particular intervention, which was displayed as a single or multiple green colored boxes across multiple disease boxes, (2) Disease-count sheet provides information on total number of different diseases for which this medicine was prescribed, (4) Medicine-list sheet shows a detailed list of diseases with the total number of patients prescribed with the intervention as well as total patients suffering from different diseases (Figure 3-34). Cross tabulation of prescribed treatments and disease group by gender was also generated. The interactive visualization was used to create a few examples for specific disease conditions or specific treatment. First example is created using Balaristham and second example is created using bhasma. An attempt was made to understand the impact of usage of Bhasma on patient visit duration. Dataset for individual patients with the following variables was created: Visit duration: cdur – "Total duration in days": cdur = End visit date – start visit date +1, "Pre bhasma duration": prebhasmadur = bhasmamin (start date of bhasma intake) – 1, "Post bhasma duration": postbhasmadur = cdur – bhasmamin (start date of bhasma intake) + 1. A simple t-test analysis was performed on the created data. The pre bhasma duration and post bhasma duration was analyzed by t-test. Analysis for disease – treatment with pre and post visit window approaches: The circular visualization allows a single page view of relation between disease – disease and / or disease – treatment across multiple time points. This view shows the following information: (1) A table on the middle row: On day 1 of a disease how many distinct diseases have been reported and how many distinct medicines prescribed, this same information is shown as the green bars inside a circle, (2) Pre and post time windows are displayed and for each of the time window a similar table is represented in the upper section of the visualization. (3) In the lower section of the visualization, 1 st row represents the co-occurrence of disease – disease and / or disease – treatment before day 1 of the reference disease. (4) Last row represents the same co- occurrence data after day 1 of the reference disease (Figure 3-37, Figure 3-38).

53 | P a g e An attempt was made to understand the disease trajectories for patients by using mathematical distances. There are numerous distance measures available in mathematics and statistics which allows understanding of similarity and dis-similarity between objects, in our case disease trajectories [110]. Following assumptions were used to derive the disease trajectory: (1) Diseases experienced by each patient were sorted by date and only first instance of a disease was retained. (2) This enabled in creation of a disease trajectory for each patient for each reference disease, before and after the occurrence of the reference disease. (3) Cartesian product of patients was created for each reference disease, so that distances could be calculated. A cartesian product is a set of all possible pairs, in this case all possible pairs of reference disease and other diseases for individual patient. (4) The similarity measure was calculated for each disease trajectory, e.g., Jaccard distance was used as a distance measure for this display [111]. The Jaccard distance highlights similarity between finite sample sets. It is defined as the size of the intersection divided by the size of the union of the sample sets [112]. For our example, this will be calculated as the common diseases reported divided all the diseases reported in each case. (5) Jaccard distance closer to 0 shows dissimilarities and closer to 1 show similarities. (6) The distances were divided into 4 categories 0 to 0.25, 0.25 to 0.5, 0.5 to 0.75 and 0.75 to 1 for data visualization perspective. (7) These calculated distances are displayed as a butterfly plot for easy comparison of underlying values [105]. This plot is a comparative bar plot to display comparison of a continuous variable across groups. Similar Analysis to understand the medicinal trajectory was performed. Radar plot representation: a multidimensional, comparative view of the different diseases was created to understand at various aspects of the diseases. The radar plot chart presents multidimensional metrics. Radar plots can convey a large amount of information [100]. They provide a standardized view of different indicators on one scale. The following information for each disease was visualized as a percentile and is represented as a dimension on a heptagon (as there are 7 parameters considered in this example): (1) Distinct number of patients for each disease, (2) Number of times a disease is reported, (3) Number for a specific disease (chronological number of disease reported by a patient) e.g. a disease is reported as the very first disease or third disease or fifth disease, etc., (4) Number of diseases before the specific disease, (5) Number of diseases after the specific disease, (6) Number of treatments before the specific disease, (7) Number of treatments after the specific disease. Trellis plot display allows multiple representations of same kind next to each other [99]. Dynamic bubble plot visualization: explanation of an algorithm using Amavaata (ACD code A6.0) as an example: (1) Identified unique patients who have had Amavaata reported at least once, (2) All the other diseases and prescribed medicines for this subset of patients, (3) Created an input Json file to be passed into a D3js java program. The underlying utility generated a dynamic bubble plot. The size of bubble is proportional to the total number of patients. The links display relationships between diseases and treatments. If a bubble is “double clicked” then all the

54 | P a g e “unrelated data” to that bubble vanishes and only relevant data is retained on the screen. Once double clicked again the complete data is displayed again (Figure 3-42). Same type of analysis was carried out for the Pre and post period, example for Amavaata (A6.0) by period [https://coursephd.github.io/nodediagram/A2_0byperiod/] Similar, views can be created for any number of diseases and treatments, links below provide similar examples for the disease Prameha (P5.0) and its treatments and comorbidities Prameha [https://coursephd.github.io/nodediagram/P5_0_Prameha/] Prameha by period [https://coursephd.github.io/nodediagram/P5_0_Pramehabyperiod/]

55 | P a g e 2.7 Summary of methods section Various methods getting employed have been explained in the earlier sections of chapter 2. These data explorations can be used by different stake holders like Hospital management (HM) from administration point of view, treating doctors (Medics), and by basic researchers (Scientists). Table below (Table 2-2) systematically presents a consolidated view of various analysis, context about different analysis and their possible relationships with each other. Table 2-2: Proposed methods and analysis use cases Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis HM, Medics, Scientis ts Clinical data understanding Tabular frequenc y table Figure 3-1: A snippet of disease table by gender (1) Data quality check based on examples of a couple of diseases. (2) Generation of ideas about how “end users” are using the system HM, Medics, Scientis ts Clinical data understanding Individu al patient level data listing Figure 3-2: Variable classification by categories (1) Understanding data generation process at each visit for each patient to gain present status of data and provide feedback into improvements of system architecture. (2) Improve operational efficiencies of the Hospital Management Information system

56 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis HM Clinical data understanding, operational efficiencies Calendar plot Figure 3-3: Visit pattern analysis (1) Learn more about patient inflow and study how to increase outreach to the society. (2) Improve operational efficiencies across various departments based on patient visit patterns Medics Patient disease and treatment journey Individu al patient level data listing Figure 3-4: Patient visit profile – Horizontal view (1) An analysis of how a patient journey is documented, how can this study be used prospectively and retrospectively to understand disease progression and treatment protocols (2) Convert treatment ideas into documented material (3) Improvements to the data standards and system architecture Medics Individu al patient level data listing Figure 3-5: Patient visit profile – Vertical view

57 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis HM, Medics, Scientis ts Study of demographics and patient specific factors Tabular frequenc y table Figure 3-6: Total Number of Patients (1) Demographic profiling analysis to understand health seeking behaviours of patients (2) Disease profiling based on preliminary disease analysis, visit patterns and types of visits (In-patient and out-patient visits) (3) Use such analysis to compare against demographic profiling of any other main-stream hospital to understand health seeking behaviours (4) Use disease profiling study to understand epidemiology (5) Use this analysis for defining public health policies to local and central authorities (6) Secondary use of this analysis is to find data quality HM, Medics, Scientis ts Data on world map Figure 3-7: Country-wise Visualization HM, Medics, Scientis ts Boxplot Figure 3-8: Age distribution by country, age distribution by gender HM, Medics, Scientis ts Tabular frequenc y table Figure 3-9: Blood-group Distribution by gender

58 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis HM Boxplot Figure 3-10: Number of Visits, and Visit Types HM, Medics, Scientis ts Descripti ve summary statistics Figure 3-11: Descriptive summary statistics by number of Diseases by Age and Gender HM, Medics Study of demographics and patient specific factors Bubble plot Figure 3-12: Data tabulation for patients reporting RMSD and Metabolic diseases Analysis similar to described above on a subset of Metabolic and Rheumatic and Musculoskeletal disease (RMSD) patients

59 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis HM, Medics Boxplot Figure 3-13: Disease distribution by age and gender HM, Medics Tabular frequenc y table Figure 3-14: Patient visit duration for Disease categories by Gender HM, Medics Disease pattern analysis Tabular frequenc y table Figure 3-15: Disease distribution by Seasonal Variations and gender (1) Diseases are differently experienced by females and males as well as natural variations affect the prevalence – can we detect if the natural variations are reported in our database (2) A byproduct of this analysis is understanding of data entry and disease classification process at each visit

60 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis HM, Medics Co-morbidity analysis Tabular frequenc y table Figure 3-16: Pre and Post Disease Classification Analysis (1) Discover disease relationships by creating chronological view (2) Differentiate pre and post diseases so that diagnosis and prognosis can be formed (3) Pre and post differentiation is carried out for the prescribed medicines so that use of medicines could be understood at a summary level HM, Medics, Scientis ts Health seeking behaviour and public policy Tabular frequenc y table Figure 3-17: ICD classification by Gender (1) ICD classification of disease is used globally to understand the disease burden. What patterns emerge via this analysis, what health seeking behaviors can be understood? HM, Medics, Scientis ts Boxplot Figure 3-18: Age distribution by ICD classification and Gender

61 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis HM, Medics, Scientis ts Boxplot Figure 3-19: Visit distribution by ICD classification and Gender HM, Medics, Scientis ts Boxplot Figure 3-20: Duration distribution by ICD classification and Gender Medics, Scientis ts Disease and Prakriti analysis Barplot Figure 3-21: Disease classification by Prakriti and Gender What does Prakriti and disease data

62 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Scientis ts Co-morbidity analysis Bubble plot + Boxplot + Descripti ve summary statistics Figure 3-22: Co-morbidity analysis approach 1 example 1: Vaatavyadhi Disease co-morbidities generate insights Scientis ts Bubble plot + Boxplot + Descripti ve summary statistics Figure 3-23: Co-morbidity analysis approach 1 example 2: Pandu

63 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Scientis ts Bubble plot + Boxplot + Descripti ve summary statistics Figure 3-24: Co-morbidity analysis approach 1 example 3: Madhumeha Scientis ts Bubble plot + Tabular frequenc y table Figure 3-25: Co-morbidity analysis approach 2

64 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Medics, Scientis ts Collapsi ble tree diagram Figure 3-26: Co-morbidity analysis approach 3: collapsible tree view Medics Patient disease and treatment journey Individu al patient level data listing + Tabular frequenc y table Figure 3-27: Patient Disease and Treatment administration by Study Day (1) An analysis of how a patient journey is documented, how can this study be used prospectively and retrospectively to understand disease progression and treatment protocols (2) This could help study the severity of the disease, co-morbidities and the number of medications prescribed to treat the condition (3) This can also provide an overview of the practicing physician's style of treatment and may be help draw parallels in treating medical conditions (4) Improvements to the data standards and system architecture is a byproduct of this analysis Medics Individu al patient level data listing + Tabular frequenc y table Figure 3-28: Patient Disease by Study Day and Treatment administration by Study Day

65 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Medics Individu al patient level data listing + Tabular frequenc y table Figure 3-29: Patient Cumulative Disease and Treatment administration by Visit Medics, Scientis ts Disease pattern analysis Area plot Figure 3-30: Area graph representation of diseases (1) Due to this analysis representation, information on many diseases can be viewed in very short space (diseases plotted side by side) (2) Diseases vary by seasons, by gender as well as some diseases are more prevalent than others which can be seen, and clinical interpretations can be drawn (3) Operational and clinical insights can be generated with help of this visualization Medics, Scientis ts Co- morbidities and concomitant medications Mosaic plot Figure 3-31: Mosaic plot: Disease and treatment representation example 1: Prameha (1) In ayurveda, a treatment is used for multiple diseases and multiple treatments are used for the same disease based on the context of disease. This relationship gives rise to many to many associations between disease and treatment (2) Many to many relationships which are hard to visualize are generated through this analysis

66 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Medics, Scientis ts Tabular frequenc y table Figure 3-32: Disease and treatment example 2: P5.0: Prameha and Oil: Kottamchukkadi (3) This analysis produces a mosaic display for diseases showing what kinds of treatments are prescribed (4) Another mosaic display is created for treatments showing what kinds of diseases are getting treated by the underlying treatment (5) Clinically meaningful as well as not so meaningful relationships are visualized, rare disease – disease combinations or disease – medicine combinations also can be studied Medics, Scientis ts Tabular frequenc y table Figure 3-33: Disease and treatment example 3: P5.0: Prameha and Vati: Diabecon DS Medics, Scientis ts Mosaic plot Figure 3-34: Mosaic plot Disease and treatment representation example 4: Treatment: Oil: Kottamchukkadi Medics, Scientis ts Tabular frequenc y table Figure 3-35: Cross tabulation of prescribed treatments and disease group by gender Example 1

67 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Medics, Scientis ts Tabular frequenc y table Figure 3-36: Cross tabulation of prescribed treatments and disease group by gender Example 2 Medics, Scientis ts Co- morbidities and concomitant medications Circular data represent ation Figure 3-37: Circular view: Co-occurrences of disease – disease Example 1 (1) This analysis provides a view for a disease – disease combination or disease – medicine combination longitudinally (day 1 of disease, diseases reported at different time points before and after day 1) (2) Disease and treatment information is represented on a circular display with green spokes representing co- occurrences of disease combination or disease – medicine

68 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Medics, Scientis ts Circular data represent ation Figure 3-38: Circular view: Co-occurrences of disease – treatment Example 2 (3) Clinically meaningful as well as not so meaningful relationships are visualized, rare disease – disease combinations or disease – medicine combinations also can be studied Scientis ts Similarity analysis in disease and medicine trajectories Butterfly plot Figure 3-39: Pre and Post distance analysis for disease: M2.0: Madhumeha (1) Disease trajectories (chronological list of diseases reported) and medicine trajectories (chronological list of medicines prescribed) are generated to study if similar diseases are experienced after an onset of a particular disease (2) This analysis helps in understanding biological changes, represented by subsequent reported diseases triggered by an underlying disease (3) Similarly, which medicines are prescribed provide insights into treatment protocols Scientis ts Butterfly plot Figure 3-40: Pre and Post distance analysis for medicines given for diseases: P5.0, V2.23, V2.63

69 | P a g e Stake holders Classification Type of analysis Figure number and analysis name (linked to the actual figure) Snapshot of the analysis (the picture is attached here to provide a quick look into the type of analysis) Context of the proposed analysis Scientis ts Multi- dimensional data analysis Radar plot Figure 3-41: Radar plot (1) This analysis developed showed 7 parameters on 7 vertices. Different diseases were displayed next to each other as trellis radar display. (2) If there are different shapes for different diseases then this suggests that there are underlying differences in the data representing differences in diseases. Scientis ts Co- morbidities and concomitant medications Dynamic bubble plot Figure 3-42: Dynamic bubble plot: Example 1: Disease: A6.0: Amavaata (1) Intricate relationship between disease and medicines in a very short space, at present this type of data representation approach may not have a direct application, but consultation with “end users” may provide appropriate use case.

70 | P a g e 3 Results 3.1 Converting real life clinical data into analyzable format 3.1.1 Details of the database The database has approximately 200 datasets (Figure 2-3). They cover various components of hospital’s day-to-day functions right from operational data to the patient level clinical information. High level of classification of data types is as below: 1. Operational datasets: a. Hospital charges – In Patient (IP), Out Patient (OP) b. Operation theater charges c. Inventory of equipment d. Doctor charges 2. Reference dictionaries a. Disease codes b. Ayurvedic services c. Medication names d. Master list of Laboratory tests e. Names of city, state, countries 3. Doctor details a. Doctor ID b. Relevant ward information c. Internal / Visiting / Part time / Full time 4. Patient information a. Patient details b. Visit details c. Vital signs d. Registration details e. Discharge details f. Lab data details g. Diet details 5. Datasets related to managing access levels, datasets related to pharmacy stocks, datasets related to scheduling appoints, datasets related to purchase orders, and other IT related contexts used by various teams in hospital For this study, the following data was not used to in accordance with the patient data protection and privacy, financial privacy as well as hospital management confidentiality thus avoiding any controversies: 1. Hospital monetary details 2. Doctor’s details (Name and ID of individual doctor)

71 | P a g e 3. Patient details of sensitive nature such as name, phone number, socio economic status, health insurance details 3.1.2 Data Extracted from Hospital Database In our study, we had different versions of data, details in the table below. Table 3-1: Versions of data used for analysis Data version Version 1 Version 2 Approach CSV files provided by the Hospital IT support Data extraction via the SQL DB connect Date time frame From start of the hospital to Oct 2016 From start of the hospital to Oct 2017 Data domains Lab Vital signs Diagnosis All the available data in the hospital database Type of extraction Full extraction of available domains Full extraction of all the available hospital data The analysis was carried out in the study using these 2 different versions of the data, version of the data has been provided along with each of the analysis for clarity. 3.2 Clinical data understanding 3.2.1 Broad checks on the datasets This paragraph summarizes observations from structural review of the datasets. In a well-defined database, patients should have the primary key as Patient ID: mr_no (in our case), but the underlying database considers unique visit for each patient as a primary key between tables (Patient_ID). In general, a variable containing same information across tables should have the same name, but in our case, each table has a different variable, making it difficult to create logical links across tables. E.g., Consultation_ID from doctor_consultation and Patient_ID from patient_registration had the same information; Visit_ID from mrd_diagnosis and Patient_ID from doctor_consultation meant the same. The case report form allowed for multiple diseases and multiple treatments to be recorded for each patient, this causes a "clinical logic" challenge – the potential 1-1 relation between a disease and a treatment is lost, this had to be derived outside of the database using expert understanding which would require investment of time and efforts from Ayurvedic vaidyas. There were multiple versions of the same table available in the database (as a programmer, it is well understood that older copies are retained in the system), but

72 | P a g e due to unavailability of the documentations increased the complexity. Potential approaches to address these challenges are provided in chapter 4 section 4.2.8. This section outlines observations from the clinical data review of individual case report forms: Vital sign dataset: Vital sign measurements include parameters like body temperature, blood pressure, body surface area, height, and weight. The existing database has various vital signs parameters listed one below the other. The current structure has one record per patient per visit per parameter. For a lot of visits vital sign information was missing, or partially filled. There were certain records with implausible values for certain parameters such as height and weight having 0 value. Blood pressure values having character data. Potential approaches to address these challenges are provided in chapter 4 sections 4.2.1, 4.2.5. Lab measurement dataset: Findings were similar to the Vital signs database. Along with the patient identifier information, only laboratory test name and laboratory measurements were present. In case a patient had the laboratory investigations outside of the hospital that data got stored in a scanned image format. Apart from this the dataset did not contain the date of sample, reference ranges, laboratory parameter units, fasting status etc. A single lab test had multiple names. E.g., Alanine Aminotransferase was captured in the dataset in the following different ways: Alanine Aminotransferase Alanine Aminotransferase ALT (SGPT)(UV Kinetic) Alanine Aminotransferase (SGPT)(UV Kinetic) Alanine Aminotransferase ALT (SGPT) Alanine Aminotransferase ALT (SGPT)(UV Kinetic) S.G.PT (UV kinetic) SGPT (UV kinetic) ERYTHROCYTE SEDIMENTATION RATE was captured in the dataset in the following different ways: ERYTHROCYTE SEDIMENTATION RATE ERYTHROCYTE SEDIMENTATION RATE (ESR) ERYTHROCYTE SEDIMENTATION RATE (ESR) Potential approaches to address these challenges are provided in chapter 4 sections 4.2.3, 4.2.5. Treatment dataset: the treatment or dosing or medication dataset does not get exported into a structured file for easy understanding and analysis. Which treatment was prescribed for which disease was not easily understandable based on the system generated report. Potential approaches to address these challenges are provided in chapter 4 sections 4.2.4, 4.2.8 .

73 | P a g e Medical coding and clinically important variables: the medical records for patients were captured differently by different doctors, nurses and other medical staff. Same information was found in more than one variable. Acronyms were used inconsistently. Answer for more than one question was captured in one variable. Due to "free text nature" of variables simple questions like Yes / No had many different data values. Potential approaches to address these challenges are provided in chapter 4 section 4.2.5. Classification and Sub-classification of the Doshas / Diseases: It was observed that the main disease classification by kapha, pitta, vata has disparity in numbers. As an example the table below show the variation in the counts of the diseases and their sub-classification. Potential approaches to address these challenges are provided in chapter 4 section 4.2.6 Figure 3-1: A snippet of disease table by gender Frequency of Prameha (P5.0) and Aamavata by gender has been displayed. The frequency of patients is substantially lower for classification by Sthula, Pidaka and Krusha as well as Kapha, Vaata, and Pitta. Data version: 2011 to Oct 2017. Link to analysis: Link 3.2.2 Contents checks The analysis below shows that for majority of the patients and for majority of the visits, the disease data and medication (Treatment /Procedure) were non-missing. Most of the other categories were not entered as consistently as they should have been. If this was the expected data collection pattern then these findings should not be considered as any issues. Figure 3-2: Variable classification by categories

74 | P a g e 500+ variables are captured for each visit and each patient are classified into the following categories, (1) Ayurvedic data, (2) Background, (3) Disease, (4) Doctor's Notes, (5) Food / Exercise, (6) Hospital Visit, (7) Lab report, (8) Measurement, and (9) Treatment / Procedure. If there is any non-missing data present in a particular category then "Yes" is assigned, if the data is missing then "No" value is assigned. This data is presented as a listing for each patient for each visit (day). When the data is available it is presented as a color-coded bar and when it is missing then it is presented as a white blank space. Data version: 2011 to Oct 2017. Link to analysis: Link Two example screenshots are shown below as to what was observed while content check was performed. CRFname_variable number_variable label Unique values Unique patients Variable classification category sec001_var008_Diabetes 1064 4124 Background sec001_var008_Diabetes: represents CRF page number 1, labelled as "History of Present Illness" and variable number 8 "Diabetes". Based on the label of the variable, this variable is considered as a part of "Background" information. This variable "Diabetes" has 1064 unique values entered by different doctors for different patients. Data is available for 4124 distinct patients. CRFname_variable number_variable label Unique values Unique patients Variable classification category sec001_var018_Associate d Complaint with Onset & Duration 5102 4549 Disease sec001_var018_Associated Complaint with Onset & Duration: presents CRF page 1 "History of Present Illness" and variable number 8 "Associated Complaint with Onset & Duration". This variable "Associated Complaint with Onset & Duration" has 5102 different values entered by different doctors for different patients. Data is available for 4549 distinct patients.
75 | P a g e The large number of unique values show that the data entry rules are not being followed consistently, and each doctor or each nurse might have a different interpretation of the rules. In addition to this, looking at the unique number of patients, the data has not been entered for all the patients, hypothetically giving rise to missing data. Potential approaches to address these challenges are provided in chapter 4 section 4.2.8. 3.2.3 Visit pattern analysis Each of the cell displayed on a calendar display was coloured in shades of blue from light blue to dark blue showing increasing frequency count of number of patients. From 2011 to 2016, the number of patients visiting hospital on weekdays was less than the number of patients visiting on weekends. In-Patients were considerably less than Out-Patients. Overall number of patients coming to hospital have been increasing year on year. Potential approaches to address additional patient burden are provided in chapter 4 section 4.2.1. Figure 3-3: Visit pattern analysis Frequency counts of 4 parameters, (1) new Out-Patients added on that day, (2) total number of patients visiting on that day, (3) total number of In-Patient visits on that day, and (4) total number of Out-Patient visits on that day are calculated for each day to understand the patient flow to hospital from year on year. Light blue to dark blue shows increasing frequency count of number of patients Data version: 2011 to Oct 2016. Link to analysis: Link 3.2.4 Patient disease and treatment journey view Multiple representations of data allow the end user to review data with different perspectives. Patient profile reports provide detailed view of individual patient's disease condition, prescribed medication, co-morbidities along with basic demographic information. Treating doctors and researchers will greatly benefit from this visual display. This representation (Figure 3-4) provides the patient an understanding of the disease chronology as well as the prescribed
76 | P a g e medication and progress. Interpretations drawn from these representations and potential approaches to improve current version of patient profiles used at the hospital are provided in chapter 4 sections 4.2.8. Figure 3-4: Patient visit profile – Horizontal view Mr No: Patient ID, Patient gender, x-axis: duration of hospital visits, Orange bar: Out-patient visit, Blue bar: In-patient visit, Metabolic: when a metabolic disease is reported, RMSD: when a Rheumatic and Musculoskeletal disease is reported, OTHER: other diseases are reported. Patients listed have at least one of the RMSD or metabolic diseases. Tooltip has a lot of information relevant to each visit. Data version: 2011 to Oct 2017. Link to analysis: Link Figure 3-5: Patient visit profile – Vertical view Mr No: Patient ID, Patient gender, baseage Age at the very first hospital visit, category: Disease and medicine, Code: ACD code, Description: disease description, x-axis: duration of disease and medicine, Tooltip has a lot of information relevant to each visit. Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset Mr No = MR000774 3.3 Studying demographics and patient specific factors Results related to complete set of patients: While exploring the basic data, the following high- level picture appeared: For the 5-year time frame from 2011 to 2016, the database contained
77 | P a g e approximately 40,000 unique patients (Figure 3-6), 90% of patients were from India and remaining 10% patients were from more than 50 different countries (Figure 3-7). The proportion of male and female patient was approximately 50%. Median age for females was marginally higher than males across all visit types (Figure 3-8). Approximately 90% of patients were Out- Patients and 10% were In-Patients (Figure 3-1). Approximately12,000+ female patients and 14,000+ male patients had reported only a single disease (Figure 3-11), these patients could have come only once to the hospital and may not have come back at all after reporting the first disease. There were a few outliers observed having more than 10 disease conditions across the years. The maximum age of 108 years was a possible case of data issue. Similar anomalies were seen in a few other groups, e.g., patients reporting 23 diseases, is this accurate? This warrants additional data checks from operational and clinical perspective. Blood group is collected only for ~32,500 out of ~40,000 patients. There was missing data for almost 20% of patients. Blood group distribution was largely in line with the Indian blood group distribution (Figure 3-9). Potential approaches to improve current situation and interpretations are provided in chapter 4 section 4.3. Figure 3-6: Total Number of Patients Grad Total: Total number of patients used in the analysis, OnlyIP: patients having only In-Patient visits, OnlyOP: patients having only Out-Patient visits, Common: Patients having both type of visits. Data version: 2011 to Oct 2016. Link to analysis: Link Figure 3-7: Country-wise Visualization
78 | P a g e Unique number of patients are plotted on the world map, the map shows that at least 1 patient data is coming from 50+ countries, 95% or more patients are from India. Data version: 2011 to Oct 2016. Link to analysis: Link Figure 3-8: Age distribution by country, age distribution by gender Boxplot representation: Age distribution is presented for each country and then by type of patient and gender, OnlyIP: patients having only In-Patient visits, OnlyOP: patients having only Out-Patient visits, Common: Patients having both type of visits Data version: 2011 to Oct 2016. Link to analysis: Link01, Link02
79 | P a g e Figure 3-9: Blood-group Distribution by gender Tabular frequency distribution table for Blood-group by gender. Data version: 2011 to Oct 2016. Link to analysis: Link Figure 3-10: Number of Visits, and Visit Types

80 | P a g e Boxplot representation of number of Visits, and Visit Types, Data version: 2011 to Oct 2016. Link to analysis: Link Figure 3-11: Descriptive summary statistics by number of Diseases by Age and Gender Descriptive summary statistics by number of diseases reported, by gender. Noofdisases: Number of diseases reported in the database. Data version: 2011 to Oct 2016. Link to analysis: Link Results for the metabolic and RMSD disease areas: Out of ~40,000 patients, there were ~14,000 patients having reported at least 1 metabolic and/or 1 RMSD disease condition. It was quite evident that there were a lot more patients in the RMSD group compared to the metabolic group

81 | P a g e (Figure 3-12). Large number of patients were visiting the hospital only for 1 visit, ~62% patients were dropping off in first month of treatment. ~15% of patients having at least one RMSD disease were still visiting the hospital after 1 year of first ever visit to the hospital. Boxplot representation of age showed variability in age across disease type and gender (Figure 3-13). Presentation of disease burden by gender, Indian seasons (rutus) and disease category provides data about possible variations reported for different diseases (Figure 3-15). (1) Prameha, (2) Madhumeha, and (3) Sthaulya were the top three most frequently reported metabolic diseases where as (1) Vaatavyaadhi – Sandhigata Vaata, (2) Vaatavyaadhi, (3) Vaatavyaadhi – Gridhrasee, (4) Sthaanabhedana Shoola – Katee Shoola and (5) Sthaanabhedana Graha – Katee Graha were the top five most frequently reported RMSD diseases. Prameha and Madhumeha were reported more by males than females. There were more female patients with disease condition Sthaulya. In general, RMSD diseases were reported in more females than males. For RMSD disease group, 51 out of 97 diseases were reported in &gt;= 10 patients. Metabolic diseases were not varying across seasons, while RMSD diseases had some seasonal variations (Figure 3-15). The before and after visualization of data allows to build a disease and medicinal trajectories (Figure 3-16). These should be useful for determining diagnostic and prognostic relationships. Figure 3-12: Data tabulation for patients reporting RMSD and Metabolic diseases Bubble plot: 1 = Patients with at least 1 metabolic diseases, 2 = Patients with at least 1 Rheumatic, Musculoskeletal (RMSD) diseases, 99 = Patients with at least one disease from each of the groups, Data version: 2011 to Oct 2016. Link to analysis: Link

82 | P a g e Figure 3-13: Disease distribution by age and gender Boxplot representation of age by disease. Orange: male, Blue: female. Individual column represents a disease. 1 = Patients with at least 1 metabolic diseases, 2 = Patients with at least 1 Rheumatic, Musculoskeletal (RMSD) diseases, 99 = Patients with at least one disease from each of the groups, Data version: 2011 to Oct 2016. Link to analysis: Link The columns in the above image are different diseases from Metabolic and RMSD categories. (Refer Link for codes and de- codes) Figure 3-14: Patient visit duration for Disease categories by Gender 1 = Patients with at least 1 metabolic diseases, 2 = Patients with at least 1 Rheumatic, Musculoskeletal (RMSD) diseases, 99 = Patients with at least one disease from each of the groups, Data version: 2011 to Oct 2016. Link to analysis: Link Figure 3-15: Disease distribution by Seasonal Variations and gender

83 | P a g e Distype: Metabolic and RMSD, Code: ACD disease code, Description: disease description, seasons are presented as: Vasant rutu, Grishma rutu, Varsha rutu, Sharad rutu, Hemant rutu, and Shishir rutu, Data version: 2011 to Oct 2017. Link to analysis: Link Figure 3-16: Pre and Post Disease Classification Analysis Example 1: Prameha Cat: category of disease and medicine, Code: ACD code, prescribed medicine types, pre and post visit window w.r.to the 1 st day of each of the reference diseases, Data version: 2011 to Oct 2017. Link to analysis: Link to get the display for Prameha, go to “Refcode, Refdesc” filter and select P5.0, Prameha. Example 2 Vaatavyadhi – Sandhigata Vaata

84 | P a g e Cat: category of disease and medicine, Code: ACD code, prescribed medicine types, pre and post visit window w.r.to the 1 st day of each of the reference diseases, Data version: 2011 to Oct 2017. Link to analysis: Link to get the display for Vaatavyaadhi – Sandhigata Vaata, go to “Refcode, Refdesc” filter and select V2.63, Vaatavyaadhi – Sandhigata Vaata. 3.4 Studying diagnostics and interventions Almost all the ICD categories were represented in the analysis. Some of the ICD classes had more patients than other categories. Age distribution showed natural variation. Visit distribution and duration for which patients were visiting hospital look like the earlier analysis. The ACD and ICD mapping exercise showed that the current hospital data demonstrates all the types of diseases being catered to at the hospital. (Figure 3-17, Figure 3-18, Figure 3-19 , Figure 3-20) The bar graph representation (Figure 3-21) provided a view of the spread of the patient population their diseases versus their prakriti classification seen at the hospital. The prakriti type as well as disease type can be filtered. It also provided a view of the combination of the gender and the prakriti manifesting into the kind of doshas and the most prevalent doshas for the combination. Observations from the first Co-morbidity approach (Figure 3-22, Figure 3-23, Figure 3-24): This display provided a comprehensive view of the disease clusters. Comorbidities were easily identified, some of them are clinically relevant, and some of them are not. Bubble size provided comparative view of number of patients reporting a specific disease. The age group distribution for each gender was available. Some diseases were reported more by males or by females, easy to spot on the graph. Box named “Number of other diseases” provided a contextual display about number of co-morbidities. Some diseases had higher number of co-morbidities, some had lower number. Variations were seen amongst gender as well. This analysis did not consider the before or after nature of time points, hence did not provide insights into the causal relationships between diseases. The second co-morbidity approach provided views on the seasonal variations of diseases as well as seasonal co-morbidities (Figure 3-25). The third co-morbidity approach provided the following: The collapsible tree showed progression of diseases as experienced by patients and reported in the database. The tree showed approximately 12,500 lines of data in very short space. Some diseases were experienced more by males than by females. Some diseases

85 | Page were only reported by one of the genders. Some diseases had many more branches than a few others. Some of the disease trajectories had very few numbers of patients. Some of the trajectories may be clinically meaningful and some may not be meaningful (Figure 3-26). Treatment and disease analysis at individual patient level: This analysis was explained using example patient MR000335: For this patient there were 17 visits in the database, there were 17 distinct diseases reported and 45 distinct treatments, services prescribed. 4 out of 17 diseases were repeated and 10 out of 45 treatments had been repeated. When a new disease was reported, usually a new treatment or treatments re-reported, if there was only a new treatment added then it could indicate, the earlier treatment may not have worked, or it described the treatment regimen. If only new diseases were added and no new treatment was added, then the same treatment could work for multiple diseases. These visualizations allow the treating doctor insights into newer diseases getting reported as well as what newer treatments have been prescribed at what time points (Figure 3-27, Figure 3-28, Figure 3-29). Area graph representation of diseases provided information about 800+ diseases, almost all diseases present in the database in very short space. Due to the data visualization scheme variations caused by day-to-day, seasons, gender, and diseases could be interpreted very easily. The interactive nature of visualization allowed for real time subset of diseases. One of the 4 diseases displayed has very few patients compared to other 3 diseases showing different nature of diseases (Figure 3-30). Cross tabulation of prescribed treatments and disease group by gender was generated. First example is created using Balaristham. The source variable captured the quantity + unit + company name in the same variable, which did not allow for 100% accurate numerical calculations, but still provided a good idea. Only 30 patients having metabolic diseases were prescribed the medicine whereas 1,142 patients with RMSD were prescribed. Second example is created using bhasma: approximately 287 patients out of 17,406, 1.5% of patients are prescribed bhasmas (at least treatments having word "bhasma") for various diseases (Figure 3-35, Figure 3-36). Summary statistics and hypothesis testing was conducted to conclude any impact of bhasma on visit duration. 514 patients were identified with at least 1 bhasma treatment. The mean duration of treatment before 1 st bhasma treatment was 14.8 days, min - max duration was reported as (1, 111) whereas the mean duration of treatment after 1 st bhasma treatment was 10.6 days, min - max duration was reported as (0, 89). The t-test at 5% significance level shows statistically significant difference between duration of treatment before bhasma treatment and duration of treatment after bhasma treatment. Circular display, how to read the visualization? Figure 3-37 and Figure 3-38 show 2 examples of 2 combinations, the first example had many green bars, and the second combination had very few green bars. Details about the display: For each reference disease 1 page was created. Each page was controlled by a combination of "Reference disease + disease", "Reference disease + medicine"

86 | Page Reference disease window: , Reference disease or medicine window: Tables displayed in the top part of the display: there were 9 columns created for each time point. 2 columns were displayed in each time point to display "count of distinct number of diseases" and "count of distinct number of medicines". Count of distinct number of diseases: Count of distinct number of medicines: There were 3 rows for "Before period", "Day 1" and "After period" with 2 lines in each period. Day 1 cell showed the start day of reference disease "V2.23: Vaatavyadhi – Gridhrasee", This example showed 137 total number of distinct diseases reported and 664 total number of distinct medicines prescribed on day 1 for this combination of reference disease "V2.23: Vaatavyadhi – Gridhrasee" and disease "A6.0: Amlapitta". The first line in the Day 1 cell showed, 1 disease – which is "A6.0: Amlapitta" and 81 distinct medicines prescribed. These 81 different treatments could have been prescribed for "A6.0: Amlapitta".

87 | Page Cells in the "Before period" line provided the following information: This example shows 1 month before "V2.23: Vaatavyadhi – Gridhrasee", there were 61 distinct diseases and 298 distinct medicines reported. The first line in the cell shows, 1 disease – which is "A6.0: Amlapitta" and 11 distinct medicines prescribed. These 11 different treatments could have been prescribed for "A6.0: Amlapitta". Cells in the "After period" line provide the following information: This example shows 1 month before "V2.23: Vaatavyadhi – Gridhrasee", there were 96 distinct diseases and 633 distinct medicines reported. The first line in the cell shows, 1 disease – which is "A6.0: Amlapitta" and 49 distinct medicines prescribed. These 49 different treatments could have been prescribed for "A6.0: Amlapitta". The bottom section follows the same structure as the top section. The following table provided distinct number of diseases reported and distinct number of medicines prescribed for this particular combination. There were 308 diseases and 1,357 medicines reported for this

88 | Page combination of reference disease "V2.23: Vaatavyadhi – Gridhrasee" and disease "A6.0: Amlapitta". Explanation about the circular view: The starting point marked the position of the other disease in this case, "A6.0: Amlapitta", The green colored spokes going from point of origin were different treatments prescribed. These were showing 664 distinct medicines prescribed on day 1. Hovering tooltip provided details about the disease, treatment name and count of number of patients: The inner circle displayed the diseases. And the outer circle displayed the treatments. Distance based Pre and post analysis: in the example for the disease M2.0 (Madhumeha), disease trajectory distances were plotted (Figure 3-39). (1) Madhumeha was reported by 1,441 patients at least once. (2) Of these 567 patients reported only Madhumeha and no other disease thereafter. For such patients, the disease trajectory calculation was not possible, hence these patients were removed from the analysis. (3) The following table showed details of patient count. The disease trajectory calculation was based on 874 patients comprising of 530 males and 344 females

89 | P a g e Out of 530 male patients: (1) 201 patients had diseases reported before the first reported instance of M2.0, (2) 460 patients had at least one other disease reported other than M2.0 on or after the first reported instance of M2.0, (3) Disease trajectory for 70 patients could not be calculated since the next reported disease was M2.0. Out of 344 female patients: (1) 140 patients had diseases reported before the first reported instance of M2.0, (2) 291 patients had at least one disease other than M2.0 reported on or after the first reported instance of M2.0, (3) Disease trajectory for 53 patients could not be calculated since the next reported disease was M2.0. The trajectories were calculated for these patients and displayed for the before and after period. More number of patients had disease trajectories in the "after onset" section. More than 73% of males lie in the score &lt;0.25 and around 36% of them lie in the score&lt;0.5 which could confirm that there were similar diseases experienced by the patients post the onset of the reference disease. Similarly, around 64% of females lie in the score &lt;0.25 and around 24% of them lie in the score&lt;0.5 which could confirm that there were similar diseases experienced by the patients post the onset of the reference disease. A few more examples of similar kind were shown for diseases: P5.0: Prameha, V2.23: Vaatavyaadhi - Gridhrasee, V2.63: Vaatavyaadhi - Sandhigata Vaata (Figure 3-40). The radar plot shows multi-dimensional data in a short space, 7 different parameters were shown on 7 vertices. Different shapes suggest that there were underlying differences to the data structure. Dynamic bubble plot visualization showed intricate relationships and details in a limited space, one more analysis method to see multi-dimensional data (Figure 3-42). Using data for Amavaata (A6.0), in the snapshot below, we could see the patients with Aamvata who received Rasayanam Madiphala and in addition what were the other diseases that these patients reported.

90 | P a g e Another view which showed the further detailed view of Aamvata patients having received Rasayanam Madiphala and further having reported Stanbheedana Kathee Shoola disease and further its treatment. Further, the big bubble here closer to Stanbheedana Kathee Shoola disease was Aamdosha which showed that Stanbheedana Kathee Shoola was also one of the diseases reported by Aamdosha patients.

91 | P a g e Figure 3-17: ICD classification by Gender

92 | P a g e ICD classification and ICD classification high level categories, frequency counts by gender. Data version: 2011 to Oct 2017. Link to analysis: Link Figure 3-18: Age distribution by ICD classification and Gender Boxplot representation of age by ICD classification and gender, Data version: 2011 to Oct 2017. Link to analysis: Link

93 | P a g e Figure 3-19: Visit distribution by ICD classification and Gender Boxplot representation of hospital number of visits by ICD classification and gender, Data version: 2011 to Oct 2017. Link to analysis: Link Figure 3-20: Duration distribution by ICD classification and Gender Boxplot representation of hospital visit duration by ICD classification and gender, Data version: 2011 to Oct 2017. Link to analysis: Link

94 | P a g e Figure 3-21: Disease classification by Prakriti and Gender Dosha: Prakriti type, x-axis: male and female grouped by individual disease, y-axis: frequency counts of unique patients, Data version: 2011 to Oct 2016. Link to analysis: Link to get the above display subset Dosha for Pitta.

95 | P a g e Figure 3-22: Co-morbidity analysis approach 1 example 1: Vaatavyadhi Summary stats section: descriptive statistics details by gender and other diseases reported, No of other disease: distinct number of other diseases reported by patients who had reported the primary disease, Bubble plot: frequency count of distinct patients by disease, Boxplot: age distribution by disease and gender, Data version: 2011 to Oct 2016. Link to analysis: Link to get the above display subset Primarycode = V2.0 Figure 3-23: Co-morbidity analysis approach 1 example 2: Pandu Summary stats section: descriptive statistics details by gender and other diseases reported, No of other disease: distinct number of other diseases reported by patients who had reported the primary disease, Bubble plot: frequency count of distinct patients by disease, Boxplot: age distribution by disease and gender, Data version: 2011 to Oct 2016. Link to analysis: Link to get the above display subset Primarycode = P2.0

96 | P a g e Figure 3-24: Co-morbidity analysis approach 1 example 3: Madhumeha Summary stats section: descriptive statistics details by gender and other diseases reported, No of other disease: distinct number of other diseases reported by patients who had reported the primary disease, Bubble plot: frequency count of distinct patients by disease, Boxplot: age distribution by disease and gender, Data version: 2011 to Oct 2016. Link to analysis: Link to get the above display subset Primarycode = M2.0 Figure 3-25: Co-morbidity analysis approach 2 Upper section: bubble plots for the reference disease and other diseases reported, bubble size is based on number of distinct patients. Lower sections: unique number of other diseases reported for the reference disease. 1, 2, ..., 12: January to December month, Data version: 2011 to Oct 2016. Link to analysis: Link to get the above display subset Primarycode = V2.0 Figure 3-26: Co-morbidity analysis approach 3: collapsible tree view Initial view of the tree

97 | P a g e After clicking on F (Female), the collapsible tree opens up An example of a disease experienced only by one gender https://coursephd.github.io/ Figure 3-27: Patient Disease and Treatment administration by Study Day

98 | P a g e The report displays individual patient data. Upper left part: Mr No: Patient ID, study day, Disease reported 1 st time and repeated, medicine reported 1 st time and repeated, Lower part of the report displays individual patient data for each day and distinguishes 1 st dose, 1 st disease and Repeat reporting for the same. Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset Mr No = MR000335 Figure 3-28: Patient Disease by Study Day and Treatment administration by Study Day The report displays individual patient data. Panel on the left hand: number of diseases reported at a particular visit, panel on the right hand: number of prescribed treatments reported at a particular visit, Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset Mr No = MR000335 Figure 3-29: Patient Cumulative Disease and Treatment administration by Visit The report displays individual patient data. Panel on the left hand: absolute values of diseases and prescribed treatments till particular visits, panel on the right hand side: % values of diseases and prescribed treatments till particular visits, Data version: 2011 to Oct 2017. Link to analysis: Link Figure 3-30: Area graph representation of diseases

99 | P a g e Area graph representation of diseases: x-axis: Month, y-axis: frequency count of unique patients, disease code: the underlying data for each disease, peach colour: counts for male, blue colour: counts for female. x-axis can be expanded to covert the monthly view to more granular unit (week, day). The original data displayed in the 1 st part of the presentation is opened for daily view for a particular disease code A2.0, Data version: 2011 to 2016. Link to analysis: Link to get the above display subset the Code for (A2.0, M2.0, P2.0, P5.0) Figure 3-31: Mosaic plot: Disease and treatment representation example 1: Prameha

100 | P a g e Mosaic plot: Each box is one disease, the selected disease is marked in Green colour, smaller boxes inside each disease display one intervention each, Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset SubDisMed = Metabolic: P5.0: Prameha The snapshot above is the zoomed version of the Prameha block which highlights one of the many treatments prescribed for the same.

101 | P a g e The Medicine count tab provides the information on total number of different medicines prescribed for Prameha. There are 732 distinct interventions. A detailed list of medicines with the total patients prescribed with them as well as total patients suffering from Prameha are displayed in the snapshot above. Figure 3-32: Disease and treatment example 2: P5.0: Prameha and Oil: Kottamchukkadi

102 | P a g e Totpatmed shows that there are 4,604 patients who have been prescribed with Oil: Kottamchukkadi; Totpatdis shows that there are 1,456 patients who are diagnosed with Prameha; this shows that the treatment is not Prameha specific. The n count of 324 shows the patients who had Prameha and were prescribed Oil: Kottamchukkadi. Percdis shows that 22.25% i.e. 324 / 1456 of patients having Prameha are prescribed this particular treatment. Percmed shows that only 7% of the time this medicine has been prescribed for Prameha patients. Figure 3-33: Disease and treatment example 3: P5.0: Prameha and Vati: Diabecon DS

103 | P a g e Totpatmed shows that there are 476 patients who have been prescribed with Vati: Diabecon DS; Totpatdis shows that there are 1,456 patients who are diagnosed with Prameha; this shows that the treatment could be more prescribed to Prameha patients. The n count of 145 shows the patients who had Prameha and were prescribed Vati: Diabecon DS. Percdis shows that 9.96% i.e. 145 / 1,456 of patients having Prameha are prescribed this particular treatment. Percmed shows that only 30.46% of the time this medicine has been prescribed for Prameha patients. Figure 3-34: Mosaic plot Disease and treatment representation example 4: Treatment: Oil: Kottamchukkadi Mosaic plot: Each box is one disease, the selected disease is marked in Green colour, smaller boxes inside each disease display one intervention each, Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset SubDisMed = Oil: Kottamchukkadi The Disease count tab provides the information on total number of different diseases for which the intervention was prescribed. There are 62 distinct diseases.

104 | P a g e A detailed list of diseases with the total patients prescribed with the treatment as well as total patients suffering from different diseases are displayed. Figure 3-35: Cross tabulation of prescribed treatments and disease group by gender Example 1 Cross tabulation of medicine, disease type and patient gender, Medicine name: source data collected on Case Report Form, Distype: Metabolic and RMSD groups derived in the analysis dataset. Patient Gender: source data. Only 30 patients having metabolic diseases were prescribed the medicine whereas 1,142 patients with RMSD were prescribed. This reflects the ayurvedic principle of who should be prescribed any aristham. Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset Medicine name = Balarishtam

105 | P a g e Figure 3-36: Cross tabulation of prescribed treatments and disease group by gender Example 2 Cross tabulation of medicine, disease type and patient gender, Medicine name: source data collected on Case Report Form, Distype: Metabolic and RMSD groups derived in the analysis dataset. Patient Gender: source data. Only 287 (1.5%) patients have been prescribed bhasma. This reflects the ayurvedic principle of using bhasma based treatment wisely. Data version: 2011 to Oct 2016. Link to analysis: Link to get the above display subset Medicine name = Bhasma Table 3-2: Summary statistics and t-test for bhasma usage

106 | P a g e Patients who were prescribed at least one "bhasma" treatments are summarized, Pre_Bhama: Duration of treatment before the 1 st bhasma treatment, Post_Bhasma: Duration of treatment after the 1 st bhasma treatment, Data version: 2011 to Oct 2017 https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R Figure 3-37: Circular view: Co-occurrences of disease – disease Example 1 Example 1: Disease: A6.0: Amavaata and Reference Disease: V2.23: Vaatavyadhi – Gridhrasee, Upper section: Pre and post time windows, count of distinct diseases and count of distinct medicines prescribed at the given time point. Lower section: 1 st row represents the co-occurrence of disease – disease and / or disease – treatment before day 1 of the reference disease. Middle row: On day 1 count of distinct diseases and count of distinct medicines prescribed. Last row represents the same co-occurrence data after day 1 of the reference disease. Green bars inside a circle show co-octene of chosen disease – disease and / or disease – treatment combination, Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset Refcode, Refdesc = V2.23 Vaatavyadhi – Gridhrasee and HighlightCode = A6.0: Amlapitta Figure 3-38: Circular view: Co-occurrences of disease – treatment Example 2

107 | P a g e Example 2: Treatment: Arishtam:Dhanwanaristham and Reference Disease: V2.23: Vaatavyadhi – Gridhrasee, Upper section: Pre and post time windows, count of distinct diseases and count of distinct medicines prescribed at the given time point. Lower section: 1 st row represents the co-occurrence of disease – disease and / or disease – treatment before day 1 of the reference disease. Middle row: On day 1 count of distinct diseases and count of distinct medicines prescribed. Last row represents the same co-occurrence data after day 1 of the reference disease. Green bars inside a circle show co-octene of chosen disease – disease and / or disease – treatment combination, Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset Refcode, Refdesc = V2.23: Vaatavyadhi – Gridhrasee and HighlightCode = Arishtam:Dhanwanaristham Figure 3-39: Pre and Post distance analysis for disease: M2.0: Madhumeha Butterfly plot display of pre and post distance analysis for disease trajectories. Refcode: reference disease, Patient gender, Allcapn: Total number of patients in each of the categories, Cut: Jaccard distance, scr Before: patients falling in a particular category before day 1 of reference disease, scr After: patients falling in a particular category after day 1 of reference disease, Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset Refcode = M2.0 Figure 3-40: Pre and Post distance analysis for medicines given for diseases: P5.0, V2.23, V2.63

108 | P a g e Butterfly plot display of pre and post distance analysis for medicine trajectories. Refcode: reference disease, Patient gender, Allcapn: Total number of patients in each of the categories, Cut: Jaccard distance, scr Before: patients falling in a particular category before day 1 of reference disease, scr After: patients falling in a particular category after day 1 of reference disease. This distance calculation was done on the basis of trajectory of prescribed treatments, Data version: 2011 to Oct 2017. Link to analysis: Link to get the above display subset for Refcode in (P5.0, V2.23, V2.63) Figure 3-41: Radar plot Radar plot showing multiple diseases displayed side by side.

109 | P a g e Example 1: Multidimensional view a single disease: A6.0: Aamavaata 7 parameters displayed on one plot: (1) Unique patients, (2) Number of times disease reported, (3) Disease chronology, (4) Number of diseases reported before the reference disease, (5) Number of disease reported after the reference disease, (6) Number of medicines prescribed before the reference disease, (7) Number of medicines prescribed after the reference disease, Data version: 2011 to Oct 2017. Link to analysis: Link Example 2: Multiple disease comparison

110 | P a g e Figure 3-42: Dynamic bubble plot: Example 1: Disease: A6.0: Amavaata

111 | P a g e This dynamic bubble plot shows relations between diseases and medicines, the bubble plot size is based on number of unique patients, Data version: 2011 to Oct 2017, Amavaata [https://coursephd.github.io/nodediagram/A2_0/]

112 | P a g e 4 Discussions 4.1 Converting real life clinical data into analyzable format The TDU / I-AIM team should be congratulated first before any discussion to create an electronic database right from the inception of the hospital. This foresight has allowed us to have significant amount of data. There are a lot of learnings from this exercise which can be beneficial to many institutes and hospitals. Conversion of real-life clinical data from an individual data point to a logical dataset was done. Logical relationships were established post inspection of the datasets and the columns. Relational datasets were identified. Observations about data capture methods and storage were noted. Some shortcomings and errors in the data were seen and noted (missing data, inconsistent values, or unresolved duplicates). This exercise of understanding technical architecture from "an end user point of view" will help in running analysis of various types. If data generation for future use is one of the top priorities for the hospital, then there should be a project plan put together and appropriate steps should be taken to plug the existing gaps. 4.2 Clinical data understanding Sections listed below offer possible technical solutions for the shortcomings identified: 4.2.1 Visit pattern analysis From 2011 to 2016, the number of patients visiting hospital on weekdays was less than the number of patients visiting on weekends. In-Patients were considerably less than Out-Patients. Overall number of patients coming to hospital have been increasing year on year. This information would help in employing staff across different departments from helpers, cleaners, nurses to doctors to adequately cover services for patients (Figure 3-3). 4.2.2 Vital sign dataset The vital signs database could have an alternative presentation of one record per patient per visit in addition to the existing presentation. The vital sign parameters can be presented as distinct columns, one each for each parameter. Table 4-1: Proposed vital sign data structure Patient ID Visit date SBP DBP Pulse Height Weight 1 01-Jan-2016 xxx xxx xxx xxx xxx 1 15-Jan-2016 xxx xxx xxx xxx xxx 1 31-Jan-2016 xxx xxx xxx xxx xxx This type of presentation would make the length of data smaller. Trends for the same patient over a period could be assessed faster. The parameter result values should be presented in

113 | P a g e numeric form, rather than character format. This will allow the data to be used for numeric calculations. In case of age and/or gender specific analysis; normal ranges can be applied in the database and these calculations could be done in the backend without affecting the end users, here doctors, nurses to name a few. 4.2.3 Lab dataset When a lab test is done from other pathology the data from scanned reports is not translated into hospital dataset, this results in missing information in the database, as it is not retrievable for any analysis. A single test has multiple names making it very difficult to create summaries on laboratory parameters. Would it be possible to get the pathology lab data in electronic dataset format? A few suggestions on updating the existing version: • A standard naming convention of lab tests should be created • A standard units look up table should be built for possible conversion from one unit to another • Lab results values should be saved as numeric and character (when some test results come out as ordinal scale measurements) variables • In case of cancer patients, should the National Cancer Institute, USA, proposed NCI CTC grading variable be created for specific parameters? – this could be created in the backend database, more useful for analysis [113] 4.2.4 Treatment dataset This is one of the most important parts of data necessary for any type of analysis. A complex SQL data extraction code along with numerous merges and complex Cartesian products via the R-code had to be performed to arrive at easily readable dataset. E.g. MRD_Diagnosis, Patient_Prescription, Patient_Medicine_Prescriptions, IP_Prescription datasets had to be merged to get a complete interventional view. Treatment database could be structured in a way that components can be collected and reported in a systematic manner: • Name of the treatment(s) • Treatment(s) prescribed for which disease • Treatment(s) start date • Treatment(s) end date • Names of medications • Type of medications (classical formulation, proprietary, etc.) • Dosing information • Route of administration (Treatment procedure, oral treatment, panchakarma, etc.) • Dose increase or decrease Due to the complex nature of the data, the structure would be one record per patient, per visit, per disease, per treatment assigned. E.g. if a patient has 2 disease conditions and 4 treatments are assigned then for that particular visit, there should be 8 records present in the database. There are numerous medicines prescribed. These medicines are classified into following broad categories:

114 | P a g e 1. Abhyanga 2. Aristham 3. Arka 4. Asavam 5. Avagha 6. Bhasma / Bhasma Cap / Bhasma Tab 7. Dhara 8. Ghritam and variations of Ghritam 9. Kashayam and variations 10. Kshar 11. Lehyam 12. Oil 13. Pichu 14. Rasayanam 15. Any additional classification which makes sense Source variable for treatment will be classified into the following categories. This should allow recreation of treatment protocol as per Ayurvedic principles. Various Ayurvedic texts have defined standard treatment protocols for different ailments. Based on the table below, can we propose a sequence of treatment for various conditions and build it in our database so that an automatic re-creation of the classical treatment text can be done? This will not only serve as a support to the practicing vaidyas but also serve as a validation tool for the given treatment regime. Figure 4-1: Treatment principles defined in different texts

115 | P a g e 4.2.5 Medical coding Medical coding is a robust method to simplify the variation in the data by uniformly categorizing the medical terms appropriately. This step allows us to maintain high quality database. Coded medical data is a standardized form of data, globally approved, and can aid in future machine learning and automation. The most used medical coding dictionaries for coding medical terms are MedDRA and WHO DDE [114] [115]. A few examples of medical dictionaries are: • COSTART: Coding Symbols for Thesaurus of Adverse Reaction Terms • ICD xx CM: International Classification of Diseases xx Revision Clinical Modification [116] • MedDRA: Medical Dictionary for Regulatory Activities [114] • WHO-ART: World Health Organization Adverse Reactions Terminology [115] • WHO-DDE: World Health Organization Drug Dictionary Enhanced [115] • ACD: Ayurvedic Classification of Diseases • NCI: National Cancer Institute Code list [113] • LOINC: Logical Observation Identifiers Names and Codes standards [117] • Any other dictionaries, as recommended by AYUSH 4.2.6 Classification and Sub-classification of the Doshas / Diseases Almost each researcher understands that research results are as good as the data using which the conclusions are drawn. Most scientists do not receive guidance in methods for controlling the quality of research data which is fundamental to clinical research. An exhaustive list of all possible diseases should be created, and a checklist of disease classification and sub- classification should be maintained so that the doctors based on their judgment can classify the dosha appropriately and use the recommended disease classification term. This should help in reducing inconsistency and disparity in reporting (Figure 3-1). The problem should be split into operational and scientific components. Identify fields which would require coding: Diagnosis codes description, Compliant, Drug description. Operational steps to be taken as follows: • The existing data should be codified in a retrospective manner • Business guidance document should be prepared on how to work in future • Number of days should be predefined to have data coded from the time of patient visit • An automatic tracking mechanism should be defined to keep track of the status: Table 4-2: Proposed idea for clinical coding timetable Number of days from patient visit to coded data Number of records coded Number of records yet to be coded &gt; 7 days XX XX 7 days to 14 days XX XX 14 days to 28 days XX XX

116 | P a g e &lt; 28 days XX XX Temporary staff should be allocated to complete the backlog. Scientific questions regarding Ayurvedic medical terminology should be answered by doctors at hospital. A team of 3-4 doctors for a period of 4 months or as appropriate, contributing 20% of their time, would help complete the categorization process as described above or post graduate students, under the guidance of a senior vaidya can take on this responsibility. 4.2.7 Patient profile module A patient profile report is a consolidation of all the data for a patient available in the database. This consolidated view at a patient level provides an easy access to the patient history (Figure 3-4, Figure 3-5). If this type of a report is electronically available for a patient then a patient's case can be handled by any doctor available. The contents of a good patient profile are outlined below: • All the demographic characteristics of a patient: age, sex, race, religion, place of residence, etc. • All the useful data for operational ease: policy number, health coverage status, in-patient, Out-patient, etc. • Visit information: number of visits to the hospital, corresponding dates and day of visit. The day should be calculated based on the first visit date (visit date − first visit date + 1). This value must never be missing and must be positive. • Background disease history: Is the patient disease history getting captured at first visit of each patient? Would it be useful to not down the background history in a systematic manner? • Vital sign measurements: a tabular view of the collected vital sign measurements. • Data collected for the diseases and diagnosis: details about the clinically relevant fields should be discussed. Some standard fields – o Complains as reported and medically coded as per section 4.2.5 o Duration of disease or start date, end date o Data collected for Ayurvedic examination: variables outlined in dash vidh pariksha (if these variables are not captured currently then how to make provisions for the same?) • Treatments administered: o Treatment start date o Treatment end date o Names of medications o Type of medications (classical formulation, proprietary, etc.) o Dosing information o Route of administration • Details of lab results

117 | P a g e • Outcomes • Patient status still ongoing or discontinued (need an algorithm) The pictorial representation (Figure 4-2), summarizes the cycle of understanding the hospital data so that a meaningful interpretation can be arrived at. 6 steps provide a way of generating very good quality data: (1) Understand the data from variable and observation point of view, (2) Collect consistent data across case report forms across visits, (3) Maintain consistency across patients, (4) Maintain consistency across disease areas, (5) Strive to maintain completeness to provide overall clinical picture, and (6) These steps should enable translation of thoughts from mind to data for future use.

118 | P a g e Figure 4-2: Data understanding from an observation – patient – disease to a clinical picture 4. Consistency within the disease area / Therapeutic area 5. Completeness to provide overall clinical picture 3. Consistency across patients (as appropriate) 2. Consistent data across case report forms across visits Data checks to understand the collected data 1. Understand data from variable and observation point of view 6. Translations of thoughts from mind to paper for future use

119 | P a g e 4.2.8 Improvements to the system architecture Overall the hospital data is not captured in a standardized format as explained in the above points (Figure 3-2). Create standardized CRF pages for consistent and correct data capture. Already established standards like Clinical Data Interchange Standards Consortium (CDISC) or International Organization for Standardization (ISO) standards can be implemented [118] [119]. Appropriate drop-down menu lists with predefined inputs to be built into the system, with the help of experts, to ensure good quality data. Usually any team in any organization sets up rules and guidelines for the implementation. Yet once the system is live, due to the lack of consistency in data entry methodology things begin to fall apart. User inputs the same data in different ways. New staff comes on board and has their own way of entering information. Inconsistent data creates inaccurate reports. Hence robust documentation and streamlined training and onboarding of the new data entry operators is a must. Building and implementing a design policy is the first step towards reinforcing the build rules. It provides documentation for Electronic Medical Record (EMR) analysts to follow which will reduce inconsistencies and improve the EMR functionality. Below are some of the aspects which if kept in check can avoid inconsistency in data • Capitalization: Monitor the use of capitalization. Create guidelines and instruct accurately when to use caps and otherwise. In case all caps are used, ensure appropriate warning message or suggestion is generated to alert the end user. • Abbreviations: Prepare a catalog of all allowed abbreviations along with their meanings. If possible, create inbuilt checks in the building forms to avoid incorrect abbreviations. • Workflows: Evaluate and monitor the workflows to check the effectiveness on an ongoing basis. During a system change a workflow audit is extremely important, since non-working workflows undermines' the system functionality as the user may create smart workarounds skipping the important steps. Along with Workflow audits it is a best practice to have planned system audits. • Naming conventions: This is the most important step. Following the appropriate naming conventions helps save time, money, and future efforts. It is easier to onboard new employees and eases future searches or any kind of analysis. • Data Quality check plan: It is best to create data validation programs during data base setup which can be run periodically to check for the correctness of data entry. • Well defined database maintenance plan: It is important to have a well-planned and periodically scheduled database maintenance program. • Regular training: Regular training and refresher programs is a must to ensure that the end users are up to the mark with the processes and systems so that a healthy database can be maintained. If any organization can follow these proposed solutions, then possible outcomes could be as follows: • The data will be closer to analysis ready format

120 | P a g e • The database will be useful in publishing case studies, case series, etc. in very short period. This will help us gain more visibility in scientific world • More empirical data will be available at our disposal • The database could become a model database for other Ayurvedic institutions to follow Variable classification analysis: Review of the database suggested that the case report form completion was carried out by different doctors differently giving rise to differences in the way the data was captured. This lack of documentation should be addressed. Hospital management, treating doctor, researchers, and insurance companies could be the key stakeholders benefitting from these improvements. 4.3 Studying demographics and patient specific factors A pictorial representation of data on world map is a convenient way to summarize large amounts of data. This form of data representation will help any public health official. If the individual state and city information is available, then additional drill down illustration is also possible – this supplementary graphic will allow us to identify the distribution of patients and diseases from different parts of India. More details related to diseases, treatments, additional demographic characteristics could be added to the visual analysis to efficiently recover key information as and when needed. This can form the basis of public health policies framed either by government or by private companies (Figure 3-6, Figure 3-7). The In-Patient and Out-Patient distribution suggests that the route of administration is simple and easily understood by the patients and the caregivers. The diseases may not be life threatening or fatal (Figure 3-8). Are these patients largely coming in for "second opinion"? Or if this data is to be looked at positively, are they getting benefitted and hence are not coming back for consultation beyond the first reported disease? On the other hand, a few patients could be having a lot of faith in Ayurvedic treatment, for them to continue with treatment, they could have found the underlying treatment effective (Figure 3-11). Blood group distribution for many patients is a great source of knowledge. Even though this does not help in day-to-day treatment options, there is undoubted epidemiological value in this presentation. There are obvious mistakes in documenting the blood groups observed via this tabulation – another secondary use of this tabulation is to build data quality related efficiencies (Figure 3-9). While finding data inconsistencies was not a primary objective of this analysis, there is this secondary usage available to the scientific community. The empirical evidence generated by such fundamental data will be very useful for the hospital management, public health officials, treating physicians. This kind of tabulation plays a key role in evidence generation and synthesis. Is there a similar analysis available for another Ayurvedic hospital, or any other private or public hospital in public domain? This can be used to understand the use and misuse of the limited medical sources across the geographies.

121 | P a g e Lesser duration of patient and hospital association may mean either the patients are benefitted by the treatment or are not happy and hence discontinue the treatment. Longer duration of association may mean that the patient is receiving benefit and hence is coming for regular follow-ups for the same condition, or the disease condition could be chronic in nature. These analyses provide a useful macro level representation of data for public health policies for these non-communicable diseases. Data driven approach of optimally utilizing resources suggest strengthening the RMSD disease treating facilities from pharmacy to Vaidyas to patient (Figure 3-12). The low rate of reporting of some of the diseases may explain the natural variations or may reveal inconsistent labelling of the diseases (Figure 3-15). Boxplot representation of age provides the distribution of diseases across age and grouped by gender. It also gives a comparative view of multiple diseases thus providing an information on the disease prevalence in the age category as well as gender (Figure 3-13). 4.4 Studying diagnostics and interventions The ACD and ICD mapping exercise shows that the current hospital data demonstrates all the types of diseases being catered to at the hospital. Large spectrum of diseases getting treated at the hospital. This provides insights into the health seeking behaviour of patients. Additionally, this data can be used by insurance companies and policy makers to strengthen ongoing efforts. If the ICD Code can be included in the data collection, then this correlation analysis can be done easily. The ICD code getting populated using the medical expertise will be more reliable than the current analysis (Figure 3-17, Figure 3-18, Figure 3-19, Figure 3-20). The Prakriti data was not available for all patients across all visits. This points to shortcomings in the data collection methods (Figure 3-21). This analysis shows natural variations in the diseases getting treated. It should be studied by treating physicians to take a deep dive into the data. The process of data collection needs to be looked at and improved. Would it be possible to create an online tool to generate prakriti information? Can this data be collected before a patient goes in for doctor's consultation? Can this data field be made a mandatory field so that there is no missing data generated? If prakriti derivation is a complex process Vikriti or Dosha current dominance must be looked at. Tag the treatment or formulations as -kara and -hara e.g., Pippali is Kaphahara and Pittakara. Co-morbidity analysis can be used to understand the disease clustering. Which disease(s) cause(s) the other disease(s) to manifest, which disease(s) could be precursor to subsequent disease(s). This analysis can be used to validate the existing hypothesis. This analysis could be further enriched for predictive abilities – turning this into a possible disease preventive tool. Some examples like prameha (causing many diseases for both the genders), pandu roga (mainly reported by females with many disease, and relatively low numbers reported by males), sandhigata vata (reported by more females), etc. have shown that meaning of shlokas can be shown in the modern data format. This type of exercises can be carried out with help of ayurvedic experts (Figure 3-22, Figure 3-23, Figure 3-24, Figure 3-25, and, Figure 3-26).

122 | P a g e Treatment and disease analysis at individual patient level: This analysis at individual patient level shows life journey of each patient. This may help in understanding the severity of the disease, co-morbidities and the number of medications prescribed to treat the condition. This can also provide an overview of the practicing physician's style of treatment and may be help draw parallels in treating medical conditions. If a new disease is reported and new treatments are added, then it is understood that this is a part of treatment regimen. But if a treatment is reduced then would it be considered as a part of treatment regimen or would it be considered as a removal due to side effect? This cannot be ascertained without an ayurvedic clinician's opinion. If a new disease is reported and if no new treatment is added, then also it raises some questions? Is it as per treatment protocol or are the existing treatments sufficient to cover this new imbalance? More detailed discussions with ayurvedic clinicians will help in understanding this analysis. This may give rise to better ways of collecting the data (Figure 3-27, Figure 3-28, Figure 3-29). Area graph representation of diseases provides information about 800+ diseases in very short space. Disease patterns are interesting due to following reasons: Diseases vary seasonally, diseases are experienced differently by gender, and it gets shown easily by looking at the distributions. This view is very useful for both operational excellence as well as clinical judgment. The interactive nature of visualization allows for real time subset of diseases. One of the 4 diseases displayed has very few patients compared to other 3 diseases showing different nature of diseases. Another interpretation could be that the disease shown with very low frequency may not be treated by very regularly by Ayurvedic treatments (Figure 3-30). Treatment and disease analysis at summary level: Mosaic plot and cross tabulation analysis: the Patient Report form or Case Report form captures diseases reported on a particular visit along with treatments and services prescribed to a patient. Due to the nature of the CRF page, multiple diseases and treatments are captured on the same visit. This creates many-to-many relationships which makes it difficult to identify the disease treatment relationship. Even though this challenge exists, the data at a summary level provides good view on treatment and disease relationship. The cross tabulations of balaristham and bhasma provide additional evidence of how these analyses can be used to validate facts and / or generate new concepts. Traditionally bhasmas of any kind are prescribed in very limited quantity and same is reflected in observed data. Naming convention and spelling correctness need to be considered while capturing data in future. The clinical utility of this analysis was shown (Figure 3-31, Figure 3-34, Figure 3-35, Figure 3-36). The t-test at 5% significance level shows statistically significant difference between duration of treatment before bhasma treatment and duration of treatment after bhasma treatment. The study was not powered to detect any specific difference in treatment duration, so the p-value and significance should be interpreted cautiously. How should one interpret ~15 days vs. ~11 days of pre and post bhasma treatment, would it be considered clinically meaningful? These discussions with experts will provide more ideas about these plain numeric observations. Interpretation from additional Disease – treatment analysis with pre and post visit window approach are as follows: in circular data representation many green lines means that there is a greater chance of diseases reported by patients, there is a greater chance of a medicine prescribed

123 | P a g e for a disease. If there are very few lines then the combination is clinically not meaningful or if it is meaningful then it is a very rare combination which needs to be studied further (Figure 3-37, Figure 3-38). On a single page there are multiple dimensions of the disease – disease and / or disease – treatment combinations are shown. Distance score-based analysis reveals the following observations: More number of patients with Jaccard distance closer to 1 was seen for the Post reference day 1 period. This could be pointing to similar biological activity caused by a particular disease. This could be a very important finding from this analysis (Figure 3-39). In the medicinal display the similarity scores are lower as compared to that for the disease trajectories. Which implies that most of the prescribed treatments are dis-similar for both the periods. It is observed that around 50% of treatments could form the base of treatment regimen and could be same for the patients. The remaining part of the treatment regimen is driven by individual patient characteristics. The before and after medicine trajectories would show such underlying data. E.g., for M2.0, there are very few patients having distance above 0.5 for both genders (Figure 3-40). This analysis should be executed using other mathematical distances to understand the consistency of results. If the disease classification and treatment tagging in the underlying data is improved then we should be able to see much better results, with lesser confounding effect. Radar plot for multiple diseases is shown next to each other. This is showing massive amounts of information immediately. Differing shapes provide differences reported in the data and an easy way of identifying differences. If there is additional data made available in a structured format, then these parameters could also be added on the radar plot. This radar + trellis combination provides a more powerful tool to visualize large amounts of data on a single page (Figure 3-41). 4.5 Use cases with in-depth illustrations This section outlines a few use cases in-depth and how can the key stake holders make use of the information generated earlier. 4.5.1 Illustration 1: In-depth review of Visit pattern analysis Data / observations: In a competitive world, understanding how patients plan their visits to a hospital and their social demographics can help in running an operation which would make the patient as the focus point of efficient operations. Operational efficiency allows businesses to build an edge and it is applicable in any field. Visit pattern analysis (Figure 3-3) allows insights into the same. This visualization depicts data from the very first of the hospital in 2010 on a calendar. In Feb 2011 there were more than 50 patients visit on a single calendar day. In May of 2011, there were more than 200 patients visit on a single day. Otherwise, the average number of visits was hovering around 30 – 50 patient visits. The underlying data shows that the patient inflow increasing year on year. Out of 2100+ days of data from year 2010 to 2016, there were 158 days on which more than 100 patient visit days were there, which accounts for 7.33% of all the days. Most of these 100+ patient visit days happened on Saturday, showing patients' tendency to visit hospital on a weekend. Figure 3-7: Country-wise Visualization shows that almost 98% of patients are coming from India. Analysis carried out as per Figure 3-25 for V2.0 (Vaatavyadhi) shows slightly a greater number of patients visiting in June, July months

124 | P a g e compared to other months, providing data on seasonal variations. The numbers start tapering down across other months. Figure 3-15 shows this data in terms of Indian rutus. Jwara (J1.0) in ACD is "flu" in Figure 3-25, subset for PrimaryCode = J1.0, should show seasonal variations, as people getting with flu is seasonal in nature. But the overall number of patients reporting Jwara is low (128 females, 140 males) not allowing for these variations to be detected. If Jwara is not reported by large number of patients in the database then that would mean less patients are taking ayurveda treatment to get rid of Jwara. If patients having certain disease conditions are not represented in same proportion in hospital database, considered as a sample, as epidemiologically seen, then the underlying data will not represent disease condition appropriate, this explains relationship between sample and population. Insights: As the sensitive information about how many doctors, how many nurses, how much money was being charged is not used in this analysis, currently there will not be any conclusions drawn w.r.to economic efficiency. As the location of the hospital is almost 4 km inside from the closest main road and the transport options were not regularly available, there were lesser number of patients seen visiting. Visit pattern analysis report can be used on a periodic basis to understand the patient patterns along with any other material hospital management must be using. Even though more than 50+ countries are represented in the database, the number of unique patients and amount of patient data is small to carry out any meaningful analysis. The visit pattern could be repeated by major disease types to get more insights into what type of doctors should be scheduled, what kind of support staff and what type of perishable pharmacy components made available in an optimal manner. Based on this information, the hospital management can have more staff working on weekends to cover patient inflow. Appropriate amount of stock of medicines in pharmacy could also be arranged. If the patient inflow is consistently lower on other days, then should hospital employ lower number of staff on these days, practically this may not be possible, but for operational efficiency this path should be explored. Some of the repairs and upgrades to the hospital infrastructure could be planned for the weekdays avoiding inconvenience to day-to-day functioning. 4.5.2 Illustration 2: In-depth review of summary statistics of number of diseases Data / observations: Data quality is an inherent requirement for any data related exercise. Some insights based on patterns of missing data, outliers and potential scientific and medical inconsistencies can suggest methods to improve the data quality which would be of essence to hospital administration. Analysis covering descriptive summary statistics by number of Diseases by age and gender was carried out (Figure 3-11) and explained in detail in the earlier chapter 3 section 3.3. This table summarizes descriptive statistics for age. The first column categorizes how many diseases were reported per patient going from 1, 2, ..., n diseases. The table shows 23 diseases as the maximum number of diseases reported by a patient. There is a large part of data showing "NA" = unknown number of diseases. Both observations point to potential data issues. Another key finding from the table is to have 12,884 and 14,375 female and male patients reporting only one disease. Other analysis related to duration and hospital visits (Figure 3-14) show that there is large dropout rate

125 | P a g e after one visit. The minimum and maximum age listed goes from 1 to 96 years for females, 2 to 108 years for males. Insights: These observations should be of interest to the hospital management as well as treating doctors. At the beginning of the hospital operations the ACD coding was not implemented and that resulted in missing disease codes, giving rise to the "NA" category. Did the hospital have a patient with 23 different diseases or does this number also point a data issue or duplicate data entries for a patient? Such kind of data points should be checked on an ongoing basis. How do hospital management as well as treating doctors view nearly 70% drop out rate after reporting first disease? This analysis will help in multiple ways. Are there operational challenges causing patients not to come back? Are the treatment mechanisms too difficult to follow for patients? What type of follow-up methods should be built to keep the patient engagement high? If this data is to be looked at positively then should we conclude that nearly 70% patients got benefitted from the treatment and hence did not need come back to the hospital. Are there truly 96- and 108-years old patients coming to hospital or are these possible data issues? Are these ages 9.6 and 10.8 years, OR 96 and 108 months / days? It is encouraging to see patients of all ages coming to the hospital. This topic should be of interest to the data managing team at hospital. If the training material is not clearly specifying an appropriate way of data entry, then this analysis helps in improving material as well as Hospital management system. Missing disease information could be understood by treating clinicians or anyone doing data entry on their behalf for having complete documentation for future use. 4.5.3 Illustration 3: In-depth review of disease table by gender Data / observations: If we can understand the disease and underlying conditions, then the insights generated can give better understanding of science behind the disease. Disease categorization is done by the treating doctor at individual patient visit using ACD classification for documentation. One such example of subset of Prameha has been listed in Figure 3-1. This subset shows disease categories as Prameha, Prameha – Sthula, Prameha – Pidaka, Prameha – Krusha. The numbers for Prameha category are 648 and 849, and for the other categories these frequency counts are in single digits or in 20s. Is this analysis pointing to less than accurate representation of disease? Mosaic / tile plot analysis for Prameha (Figure 3-31 subset for Metabolic: P5.0: Prameha) shows that there are 732 unique treatments prescribed, for Prameha – Sthula there are 114 unique treatments prescribed, for Prameha – Pidaka there are 48 unique treatments prescribed and, for Prameha – Krusha there are 27 unique treatments prescribed. Figure 3-1 has data Aamavata related data represented. Aamavata – Vaataja, Kaphaja and Pittaja categories are tabulated in the analysis. The frequency counts are quite different for each category. 652 females in only Aamavata category followed by 27, 13 and 4 patients in Aamavata – Vaataja, Kaphaja and Pittaja categories. Same is the case for male category. Insights: Should we assume that reporting just "Prameha" was a lot easier than reporting more details about it in the ACD class? The treating doctor must have treated patients as accurately as 126 | P a g e possible but thoughts have not be transferred accurately from "mind" to "paper" for future usage. Any analysis carried out using this data for which treatments were prescribed for which variation of the underlying "Prameha" could be a suboptimal approximation. This observation points to a possible shortcoming in usage of ACD and maybe a hospital level training is needed. Is the current classification accurately expressing the underlying disease, perhaps no. There is a need to have an in-depth discussion with treating doctors to get more insights into this mislabelling. The ayurvedic experts from hospital and if needed experts from other institutions could come together and define guidance documents on how to document a disease. Due to inaccurate representation of disease data, some of these interpretations will not be fully reliable, even though the big picture view may not be adversely impacted (this sentence should be validated by an ayurvedic vaidya). Can carefully labelling of disease at each visit be done to increase the overall quality of data and possible by products from them. This observation is not only for ayurveda but must be happening in any clinical assessment across the world. These nuances are of great importance to basic researchers. We can find similar examples for other diseases in our existing dataset. If disease classification by Vata, Pitta, Kapha by a treating doctor is not easy to be achieved at each visit to produce "close to 100% real picture of patient" then what could be possible solutions proposed? If there are existing validated tools for treating doctors to use then they should be used. 4.5.4 Illustration 4: In-depth review of individual patient disease journey Data / observations: If we can understand the patient journey which includes, diagnoses, treatments, outcomes over time, it will allow the treating doctor to adjust and do better for the patient. The insights can as well give more understanding of the disease and the science behind it. This in-depth analysis shows us how to combine three analyses to understand individual patient journey. Figure 3-5: Patient visit profile – Vertical view, Figure 3-27: Patient Disease and Treatment administration by Study Day, and Figure 3-29: Patient Cumulative Disease and Treatment administration by Visit are combined to show how can a treating doctor use these three tools simultaneously. Subset these 3 analyses for patient MR000774. This data is for a 68- year-old female patient who has been coming to the hospital for close to 3 years (960 days). In this period, she has visited hospital for 29 times. There were 9 different diseases reported and 35 prescribed medicines in the hospital database. For a few of these diseases ACD code has not been reported, giving rise to missing data. Madhumeha and Prameha have been reported as first two diseases reported. Is this a correct representation having both Pramhea and Madhumeha being reported? Diabetes condition is treated by DNil and Nisakathakadi kashayam. Prameha was treated by Nisakathakadi, kathaka kathiradi. Nisakathakadi was prescribed as Choornam / powder and Kashayam, Nisakathakadi treatment was stopped and then Glokustat and Diabecon DS were prescribed. This line-by-line data review provides information about treatment protocol being followed. The analysis carried out in Figure 3-29 helps in understanding first time disease reported and repeat disease representation may be used to understand the co-morbidities. 26 times diseases

127 | P a g e were reported and only 9 were unique showing a smaller number of complications, if we ignore "Not coded terms" then there are only 5 unique reported. 95 times treatments were prescribed out of which 35 were unique. There are 2 vaatavyadhis, arthritic conditions, reported after a few visits. Similar details about the treatment can be found in the medicines section. The analysis carried out in Figure 3-27 provides a tabular and line-by-line listing view at one go, providing one more variation of the same analysis. This report shows reported diseases and medicines as well as updates across timepoints. These three patient journey analyses should be good additions to existing tools to study patient status before a patient visit for any doctor. A subset of Figure 3-5 for patient = MR000774

128 | Page A subset of Figure 3-29 for patient = MR000774 A subset of Figure 3-27 for patient = MR000774 Insights: The spelling was different in different prescriptions which does not change the underlying treatment but creates a perception about extra treatments prescribed to tackle the same disease. If this can be controlled by the coded medicine dictionary and type of formulation, then this will help in data analysis. The arthritic conditions reported by the patient, could be just

129 | Page age-related ailments reported by the patient. This elderly female patient does not have many co- morbidities reported which is a good sign from health point of view.

130 | Page 5 Conclusion The introduction of this thesis outlined the need to undertake such a study, by providing perspectives on medicine, pharmacy, development of hospitals throughout the world, internet era, and the Indian context relating to modern medicine as well as Ayurveda. Insights ts from the thought leaders in the field of Ayurveda are profound and they call for modern methods, new approaches and innovative strategies to be attempted to take Ayurveda science forward. There is a reflection on the type of evidence generated through controlled experiments (RCTs) and life experiments (Observational, Experiential) – some evidence about how these multiple approaches could yield meaningful results. This chapter asks a few questions from diverse points of views and seeks answers – some of these answers are hidden in everyday Ayurvedic clinical practice – which is still largely untapped, prompting the following: how can data analysis of electronically captured data help in advancing our understanding of Ayurveda as a practice and science? Next section of this thesis elaborated on the technical details of a hospital HMIS based database and the associated EMR. We highlighted technical details about the hospital database: how many source tables, how they stored in ~200+ tables, out of which ~20 to 25 tables were used to generate datasets useful for further analyses. Subsequently we displayed flowcharts outlining ~50+ steps to go from Live source to Staging data to transformed data to ~30+ source variables + ~30+ derived variables in 01adsl_met_rmsd dataset: patient level data covering treatment and disease information. This formed the basis of possible operational and clinical analyses going forward. Clinical data understanding section showed how individual observations can be transformed into meaningful patient narratives. This section explained how the usage of operational and clinical part of the data can benefit varied stake holders and emphasized the need to convert “a thought from a doctor’s mind” into an “actionable and consistent data point” in the database for future use. While studying both the structure and the content of the hospital database, it was observed, that standardization of database along with effective curation towards good data quality is needed. It was highlighted that this type of data could provide a gold mine of information which when summarized could lead us to many insights which could potentially increase operational efficiencies and progress the Ayurvedic practice. It laid down the foundation for “understanding demographics and patient characteristics” as a basis. The demographic and patient characteristic analysis provided good insights into different components of the data which can feed more generally into the public health domain. It was observed that the health and healthcare requirements of a population can benefit through disease surveillance and population health insights. It also provided actionable inputs to hospital management on efficient operations, practicing doctors and for research publications. Diagnostics and interventions section showed complex relations between diseases and interventions. Comorbidities as well as combinations of interventions showed the complex clinical decision-making process as followed by Ayurveda physicians. The disease and treatment comorbidity analyses was performed and presented using a variety of plots and heatmaps. These

131 | Page analyses showed how individual observations can be transformed into meaningful insights and data stories. Day-to-day transactions at hospitals involve people from many backgrounds like, hospital administration, patients, doctors, nurses, pharmacists, pathologists, representative from insurance companies, lawyers, etc. These interactions generate a lot of information and are the primary data generators. Same set of people and a few additional professionals are the end users of the data e.g., scientists, statisticians, database developers, etc. This study has provided preliminary insights into various aspects of HMIS based EMR data generated during real time consultation at I-AIM. A variety of analysis and summarization of the hospital data was conducted with a view to derive meaningful outcomes which confirm the Ayurvedic principles. As a final summary of all that has been said previously is represented in the figure below (Figure 5-1). First part of the diagram covers how to define the underlying question, which is followed by defining the hypothesis. Researchers should logically think about clinical context and methodological context. For converting this theoretical thinking into a real-world study, we need to have the necessary IT infrastructure which will enable data related components. Middle of the diagram shows potential stakeholders. This concise representation shows how the HMIS based EMR can shape up Real World Evidence generation.

132 | Page Figure 5-1: Real World Evidence life cycle Is the study specific to a treatment? Is the treatment an approved one? Is it a comparative study? Has treatment been assigned by study protocol? Is data available in existing sources? Causal diagram: Specify causal relations & supporting evidence among treatment, outcome(s), & other variables to control confounding All Data Sources Data protection Different data types IT infrastructure Data quality Information governance Valid sample size Clinical outcome Disease registry Patient registry Patient charts Sensor data, Mobile App Low recall bias Medical practitioner bias Patient reported outcome Real World Data Individual Patient Data (pragmatic trials, cohort trials, observational) Effectiveness in wider population Stakeholders: Regulatory Authorities, Policy Makers, Government and Payers Longitudinal data Co- morbidities and Cost effectiveness Methodology context Low adherence Quality of life outcome Health surveys Preference of other medicine Clinical context Confounding and Population homogeneity Economic outcome Hospital EHRs Real life data, clarity of treatment impact and AEs Un-blinded treatment and Treatment switch Primary / Secondary data Retrospective / Prospective study Big data, large sample size Social media Individual practice More data available on drug and life style interaction Operational challenges Comparable data Patient level data access GDPR and Anonymization Incomplete data Data context

133 | P a g e Due to the above-mentioned outcomes, the following contributions can be possible: 1. Contribution to Public health data creation based on large data at our disposal which is not marred by artificial boundaries imposed on patient disease conditions and treatments prescribed as followed in a designed randomized clinical trial. 2. Make recommendations to the practitioners for standardized way of data collection, analyses and reporting which will support future EMR based RWD studies 3. Understand the hidden wealth of data for Transdisciplinary expansion of thoughts a. Sustainable treatment solutions for diseases readily available b. Thought provoking work to generate new needs through unconventional use of the data c. Expand the use of modern IT solutions like IT infrastructure, electronic health records, cloud, etc. within Ayurvedic area where appropriate – Ayur IT solutions. d. Take advantage of freely available cutting-edge software(s) to create new approaches e. Introduce statistical programming (Ayurdata analyst) as a tool to Ayurvedic area A lot of work carried is out by health authorities, pharmaceutical companies, and nonprofit organizations. Some of these resources could be used as reference to become a world class data generator:

97% **MATCHING BLOCK 13/51** **W**

Clinical Data Interchange Standards Consortium (CDISC) is a global nonprofit charitable organization with administrative offices in Austin, Texas, with

many people contributing across the world. CDISC brings experts together

54% **MATCHING BLOCK 14/51** **W**

to create and advance data standards. This allows for accessibility, interoperability, and reusability of data for competent research that has greater impact on global health [118].

Many of the leading health authorities use the standards developed by the CDISC teams in various parts of drug applications.

78% **MATCHING BLOCK 15/51** **W**

TransCelerate BioPharma Inc. is a nonprofit organization with a mission to collaborate across the global biopharmaceutical research and development community to identify, prioritize, design, and facilitate the implementation of solutions designed to deliver high quality new medicines.

They have many open-source solutions which could be used to improve delivery model [120]. Based on the 21 st century act, 2016, the US FDA has created a regulatory framework for the Real-world data (RWD) and real-world evidence (RWE) which are playing increasing role in the health care decisions [121], [122]. The European Medicines Agency (EMA) has established a center to provide timely and reliable evidence from real world healthcare databases on the use, safety, and effectiveness of medicines for human use, including vaccines, across the European Union (EU). This capability is called the Data Analysis and Real-World Interrogation Network (DARWIN EU®) [123]. These resources provide a lot of material to enhance overall understanding and allows researchers to be compliant with the regulatory requirements. This thesis outlines many tools which can be used by various stakeholders. They are free and easy to use. They allow multi-dimensional display of complex data in a very short amount of
134 | P a g e space. The tools can create evidence for multiple stakeholders. Free softwares like, R, python, Java, tableau and many more have made it possible to harness the power of data in many ways. There is a need to have a profession of a "Statistical programmer" or a "clinical programmer" or an "Ayurdata expert". This role can contribute to database development, data collection, data cleaning aspects, creating analyses ready datasets, and to finally analyses and reporting. This role should have capabilities related to information technology, data management techniques for generating quality data, in addition to knowing basic and advanced statistical and data science concepts. The computational advances in the world of computer science could be leveraged via appropriate software. Theoretical ideas can be converted into practical interactive visualizations and interactive analyses using multiple technologies. These will help convert individual data observations into summaries then into stories thus enabling knowledge generation. Ayurdata expert can contribute to creating documents for medical journalism, medical education, medical marketing of healthcare products, publications, research documents, and regulatory documents by collaborating with other experts (Table 5-1). We believe that this would be a pioneering effort within ayurvedic EMR area. Table 5-1: Different types of documents Medical Reporting Medical Teaching Medical advertising of products Publication • Newspaper & magazine articles • Mostly for public • Written in simple, non- technical language For doctors • Textbooks, • Continued Medical Education programs, • Slide decks, • Online learning material For Patients learning material • Promotional information for healthcare professionals • Product profiles • Brochures • Sales force training • Online learning material • Abstracts • Journal articles, case reports, review articles • Posters & presentations for scientific conferences Research Documents Regulatory documents # • Research proposals • Clinical trial protocols • Investigators' Brochure • Informed Consent Documents • Package Inserts • Patient Information Leaflets • Clinical study reports • Web synopses Aggregate safety reports such as • Periodic Safety Update Reports

135 | P a g e • Study reports • Subject narratives Common Technical Document (CTD) modules such as • nonclinical and clinical overviews & summaries • expert reports • PK, Safety, Efficacy summaries • Periodic Adverse Drug Experience Reports • Annual safety reports • Policy papers #: Presently, some of these documents are not applicable within Ayurvedic area Work carried out for this thesis is typically reflective of work done by a team of people.In a mid to large sized pharmaceutical organization, this type of work is carried out by (1) Clinicians and statisticians design clinical protocol, (2) Database development team creates database and data flow components, (3) Data management team reviews and cleans the data on an ongoing basis, (4) Statistical programming team and statisticians create the necessary analyses, (5) Writing team generates Clinical Study Report / Publication, and last but not the least (6) IT team handles various systems so that the data and information flow is managed appropriately. Much more details about the database and programming done for analyses and visualization are available in the appendices. This is not an end but just a beginning of Ayurdata experts ...

136 | P a g e 6 Appendix 6.1 Approval from the hospital management to carry out the retrospective study 11.1 ANNEXURE 1 –NOC FROM IAIM MEDICAL DIRECTOR. **
************************************* IAIM/2020/NOC/01 Date: 29.05.2020 LETTER OF PERMISSION AND NO OBJECTION CERTIFICATE TO WHOM IT MAY CONCERN This is to grant permission to Mr. Vinay Mahajan to conduct the research study "Review of hospital based Ayurvedic Electronic Health Records to gain real world knowledge - a retrospective data analysis" using I-AIM anonymized Electronic Health Records as per the protocol approved by the IEC. I am assured that Mr. Vinay Mahajan will maintain confidentiality of the data. Further, it is also agreed that any presentation and publication of the results arising from the study will be done after due permission from the authorities of IAIM and TDU. Dr. Prasan Shankar Medical director IAIM Bangalore
** *************************************

137 | P a g e 6.2 Details of analysis dataset Table 6-1: Details of the Reference Dataset "01adsl_met_rmsd" 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 In-Patient visits, Out-Patient visits and Service related visits are combined from source datasets into a dataset, unique visit and date combinations are created. newdt Date of visit to hospital Character version of newdt0

138 | P a g e Variable name Description Derivation vis Visit 1. Based on all the In-Patient visits, Out-Patient 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 visit for each patient. all_vis = max(vis) grouped by each mr_no all_ip All IP visits This variable contains maximum number of visits 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 visits 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

139 | P a g e Variable name Description Derivation 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

140 | P a g e Variable name Description Derivation 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

141 | P a g e Variable name Description Derivation • 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

142 | P a g e Variable name Description Derivation 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

143 | P a g e Variable name Description Derivation • Tablet / Gulika / Vati • ... cat_id Identification of categories 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 Metabolic disease group has 10 diseases (Refer 2.4.1.6.1) RMSD RMSD If a patient has reported any RMSD disease at least once then that patient is given value RMSD = 1, else RMSD =0 RMSD disease group has 97 diseases (Refer 2.4.1.6.1) 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

144 | P a g e Variable name Description Derivation 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.

145 | P a g e Table 6-2: Metabolic and RMSD disease code and de-code Code Description Distype M10.0 Medoroga Metabolic M10.1 Medoroga - Sthula medho roga Metabolic M10.2 Medoroga - Sukshma medho roga Metabolic M2.0 Madhumeha Metabolic P5.0 Prameha Metabolic P5.1 Prameha - Krusha Metabolic P5.2 Prameha - Pidaka Metabolic P5.3 Prameha - Sthula Metabolic P5.4 Prameha - Upadrava Metabolic S16.0 Sthaulya Metabolic A2.0 Aamavaata RMSD A2.1 Aamavaata - Kaphaja RMSD A2.2 Aamavaata - Pittaja RMSD A2.3 Aamavaata - Vaataja RMSD A3.0 Abhighataja Shoola RMSD S10.0 Stambha RMSD S10.1 Stambha - Baahu Stambha RMSD S10.10 Stambha - Prishtha Stambha RMSD S10.12 Stambha - Sandhi Stambha RMSD S10.13 Stambha - Siraa Stambha RMSD S10.14 Stambha - Uru Stambha RMSD S10.4 Stambha - Greevaa Stambha RMSD S10.5 Stambha - Hanu Stambha RMSD S10.6 Stambha - Hridaya Stambha RMSD

146 | P a g e S13.0 Sthaanabhedena Graha RMSD S13.1 Sthaanabhedena Graha - Anga Graha RMSD S13.11 Sthaanabhedena Graha - Katee Graha RMSD S13.13 Sthaanabhedena Graha - Manyaa Graha RMSD S13.14 Sthaanabhedena Graha - Marma Graha RMSD S13.17 Sthaanabhedena Graha - Paada Graha RMSD S13.18 Sthaanabhedena Graha - Paarshva Graha RMSD S13.19 Sthaanabhedena Graha - Prishtha Graha RMSD S13.20 Sthaanabhedena Graha - Shiro Graha RMSD S13.22 Sthaanabhedena Graha - Uro Graha RMSD S13.23 Sthaanabhedena Graha - Vaak Graha RMSD S13.3 Sthaanabhedena Graha - Gala Graha RMSD S13.5 Sthaanabhedena Graha - Hanu Graha RMSD S13.6 Sthaanabhedena Graha - Hrid Graha RMSD S13.7 Sthaanabhedena Graha - Jaanugraha RMSD S13.8 Sthaanabhedena Graha - Janghaa Graha RMSD S14.0 Sthaanabhedena Shoola RMSD S14.11 Sthaanabhedena Shoola - Guda Shoola RMSD S14.13 Sthaanabhedena Shoola - Gulpha Shoola RMSD S14.14 Sthaanabhedena Shoola - Hanu Shoola RMSD S14.15 Sthaanabhedena Shoola - Hasta Shoola RMSD S14.16 Sthaanabhedena Shoola - Hrid Shoola RMSD S14.17 Sthaanabhedena Shoola - Jaanu Shoola RMSD S14.18 Sthaanabhedena Shoola - Janghaa Shoola RMSD S14.19 Sthaanabhedena Shoola - Kantha Shoola RMSD S14.21 Sthaanabhedena Shoola - Katee Shoola RMSD S14.23 Sthaanabhedena Shoola - Kukshi Shoola RMSD

147 | P a g e S14.24 Sthaanabhedena Shoola - Manyaa Shoola RMSD S14.3 Sthaanabhedena Shoola - Amsa Shoola RMSD S14.4 Sthaanabhedena Shoola - Anga Shoola RMSD S14.5 Sthaanabhedena Shoola - Anguli Shoola RMSD S14.6 Sthaanabhedena Shoola - Asthi Shoola RMSD S14.7 Sthaanabhedena Shoola - Baahu Shoola RMSD S15.28 Sthaanabhedena Shoola - Nakha Shoola RMSD S15.31 Sthaanabhedena Shoola - Paada Shoola RMSD S15.32 Sthaanabhedena Shoola - Paarshni Shoola RMSD S15.34 Sthaanabhedena Shoola - Parva Shoola RMSD S15.36 Sthaanabhedena Shoola - Prishtha Shoola RMSD S15.41 Sthaanabhedena Shoola - Sakthi Shoola RMSD S15.42 Sthaanabhedena Shoola - Sandhi Shoola RMSD S15.43 Sthaanabhedena Shoola - Skandha Shoola RMSD S15.44 Sthaanabhedena Shoola - Snaayu Shoola RMSD S15.45 Sthaanabhedena Shoola - Sphik Shoola RMSD S15.46 Sthaanabhedena Shoola - Stanaanta Shoola RMSD S15.47 Sthaanabhedena Shoola - Trika Shoola RMSD S15.48 Sthaanabhedena Shoola - Urah Shoola RMSD S1A.0 Shoola RMSD V1.0 Vaatarakta RMSD V1.1 Vaatarakta - Dvandvaja RMSD V1.2 Vaatarakta - Gambheera RMSD V1.3 Vaatarakta - Kapha Vaataja RMSD V1.4 Vaatarakta - Kaphaadhika Vaatarakta RMSD V1.5 Vaatarakta - Pittaadhika Vaatarakta RMSD V1.7 Vaatarakta - Uttaana RMSD

148 | P a g e V1.8 Vaatarakta - Vaata Kaphaja RMSD V1.9 Vaatarakta - Vaataadhika Vaatarakta RMSD V2.0 Vaatavyaadhi RMSD V2.12 Vaatavyaadhi - Stabdhagaatra RMSD V2.16 Vaatavyaadhi - Baahugata Vaata RMSD V2.23 Vaatavyaadhi - Gridhrasee RMSD V2.30 Vaatavyaadhi - Jaanugata Vaata RMSD V2.31 Vaatavyaadhi - Janghaagata Vaata RMSD V2.36 Vaatavyaadhi - Kateegata Vaata RMSD V2.42 Vaatavyaadhi - Maamsagata Vaata RMSD V2.43 Vaatavyaadhi - Maamsamedogata Vaata RMSD V2.44 Vaatavyaadhi - Majjaagata Vaata RMSD V2.45 Vaatavyaadhi - Majjaasthigata Vaata RMSD V2.46 Vaatavyaadhi - Manyaagata Vaata RMSD V2.47 Vaatavyaadhi - Manyaastambha RMSD V2.48 Vaatavyaadhi - Medogata Vaata RMSD V2.61 Vaatavyaadhi - Prishthagata Vaata RMSD V2.63 Vaatavyaadhi - Sandhigata Vaata RMSD V2.64 Vaatavyaadhi - Sarvaangagata Vaata RMSD V2.65 Vaatavyaadhi - Shaakhaagata Vaata RMSD V2.68 Vaatavyaadhi - Siraagata Vaata RMSD V2.69 Vaatavyaadhi - Siraagraha RMSD V2.70 Vaatavyaadhi - Snaayugata Vaata RMSD V2.72 Vaatavyaadhi - Trikgata Vaata RMSD V2.73 Vaatavyaadhi - Tvaggata Vaata RMSD V2.74 Vaatavyaadhi - Urugata Vaata RMSD V2.75 Vaatavyaadhi - Vaatakantaka RMSD

149 | P a g e V2.77 Vaatavyaadhi - Vishvaachee RMSD V2.9 Vaatavyaadhi - Asthigata Vaata RMSD 6.3 All variables in the source database Table 6-3: All variables in the source database 001_all variables in database.xlsx

150 | P a g e 6.4 Details of analysis Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program Figure 3-1: A snippet of disease table by gender Link 01adsl_met_rmsd 100_adsl.R Figure 3-2: Variable classification by categories Link 03_typesOfassessent 03_typesOfassessment.R Figure 3-3: Visit pattern analysis Link 01adsl_met_rmsd 100_adsl.R Figure 3-4: Patient visit profile – Horizontal view Link 01adsl_met_rmsd 100_adsl.R Figure 3-5: Patient visit profile – Vertical view Link 01adsl_met_rmsd 100_adsl.R

151 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program Figure 3-6: Total Number of Patients Link 04_patient_analysis_tableu_adsl 04_patients_analysis_tableu_adsl.R Figure 3-7: Country-wise Visualization Link 04_patient_analysis_tableu_adsl 04_patients_analysis_tableu_adsl.R Figure 3-8: Age distribution by country, age distribution by gender Link01 Link02 04_patient_analysis_tableu_adsl 04_patients_analysis_tableu_adsl.R Figure 3-9: Blood-group Distribution by gender Link 04_patient_analysis_tableu_adsl 04_patients_analysis_tableu_adsl.R Figure 3-10: Number of Visits, and Visit Types Link 04_patient_analysis_tableu_adsl 04_patients_analysis_tableu_adsl.R

152 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program Figure 3-11: Descriptive summary statistics by number of Diseases by Age and Gender Link 04_patient_analysis_tableu_adsl 04_patients_analysis_tableu_adsl.R Figure 3-12: Data tabulation for patients reporting RMSD and Metabolic diseases Link rmsd_met_primary_diag rmsd_metabolic_all.R Figure 3-13: Disease distribution by age and gender Link rmsd_met_primary_diag rmsd_metabolic_all.R Figure 3-14: Patient visit duration for Link rmsd_met_primary_diag rmsd_metabolic_all.R

153 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program Disease categories by Gender Figure 3-15: Disease distribution by Seasonal Variations and gender Link rmsd_met_primary_diag rmsd_metabolic_all.R Figure 3-16: Pre and Post Disease Classification Analysis Link 085_dis_1st_time_refCal_NodesEdges 085_dis_1st_time_refCal_NodesEdges.R Figure 3-17: ICD classification by Gender Link 060_allopathic_diag 060_allopathic_diag.R Figure 3-18: Age distribution Link 060_allopathic_diag 060_allopathic_diag.R

154 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program by ICD classification and Gender Figure 3-19: Visit distribution by ICD classification and Gender Link 060_allopathic_diag 060_allopathic_diag.R Figure 3-20: Duration distribution by ICD classification and Gender Link 060_allopathic_diag 060_allopathic_diag.R Figure 3-21: Disease classification by Prakriti and Gender Link Disease_by_dosha_type Disease_by_dosha_type.R

155 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program Figure 3-22: Co-morbidity analysis approach 1 example 1: Vaatavyadhi Link prim_diag diagnosis_primary.R Figure 3-23: Co-morbidity analysis approach 1 example 2: Pandu Link prim_diag diagnosis_primary.R Figure 3-24: Co-morbidity analysis approach 1 example 3: Madhumeha Link prim_diag diagnosis_primary.R Figure 3-25: Co-morbidity Link prim_diag_mon diagnosis_primary_month.R

156 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program analysis approach 2 Figure 3-26: Co-morbidity analysis approach 3: collapsible tree view Link 085_dis_count_edges_3rd_byPeriod_A2_bruce 085_dis_counts_bruce_java.R Figure 3-27: Patient Disease and Treatment administration by Study Day Link 080_disease_repeat_prop 080_medicine_dis_repeat_prop.R Figure 3-28: Patient Disease by Study Day and Treatment administration by Study Day Link 080_medicine_dis_all_met_rmsd_prop 080_medicine_dis_repeat_prop.R

157 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program Figure 3-29: Patient Cumulative Disease and Treatment administration by Visit Link 080_medicine_dis_repeat_prop_cumulative 080_medicine_dis_repeat_prop.R Figure 3-30: Area graph representation of diseases Link adiag diagnosis.R Figure 3-31: Mosaic plot: Disease and treatment representation example 1: Prameha Link 305_medicine_duration_by_dis_xyplot 305_medicine_duration_by_dis.R Figure 3-32: Disease and treatment This is a part of 3-31 analysis so no other explicit link or program, use the above link

158 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program example 2: P5.0: Prameha and Oil: Kottamchukkadi Figure 3-33: Disease and treatment example 3: P5.0: Prameha and Vati: Diabecon DS This is a part of 3-31 analysis so no other explicit link or program, use the above link Figure 3-34: Mosaic plot Disease and treatment representation example 4: Treatment: Oil: Kottamchukkadi Link 305_medicine_duration_by_dis_xyplot 305_medicine_duration_by_dis.R Figure 3-35: Cross tabulation of prescribed Link 01adsl_met_rmsd 100_adsl.R

159 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program treatments and disease group by gender Example 1 Figure 3-36: Cross tabulation of prescribed treatments and disease group by gender Example 2 Link 01adsl_met_rmsd 100_adsl.R Figure 3-37: Circular view: Co-occurrences of disease – disease Example 1 Link 085_dis_counts_edges_3rdbyPeriod_circular17 085_dis_counts_edges_3rdbyPeriod_circular17.R Figure 3-38: Circular view: Co-occurrences of disease – Link 085_dis_counts_edges_3rdbyPeriod 085_dis_counts_edges.R

160 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program treatment Example 2 Figure 3-39: Pre and Post distance analysis for disease: M2.0: Madhumeha Link 086time_dis_refcode_max 086time_dis_patterns_combinations_gender_Macro.R Figure 3-40: Pre and Post distance analysis for medicines given for diseases: P5.0, V2.23, V2.63 Link 086time_dis_refcode_max 086_med_patterns_combinations.R Figure 3-41: Radar plot Link 300_radar_plot 300_radar_plot_tableu.R Figure 3-42: Dynamic bubble Link decode_gender.csv file is converted to Json files 06_d3tree_diagram.R

161 | P a g e Figure number and analysis name Link to visualization Dataset(s) used, (if needed then the datasets will be made available) R / SQL / D3js program plot: Example 1: Disease: A6.0: Amavaata

162 | Page 6.5 Programs for the different parts of analysis 6.5.1 Data extraction from SQL database: 01adsl.sql SQL program to get the source data from hospital database, combine major components of data in a logical manner /* SQL version with UNION of data */ /*==*/ /* Execute the code in following manner; */ /* iaim=&lt; \i /cygdrive/d/Hospital_data/ProgresSQL/prgm/100_adsl_sqlpart.sql; */ /*==*/ drop table if exists temp0pat_demog, temp1pat_reg, temp1doc_cons, temp2reg_cons, temp2diag, temp3pat_presc, temp4pat_med, temp20, temp30, temp30_1, temp30_5, temp100ip, temp350, temp100ser, temp100ser2, temp360, base01_op0, base01_op, base01_ip, base01_ser, base_all ; /* Country and state names */ create temp table state as select city.city_id, city.city_name, city.state_id, state.state_name, state.country_id from iaim.city as city, iaim.state_master as state where city.state_id = state.state_id; create temp table cou as select state.*, country.country_name from state, iaim.country_master as country where state.country_id = country.country_id; /* Create demog table */ create temp table temp0pat_demog0 as

163 | Page select distinct mr_no as mrno, patient_gender, patient_city, patient_state, dateofbirth, country, /*oldmrno, remarks,*/ death_date from iaim.patient_details; create temp table temp0pat_demog as select cou.city_name, cou.state_name, cou.country_name, temp0pat_demog0.* from cou, temp0pat_demog0 where cou.city_id = temp0pat_demog0.patient_city and cou.state_id = temp0pat_demog0.patient_state and cou.country_id = temp0pat_demog0.country; create temp table temp1pat_reg as select mr_no, patient_id, visit_type, reg_date, bed_type, dept_name, admitted_dept, main_visit_id from iaim.patient_registration order by mr_no, patient_id; create temp table temp1doc_cons as select distinct mr_no as con_mrno, patient_id as con_patient_id, consultation_id as consult_id, doctor_name, date(visited_date) as visdate from iaim.doctor_consultation order by mr_no, patient_id; create temp table temp2reg_cons as select temp1pat_reg.*, temp1doc_cons.* from temp1pat_reg full join temp1doc_cons on temp1pat_reg.mr_no = temp1doc_cons.con_mrno and temp1pat_reg.patient_id = temp1doc_cons.con_patient_id; /* Create diagnosis table */ create temp table temp2diag as select distinct visit_id, id, description, icd_code, diag_type, doctor_id, diagnosis_datetime::timestamptz::date as diagdate

164 | Page from iaim.mrd_diagnosis; /* Full join temp10 and temp2diag on temp10.patient_id and temp2diag.visit_id */ create temp table temp20 as select temp2reg_cons.*, temp2diag.* from temp2reg_cons full join temp2diag on temp2reg_cons.patient_id = temp2diag.visit_id; /* patient_prescription = consultation_id */ /* A Subset is required for presc_type */ create temp table temp3pat_presc as select patient_presc_id, consultation_id, presc_type, status, date(prescribed_date) as dateonly from iaim.patient_prescription /*where presc_type in ('Medicine') */ order by patient_presc_id, consultation_id; create temp table temp30 as select temp20.*, temp0pat_demog.* from temp20 full join temp0pat_demog on temp20.mr_no = temp0pat_demog.mrno; create temp table temp30_1 as select temp30.*, temp3pat_presc.* from temp30 full join temp3pat_presc on temp30.consult_id = temp3pat_presc.consultation_id; create temp table temp30_5 as select mr_no, patient_id, patient_gender, city_name, state_name, dateofbirth, country_name, death_date,

165 | Page consult_id, description, icd_code, diag_type, diagdate, patient_presc_id from temp30_1; /* patient_medicine_prescriptions = medicine_id */ create temp table temp4pat_med as select medicine_id as cat_id, op_medicine_pres_id, duration, duration_units, mod_time::timestamptz::date as prescdate, frequency, medicine_quantity as quantity, medicine_remarks as remarks from iaim.patient_medicine_prescriptions order by medicine_id, op_medicine_pres_id; /* BASE 1 data for the OP medication */ create temp table base01_op as select temp30_5.*, temp4pat_med.* from temp4pat_med left join temp30_5 on temp30_5.patient_presc_id = temp4pat_med.op_medicine_pres_id order by mr_no, patient_id; /* IP medications */ create temp table temp100ip as select prescription_id as consultation_id, patient_id as ippatient_id, doctor_id, prescription_date::timestamptz::date as prescdate, presc_type,

166 | Page item_id as cat_id, item_name, med_dosage as quantity, med_route, med_form_id, generic_code, remarks, recurrence_daily_id as frequency from iaim.ip_prescription order by patient_id; create temp table base01_ip as select temp30_5.*, temp100ip.* from temp100ip left join temp30_5 on temp30_5.patient_id = temp100ip.ippatient_id; /* Services Create Base01_ser */ create temp table base01_ser as select mr_no, patient_id, service_id as cat_id, presc_date::timestamptz::date as prescdate, conducted, conductedby, conducteddate::timestamptz::date as sercond_date, prescription_id as consultation_id from iaim.services_prescribed; create temp table services as select service_id as medicine_id, service_name as medicine_name from iaim.services;

167 | Page create temp table med as select distinct medicine_name, medicine_id from iaim.medicine_sales_view; \copy temp30_5 TO 'd:/hospital_data/ProgresSQL/source/pat_diag_vis.csv' CSV HEADER DELIMITER ','; \copy base01_ip TO 'd:/hospital_data/ProgresSQL/source/base01_ip.csv' CSV HEADER DELIMITER ','; \copy base01_op TO 'd:/hospital_data/ProgresSQL/source/base01_op.csv' CSV HEADER DELIMITER ','; \copy base01_ser TO 'd:/hospital_data/ProgresSQL/source/base01_ser.csv' CSV HEADER DELIMITER ','; \copy services TO 'd:/hospital_data/ProgresSQL/source/services.csv' CSV HEADER DELIMITER ','; \copy med TO 'd:/hospital_data/ProgresSQL/source/med.csv' CSV HEADER DELIMITER ','; /*==*/ /* End of program */ /*==*/ 6.5.2 Primary dataset creation program: 100_adsl.R This R program generates analysis dataset which is used as a primary dataset ### # Create calculations using base01_ip and base01_op ### #C- Cancelled #U - Condn. Unnecessary

168 | P a g e #Y -Conducted #N - Not Conducted #P - Partially Conducted library(data.table) library(dplyr) library(anytime) # Get all the data IP, OP and Service base01_ip &gt;- fread("D:/Hospital_data/ProgresSQL/source/base01_ip.csv") base01_op &gt;- fread("D:/Hospital_data/ProgresSQL/source/base01_op.csv") base01_ser &gt;- fread("D:/Hospital_data/ProgresSQL/source/base01_ser.csv") pat_diag_vis &gt;- fread("D:/Hospital_data/ProgresSQL/source/pat_diag_vis.csv") # Get the disease category list for MCSD and Metabolic discat &gt;- data.table(fread ("D:/Hospital_data/ProgresSQL/analysis/discategory.csv")) # Get the medication and service list med &gt;- data.table(fread ("D:/Hospital_data/ProgresSQL/source/med.csv")) ser &gt;- data.table(fread ("D:/Hospital_data/ProgresSQL/source/services.csv")) medall &gt;- rbind(med, ser, fill = TRUE) rm(med, ser) ### # Work on the services data # get the date converted to numeric date # get the minimum and maximum date for each visit # get the frequency count for each type of service ###
169 | P a g e base01_ser0 &gt;- base01_ser [,c("mr_no", "patient_id", "prescdate", "sercond_date", "cat_id", "conducted"), with =FALSE] base01_ser0 &gt;- base01_ser0 [, `:=` (newdt = anydate(prescdate), serdt = anydate(sercond_date))] [order(mr_no, newdt, patient_id)] base01_ser01 &gt;- base01_ser0[, .(serstdt = min(newdt), serendt = max(newdt), freq = .N), by = .(mr_no, patient_id, cat_id, conducted)] base01_ser01t &gt;- dcast(data = base01_ser01, mr_no + patient_id + cat_id + serstdt + serendt ~ conducted, value.var = c("freq"), fill = "") base01_ser01t &gt;- merge (x = base01_ser01t, y = medall, by.x = "cat_id", by.y = "medicine_id", all.x = TRUE) base01_ser01t &gt;- base01_ser01t [order(mr_no, serstdt, patient_id)] base01_ser01t &gt;- base01_ser01t [, newdt := serstdt] l = list(IP = base01_ip, OP = base01_op) base01_all &gt;- rbindlist(l, idcol = "Type", use.names = TRUE, fill = TRUE) base01_all &gt;- base01_all [, `:=` (newdt = anydate(prescdate))] [order(mr_no, newdt, patient_id)] ## # create visit numbers and total number of visits # Individual visits: merge the data on base01_all
170 | P a g e # IP visits # OP visits # Total number of visits IP + OP ## vis &gt;- unique (rbind(base01_all [, c("mr_no", "patient_id", "newdt"), with =FALSE], base01_ser01t[, c("mr_no", "patient_id", "newdt"), with =FALSE], fill=TRUE)) vis &gt;- vis [, Type := substr(patient_id, 1, 2)] [order (mr_no, newdt, patient_id)] vis &gt;- vis [, `:=` (vis =1:.N, all_vis = max(seq_len(.N))), by = .(mr_no)] vis02 &gt;- vis [, .(vistype =.N), by = .(mr_no, Type, all_vis)] vis02t &gt;- dcast(data = vis02, mr_no +all_vis ~ paste("all_", tolower(Type), sep =""), value.var =c("vistype"), fill="") vis03 &gt;- merge (vis [, -c("all_vis")], vis02t, by = "mr_no") ### # Start and end date for each type OP and IP # Start and end date for overall visit dates ### base01_all01 &gt;- vis[, .(stdt = min(newdt), endt = max(newdt), dur = max(newdt) - min(newdt) + 1), by = .(mr_no, Type)] base01_all01t &gt;- dcast(data = base01_all01, mr_no ~ Type, value.var = c("stdt", "endt", "dur"), fill = "") #############################
171 | P a g e # Start for the overall study ############################# base01_all020 &gt;- vis[, .(cstdt = min(newdt), cendt = max(newdt), cdur = max(newdt) - min(newdt) + 1), by = .(mr_no)] ### # Create one large dataset with all the dates ### dates_dur &gt;- merge (x = base01_all020, y = base01_all01t, by = c("mr_no"), all.x = TRUE) vis03dates_dur &gt;- merge (x = dates_dur, y = vis03, by = c("mr_no"), all.x = TRUE) vis03dates_dur &gt;- vis03dates_dur [, studyday := newdt - cstdt + 1] ## # Merge the Medication information # Merge the visit information and day calculations # Merge this information on SERVICEs data as well ### base01_all01 &gt;- merge (x = base01_all, y = medall, by.x = "cat_id", by.y = "medicine_id", all.x = TRUE)
172 | P a g e base01_all011 &gt;- merge (x = base01_all01, y = vis03dates_dur [, -c("Type")], by = c("mr_no", "patient_id", "newdt"), all.x = TRUE) ## # This should be moved after the VIS calculations # Add the patient_info ## base01_ser02t &gt;- merge (x = base01_ser01t, y = vis03dates_dur, by = c("mr_no", "patient_id", "newdt"), all.x = TRUE) base01_ser02t &gt;- merge (x = base01_ser02t, y = pat_diag_vis, by = c("mr_no", "patient_id"), all.x = TRUE) all &gt;- rbind(base01_all011, base01_ser02t, fill =TRUE, use.names = TRUE) all02 &gt;- all [, -c("ippatient_id", "consult_id", "consultation_id" ,"patient_presc_id", "med_form_id", "op_medicine_pres_id", "doctor_id", "diagdate", "prescdate")] [order(mr_no, studyday, patient_id, newdt, vis, cat_id)] ## # Calculations for # Get the disease category list for RMSD and Metabolic ## tmpall &gt;- merge (x = discat[, -c("Description"), with =FALSE], y = all02, by.x = "Code", by.y = "icd_code")
173 | P a g e # create a dummy variable tmpall &gt;- tmpall[,val:=1] subset2 &gt;- tmpall [, c("mr_no", "distype", "val"), with =FALSE] subset2 &gt;- unique(subset2) subset3 &gt;- dcast (data = subset2, fill =0, mr_no ~ distype, value.var="val") # Create an indicator variable to determine # Both Metabolic and RMSD = 99 # Only Metabolic = 1 # Only RMSD = 2 subset3 &gt;- subset3 [Metabolic == 1 & RMSD == 1, combine := "Metabolic and RMSD"] subset3 &gt;- subset3 [Metabolic == 1 & RMSD == 0, combine := "Metabolic"] subset3 &gt;- subset3 [Metabolic == 0 & RMSD == 1, combine := "RMSD"]

61% **MATCHING BLOCK 16/51** W

all_met_rmsd &gt;- merge (x = subset3, y = all02, by = "mr_no", all.x = TRUE) all_met_rmsd &gt;-

merge (x = discat[, -c("Description" , "date"), with =FALSE], y =

34% MATCHING BLOCK 23/51 W

all_met_rmsd, all = TRUE, by.x = "Code", by.y = "icd_code") all_met_rmsd$distype[is.na(all_met_rmsd$distype)] &gt;- "OTHER" all_met_rmsd &gt;- all_met_rmsd [order(mr_no, studyday,

patient_id, newdt, vis, cat_id)]
174 | P a g e # Calculation of first RMSD or Metabolic disease date minday &gt;- all_met_rmsd[distype != "OTHER", .(minday = min(studyday)), by =.(mr_no, distype)] mindayt &gt;- dcast (data = minday, mr_no ~ paste("minday", distype, sep=""), value.var="minday")

38% MATCHING BLOCK 17/51 W

all_met_rmsd &gt;- merge (all_met_rmsd, mindayt, by = "mr_no") # Calculate the age variable for non-missing dates all_met_rmsd &gt;- all_met_rmsd [, `:=`(age = ifelse (!

is.na(anydate(dateofbirth)) , round((anydate(newdt) - anydate(dateofbirth) + 1)/365.25, digits = 0), NA), newdt0 = anydate(newdt)),] # Add Indian rutus as new variables # https://www.drikpanchang.com/seasons/season-tropical-timings.html?geoname-id=1277333&year=2010 rutus &gt;- fread("D:/Hospital_data/ProgresSQL/analysis/rutus.csv") rutus &gt;- rutus [, `:=`(startdt = as.POSIXct(startdate, format="%d-%m-%Y"), enddt = as.POSIXct(enddate, format="%d-%m-%Y"))] rutus02 &gt;- rutus[, list(season = season, year = year, newdt0 = anydate(seq(startdt, enddt, by = "day"))), by = 1:nrow(rutus)] all_met_rmsd &gt;- merge (x = all_met_rmsd, y = rutus02 [, c("newdt0", "year", "season")], by = c("newdt0"), all.x = TRUE) rm (base01_ip, base01_op, base01_ser, l)
175 | P a g e all_met_rmsd &gt;- all_met_rmsd [, `:=` (baseage = min(age)), by =.(mr_no)] ## # Update the data by re-coded Medicine names ## lookup_medicine &gt;- fread("D:/Hospital_data/ProgresSQL/analysis/lookup_medicine.txt", sep="|") all_met_rmsd &gt;- merge(x = all_met_rmsd, y = lookup_medicine, all.x = TRUE, by.x = c("medicine_name"), by.y = c("medicine_name")) fwrite(all, "D:/Hospital_data/ProgresSQL/analysis/01adsl.csv") fwrite(

58% MATCHING BLOCK 18/51 W

all_met_rmsd, "D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.csv") saveRDS (all_met_rmsd, "D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") dis_rutu &gt;- all_met_rmsd [Code != "", .(

cnt = uniqueN(mr_no)), by = .(season, Code, description)] [order(season, -cnt, Code)] dis_rutu_yr &gt;- all_met_rmsd [Code != "", .(cnt = uniqueN(mr_no)), by = .(year, season, Code, description)][order(year, season, -cnt, Code)] dis_rutu_yr02 &gt;- dcast(dis_rutu_yr, season + Code + description ~ paste("yr", year, sep=""), value.var = c("cnt"), fill=" ") fwrite(dis_rutu, "D:/Hospital_data/ProgresSQL/analysis/dis_rutu.csv") fwrite(dis_rutu_yr02, "D:/Hospital_data/ProgresSQL/analysis/dis_rutu_yr.csv") /*==*/ /* End of program */
176 | P a g e /*==*/ 6.5.3 R and SQL programs for other datasets from SQL database: 02other_data.R This program creates datasets for various CRFs which are not covered in the first program. This program creates 1 csv file per CRF. SQL code is added at the top of the file and then followed by R code #== drop table if exists patient_section_details, patient_section_values; drop table if exists patient_section_details, patient_section_values, base10_other0; create table patient_section_details as select mr_no, patient_id, section_id, section_detail_id, section_item_id, item_type from iaim.patient_section_details order by mr_no, patient_id; create table patient_section_values as select section_detail_id, field_id, option_id, option_remarks from iaim.patient_section_values order by section_detail_id; ## Not working /* create table base10_other0 as select patient_section_details.*, patient_section_values.field_id, patient_section_values.option_id, patient_section_values.option_remarks from patient_section_details full join patient_section_values on patient_section_details.section_detail_id = patient_section_values.section_detail_id and patient_section_details.section_item_id = patient_section_values.field_id and patient_section_details.section_id = patient_section_values.option_id;
177 | P a g e */ ## Working: create table base10_other11 as select a.*, b.field_id, b.option_id, b.option_remarks from patient_section_details as a, patient_section_values as b where a.section_detail_id=b.section_detail_id ; \copy base10_other11 TO 'd:/hospital_data/ProgresSQL/data_chk/base10_other11.csv' CSV HEADER DELIMITER ','; \copy iaim.section_master TO 'd:/hospital_data/ProgresSQL/data_chk/section_master.csv' CSV HEADER DELIMITER ','; \copy iaim.section_field_options TO 'd:/hospital_data/ProgresSQL/data_chk/section_field_options.csv' CSV HEADER DELIMITER ','; \copy iaim.section_field_desc TO 'd:/hospital_data/ProgresSQL/data_chk/section_field_desc.csv' CSV HEADER DELIMITER ','; \copy iaim.patient_consultation_field_values TO 'd:/hospital_data/ProgresSQL/data_chk/patient_consultation_field_values.csv' CSV HEADER DELIMITER ','; ################################## Check the _orig dataset create table patient_section_details_orig as select mr_no, patient_id, section_id, section_detail_id, section_item_id, item_type from iaim.patient_section_details order by mr_no, patient_id; ## Working: create table base10_other11_orig as select a.*,

178 | P a g e b.field_id, b.option_id, b.option_remarks from patient_section_details_orig as a, patient_section_values as b where a.section_detail_id=b.section_detail_id ; \copy base10_other11_orig TO 'd:/hospital_data/ProgresSQL/data_chk/base10_other11_orig.csv' CSV HEADER DELIMITER ',';
#== library(data.table) library(stringi) library(stringr) # Read the data base01_other &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/base10_other11.csv") base01_other02 &gt;- base01_other [nchar(option_remarks)&lt; 0] # CRF names section_master &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/section_master.csv") section_master &gt;- section_master[, c("section_id", "section_title"), with = FALSE] base01_other02 &gt;- merge (x = base01_other02, y = section_master, by = "section_id", all.x = TRUE) # variable names section_field_options &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/section_field_options.csv") base01_other022 &gt;- merge (x = base01_other02 [option_id &lt;= 0], y = section_field_options ,
179 | P a g e by = c("option_id", "field_id"), #by = c("section_id", "field_id"), all.x = TRUE) # Keep Unique records #base01_other022 &gt;- unique (base01_other02 [, c("mr_no", "patient_id", "section_id", "option_id", "field_id", "option_remarks", "section_title", "display_order", "option_value"), with =FALSE]) base01_other022 &gt;- unique (base01_other022 [, c("mr_no", "patient_id", "section_id", "field_id", "option_remarks", "section_title", "display_order", "option_value"), with =FALSE]) # Sort the data by patient and visits base01_other022 &gt;- base01_other022 [order(mr_no, patient_id, section_id, field_id, display_order)] section_field_desc &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/section_field_desc.csv") section_field_desc &gt;- section_field_desc[, c("section_id", "field_id", "display_order", "field_name", "no_of_lines"), with = FALSE] base01_other044 &gt;- merge (x = base01_other02 [option_id &gt; 0], y = section_field_desc, by = c("section_id", "field_id"), all.x = TRUE) base01_other044 &gt;- unique (base01_other044 [, c("mr_no", "patient_id", "section_id", "option_id", "field_id", "option_remarks", "section_title", "display_order", "field_name", "no_of_lines"), with =FALSE]) # Sort the data by patient and visits base01_other044 &gt;- base01_other044 [order(mr_no, patient_id, section_id, field_id, display_order)] setnames(base01_other044, "field_name", "option_value")
180 | P a g e base01_all &gt;- rbind(base01_other022, base01_other044, fill =TRUE)
Need to consolidate variable names and combine # base01_other044 # base01_other022 # Create a counter variable for transposing
Create a counter variable for transpose base01_other030 &gt;- base01_all [, subvis := 1:.N, by = .(mr_no, patient_id, section_id, field_id, option_value)] #fwrite(base01_other030, "D:/Hospital_data/ProgresSQL/analysis/complete_other_data.csv")
################################## # Subset for Metabolic

77% MATCHING BLOCK 19/51 W

RMSD data ################################## all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") subpat &gt;- unique(all_met_rmsd [,

c("mr_no", "Metabolic", "RMSD", "combine", "all_vis", "city_name", "state_name", "dateofbirth", "country_name", "death_date")]) vispat &gt;- unique(all_met_rmsd [, c("mr_no", "studyday", "patient_id", "newdt", "vis", "Type", "Code", "distype", "description", "all_ip", "all_op")]) # Only keep Metabolic and RMSD patients base01_met_rmsd &gt;- merge (x = base01_other030,
181 | P a g e y = subpat [,c("mr_no")], by = c("mr_no"), all.y = TRUE) sub &gt;- unique(base01_met_rmsd [, c("section_id", "section_title", "field_id", "display_order", "option_value")]) [order(section_id, field_id, display_order, option_value)] sub &gt;- sub [, varnum:=seq_len(.N), by =.(section_id)] sub &gt;- sub [, trnvar := paste("sec", str_pad(section_id, 3, side = "left", pad = 0), "_var", str_pad(varnum, 3, side = "left", pad = 0), "_", option_value, sep="")] base01_met_rmsd &gt;- merge (x = base01_met_rmsd, y = sub, by = c("section_id", "section_title", "field_id", "display_order", "option_value"), all.x = TRUE) # Transpose the data as per CRF pages base01_met_rmsd_trn &gt;- dcast(data = base01_met_rmsd, mr_no + patient_id + subvis ~ trnvar, value.var = c("option_remarks")) # Add visit information and disease information: base01_met_rmsd_trn &gt;- merge (x = base01_met_rmsd_trn, y = vispat, by = c("mr_no", "patient_id"), all.x = TRUE) # Add patient demog + visit + duration information base01_met_rmsd_trn &gt;- merge (x = base01_met_rmsd_trn, y = subpat, by = c("mr_no"),
182 | P a g e all.x = TRUE) # Keep variables by section df = base01_met_rmsd_trn[,(names(base01_met_rmsd_trn) %in% c("mr_no", "patient_id", "Metabolic", "RMSD", "combine", "subvis", "city_name", "state_name", "dateofbirth", "country_name", "death_date", "Type", "Code", "distype", "description", "studyday", "patient_id", "newdt", "vis", "all_vis", "all_ip", "all_op") | grepl("^sec004",names(base01_met_rmsd_trn))), with =FALSE] sections &gt;- unique(sub$section_id) for (ii in sections){ jj &gt;- str_pad(ii, 3, side = "left", pad = 0) kk &gt;- paste0("^sec", jj, sep="") print(jj) print(kk) fwrite(file = paste0("D:/Hospital_data/ProgresSQL/analysis/sec", jj, ".csv"), x = base01_met_rmsd_trn [,(names(base01_met_rmsd_trn) %in% c("mr_no", "patient_id", "Metabolic", "RMSD", "combine", "subvis","city_name", "state_name", "dateofbirth", "country_name", "death_date", "Type", "Code", "distype", "description", "studyday", "patient_id", "newdt", "vis", "all_vis", "all_ip", "all_op") | grepl(kk,names(base01_met_rmsd_trn))), with =FALSE]
183 | P a g e) } # dtable &gt;- df[, fwrite(.SD, paste0("./output/"), Name, ".csv"), by = Name]
##
End of program
##
#### 6.5.4 Analysis program for Figure 3-1 Refer to the 100_adsl.R program 6.5.5 Analysis program for Figure 3-2 R program: 03_typesOfassessment.R library(Hmisc) library(data.table) library(stringi) library(stringr) library(sqldf)

100% **MATCHING BLOCK 20/51** W

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") ################################### #

Subset for Metabolic RMSD data
184 | P a g e ###################################

83% **MATCHING BLOCK 21/51** W

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") subpat &gt;- unique(all_met_rmsd [,

c("mr_no", "Metabolic", "RMSD", "combine", "all_vis", "patient_gender", "baseage", "city_name", "state_name", "dateofbirth", "country_name", "death_date")]) vispat &gt;- unique(all_met_rmsd [, c("mr_no", "studyday", "patient_id", "newdt", "vis", "Type", "Code", "distype", "description", "all_ip", "all_op")]) ### # Get records per visit for Treatment / Procedure # Med start date, end date non missing and # name non missing ### med_ip &gt;- unique(na.omit(all_met_rmsd, cols = c("stdt_IP")))
185 | P a g e med_ip &gt;- unique(med_ip [Type == "IP", c("mr_no", "vis", "studyday", "Metabolic", "RMSD", "combine", "all_vis", "patient_gender", "baseage"),]) med_ip &gt;- med_ip [, cat := "Treatment - IP"] med_op &gt;- unique(na.omit(all_met_rmsd, cols = c("stdt_OP"))) med_op &gt;- unique(med_op [Type == "OP", c("mr_no", "vis", "studyday", "Metabolic", "RMSD", "combine", "all_vis", "patient_gender", "baseage"),]) med_op &gt;- med_op [, cat := "Treatment - OP"] ser &gt;- unique(na.omit(all_met_rmsd, cols = c("serstdt"))) ser &gt;- unique(ser [, c("mr_no", "vis", "studyday", "Metabolic", "RMSD", "combine", "all_vis", "patient_gender", "baseage"),]) ser &gt;- ser [, cat := "Treatment - Procedure"] dis &gt;- unique(all_met_rmsd [Code != " " | description != " ", c("mr_no", "vis", "studyday", "Metabolic", "RMSD", "combine", "all_vis", "patient_gender", "baseage"),]) dis &gt;- dis [, cat := "Disease"] catall &gt;- rbind(med_ip, med_op, ser, dis, fill = TRUE) # Read the data
186 | P a g e base01_other &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/base10_other11.csv") base01_other02 &gt;- base01_other [nchar(option_remarks)&lt; 0] # CRF names section_master &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/section_master.csv") section_master &gt;- section_master[, c("section_id", "section_title"), with = FALSE] base01_other02 &gt;- merge (x = base01_other02, y = section_master, by = "section_id", all.x = TRUE) # variable names section_field_options &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/section_field_options.csv") base01_other022 &gt;- merge (x = base01_other02 [option_id &lt;= 0], y = section_field_options , by = c("option_id", "field_id"), #by = c("section_id", "field_id"), all.x = TRUE)
187 | P a g e # Keep Unique records base01_other022 &gt;- unique (base01_other022 [, c("mr_no", "patient_id", "section_id", "field_id", "option_remarks", "section_title", "display_order", "option_value"), with =FALSE]) # Sort the data by patient and visits base01_other022 &gt;- base01_other022 [order(mr_no, patient_id, section_id, field_id, display_order)] section_field_desc &gt;- fread("D:/Hospital_data/ProgresSQL/data_chk/section_field_desc.csv") section_field_desc &gt;- section_field_desc[, c("section_id", "field_id", "display_order", "field_name", "no_of_lines"), with = FALSE] base01_other044 &gt;- merge (x = base01_other02 [option_id &gt; 0], y = section_field_desc, by = c("section_id", "field_id"), all.x = TRUE) base01_other044 &gt;- unique (base01_other044 [, c("mr_no", "patient_id", "section_id", "option_id", "field_id", "option_remarks", "section_title", "display_order", "field_name", "no_of_lines"), with =FALSE])
188 | P a g e # Sort the data by patient and visits base01_other044 &gt;- base01_other044 [order(mr_no, patient_id, section_id, field_id, display_order)] setnames(base01_other044, "field_name", "option_value") base01_all &gt;- rbind(base01_other022, base01_other044, fill =TRUE) ## # Need to consolidate variable names and combine # base01_other044 # base01_other022 # Create a counter variable for transposing ## # Create a counter variable for transpose base01_other030 &gt;- base01_all [, subvis := 1:.N, by = .(mr_no, patient_id, section_id, field_id, option_value)] # Only keep Metabolic and RMSD patients base01_met_rmsd &gt;- merge (x = base01_other030, y = subpat [,c("mr_no")],
189 | P a g e by = c("mr_no"), all.y = TRUE) sub &gt;- unique(base01_met_rmsd [, c("section_id", "section_title", "field_id", "display_order", "option_value")]) [order(section_id, field_id, display_order, option_value)] sub &gt;- sub [, varnum:=seq_len(.N), by =.(section_id)] sub &gt;- sub [, trnvar := paste("sec", str_pad(section_id, 3, side = "left", pad = 0), "_var", str_pad(varnum, 3, side = "left", pad = 0), "_", option_value, sep="")] base01_met_rmsd &gt;- merge (x = base01_met_rmsd, y = sub, by = c("section_id", "section_title", "field_id", "display_order", "option_value"), all.x = TRUE) base01_met_rmsd02 &gt;- unique(base01_met_rmsd [option_remarks != " ", c("mr_no", "option_value", "patient_id", "trnvar")]) # Add visit information and disease information:
190 | P a g e base01_met_rmsd02 &gt;- merge (x = base01_met_rmsd02, y = unique(vispat [, c("mr_no", "patient_id", "vis", "studyday")]), by = c("mr_no", "patient_id"), all.x = TRUE, allow.cartesian = TRUE) # Add patient demog + visit + duration information base01_

58% **MATCHING BLOCK 22/51** W

met_rmsd02 &gt;- merge (x = base01_met_rmsd02, y = subpat, by = c("mr_no"), all.x = TRUE) base01_met_rmsd02 &gt;- unique(base01_met_rmsd02) #

```
variable names types &gt;- fread("D:/Hospital_data/ProgresSQL/analysis/lookup_03types.csv") base01_met_rmsd02 &gt;- merge (x =
base01_met_rmsd02, y = types [, c("trnvar", "cat")],
191 | P a g e by = c("trnvar"), all.x = TRUE) base01_met_rmsd03 &gt;- unique( base01_met_rmsd02 [ , -c("option_value", "patient_id",
"trnvar" )]) catall02 &gt;- rbind(catall, base01_met_rmsd03, fill = TRUE) catall02 &gt;- catall02 [, val :=1] fwrite(catall02,
"D:/Hospital_data/ProgresSQL/analysis/03_typesOfassessent.csv")
####################################################################################
#### # End of program
####################################################################################
#### 6.5.6 Analysis program for Figure 3-3 Refer to the 100_adsl.R program 6.5.7 Analysis program for Figure 3-4 Refer to the
100_adsl.R program 6.5.8 Analysis program for Figure 3-5 Refer to the 100_adsl.R program 6.5.9 Analysis program for Figure 3-6 R
program: 04_patients_analysis_tableu_adsl.R library(zoo)
192 | P a g e setwd("C:\\Users\\mahajvi1\\Desktop\\Desktop - copied on 18August2014\\Backup\\Ayur guidelines\\FRLHT\\01
Hospital data 30July2016\\") # Create Vital sign data from 31st July 2016 version of the data vitals &gt;- read.csv("Vitals.csv") vitals
&gt;- data.frame ( cbind(vitals, data="vital", type = substr(vitals$Patient.Id, 1, 2) ) ) vitals &gt;- cbind ( data.frame ( subset (vitals, select
=c(MR.No., type, data, Age, Gender, City, Country, Blood.Group, First.Visit.Date) )), visdate = vitals$Vital.Date) Diagnosis &gt;-
read.csv("Diagnosis.csv") Diagnosis &gt;- data.frame ( cbind(Diagnosis, data="diag", type = substr(Diagnosis$Patient.Id, 1, 2) ) )
Diagnosis &gt;- cbind ( data.frame ( subset (Diagnosis, select =c(MR.No., type, data, Age, Gender, City, Country, Blood.Group,
First.Visit.Date, Code) )), visdate = Diagnosis$Admission.Date) Doctor_consultation &gt;- read.csv("Doctor_consultation.csv")
Doctor_consultation &gt;- data.frame ( cbind(Doctor_consultation, data="Doccon", type = substr(Doctor_consultation$Patient.Id, 1,
2) ) )
193 | P a g e Doctor_consultation &gt;- cbind ( data.frame ( subset (Doctor_consultation, select =c(MR.No., type, data, Age, Gender,
City, Country, Blood.Group, First.Visit.Date) )), visdate = Doctor_consultation$Cons..Aptmt.Date) Lab &gt;- read.csv("Lab.csv") Lab &gt;-
data.frame ( cbind(Lab, data="Lab", type = substr(Lab$Patient.Id, 1, 2) ) ) Lab &gt;- cbind ( data.frame ( subset (Lab, select =c(MR.No.,
type, data, Age, Gender, City, Country, Blood.Group, First.Visit.Date) )), visdate = Lab$Conducted.Date) # Combine all the data into 1
all0 &gt;- rbind (vitals, subset(Diagnosis, select =-c(Code)), Doctor_consultation, Lab) all01 &gt;- data.frame (unique ( subset(all0,
select =c(MR.No., type) ) )) tmp01op &gt;- data.frame( unique ( subset(all01, select =c(MR.No.), type == "OP"))) tmp01ip &gt;-
data.frame( unique ( subset(all01, select =c(MR.No.), type == "IP"))) common &gt;- data.frame ( MR.No.= intersect(tmp01op$MR.No.,
tmp01ip$MR.No.) ) onlyip &gt;- data.frame ( merge (x = tmp01op, y =tmp01ip, by ="MR.No.", all.y=TRUE) ) onlyop &gt;- data.frame (
merge (x = tmp01ip, y =tmp01op, by ="MR.No.", all.y=TRUE) ) demog02 &gt;- data.frame (unique ( subset (all0, select=-c(data, visdate,
First.Visit.Date) ) ) )
194 | P a g e common_demog &gt;- cbind ( data.frame ( merge (x = demog02, y =common, by ="MR.No.", all.y=TRUE) ),
grp="Common") onlyip_demog &gt;- cbind ( data.frame ( merge (x = demog02, y =onlyip, by ="MR.No.", all.y=TRUE) ), grp="OnlyIP")
onlyop_demog &gt;- cbind ( data.frame ( merge (x = demog02, y =onlyop, by ="MR.No.", all.y=TRUE) ), grp="OnlyOP") all_age02
&gt;- rbind ( common_demog, onlyip_demog, onlyop_demog) all_age02 &gt;- cbind (all_age02, age2= as.numeric(all_age02$Age) )
rm(list=ls(pattern="vitals")) #rm(list=ls(pattern="Diagnosis")) rm(list=ls(pattern="Doctor_")) rm(list=ls(pattern="Lab")) tmp02 &gt;-
data.frame (cbind (all0, adm = gsub("-", "/", all0$visdate), fdt = gsub("-", "/", all0$First.Visit.Date) )) tmp02 &gt;- data.frame (cbind
(tmp02,
195 | P a g e adm2 = as.POSIXct(tmp02$adm, format="%d/%m/%Y"), fdt2 = as.POSIXct(tmp02$fdt, format="%d/%m/%Y") )) tmp03
&gt;- data.frame (cbind (tmp02, visday = difftime(tmp02$adm2, tmp02$fdt2, units="days") + 1, mondt= as.yearmon(tmp02$adm2,
format="%Y-%m" ) )) tmp04 &gt;- data.frame ( unique ( subset (tmp03, select =c (MR.No., First.Visit.Date, visdate, adm, fdt, data) )) ) #
Count the number of visits per patient tmp05 &gt;- data.frame (table (tmp04$MR.No.)) tmp05 &gt;- data.frame (cbind (tmp05, MR.No.
= tmp05$Var1, Novisits = tmp05$Freq)) # Count the number of diagnosis tmp06 &gt;- droplevels (unique (data.frame ( subset
(Diagnosis, Code !="", select = c(MR.No., Code ) ) )) ) tmp07 &gt;- subset (data.frame (table (tmp06$MR.No.)), Freq &lt; 0) tmp07 &gt;-
data.frame (cbind (tmp07, MR.No. = tmp07$Var1, Nodiseases = tmp07$Freq))
196 | P a g e # Put all data into 1 with 1 record per patient final &gt;- data.frame (merge (x= all_age02, y =tmp05, by = "MR.No.",
all=TRUE) ) final02 &gt;- data.frame (merge (x= final, y =tmp07, by = "MR.No.", all=TRUE) ) final03 &gt;- data.frame ( subset (final02,
select =-c ( type, age2, Freq.x, Freq.y, Var1.x, Var1.y) ) ) write.csv(final03, file="04_patient_analysis_tableu_adsl.csv") # Create a table by
one record per patient per date tmp08 &gt;- data.frame (unique ( subset (tmp03, select=c(MR.No., fdt2) ) ) ) tmp09 &gt;- subset
(data.frame (table (tmp08$fdt2)), Freq &lt; 0) tmp09 &gt;- data.frame (cbind (tmp09, Newpatients = tmp09$Freq)) tmp09 &gt;-
data.frame ( subset (tmp09, select =-c (Freq) ) ) # Create 1 record per date for number of patients on a day tmp10 &gt;- data.frame
(unique ( subset (tmp03, select=c(MR.No., adm2) ) ) ) tmp11 &gt;- subset (data.frame (table (tmp10$adm2)), Freq &lt; 0) tmp11 &gt;-
data.frame (cbind (tmp11, Visitpatients = tmp11$Freq)) tmp11 &gt;- data.frame ( subset (tmp11, select =-c (Freq) ) )
197 | P a g e # Create 1 record per date per patient type for first visit date tmp12 &gt;- data.frame (unique ( subset (tmp03,
select=c(MR.No., fdt2, type) ) ) ) tmp13 &gt;- subset (data.frame (table (tmp12$fdt2, tmp12$type)), Freq &lt; 0) tmp13_tran &gt;-
reshape (tmp13, direction="wide", idvar= c("Var1"), timevar="Var2" ) tmp13_tran &gt;- data.frame (cbind (tmp13_tran, newIP =
tmp13_tran$Freq.IP, newOP = tmp13_tran$Freq.OP)) tmp13_tran &gt;- data.frame ( subset (tmp13_tran, select =-c (Freq.IP, Freq.OP) ) )
# Create 1 record per date per patient type for other visit dates tmp14 &gt;- data.frame (unique ( subset (tmp03, select=c(MR.No.,
adm2, type) ) ) ) tmp15 &gt;- subset (data.frame (table (tmp14$adm2, tmp14$type)), Freq &lt; 0) tmp15_tran &gt;- reshape (tmp15,
direction="wide", idvar= c("Var1"), timevar="Var2" ) tmp15_tran &gt;- data.frame (cbind (tmp15_tran, visitsIP = tmp15_tran$Freq.IP,
visitsOP = tmp15_tran$Freq.OP))
```

198 | P a g e tmp15_tran &gt;- data.frame (subset (tmp15_tran, select =-c (Freq.IP, Freq.OP))) # Create a table by one record per patient per date for determining what measurements # are done on which dates tmp16 &gt;- data.frame (unique (subset (tmp03, select=c(MR.No., adm2, data)))) tmp17 &gt;- subset (data.frame (table (tmp16$adm2, tmp16$data)), Freq &lt; 0) tmp17_tran &gt;- reshape (tmp17, direction="wide", idvar= c("Var1"), timevar="Var2") finalcal &gt;- data.frame (merge (x= tmp09, y =tmp11, by = "Var1", all=TRUE)) finalcal02 &gt;- data.frame (merge (x= finalcal, y =tmp13_tran, by = "Var1", all=TRUE)) finalcal03 &gt;- data.frame (merge (x= finalcal02, y =tmp15_tran, by = "Var1", all=TRUE)) finalcal04 &gt;- data.frame (merge (x= finalcal03, y =tmp17_tran, by = "Var1", all=TRUE)) write.csv(finalcal04, file="04_patient_analysis_tableu_calendar.csv") # Create 1 record per patient per disease with all other variables # This will help us understand age group as well as other demog factors # for different diseases

199 | P a g e dis &gt;- droplevels (unique (data.frame (subset (Diagnosis, Code !="", select = c(MR.No., Code))))) dis02 &gt;- data.frame (merge (x= dis, y =all_age02, by = "MR.No.", all.x=TRUE)) write.csv(dis02, file="04_patient_analysis_disease.csv") # Create 1 record per patient per date per diagnosis # Print this using the calendar display # Display this with patient ID as a filter dis03 &gt;- droplevels (unique (data.frame (subset (Diagnosis, Code !="", select = c(MR.No., Code, visdate))))) dis04 &gt;- data.frame (merge (x= dis03, y =all_age02, by = "MR.No.", all.x=TRUE)) write.csv(dis04, file="04_patient_analysis_disease_cal.csv") ### # End of program ### 6.5.10 Analysis program for Figure 3-7 Refer to program: 04_patients_analysis_tableu_adsl.R 6.5.11 Analysis program for Figure 3-8 Refer to program: 04_patients_analysis_tableu_adsl.R 6.5.12 Analysis program for Figure 3-9 Refer to program: 04_patients_analysis_tableu_adsl.R

200 | P a g e 6.5.13 Analysis program for Figure 3-10 Refer to program: 04_patients_analysis_tableu_adsl.R 6.5.14 Analysis program for Figure 3-11 Refer to program: 04_patients_analysis_tableu_adsl.R 6.5.15 Analysis program for Figure 3-12 R program: rmsd_metabolic_all.R ###################### ## Call all programs ##################### source("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\prgm\\rmsd_metabolic_subset.R") source("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\prgm\\diagnosis_primary.R") source("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\prgm\\diagnosis_primary_month.R") source("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\prgm\\diagnosis.R") source("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\prgm\\vital_sign.R") source("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\prgm\\lab.R")

201 | P a g e ## Calculate all the common variables from all datasets ## merge them back onto the individual datasets keep &gt;- c("MRNo", "Age", "AgeIn", "Gender", "City", "Country", "Bloodgrp", "data", "type", "visday") # Combine all datasets nvisall &gt;- data.table (rbind (diag2 [, keep, with = FALSE], vitals2[, keep, with = FALSE], lab2 [, keep, with = FALSE][MRNo !=""])) # Add a dummry variable nvisall &gt;- nvisall[, cal :=1] # Unique patient values n1unq &gt;- unique (nvisall [, c("MRNo", "Age", "AgeIn", "Gender", "City","Country"), with = FALSE]) # Count distinct visits for each domain based on diag2, vitals2 and lab2: n1vis &gt;- nvisall[, .(novis = uniqueN(visday)), by = .(MRNo, data)] n2vis &gt;- dcast(n1vis, MRNo ~ data, value.var = "novis")

202 | P a g e # No of diseases n1dis &gt;- unique (diag2 [, c("MRNo", "noofdis"), with = FALSE]) # Blood group creation n1blood &gt;- unique(nvisall [Bloodgrp != "", c("MRNo", "Bloodgrp"), with =FALSE]) # OP, IP creation n1type &gt;- unique (nvisall [type != "", c("MRNo", "type", "cal"), with =FALSE]) n2type &gt;- dcast(n1type, MRNo ~ type, value.var = "cal") # Combine the diag2 dataset and discat data and # determine diseases which are in RMSD and Metabolic categories vs. OTHER categories subset5 &gt;- merge (x = discat[, -c("Description"), with =FALSE], y = diag2[, -c("Code"), with =FALSE], all = TRUE, by.x = "Code", by.y = "Code2") subset5 &gt;- subset5 [,-c("Age", "AgeIn", "Gender", "City", "Country",

203 | P a g e "Bloodgrp", "data", "type", "noofdis"), with =FALSE] # Code non metabolic and non RMSD diseases as OTHER subset5$distype[is.na(subset5$distype)] &gt;- "OTHER" # Count number of diseases and freq count for each patient # ??? # ???? Understand why Code2 is not coming out correctly # ??? subset6 &gt;- subset5[, .(cnt = uniqueN(Code), frq = .N), by =.(MRNo, distype)] # Transpose the data and get in 1 row per patient subset7 &gt;- dcast(subset6, MRNo ~ distype, value.var = c("cnt", "frq"), fill = 0) # Combine all variables into 1 dataset ADSL adsl &gt;- Reduce(function(...) merge(..., all = TRUE, by = "MRNo"),

204 | P a g e list(n1unq, n1dis, n1blood, n2type, n2vis, subset7)) setnames(adsl, "vital", "nvis_vital") setnames(adsl, "diag", "nvis_diag") setnames(adsl, "lab", "nvis_lab") ## # Create a subsetted data for RMSD and Metabolic diseases # For analysis create a few additional variables ## adsl_sub &gt;- Reduce(function(...) merge(..., all.x = TRUE, by = "MRNo"), list(subset3, adsl)) # Merge this information onto diag2 dataset diag8 &gt;- Reduce(function(...) merge(..., all.x = TRUE, by = "MRNo"), list(adsl_sub, subset5)) write.csv(diag8, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_diag.csv", na=" ")

205 | P a g e # Merge this information onto vitals5 dataset vitals8 &gt;- Reduce(function(...) merge(..., all.x = TRUE, by = "MRNo"), list(adsl_sub, vitals5)) write.csv(vitals8, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_vital.csv", na=" ") # http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges # Version 2 of the code ## # Create a vertical vesion of diag_vital ## vitals5[, tempday := visday] ##

57% **MATCHING BLOCK 24/51** W

Add a redundant day column to use as the end range setkey(diag2, MRNo, min, max) ## Set the key for

patient IDs ("

69% **MATCHING BLOCK 49/51** W

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: ans2 &gt;- foverlaps(vitals5, diag2, by.x=c("MRNo", "visday", "tempday"))[, `:=`(inPolicy=T, tempday=NULL)] 206 |

Page ##

79% **MATCHING BLOCK 25/51** W

Update rows where the claim was out of policy: ans2[is.na(min), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(ans2, MRNo,Code2,visday, min) ## order the results setkey(ans2, MRNo, Code2) ## set the key to identify unique values ans2 &gt;- rbindlist(list(ans2, ## select only the unique values diag2[!.(ans2[, unique(MRNo)])] ## policies with no claims), fill=T) ans20 &gt;-

ans2 [, -c("Age", "AgeIn", "City", "Country", "Bloodgrp", "Gender","noofdis"), with = FALSE] ans3 &gt;- Reduce(function(...) merge(..., all.y = TRUE, by = "MRNo"), list(ans20, adsl_sub))
207 | Page write.csv(ans3, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_vital_vertical.csv", na=" ")
Create primary disease data, combination of 1 disease # considered as primary disease and display all other # diseases
prim_diag2 &gt;- Reduce(function(...) merge(..., all.y = TRUE, by = "MRNo"), list(prim_diag, adsl_sub)) write.csv(prim_diag2, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_primary_diag.csv", na=" ")
Create primary disease data, combination of 1 disease # considered as primary disease and display all other # diseases
##
208 | Page prim_diag_mon2 &gt;- Reduce(function(...) merge(..., all.y = TRUE, by = "MRNo"), list(prim_diag_mon, adsl_sub)) write.csv(prim_diag_mon2, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_primary_diag_mon.csv", na=" ") # Create a lookup table to get the cumulative view of the patients # Merge this onto individual datasets to create multiple records for patients # Use vismon variable and create many to many join dur &gt;- c("&lt;=1 day", "&lt;=1 month", "&lt;=2 months", "&lt;=3 months", "&lt;=6 months", "&lt;=1 year", "&lt;=2 years", "&lt;=3 years", "&lt;=4 years", "&lt;=5 years") durlwr &gt;- c(0, 1, 2, 3, 6, 12, 24, 36, 48, 60) durupr &gt;- c(999, 999, 999, 999, 999, 999, 999, 999, 999, 999) ref &gt;- data.table (cbind.data.frame (durlwr, durupr, dur))
209 | Page # http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges # Version 2 of the code ##

71% **MATCHING BLOCK 26/51** W

The foverlaps function requires both tables to have a start and end range, # and the "y" table to be keyed diag8[, tempmon := vismon] ## Add a redundant day column to use as the end range setkey(ref, durlwr, durupr) ## Set the key for

patient IDs ("

51% **MATCHING BLOCK 27/51** W

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: diag8rpt &gt;- foverlaps(diag8, ref, by.x=c("vismon", "tempmon"))[, `:=`(inPolicy=T, tempmon=NULL)] ## Update rows where the claim was out of policy: diag8rpt[is.na(durlwr), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(diag8rpt, MRNo, Code, vismon, durlwr) ## order the results 210 | Page setkey(diag8rpt, MRNo, Code, dur) ## set the key to identify unique values diag8rpt &gt;- rbindlist(list(diag8rpt, ## select only the unique values diag8[!.(diag8rpt[, unique(MRNo)])] ## policies with no claims), fill=T) #

Count number of unique patients, with only 1 visit, 2 visits,3 visits, etc. diag9rpt &gt;- diag8rpt [, unqvisit := uniqueN(dur), by = .(MRNo)] [order(MRNo, durlwr)] diag10rpt &gt;- diag9rpt [, .(nopat = uniqueN(MRNo)), by = .(unqvisit)] [order(unqvisit)] # Create a file for write.csv(diag9rpt, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_diag_repeat.csv", na=" ") # Cumulative view on the vital signs data vitals8[, tempmon := vismon] ##

60% **MATCHING BLOCK 28/51** W

Add a redundant day column to use as the end range setkey(ref, durlwr, durupr) ## Set the key for

patient IDs ("

52% **MATCHING BLOCK 29/51** W

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: 211 | P a g e vitals8rpt &gt;- foverlaps(vitals8 [vismon &lt; 0], ref, by.x=c("vismon", "tempmon"))[, `:=`(inPolicy=T, tempmon=NULL)] ## Update rows where the claim was out of policy: vitals8rpt[is.na(durlwr), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(vitals8rpt, MRNo, vitalparam, vismon, durlwr) ## order the results setkey(vitals8rpt, MRNo, vitalparam, dur) ## set the key to identify unique values vitals8rpt &gt;- rbindlist(list(vitals8rpt, ## select only the unique values vitals8[!.(vitals8rpt[, unique(MRNo)])] ## policies with no claims), fill=T) #

Create a file for vital signs observations repeated for # cumulative time point write.csv(vitals8rpt, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_vital_vertical_repeat.csv", na=" ")
212 | P a g e # Create primary diagnosis and all other diagnosis view by year # This gives a view about patient having many diseases simultaneously # along with other diseases setkeyv (diag8rpt, c("MRNo", "Code", "Description", "dur", "durlwr")) diag30 &gt;- unique(diag8rpt) diag40 &gt;- diag30[, `:=`(primarycode = Code, primarydesc = Description, primarydur = dur, primarydurlwr = durlwr),] diag50 &gt;- diag40[, c("MRNo", "Age", "AgeIn", "Gender", "City","Country", "Bloodgrp", "Code","distype", "Description", "dur", "durlwr", "durupr", "Metabolic", "RMSD", "combine"), with =FALSE] diag60 &gt;- diag40[, c("MRNo", "primarycode", "primarydesc", "primarydur", "primarydurlwr"), with =FALSE]
213 | P a g e # set the ON clause as keys of the tables: setkey(diag50,MRNo, dur) setkey(diag60,MRNo, primarydur) # perform the join prim_diag_dur_repeat &gt;- data.table(merge(diag50,diag60, all=TRUE, allow.cartesian = TRUE)) # Clean up some space rm (list = ls(pattern = "lab*")) rm (list = ls(pattern = "n1*")) # Create a file for vital signs observations repeated for # cumulative time point write.csv(prim_diag_dur, file ="C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\analysis\\rmsd_met_primary_diag_dur_repeat.csv", na=" ") # Find difference between 2 consecutive visits # This may give us some idea about the data
214 | P a g e diff &gt;- diag8 [, c("MRNo", "visday"), with =FALSE] setkey(diag8, MRNo, visday) diff &gt;- unique(diff) [order (MRNo, visday)] diff &gt;- diff[,diff:=c(NA,diff(visday)),by=MRNo] summary(diff$diff)
##
End of program
##
#### 6.5.16 Analysis program for Figure 3-13 Refer to program: rmsd_metabolic_all.R 6.5.17 Analysis program for Figure 3-14 Refer to program: rmsd_metabolic_all.R 6.5.18 Analysis program for Figure 3-15 Refer to program: rmsd_metabolic_all.R 6.5.19 Analysis program for Figure 3-16 R program: 085_dis_1st_time_refCal_NodesEdges.R library(data.table) library(stringi) library(stringr) library(sqldf)
215 | P a g e

83% **MATCHING BLOCK 30/51** W

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") unqdis &gt;- unique(all_met_rmsd [,

c("mr_no", "studyday", "Code", "description"),]) unqdis &gt;- unqdis [!Code %in% c("" , " ")] unqdis &gt;- unqdis [, mindisday := min(studyday), by = .(mr_no, Code, description)] unqdis &gt;- unique(unqdis [, c("mr_no", "mindisday", "Code", "description"),]) setnames(unqdis, "Code", "refcode") setnames(unqdis, "description", "refdesc")
Create this data with min refday for each disease
###

59% **MATCHING BLOCK 31/51** W

all_met_rmsd02 &gt;- merge(x = all_met_rmsd, y = unqdis, by = c("mr_no"), allow.cartesian = TRUE) 216 | P a g e all_met_rmsd02 &gt;- all_met_rmsd02 [,

c("mr_no", "Code", "description", "combine", "RMSD", "Metabolic", "newdt0", "Type_med", "Coded_med", "studyday", "mindisday", "refcode", "refdesc", "patient_gender", "age", "baseage", "distype", "cdur"),]
Calculate reference day for each disease as before and after # studyday and mindisday
###

45% **MATCHING BLOCK 32/51** W

all_met_rmsd02 &gt;- all_met_rmsd02 [, refday := ifelse(studyday &lt;= mindisday, studyday - mindisday + 1, studyday - mindisday),] all_met_rmsd02 &gt;- all_met_rmsd02[, refmnyr := ifelse(

refday &lt;= 1, as.numeric(ceiling (refday / 30.4375)), as.numeric(floor (refday / 30.4375))),]
217 | P a g e period01 &gt;- fread("D:/Hospital_data/ProgresSQL/analysis/lookup_1st_nodesedges.csv") period02 &gt;- period01[, list(period = period, periodn = periodn, refmnyr = seq(as.numeric(start), as.numeric(end))), by = 1:nrow(period01)] all_met_rmsd02 &gt;- merge (x = all_met_rmsd02, y = period02 [, c("refmnyr", "period", "periodn")], by = c("refmnyr"), all.x = TRUE) ##################################### # Post process to get day 1 as Day 1 #####################################

32% **MATCHING BLOCK 33/51** W

all_met_rmsd02 &gt;- all_met_rmsd02[, period := ifelse(refday == 1, 1, period),] all_met_rmsd02 &gt;- all_met_rmsd02[, periodn := ifelse(refday == 1, "Day 1", periodn),] saveRDS (all_met_rmsd02, "D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") disease &gt;- unique(all_met_rmsd02 [, -c("

Type_med", "Coded_med"),]) disease &gt;- disease [, cat :="Disease"]
218 | P a g e meds &gt;- unique(all_met_rmsd02 [, -c("Code", "description"),]) meds &gt;- meds [, cat :="Medicine"] meds &gt;- meds [, Coded_med := paste(Type_med, Coded_med, sep=":"),] setnames(meds, "Type_med", "Code") setnames(meds, "Coded_med", "description") all &gt;- rbind(disease, meds) bfraftr &gt;- all [, .(min = min(refday), max = max(refday)), by = .(mr_no, refcode, refdesc, Code, description)] bfraftr &gt;- sqldf("select *, case When min &gt;= 0 and max &gt;= 0 then 'Reported only before' When min &lt;= 1 and max &lt;= 1 then 'Reported on or after' When min &gt;= 0 and max &lt;= 1 then 'Reported before and after' end as classification from bfraftr")
219 | P a g e all02 &gt;- merge (x = all, y = bfraftr , by = c("mr_no", "refcode", "refdesc", "Code", "description"), all.x = TRUE) all02 &gt;- all02 [, -c("min", "max"),] fwrite(all02, "D:/Hospital_data/ProgresSQL/analysis/085_dis_1st_time_refCal_NodesEdges.csv") saveRDS (all02, "D:/Hospital_data/ProgresSQL/analysis/085_dis_1st_time_refCal_NodesEdges.rds") ## # Save disease and medicine version of the data ##

66% **MATCHING BLOCK 34/51** W

all_met_rmsd02 &gt;- merge (x = all_met_rmsd02, y = bfraftr , by = c("mr_no", "refcode", "refdesc", "Code", "description"), all.x = TRUE) all_met_rmsd02 &gt;- all_met_rmsd02 [, -

c("min", "max"),] saveRDS (

83% **MATCHING BLOCK 35/51** W

all_met_rmsd02, "D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") 220 |

P a g e ## # End of program ## 6.5.20 Analysis program for Figure 3-17 R program: 060_allopathic_diag.R library(dplyr) library(data.table) library(fuzzyjoin) library(stringr) library(stringi) library(stringdist) library(quanteda) library(tm) library(tidyr) library(sqldf) ## # This section creates the allopathic diagnosis as per ICD 10 dictionary ##

54% **MATCHING BLOCK 36/51** W

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") 221 | P a g e all_met_rmsd &gt;- all_met_rmsd [, `:=` (baseage = min(age)), by =.(mr_no)] all_met_rmsd &gt;- all_met_rmsd [, `:=` (

```
vismon = round( cdur/30.4375, digits = 0))] # Baseline age age01 &gt;- unique( all_met_rmsd [, c("mr_no", "baseage", "patient_gender",
"cdur")] ) lookup_allopathic_diag &gt;- fread("D:/Hospital_data/ProgresSQL/analysis/lookup_allopathic_diag.txt", sep="|") chkpat &gt;-
function (dname, var, dataout = "unqsec") { sec &gt;- readRDS( paste("D:/Hospital_data/ProgresSQL/analysis/", noquote(dname) ,
".rds", sep="") ) sec011 &gt;- sec [, `:=` (orig = get(var), all_diag = toupper( get( var))), ]
############################################################ # Replace multiple diseases in 1 row to
multiple rows, # Seperate_rows allows: keeping all other rows as is
222 | P a g e ############################################################ sec011_1 &gt;-
separate_rows(sec011, all_diag, sep =",|\r\n|\\?|\\bAND\\b|;" ) #sec011_1 &gt;- sec011_1 [, all_diag := trimws(all_diag), ] #sec011_1
&gt;- sec011_1 [, dname := paste(var, sep = ""), ] sec011_1 [, all_diag := trimws(all_diag)] sec011_1 [, dname := paste(var, sep = "")]
###################################################################### # Create unique row per
disease, this will be matched against ICD 10
###################################################################### #unqsec &gt;- unique(
sec011_1 [, c("all_diag", "dname"), ] ) tmp &gt;- sec011_1 assign(dataout, tmp, envir=.GlobalEnv) } chkpat(dname = "sec011", var=
"sec011_var001_Allopathic Diagnosis", dataout = "dsec011") chkpat(dname = "sec082", var= "sec082_var001_Allopathic Diagnosis",
dataout = "dsec082") chkpat(dname = "sec122", var= "sec122_var001_Allopathic Diagnosis", dataout = "dsec122")
223 | P a g e chkpat(dname = "sec123", var= "sec123_var001_Allopathic Diagnosis", dataout = "dsec123") dislist &gt;-
lapply(ls(pattern="dsec*"), get) dis_all &gt;- data.table( rbindlist (dislist)) dis_all02 &gt;- merge(x = dis_all, y = lookup_allopathic_diag,
all.x = TRUE, allow.cartesian=TRUE, by = c("all_diag", "dname") ) ########################## # Subset for coded records
########################## dis_pat &gt;- dis_all02 [ nchar(code01) &lt; 0 ] dis_pat &gt;- dis_pat[ code01 != c("** Can
not be coded")] dis_pat02 &gt;- merge (x = dis_pat, y = age01, by = c("mr_no"))
224 | P a g e unqpat &gt;- dis_pat02 [, .(npat = uniqueN (mr_no)), by = .(combine)] unqpat_gender &gt;- dis_pat02 [, .(npat = uniqueN
(mr_no)), by = .(combine, patient_gender)] chk01 &gt;- unique( dis_pat02 [, c("mr_no", "code01", "text01", "baseage", "patient_gender",
"combine", "Metabolic", "RMSD", "cdur", "all_vis")]) chk01 &gt;- chk01 [, high := substr(code01, 1, 3)] ################# # ICD
dictionary ################# icd10 &gt;- fread("D:/Hospital_data/ProgresSQL/analysis/icd10cm_order_2018.csv", header=
FALSE) cats &gt;- data.table( expand.grid( cat1 = LETTERS, cat2 = seq (0, 99) ) ) cats &gt;- cats [, high := paste( cat1, str_pad(cat2, 2,
side = "left", pad = 0), sep=""), ] cats &gt;- sqldf("select *,
225 | P a g e case When cat1 == 'A' OR cat1 == 'B' then 'Certain infectious and parasitic diseases' When (cat1 == 'C' OR (cat1 == 'D'
AND cat2 &gt; 50)) then 'Neoplasms' When (cat1 == 'D' AND cat2 &lt;=50) then 'Diseases of the blood and blood- forming organs and
certain disorders involving the immune mechanism' When (cat1 == 'E') then 'Endocrine, nutritional and metabolic diseases' When (cat1
== 'F') then 'Mental and behavioural disorders' When (cat1 == 'G') then 'Diseases of the nervous system' When (cat1 == 'H' and cat2
&gt;= 59) then 'Diseases of the eye and adnexa' When (cat1 == 'H' and cat2 &lt; 59) then 'Diseases of the ear and mastoid process'
When (cat1 == 'I') then 'Diseases of the circulatory system' When (cat1 == 'J') then 'Diseases of the respiratory system' When (cat1 ==
'K') then 'Diseases of the digestive system' When (cat1 == 'L') then 'Diseases of the skin and subcutaneous tissue' When (cat1 == 'M')
then 'Diseases of the musculoskeletal system and connective tissue' When (cat1 == 'N') then 'Diseases of the genitourinary system'
When (cat1 == 'O') then 'Pregnancy, childbirth and the puerperium' When (cat1 == 'P') then 'Certain conditions originating in the
perinatal period'
226 | P a g e When (cat1 == 'Q') then 'Congenital malformations, deformations and chromosomal abnormalities' When (cat1 == 'R')
then 'Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified' When (cat1 == 'S' OR cat1 == 'T') then
'Injury, poisoning and certain other consequences of external causes' When (cat1 == 'V' OR cat1 == 'W' OR cat1 == 'X' OR cat1 == 'Y')
then 'Injury, poisoning and certain other consequences of external causes' When (cat1 == 'Z') then 'Factors influencing health status
and contact with health services' When (cat1 == 'U') then 'Codes for special purposes' end as icd from cats")
############################################## # Get the 2nd level terms from ICD dictionary
############################################## icd_sub &gt;- icd10 [ V2 %in% unique (chk01$high)]
###################################################### # Merge the high level terms and
2nd level terms with the data ######################################################
227 | P a g e chk02 &gt;- merge (x = chk01, y = cats, all.x = TRUE, by = c("high")) chk02 &gt;- merge (x = chk02, y = icd_sub [, c("V2",
"V5")], all.x = TRUE, by.x = c("high"), by.y = c("V2")) fwrite(chk02, "D:/Hospital_data/ProgresSQL/analysis/060_allopathic_diag.csv")
###########################################################################
#### # End of program
###########################################################################
#### 6.5.21 Analysis program for Figure 3-18 Refer to R program: 060_allopathic_diag.R 6.5.22 Analysis program for Figure 3-19
Refer to R program: 060_allopathic_diag.R
228 | P a g e 6.5.23 Analysis program for Figure 3-20 Refer to R program: 060_allopathic_diag.R 6.5.24 Analysis program for Figure 3-
21 Refer to R program 6.5.25 Analysis program for Figure 3-22 R program: diagnosis_primary.R
################################## # Code to generate DIAGNOSIS data
################################## library(data.table) setwd ("C:\\Users\\Lucky\\Documents\\Hospital
data\\01_31JUL2016\\source") diag &gt;- fread("Diagnosis.csv", check.names = FALSE) diag2 &gt;- diag[, c(1, 8, 9), with=FALSE ]
setnames(diag2, "MR No.", "MRNo") # Code ** NOT YET CODED for the missing code values # Sort the data by patient and day diag2
&gt;- diag2[, Code2 := ifelse(Code =="", "ZZZ999", Code), ] [order(MRNo, Code2)] setkeyv (diag2, c("MRNo", "Code2", "Description"))
diag3 &gt;- unique(diag2)
```

229 | P a g e diag4 &gt;- diag3[, `:=`(primarycode = Code2, primarydesc = Description),] diag3 &gt;- diag3[, c(1, 3, 4), with =FALSE] diag4 &gt;- diag4[, c(1, 5, 6), with =FALSE] # set the ON clause as keys of the tables: setkey(diag3,MRNo) setkey(diag4,MRNo) # perform the join prim_diag &gt;- merge(diag3,diag4, all=TRUE, allow.cartesian = TRUE) ## #### # End of program ## #### 6.5.26 Analysis program for Figure 3-23 Refer to R program: diagnosis_primary.R 6.5.27 Analysis program for Figure 3-24 Refer to R program: diagnosis_primary.R

230 | P a g e 6.5.28 Analysis program for Figure 3-25 R program: diagnosis_primary_mon.R ##################################### # Code to generate DIAGNOSIS data ##################################### library(data.table) setwd ("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\source") diag &gt;- fread("Diagnosis.csv", check.names = FALSE) diag2 &gt;- diag[, c(1, 8, 9, 10), with=FALSE] setnames(diag2, "MR No.", "MRNo") setnames(diag2, "Admission Date", "visdate") # Code ** NOT YET CODED for the missing code values # Sort the data by patient and day # Create date variables and find the difference diag2 &gt;- diag2[, visdate := as.POSIXct(gsub("-", "/", visdate), format="%d/%m/%Y")] diag2 &gt;- diag2[, Code2 := ifelse(Code =="", "ZZZ999", Code),] [order(MRNo, Code2)]

231 | P a g e # Only extract month part diag2 &gt;- diag2[, month := format(as.Date(visdate), "%m"),] setkeyv (diag2, c("MRNo", "Code2", "Description", "month")) diag3 &gt;- unique(diag2) diag4 &gt;- diag3[, `:=`(primarycode = Code2, primarydesc = Description, primarymon = month),] diag5 &gt;- diag4[, c(1, 3, 5, 6), with =FALSE] diag6 &gt;- diag4[, c(1, 7, 8, 9), with =FALSE] # set the ON clause as keys of the tables: setkey(diag5,MRNo, month) setkey(diag6,MRNo, primarymon) # perform the join prim_diag_mon &gt;- merge(diag5,diag6, all=TRUE, allow.cartesian = TRUE) ## ####

232 | P a g e # End of program ## #### 6.5.29 Analysis program for Figure 3-26 R program: 085_dis_counts_bruce_java.R ### # This is used for 085_dis_count_edges_3rd_byPeriod Tableau display ### library(data.table) library(stringi) library(stringr) library(sqldf) library(tidyr) library(rjson) library(jsonlite) library(dplyr) ### # These 2 are created using 085_dis_1st_time_refCal_NodeEdges.R program

233 | P a g e ##

55% **MATCHING BLOCK 37/51** W

all_met_rmsd02 &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Coded_med := paste(Type_med, Coded_med, sep=":"),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Coded_med := str_replace_all(Coded_med, "\"", ""),] chk01 &gt;- all_met_rmsd02 [, .(

cnt = uniqueN(mr_no)), by = .(refcode, refdesc, Code, description, Type_med, Coded_med)] chk01 &gt;- chk01[Code != "" & Coded_med != ""] chk01 &gt;- chk01 [, `:=` (Code02 = paste(Code, ":", description, "-&lt;", Coded_med, sep =""), name = paste(refcode, refdesc, sep =","), key = paste(Code, description, sep=",")),] med &gt;- unique(chk01 [Coded_med != c("", " "), c("Coded_med"),]) setnames(med, "Coded_med", "name") dis &gt;- unique(chk01 [name != c("", " "), c("name"),]) meddis &gt;- rbind(med, dis) meddis &gt;- meddis [, nrow := .I,]

234 | P a g e ### # Create a version of data as follows: # Fixed nodes as relation between # (1) Period + Reference disease &gt;--&lt; other diseases # (2) other diseases &gt;--&lt; Medicine # Use the other diseases section for creating the moving nodes ### part01 &gt;- chk01 [, c("name", "key", "Code02", "cnt", "refdesc", "refcode"),] part02 &gt;- chk01 [, c("Coded_med", "key", "Code02", "cnt", "refdesc", "refcode"),] setnames(part02, "Coded_med", "name") part03 &gt;- rbind (part01, part02) part03 &gt;- merge(part03, meddis, by = c("name"), all = TRUE) part03 &gt;- part03 [, num := .N, by =.(refdesc, key)]

235 | P a g e part03 &gt;- part03 [, maxnum := max(num), by =.(refdesc)] part03 &gt;- part03 [, pernum := (num / maxnum) * 100,] ### # Create the Json file # # [{ # "name": "addons", # "count": 1, # "key": "addons", # "pages": [{ # "name": "A year in apps script and my bucket list", # "key": "4478459723408930641", # "title": "A year in apps script and my bucket list", # "url": "http://excelramblings.blogspot.com/2015/01/a-year-in-apps-script-and-my- bucket-list.html" # }] # } # # "name" : use key

```
236 | P a g e # "count" : use num # "key" : use key # "pages" : # "names" : use name # "key" : use nrow # "title" : use name # "url" :
Code02 ######################################################################## part04 &gt;-
part03 [, frstprt := paste('{"name" :"', key, '", "count" :', pernum, ', "key" :"', key, '",', sep ="" ), ] part04 &gt;- part04 [, scndprt :=
paste('{"name" :"', name, '", "key" :', nrow, ', "title" :"', name, '", "url" :"', Code02, '"}', sep = ""), ]
######################################################################### # Combine the
scndprt variable into 1 row per refdesc + key combination
######################################################################### part05 &gt;- part04
[, .(scndprt02 = paste(scndprt, collapse = ",", sep = " " )), by = .(refcode, refdesc, frstprt)] part05 &gt;- part05 [ order(refcode, refdesc,
frstprt)] part05 &gt;- part05 [, rowrecal := .I, by = .(refcode, refdesc)]
237 | P a g e part05 &gt;- part05 [, scndprt03 := paste('"pages": [', scndprt02, "]},", sep=""), ] chk02 &gt;- part05 [ refcode == "A2.0"]
fwrite(chk02 [ scndprt02 != "...", c("frstprt", "scndprt03"), ], "D:/Hospital_data/ProgresSQL/analysis/085d3concept.json", col.names =
FALSE, quote = FALSE, sep = " ") chk02 &gt;- part05 [ refcode == "P5.0"] fwrite(chk02 [ scndprt02 != "...", c("frstprt", "scndprt03"), ],
"D:/Hospital_data/ProgresSQL/analysis/085d3concept_P5_0.json", col.names = FALSE, quote = FALSE, sep = " ")
######################################################################################
#### # End of program
######################################################################################
####
238 | P a g e chk02 &gt;- chk02 [, `:=` (name = paste(period, periodn, refcode, refdesc, sep =","), count = cnt, key = paste(Code,
description, sep=","), pages = Coded_med, url = Code02, title = Code02),] chk03 &gt;- chk02 [, c("name", "count", "key", "pages", "url",
"title", "nrow"), ] write_json(chk03, "D:/Hospital_data/ProgresSQL/misc/bruce_approach/085d3concept.json") # Validate Json using
https://jsonlint.com/ fwrite(chk01, "D:/Hospital_data/ProgresSQL/analysis/085_dis_count_edges_3rd_byPeriod_.csv") chk01 &gt;-
all_met_rmsd02 [, .(cnt = uniqueN(mr_no)),
239 | P a g e by = .(refcode, refdesc, Code, description, Type_med, Coded_med )] chk01 &gt;- chk01[Code != "" & Coded_med != ""]
chk01 &gt;- chk01 [, Code02 := paste(Code, ":", description, "-&lt;", Coded_med, sep =""),] chk02 &gt;- chk01 [ refcode == "A2.0"]
fwrite(chk02, "D:/Hospital_data/ProgresSQL/analysis/085_dis_count_edges_3rd_byPeriod_A2_bruce.csv")
######################################################################################
#### # End of program
######################################################################################
#### 6.5.30 Analysis program for Figure 3-27 R program: 080_medicine_repeat_prop.R library(data.table) library(stringi)
library(stringr) library(sqldf) library(scales)
```

100% MATCHING BLOCK 38/51 W

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") 240 |

```
P a g e #substr(cat_id, 1, 3) != "SER" #c("mr_no", "medicine_name", "studyday", "remarks", "frequency", "duration", "duration_units",
"Coded_med", "Type_med", "quantity", "patient_id", "cat_id")] ) ############################ # Data related to medicines
############################ meds0 &gt;- unique( all_met_rmsd [medicine_name != " ", c("mr_no", "studyday",
"Coded_med", "Type_med")] ) ############################################################# # Get
the minimum day (minday) for any medicine and # Get the minimum day (minmedday) for individual medicine
############################################################# meds0 &gt;- meds0 [order(mr_no,
studyday)] meds0 &gt;- meds0 [, minday := min(studyday), by = .(mr_no)] meds0 &gt;- meds0 [, minmedday := min(studyday), by = .
(mr_no, Type_med, Coded_med)] #############################################################
241 | P a g e # Get group (each day of treatment) as a grouping variable # Get individual sequential rows within each group
############################################################# time &gt;- unique(all_met_rmsd [,
c("mr_no", "studyday")] ) time &gt;- time [order(mr_no, studyday)] time &gt;- time [, grpday := 1:.N, by = .(mr_no)] time &gt;- time [,
grpmaxday := max(grpday), by = .(mr_no)] ######################################################## #
Merge the grouping variables for further calculations # Sort the data
######################################################### meds0 &gt;- merge (x = meds0, y = time, by
= c("mr_no", "studyday") ) meds0 &gt;- meds0[order(mr_no, studyday, grpday)]
################################################################################# # Sort
the data to get prescription number for each medicine # If the prescription number is &lt; 1 then that medicine is given more than
once # # There are 2 sequence variables: one for day and one for medicine
```

242 | P a g e ### cum01 &gt;- meds0 [, presc := 1:.N, by = .(mr_no, Type_med, Coded_med)] cum01 &gt;- cum01 [order(mr_no, studyday, minday, Type_med, Coded_med, presc)] cum01 &gt;- cum01 [, grpall := 1:.N, by = .(mr_no)] ## # If the prescription = 1 and studyday = minmedday then Start # If prescription &lt; 1 then Old (already given and not a medicine) # If prescription group number is &lt; 1 then Start ## cum01 &gt;- cum01 [, newold := ifelse (studyday == minmedday, "1st dose", ""),] cum02 &gt;- cum01 [, newold2 := ifelse(presc &lt; 1 & grpday &lt; 1 & studyday &lt; minmedday & newold != "1st dose", "Repeat", newold), by =.(mr_no)] cum02 &gt;- cum02 [, cat := "Medicine",] ### # Duplicate the medication and see which medications are given multiple times # This gives a cumulative view of what has been prescribed till a certain # Visit, how many medicines are 1st time given and how many are Repeated ### 243 | P a g e cum03 &gt;- cum02 [, (list(cumday = (grpday: grpmaxday))), by = .(mr_no, presc, Type_med, Coded_med, studyday, grpday, grpmaxday, minmedday, newold2, cat)] cum03 &gt;- cum03 [, cumday2 := paste("Till visit", cumday, sep = " "),] ## #### # Execute similarly for the diseases area # check if it is easy to combine disease and medicine like 01_Primary_madhumeha # display ## #### #substr(cat_id, 1, 3) != "SER" # nchar(Code) &lt; 0 ############################ # Data related to diseases ############################ meds0 &gt;- unique(all_met_rmsd [, c("mr_no", "Code", "studyday", "description")]) 244 | P a g e meds0 &gt;- data.table(meds0 [,

65% MATCHING BLOCK 39/51 W

Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),]) meds0 &gt;- data.table(meds0 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),]) ### #

Get the minimum day (minday) for any medicine and # Get the minimum day (minmedday) for individual medicine ### meds0 &gt;- meds0 [order(mr_no, studyday, Code, description)] meds0 &gt;- meds0 [, minday := min(studyday), by = .(mr_no)] meds0 &gt;- meds0 [, minmedday := min(studyday), by = .(mr_no, Code, description)] ### # Merge the grouping variables for further calculations # Sort the data ## meds0 &gt;- merge (x = meds0, y = time, by = c("mr_no", "studyday")) meds0 &gt;- meds0[order(mr_no, studyday, grpday)] ### 245 | P a g e # Sort the data to get prescription number for each medicine # If the prescription number is &lt; 1 then that medicine is given more than once # # There are 2 sequence variables: one for day and one for medicine ### cum01 &gt;- meds0 [, presc := 1:.N, by = .(mr_no, Code, description)] cum01 &gt;- cum01 [order(mr_no, studyday, minday, presc)] cum01 &gt;- cum01 [, grpall := 1:.N, by = .(mr_no)] ## # If the prescription = 1 and studyday = minmedday then Start # If prescription &lt; 1 then Old (already given and not a medicine) # If prescription group number is &lt; 1 then Start ## cum01 &gt;- cum01 [, newold := ifelse (studyday == minmedday, "1st time disease", ""),] cum02dis &gt;- cum01 [, newold2 := ifelse(presc &lt; 1 & grpday &lt; 1 & studyday &lt; minmedday & newold != "1st time dose", "Repeat", newold), by =.(mr_no)] cum02dis &gt;- cum02dis [, cat := "Disease",] setnames (cum02dis, "Code", "Type_med") 246 | P a g e setnames (cum02dis, "description", "Coded_med") ### # Duplicate the medication and see which medications are given multiple times # This gives a cumulative view of what has been prescribed till a certain # Visit, how many medicines are 1st time given and how many are Repeated ### cum03dis &gt;- cum02dis [, (list(cumday = (grpday: grpmaxday))), by = .(mr_no, Type_med, presc, Coded_med, studyday, grpday, grpmaxday, minmedday, newold2, cat)] cum03dis &gt;- cum03dis [, cumday2 := paste("Till visit", cumday, sep = " "),] ### # Combine all disease and medicine information # for individual visits as well as cumulative visit data ### cum02all &gt;- rbind (cum02, cum02dis, fill = TRUE) cum02all &gt;- cum02all[, -c("newold"),] cum03all &gt;- rbind (cum03, cum03dis, fill = TRUE) 247 | P a g e fwrite(cum02all, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_dis_repeat_prop.csv") fwrite(cum03all, "

65% **MATCHING BLOCK 40/51** W

D:/Hospital_data/ProgresSQL/analysis/080_medicine_dis_repeat_prop_cumulative.csv") all_met_rmsd0 &gt;- data.table(all_met_rmsd [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),]) all_met_rmsd0 &gt;- data.table(all_met_rmsd0 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),])

keep &gt;- c("mr_no", "studyday", "patient_gender", "baseage", "age", "Code", "description", "Coded_med", "Type_med", "combine", "Metabolic", "RMSD", "vis", "season", "newdt0", "distype") all_met_rmsd_unq &gt;- unique(all_met_rmsd0 [, ..keep,])

52% **MATCHING BLOCK 41/51** W

all_met_rmsd_unq02 &gt;- merge(x = all_met_rmsd_unq, y = cum02 [, -c("newold", "cat"),], by = c("mr_no", "

studyday", "Coded_med", "Type_med"), all.x = TRUE)

248 | P a g e ## # Should look at this syntax for these 2 variables ## setnames (cum02dis, "Type_med", "Code") setnames (cum02dis, "Coded_med", "description") setnames (cum02dis, "presc", "prescdis") setnames (cum02dis, "newold2", "newold2dis") setnames (cum02dis, "grpday", "grpdaydis") all_met_rmsd_unq03 &gt;- merge(x = all_met_rmsd_unq02, y = cum02dis [, -c("newold", "cat", "grpall", "minday", "minmedday", "grpmaxday"),], by = c("mr_no", "studyday", "Code", "description"), all.x = TRUE) fwrite(all_met_rmsd_unq03, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_dis_all_met_rmsd_prop.csv")

249 | P a g e all_met_rmsd_unq04 &gt;- all_met_rmsd_unq03 [grpday &lt; 0, `:=`(cumday = grpmaxday, cumday3 = max(studyday), cumday2 = paste("Till visit", grpmaxday, sep = " ")), by = .(mr_no)] ### # Duplicate the medication and see which medications are given multiple times # This gives a cumulative view of what has been prescribed till a certain # Visit, how many medicines are 1st time given and how many are Repeated ### all_met_rmsd_unq05 &gt;- all_met_rmsd_unq03 [grpday &lt; 0, (list(cumday = (grpday: grpmaxday))), by = .(mr_no, presc, prescdis, Type_med, Coded_med, Code, description, baseage, age, combine, Metabolic, RMSD, studyday, grpday, grpmaxday, minmedday, newold2, newold2dis)] all_met_rmsd_unq05 &gt;- all_met_rmsd_unq05 [, cumday2 := paste("Till visit", cumday, sep = " "),]

250 | P a g e all_met_rmsd_unq05 &gt;- all_met_rmsd_unq05 [, cumday3 := max(studyday), by = .(mr_no, cumday2)] ### # Count number of 1st and repeat diseases # for individual patient # # Count number of 1st and repeat doses # for individual patient # # Transpose the ### a0dis &gt;- all_met_rmsd_unq04 [, .(cntdis = uniqueN(paste(Code, description, sep=" "))), by = .(mr_no, grpday, cumday, cumday2, cumday3, newold2dis)] a0dis_t &gt;- dcast(data = a0dis, mr_no + grpday + cumday + cumday2 + cumday3 ~ newold2dis, value.var = c("cntdis"), fill = 0)

251 | P a g e setnames(a0dis_t, "Repeat", "Repeatdis") a0dose &gt;- all_met_rmsd_unq04 [, .(cntdose = uniqueN(paste(Type_med, Coded_med, sep=" "))), by = .(mr_no, grpday, cumday, cumday2, cumday3, newold2)] a0dose_t &gt;- dcast(data = a0dose, mr_no + grpday + cumday + cumday2 + cumday3 ~ newold2, value.var = c("cntdose"), fill = 0) setnames(a0dose_t, "Repeat", "Repeatdose") ### # Count total number of diseases and doses for individual patients ### a0distot &gt;- all_met_rmsd_unq05 [, .(totdis = uniqueN(paste(Code, description, sep=" "))), by = .(mr_no, cumday, cumday2, cumday3)] a0dosetot &gt;- all_met_rmsd_unq05 [, .(totdose = uniqueN(paste(Type_med, Coded_med, sep=" "))), by = .(mr_no, cumday, cumday2, cumday3)]

252 | P a g e a01small &gt;- Reduce(function(...) merge(..., all.y = TRUE, by = c("mr_no", "grpday", "cumday", "cumday2", "cumday3")), list(a0dis_t, a0dose_t)) a01cap &gt;- Reduce(function(...) merge(..., all.y = TRUE, by = c("mr_no", "cumday", "cumday2", "cumday3")), list(a0distot, a0dosetot)) a01all &gt;- merge(x = a01small [, -c("cumday", "cumday2", "cumday3"),], y = a01cap, by.x = c("mr_no", "grpday"), by.y = c("mr_no", "cumday")) a01all &gt;- a01all [, `:=` (perc1dis = percent(`1st time disease` / totdis), percrepdis = percent(`Repeatdis` / totdis), perc1dose = percent(`1st dose` / totdose), percrepdose = percent(`Repeatdose` / totdose)) ,] ## # Get the diseases and doses collapsed into 1 row ##

253 | P a g e dis &gt;- unique(all_met_rmsd_unq03 [grpday &lt; 0, c("mr_no", "grpday", "Code", "description", "distype", "studyday", "newold2dis"),]) dis &gt;- dis [, `:=` (disall = paste(distype, Code, sep= ":"), desall = paste(distype, description, sep= ":"))] discomb &gt;- dis [grpday &lt; 0, .(discomb = paste(disall, collapse = ";", sep = " "), descomb = paste(desall, collapse = ";", sep = " ")), by = .(mr_no, grpday, newold2dis)] discomb_t &gt;- dcast(data = discomb, mr_no + grpday ~ newold2dis, value.var = c("discomb", "descomb")) dose &gt;- unique(all_met_rmsd_unq03 [grpday &lt; 0, c("mr_no", "grpday", "Type_med", "Coded_med", "studyday", "newold2"),]) dose &gt;- dose [, doseall := paste(Type_med, Coded_med, sep= ":")] doscomb &gt;- dose [grpday &lt; 0, .(dosecomb = paste(doseall, collapse = ";", sep = " ")), by = .(mr_no, grpday, newold2)]

254 | P a g e doscomb_t &gt;- dcast(data = doscomb, mr_no + grpday ~ trimws(paste("Combine", newold2, sep="")), value.var = c("dosecomb")) adsl &gt;- unique(all_met_rmsd_unq03 [grpday &lt;0, c("mr_no", "patient_gender", "grpday", "season", "age", "baseage", "combine", "Metabolic", "RMSD"),]) a01all &gt;- Reduce(function(...) merge(..., all.y = TRUE, by = c("mr_no", "grpday")), list(a01all, discomb_t, doscomb_t)) a01all &gt;- merge(x = a01all, y = adsl, by = c("mr_no", "grpday")) fwrite(a01all, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_repeat_prop_cumulative_Rcal.csv") saveRDS (a01all, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_repeat_prop_cumulative_Rcal.rds")

255 | P a g e ## # Create disease and medicine combination # by patient ## dismed &gt;- all_met_rmsd_unq04 [, .(cntdismed = .N), by = .(mr_no, Code, description, Type_med, Coded_med)] dis100 &gt;- all_met_rmsd_unq04 [, .(cntdis = .N), by = .(mr_no, Code, description)] med100 &gt;- all_met_rmsd_unq04 [, .(cntmed = .N), by = .(mr_no, Type_med, Coded_med)] dismed01 &gt;- merge(x = dismed, y = dis100, by = c("mr_no", "Code", "description"), all = TRUE) dismed02 &gt;- merge(x = dismed01, y = med100,

256 | P a g e by = c("mr_no", "Type_med", "Coded_med"), all = TRUE) fwrite(dismed02, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_bymr_no_dismed_comb_Rcal.csv") saveRDS (dismed02, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_bymr_no_dismed_comb_Rcal.rds") ## # Create disease and medicine combination # by medicine and disease # count number of combinations and number of # patients ## a_dismed &gt;- all_met_rmsd_unq04 [, .(cntdismed = .N, unqdismedpat = uniqueN(mr_no)), by = .(Code, description, Type_med, Coded_med)] a_dis100 &gt;- all_met_rmsd_unq04 [, .(cntdis = .N, unqdispat = uniqueN(mr_no)), by = .(Code, description)]

257 | P a g e a_med100 &gt;- all_met_rmsd_unq04 [, .(cntmed = .N, unqmedpat = uniqueN(mr_no)), by = .(Type_med, Coded_med)] a_dismed01 &gt;- merge(x = a_dismed, y = a_dis100, by = c("Code", "description"), all = TRUE) a_dismed02 &gt;- merge(x = a_dismed01, y = a_med100, by = c("Type_med", "Coded_med"), all = TRUE) fwrite(a_dismed02, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_byoverall_dismed_comb_Rcal.csv") saveRDS (a_dismed02, "D:/Hospital_data/ProgresSQL/analysis/080_medicine_byoverall_dismed_comb_Rcal.rds") ### # End of program ##

258 | P a g e 6.5.31 Analysis program for Figure 3-28 Refer to R program: 080_medicine_repeat_prop.R 6.5.32 Analysis program for Figure 3-29 Refer to R program: 080_medicine_repeat_prop.R 6.5.33 Analysis program for Figure 3-30 R program: diagnosis.R ################################ # Code to generate DIAGNOSIS data ################################# library(data.table) setwd ("C:\\Users\\Lucky\\Documents\\Hospital data\\01_31JUL2016\\source") diag &gt;- fread("Diagnosis.csv", check.names = FALSE) diag &gt;- diag[, `:=` (data = "diag", type = substr(`Patient Id`, 1, 2)),] diag2 &gt;- diag[, c(1, 4, 5, 6, 8, 9, 10, 23, 29, 34, 44, 118, 119), with=FALSE]

259 | P a g e setnames(diag2, "MR No.", "MRNo") setnames(diag2, "Admission Date", "visdate") setnames(diag2, "Blood Group", "Bloodgrp") setnames(diag2, "Age In", "AgeIn") setnames(diag2, "First Visit Date", "fvisdate") #setnames(diag2, "Diagnosis Type", "diagtype") # Create date variables and find the difference diag2 &gt;- diag2[, visdate := as.POSIXct(gsub("-", "/", visdate), format="%d/%m/%Y")] diag2 &gt;- diag2[, fvisdate := as.POSIXct(gsub("-", "/", fvisdate), format="%d/%m/%Y")] diag2 &gt;- diag2[, visday := as.Date(visdate) - as.Date(fvisdate) + 1] diag2 &gt;- diag2[, vismon := round(visday /30.4375, 1)] diag2 &gt;- diag2[, Age := ifelse(AgeIn =="M", Age/12, Age),] diag2 &gt;- diag2[, AgeIn := ifelse(AgeIn =="M", "Y", AgeIn),] # Code ** NOT YET CODED for the missing code values # Sort the data by patient and day diag2 &gt;- diag2[, Code2 := ifelse(Code =="", "ZZZ999", Code),] [order(MRNo, visday, Code2)]

260 | P a g e # Create number of diagnosis per patient and a counter for each diagnosis diag2 &gt;- diag2[, noofdis := uniqueN(Code2), by = .(MRNo)] # Count number of unique diagnosis per patient per day # Sort the data by patient and day diag2 &gt;- diag2[, `:=` (IDX = 1: .N), by = .(MRNo, visdate, visday)] [order(MRNo, visday, IDX, Code2)] # Get the first date of the diagnosis by each code diag2 &gt;- diag2 [, min := min(visday), by = .(MRNo, Code2)] # Get the maximum date of the diagnosis (end date for each patient) # Use this data with vital sign data to get diagnosis attached to vital sign measurements diag2 &gt;- diag2 [, max := max(visday), by = MRNo] setkeyv (diag2, c("MRNo", "Code2", "Description", "min", "max", "type", "IDX", "noofdis")) diag2 &gt;- unique(diag2) write.csv(diag2, file ="C:\\Users\\mahajvi1\\Desktop\\adiag.csv", na=" ") ##

261 | P a g e # End of program ## 6.5.34 Analysis program for Figure 3-31 R program: 305_medicine_duration_by_dis.R ### # Medicine duration # Medicine duration by disease # Summary statistics should show the most frequently used medicines # Indirect relationship builiding # More usage stronger the relationship # Less usage may be no relationship or rare usage ### library(data.table) library(tidyverse) library(sqldf)

86% **MATCHING BLOCK 42/51** W

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd &gt;- all_met_rmsd [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] 262 | P a g e all_met_rmsd &gt;- all_met_rmsd [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd &gt;- all_met_rmsd [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd &gt;- all_met_rmsd [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] med01 &gt;- unique(all_met_rmsd [Med02 != "NA:NA" , c("mr_no", "

Med02", "Type_med", "Coded_med", "studyday", "frequency", "duration", "duration_units", "cat_id", "patient_gender"),]) med01 &gt;- med01 [duration &lt; 0] med01 &gt;- med01 [, duration := as.numeric(duration),] med01 &gt;- med01 [, numdays := case_when(duration_units == "D" ~ duration, duration_units == "W" ~ duration * 7, duration_units == "M" ~ duration * 30),] # Create 1 record per patient per medication with sum of durations med011 &gt;- med01 [, .(numdays = sum(numdays)), by =.(mr_no, Med02, Type_med, Coded_med, patient_gender)] med02 &gt;- med011 [, .(n=uniqueN(mr_no), mean = round(mean(numdays, na.rm = TRUE), digits =1), 263 | P a g e median= round(median(numdays, na.rm = TRUE), digits =2), SD = round(sd(numdays, na.rm = TRUE), digits =2), min = round(min(numdays, na.rm = TRUE), digits =0), max = round(max(numdays, na.rm = TRUE), digits =0), sum = round(sum(numdays, na.rm = TRUE), digits =0)), by = .(Type_med)] med02_med &gt;- med011 [, .(n=uniqueN(mr_no), mean = round(mean(numdays, na.rm = TRUE), digits =1), median= round(median(numdays, na.rm = TRUE), digits =2), SD = round(sd(numdays, na.rm = TRUE), digits =2), min = round(min(numdays, na.rm = TRUE), digits =0), max = round(max(numdays, na.rm = TRUE), digits =0), sum = round(sum(numdays, na.rm = TRUE), digits =0)), by = .(Type_med, Med02)] dismed01 &gt;- unique(all_met_rmsd [Med02 != "NA:NA" , c("mr_no", "Med02", "Type_med", "Coded_med", "studyday", "Code02",
264 | P a g e "frequency", "duration", "duration_units", "cat_id", "patient_gender"),]) dismed01 &gt;- dismed01 [duration &lt; 0] dismed01 &gt;- dismed01 [, duration := as.numeric(duration),] dismed01 &gt;- dismed01 [, numdays := case_when(duration_units == "D" ~ duration, duration_units == "W" ~ duration * 7, duration_units == "M" ~ duration * 30),] # Create 1 record per patient per medication with sum of durations dismed011 &gt;- dismed01 [, .(numdays = sum(numdays)), by =.(mr_no, Code02, Med02, Type_med, Coded_med, patient_gender)] # Create count of patients with # totpatdis: total patients having the disease # totpatmed: total patients having the medicine prescribed dismed011 &gt;- dismed01 [, totpatdis := uniqueN(mr_no), by =.(Code02)] dismed011 &gt;- dismed01 [, totpatmed := uniqueN(mr_no), by =.(Med02)] dismed02 &gt;- dismed011 [, .(n=uniqueN(mr_no), mean = round(mean(numdays, na.rm = TRUE), digits =1),
265 | P a g e median= round(median(numdays, na.rm = TRUE), digits =2), SD = round(sd(numdays, na.rm = TRUE), digits =2), min = round(min(numdays, na.rm = TRUE), digits =0), max = round(max(numdays, na.rm = TRUE), digits =0), sum = round(sum(numdays, na.rm = TRUE), digits =0)), by = .(Type_med, Code02, totpatdis, totpatmed)] dismed02_med &gt;- dismed011 [, .(n=uniqueN(mr_no), mean = round(mean(numdays, na.rm = TRUE), digits =1), median= round(median(numdays, na.rm = TRUE), digits =2), SD = round(sd(numdays, na.rm = TRUE), digits =2), min = round(min(numdays, na.rm = TRUE), digits =0), max = round(max(numdays, na.rm = TRUE), digits =0), sum = round(sum(numdays, na.rm = TRUE), digits =0)), by = .(Code02, totpatdis, Type_med, Med02, totpatmed)] # n: total number of patients having the disease and medicine prescribed # So n and totpatmed: calculate the % dismed02_med &gt;- dismed02_med [, perc := round (n / totpatmed * 100, digits = 2),]
##
266 | P a g e # End of program
6.5.35 Analysis program for Figure 3-32 Refer to R program: 305_medicine_duration_by_dis.R 6.5.36 Analysis program for Figure 3-33 Refer to R program: 305_medicine_duration_by_dis.R 6.5.37 Analysis program for Figure 3-34 Refer to R program: 305_medicine_duration_by_dis.R 6.5.38 Analysis program for Figure 3-35 Refer to R program: 100_adsl.R 6.5.39 Analysis program for Figure 3-36 Refer to R program: 100_adsl.R 6.5.40 Analysis program for Figure 3-37 R program: 085_dis_counts_edges_3rdbyPeriod.R
This is used for 085_dis_count_edges_3rd_byPeriod Tableau display
library(data.table) library(stringi) library(stringr)
267 | P a g e library(sqldf) library(tidyr)
These 2 are created using 085_dis_1st_time_refCal_NodeEdges.R program
###

92% **MATCHING BLOCK 43/51** W

all_met_rmsd02 &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Coded_med := paste(Type_med, Coded_med, sep=":"),] edges &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/085_dis_1st_time_refCal_NodesEdges.rds") ### # Get unique diseases in RMSD and Metabolic # Keep refcode from these 2 areas 107 unique values ### discat &gt;- unique(all_met_rmsd02 [distype %in% c("RMSD", "Metabolic"), c("Code", "description"),]) ### # Get the unique number of reference diseases and medicines 268 | P a g e # Create this for each of the periods before and after 1st # occurrence of the disease ### unqref &gt;- unique(edges [, c("period", "periodn")]) dismed &gt;- unique(edges [, c("cat", "refcode", "refdesc", "Code", "description"),]) dismed &gt;- dismed [nchar(Code) &lt; 0] [order(cat, refcode, refdesc, Code, description)] dismed &gt;- dismed [, `:=` (npoints = 1:.N, tot = .N), by = .(cat, refcode, refdesc)] dismed &gt;- dismed [, `:=` (radius = ifelse (cat == "Disease", 20, 40), angle = 360 / tot),] dismed &gt;- dismed [, cumulative := cumsum(angle), by = .(cat, refcode, refdesc)] ### # Function for the degrees and radian conversion ### deg2rad &gt;- function(deg) {(deg * pi) / (180)} dismed &gt;- dismed [, radian := deg2rad(cumulative),] dismed &gt;- dismed [, `:=` (xaxis = cos(radian)*radius, yaxis = sin(radian)*radius),] 269 | P a g e #dismed &gt;- dismed [, Code02 := paste(Code, ":", description),] #dismed &gt;- dismed [, cnt := 0,] ## # create a complete dataset # this is to ensure, circle is displayed all the time # Combine with the individual period for replication ## dismed_all &gt;- crossing(unqref, dismed) chk01 &gt;- all_met_rmsd02 [, .(cnt = uniqueN(mr_no)), by = .(period, periodn, refcode, refdesc, Code, description, Type_med, Coded_med)] chk01 &gt;- chk01[Code != "" & Coded_med != ""] chk01 &gt;- chk01 [, Code02 := paste(Code, ":", description, "-&lt;", Coded_med, sep =""),] # Merge the x and y coordiantes chk02dis &gt;- unique(chk01 [, c("period","periodn", "refcode", "refdesc", "Code", "description", "cnt", "Code02")]) 270 | P a g e chk02dis &gt;- chk02dis [, cat := "Disease"] chk02med &gt;- unique(chk01 [, c("period","periodn", "refcode", "refdesc", "Type_med", "Coded_med", "cnt", "Code02")]) setnames (chk02med, "Type_med", "Code") setnames (chk02med, "Coded_med", "description") chk02med &gt;- chk02med [, cat := "Medicine"] chk02all &gt;- rbind(chk02dis, chk02med) chk02all &gt;- chk02all [nchar(Code) &lt;0 & nchar(description) &lt; 0] path01 &gt;- merge (x = chk02all, y = dismed_all, by = c("cat", "period","periodn", "refcode", "refdesc", "Code", "description"), all = TRUE) path01 &gt;- path01 [, `:=` (cat = "DiseaseMedicine", Code = Code02),] path01 &gt;- path01 [, c("TabCode", "TabMed") := tstrsplit(Code, "-&lt;", fixed = TRUE),] chk03all &gt;- rbind(path01, dismed_all, fill = TRUE) chk04all &gt;- chk03all [refcode %in% discat$Code] 271 | P a g e fwrite(chk04all, "D:/Hospital_data/ProgresSQL/analysis/085_dis_count_edges_3rd_byPeriod.csv") ### ##### 6.5.41 Analysis program for Figure 3-38 Refer to R program: 085_dis_counts_edges_3rdbyPeriod.R 6.5.42 Analysis program for Figure 3-39 R program: 086time_dis_patterns_combinations_gender_Macro.R ### # 086time_dis_patterns_combinations_gender_Macro.R ### library(tidyverse) library(tidytext) #library(stringr) library(stringi) library(data.table) library(stringdist) library(scales) 272 | P a g e # https://stackoverflow.com/questions/43706729/expand-dates-in-data-table dis &gt;- fread("D:/Hospital_data/ProgresSQL/analysis/discategory.csv") setnames (dis, "Code", "refcode")

88% **MATCHING BLOCK 44/51** W

all_met_rmsd &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") ## #

Find patients with only the disease # same as reference disease # 1 = patients with only disease # 99 = patients with more than 1 disease in # a reference disease category ## addmr &gt;- unique(all_met_rmsd [!Code %in% c(" ", ""), c("mr_no", "refcode", "Code", "distype"),]) addmr &gt;- addmr [, cnt := uniqueN(refcode), by = .(mr_no)] addmr &gt;- addmr [, dis := ifelse(refcode == Code, 1, 0),] addmr &gt;- addmr [, calc := ifelse(cnt == 1 & dis == 1, 1, 99),] 273 | P a g e addmr02 &gt;- addmr [, .(cntr = uniqueN(mr_no)), by = .(distype, refcode, Code, calc)] addmr03 &gt;- addmr [, .(cntot = uniqueN(mr_no)), by = .(refcode)] addmr04 &gt;- merge(addmr02, addmr03, by = c("refcode")) addmr04 &gt;- addmr04 [, perc := percent(cntr / cntot),] addmr05 &gt;- addmr04 [refcode == Code] addmr06 &gt;- merge (addmr05, dis, by = c("refcode"), all.y = TRUE) unq &gt;- unique(addmr06 [cntot &lt; 5, c("refcode"),]) unqdis &gt;- unique(unq$refcode) count &gt;- 1 for (dis in unqdis[1:uniqueN(unqdis)]) { print (dis) print (count) a2 &gt;- all_met_rmsd [!Code %in% c("", " ", dis) & refcode == dis] a2 &gt;- a2 [, Code := paste(period, Code, sep="_"),] 274 | Page #a2med &gt;- a2 [, description := paste(Type_med, Coded_med),] #a2med &gt;- a2med [order(description)] #a2med &gt;- a2med [, Code := paste("M", str_pad(.N, width =4, pad="0"), sep =""), by = .(description)] #a2all &gt;- rbind(a2 [, c("mr_no", "studyday", "refday", "Code", "description", "refcode", "refdesc", "patient_gender")], # a2med [, c("mr_no", "studyday", "refday", "Code", "description", "refcode", "refdesc", "patient_gender")]) # Change a2 to a2all dis &gt;- unique(a2[, c("mr_no", "studyday", "refday", "Code", "description", "refcode", "refdesc", "patient_gender")]) dis &gt;- dis [, `:=` (refday2 = ifelse(refday &lt;=1, "After", "Before"), Code = str_replace_all (Code, " ", ""), description = str_replace_all(description, " ", "")),] dis &gt;- dis [order(mr_no, studyday, Code, refcode, refdesc)] dis &gt;- dis [, `:=` (alldis = uniqueN(Code), nrow = seq_len(.N), nrowend = seq_len(.N) + 4, totrow = .N), by = .(mr_no, refcode, refdesc)]

275 | Page dis &gt;- dis [, `:=` (alldisbfraftr = uniqueN(Code), nrowbfraftr = seq_len(.N)), by = .(mr_no, refcode, refdesc, refday2, patient_gender)] dis &gt;- dis [, `:=` (total = uniqueN(mr_no)), by = .(refcode, refdesc, refday2, patient_gender)] dis &gt;- dis [, `:=` (allcapn = uniqueN(mr_no)), by = .(refcode, refdesc, patient_gender)] dis02 &gt;- dis [, .(combdis = paste(unique(Code), collapse = ",", sep = " "), combdesc = paste(unique(description), collapse = ",", sep = " ")), by = .(mr_no, refcode, refdesc, refday2, patient_gender, alldis, totrow, total, allcapn)] unq01comb &gt;- unique(dis02 [, c("mr_no", "refcode", "refdesc", "alldis", "refday2","patient_gender", "totrow", "combdis", "combdesc", "total", "allcapn"),]) unq01comb &gt;- unq01comb [, x := 1,] # create a copy unq02comb &gt;- copy(unq01comb) setnames(unq02comb, "mr_no", "mr_no2")

276 | Page setnames(unq02comb, "combdis", "combdis2") unq01comb &gt;- unq01comb [, combdis := str_replace_all(combdis, ",", "|"),] # Merge the datasets on x to get all the combinations unq03comb &gt;- merge(x = unq01comb, y = unq02comb [, -c("refcode", "refdesc", "totrow", "alldis", "total", "allcapn", "combdesc"),], by = c("x", "refday2", "patient_gender"), allow.cartesian = TRUE) ## # Using str_count function to count the common diseases # Create tempdis and tempdis2 # # Consider mr_no as the reference patient # tempdis: should be lookup # a: common in both the strings # b: only present in reference patient (mr_no)

277 | Page # c: only present in other patient (mr_no2) # d: complete absence -- not sure how to calculate this ## unq03comb &gt;- unq03comb [, `:=` (tempdis = str_replace_all(combdis, ",", "|"), tempdis2 = str_replace_all(combdis2, ",", "|")),] unq03comb &gt;- unq03comb [, `:=` (cntdis = str_count(tempdis, "\\|") + 1, cntdis2 = str_count(tempdis2, "\\|") + 1),] unq03comb &gt;- unq03comb[, `:=` (a = str_count(combdis2, tempdis)),] unq03comb &gt;- unq03comb [, `:=` (b = cntdis - a, c = cntdis2 - a),] unq03comb &gt;- unq03comb[, `:=` (a01jac = (a / (a + b + c)), a02dice = (2 * a / (2* a + b + c)), a03CZEKANOWSKI = (2 * a / (2* a + b + c)), a04jac3w = (3 * a / (3* a + b + c)), a05nei_li = (2 * a / (a + b + a + c)),

278 | Page a06sokalsneath1 = (a / (a + 2 * b + 2 * c))),] unq03comb &gt;- unq03comb [mr_no != mr_no2] maxscr &gt;- unq03comb[, .(maxscr = max(a01jac)), by = .(mr_no, refcode, total, allcapn, totrow, alldis, refday2, patient_gender, combdis, combdesc)] maxscr_t &gt;- dcast (data = maxscr, mr_no + patient_gender + refcode + totrow + alldis ~ refday2, value.var = c("maxscr", "combdis", "combdesc")) maxscr02 &gt;- maxscr [, .(scr = uniqueN(mr_no)), by = .(refcode, total, allcapn, refday2, patient_gender, cut(maxscr, seq(0, 1, .25), include.lowest = TRUE, ordered_result = TRUE))] maxscr02_t &gt;- dcast(data = maxscr02, refcode + patient_gender + allcapn + cut ~ refday2, value.var = c("scr", "total"))

279 | Page maxscr03 &gt;- maxscr [, .(scr = uniqueN(mr_no)), by = .(refcode, total, allcapn, maxscr, combdis, combdesc, refday2, patient_gender)] maxscr03_t &gt;- dcast(data = maxscr03, refcode + patient_gender + allcapn + combdis + combdesc + maxscr ~ refday2, value.var = c("scr", "total")) maxscr04_t &gt;- unq03comb [, .(scr = .N), by = .(mr_no, refcode, total, allcapn, refday2, patient_gender, cut(a01jac, seq(0, 1, .25), include.lowest = TRUE, ordered_result = TRUE))] maxscr04_t &gt;- maxscr04_t [, numrow := .N, by = . (mr_no, refcode, refday2, patient_gender)] totscr &gt;- unq03comb [, .(rowcnt = .N), by = .(refcode, total, allcapn, refday2, patient_gender, cut(a01jac, seq(0, 1, .25), include.lowest = TRUE,

280 | Page ordered_result = TRUE))] totscr02 &gt;- unq03comb [, .(totn = .N), by = .(refcode, refday2, patient_gender, total, allcapn)] totscr02 &gt;- merge (totscr, totscr02, by = c("refcode", "refday2", "patient_gender", "total", "allcapn")) totscr02 &gt;- totscr02 [, perc := percent(rowcnt / totn),] totscr02_t &gt;- dcast(data = totscr02, refcode + patient_gender + allcapn + cut ~ refday2, value.var = c("perc", "totn", "rowcnt", "total")) assign (paste("D01maxscr_t", count, sep="") , maxscr_t) assign (paste("D02maxscr02_t", count, sep="") , maxscr02_t) assign (paste("D03maxscr03_t", count, sep="") , maxscr03_t) assign (paste("D03maxscr04_t", count, sep="") , maxscr04_t) assign (paste("t02totscr02_t", count, sep="") , totscr02_t) count = count + 1 }

281 | Page allD01maxscr_t &gt;- rbindlist(mget(ls(pattern = "D01maxscr_t*")), fill = TRUE) allD02maxscr02_t &gt;- rbindlist(mget(ls(pattern = "D02maxscr02_t*")), fill = TRUE) allD03maxscr03_t &gt;- rbindlist(mget(ls(pattern = "D03maxscr03_t*")), fill = TRUE) allD03maxscr04_t &gt;- rbindlist(mget(ls(pattern = "D03maxscr04_t*")), fill = TRUE) allt02maxscr02_t &gt;- rbindlist(mget(ls(pattern = "t02totscr02_t*")), fill = TRUE) rm(list = ls(pattern='^D01maxscr_t*')) rm(list = ls(pattern='^D02maxscr02_t*')) rm(list = ls(pattern='^D03maxscr03_t*')) rm(list = ls(pattern='^D03maxscr04_t*')) rm(list = ls(pattern='^t02totscr02_t*')) fwrite(allD01maxscr_t, "D:/Hospital_data/ProgresSQL/analysis/086time_dis_indPat_max.csv") fwrite(allD02maxscr02_t, "D:/Hospital_data/ProgresSQL/analysis/086time_dis_refcode_max.csv") fwrite(allD03maxscr03_t, "D:/Hospital_data/ProgresSQL/analysis/086time_dis_indtrajectory.csv") fwrite(allD03maxscr04_t, "D:/Hospital_data/ProgresSQL/analysis/086time_dis_indPat_freqcat.csv")

```
282 | P a g e fwrite(allt02maxscr02_t, "D:/Hospital_data/ProgresSQL/analysis/086time_dis_refcode_allfreq_max.csv")
################################################################################
#### # End of program
################################################################################
#### 6.5.43 Analysis program for Figure 3-40 R program: 086_med_patterns_combinations.R
######################################################################## #
086_med_patterns_combinations
######################################################################## library(tidyverse)
library(tidytext) library(stringr) library(stringi) library(data.table) library(stringdist)
```

82% MATCHING BLOCK 45/51 W

```
all_met_rmsd &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") a2 &gt;- all_met_rmsd [
```

```
refcode == "A2.0" & Coded_med != " "]
283 | P a g e a2 &gt;- a2 [, description := paste(Type_med, Coded_med),] a2 &gt;- a2 [ order(description)] a2 &gt;- a2 [, Code :=
paste("M", str_pad(.N, width =4, pad="0"), sep =""), by = .(description) ] dis &gt;- unique(a2[!Code %in% c("", " ", "A2.0") & refcode ==
"A2.0", c("mr_no", "studyday", "refday", "Code", "description", "refcode", "refdesc")]) dis &gt;- dis [, `:=` (refday2 = ifelse(refday &lt;=1,
"After", "Before"), Code = str_replace_all (Code, " ", ""), description = str_replace_all(description, " ", "")),] dis &gt;- dis [ order(mr_no,
studyday, Code, refcode, refdesc)] dis &gt;- dis [, `:=` (alldis = uniqueN(Code), nrow = seq_len(.N), nrowend = seq_len(.N) + 4, totrow
= .N), by = .(mr_no, refcode, refdesc)] dis &gt;- dis [, `:=` (alldisbfraftr = uniqueN(Code), nrowbfraftr = seq_len(.N) ), by = .(mr_no,
refcode, refdesc, refday2)] dis02 &gt;- dis [, .(combdis = paste(unique(Code), collapse = ",", sep = " " )), by = .(mr_no, refcode, refdesc,
refday2, alldis, totrow)]
284 | P a g e unq01comb &gt;- unique( dis02 [, c("mr_no", "refcode", "refdesc", "alldis", "refday2", "totrow", "combdis"), ]) unq01comb
&gt;- unq01comb [, x := 1, ] # create a copy unq02comb &gt;- copy(unq01comb) setnames(unq02comb, "mr_no", "mr_no2")
setnames(unq02comb, "combdis", "combdis2") unq01comb &gt;- unq01comb [, combdis := str_replace_all(combdis, ",", "|"), ] # Merge
the datasets on x to get all the combinations unq03comb &gt;- merge(x = unq01comb, y = unq02comb [, -c("refcode", "refdesc",
"totrow", "alldis"), ], by = c("x", "refday2"), allow.cartesian = TRUE)
########################################################
285 | P a g e # Using str_count function to count the common diseases # Create tempdis and tempdis2 # # Consider mr_no as the
reference patient # tempdis: should be lookup # a: common in both the strings # b: only present in reference patient (mr_no) # c:
only present in other patient (mr_no2) # d: complete absence -- not sure how to calculate this
######################################################## unq03comb &gt;- unq03comb [, `:=`
(tempdis = str_replace_all(combdis, ",", "|"), tempdis2 = str_replace_all(combdis2, ",", "|")),] unq03comb &gt;- unq03comb [, `:=`
(cntdis = str_count(tempdis, "\\|") + 1, cntdis2 = str_count(tempdis2, "\\|") + 1), ] unq03comb &gt;- unq03comb[, `:=` (a =
str_count(combdis2, tempdis)),] unq03comb &gt;- unq03comb [, `:=` (b = cntdis - a, c = cntdis2 - a), ]
286 | P a g e unq03comb &gt;- unq03comb[, a01jac := (a / (a + b + c)),] distraj &gt;- unique(unq03comb [tempdis != tempdis2 ,
c("tempdis", "tempdis2", "a01jac", "refday2"),]) distraj01 &gt;- distraj [, .(distraj = .N), by = .(refday2, a01jac)] common &gt;- unq03comb [,
.(cmn = (.N / nrow(unq03comb)) * 100), by = .(a)]
################################################################################
#### # End of program
################################################################################
#### 6.5.44 Analysis program for Figure 3-41 R program: 300_radar_plot_tableu.R # Tableu help # Use
https://www.tableau.com/about/blog/2015/7/use-radar-charts-compare-dimensions-over- several-metrics-41592 library(data.table)
library(tidyverse) library(sqldf) # Install the ggradar library #devtools::install_github("ricardo-bion/ggradar", dependencies = TRUE)
287 | P a g e library(ggradar)
```

100% MATCHING BLOCK 46/51 W

```
all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd &gt;- all_met_rmsd [,
Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd &gt;- all_met_rmsd [, description:= ifelse
(description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd &gt;- all_met_rmsd [, Code02 := paste(distype,
":", Code, ":", description, sep =""), ] all_met_rmsd &gt;- all_met_rmsd [, Med02 := paste(Type_med, ":", Coded_med, sep =""), ]
all_met_rmsd02 &gt;-
```

97% MATCHING BLOCK 47/51 W

```
readRDS("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse
(Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == ""
| description ==" ", "** Not yet coded", description),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Code02 := paste(distype, ":", Code, ":",
description, sep =""), ] all_met_rmsd02 &gt;- all_met_rmsd02 [, Med02 := paste(Type_med, ":", Coded_med, sep =""), ] #
```

Disease: # (1) Distinct number of patients
288 | P a g e t10 &gt;- all_met_rmsd02 [, .(cal10 = uniqueN(mr_no)), by =.(refcode, refdesc)] t10 &gt;- t10 [, perc10 := as.numeric(ntile(cal10, 100)),] # (2) Number of times a disease is reported totdis &gt;- unique(all_met_rmsd [, c("Code", "description", "patient_id"),]) t20 &gt;- totdis [, .(cal20 = .N), by =.(Code, description)] t20 &gt;- t20 [, perc20 := as.numeric(ntile(cal20, 100)),] # (3) Number for a specific disease (chronological number of disease reported by a patient) # Calculate median number of disease reported for each disease # Then calculate the percentile for each disease numdis &gt;- unique(all_met_rmsd [, c("mr_no", "Code", "description", "studyday"),]) numdis &gt;- numdis [order(mr_no, studyday, Code)] numdis &gt;- numdis [, `:=`(COUNT = .N , ndis = 1:.N), by = .(mr_no, Code, description)] t30 &gt;- numdis [, .(cal30 = median(ndis)), by =.(Code, description)]
289 | P a g e t30 &gt;- t30 [, perc30 := as.numeric(ntile(cal30, 100)),] # (4) Number of diseases before and after the specific disease banumdis &gt;- unique(all_met_rmsd02 [, c("mr_no", "Code", "description", "period", "periodn", "refcode", "refdesc"),]) banumdis &gt;- banumdis [, classification := ifelse (period &lt;=1 , "After", "Before"),] banumdis &gt;- banumdis [order(mr_no, refcode, refdesc, classification)] banumdis &gt;- banumdis [, `:=`(ndis = uniqueN(Code)), by = .(mr_no, refcode, refdesc, classification)] t40 &gt;- banumdis [, .(cal40 = median(ndis)), by =.(refcode, refdesc, classification)] t40 &gt;- t40 [, perc40 := as.numeric(ntile(cal40, 100)),] t40_trn &gt;- dcast(data = t40, refcode + refdesc ~ classification, value.var = c("cal40", "perc40"), fill ="0") # (5) Number of treatments before and after the specific disease banummed &gt;- unique(all_met_rmsd02 [, c("mr_no", "Med02", "period", "periodn", "refcode", "refdesc"),]) banummed &gt;- banummed [, classification := ifelse (period &lt;=1 , "After", "Before"),]
290 | P a g e banummed &gt;- banummed [order(mr_no, refcode, refdesc, classification)] banummed &gt;- banummed [, `:=`(nmed = uniqueN(Med02)), by = .(mr_no, refcode, refdesc, classification)] t50 &gt;- banummed [, .(cal50 = median(nmed)), by =.(refcode, refdesc, classification)] t50 &gt;- t50 [, perc50 := as.numeric(ntile(cal50, 100)),] t50_trn &gt;- dcast(data = t50, refcode + refdesc ~ classification, value.var = c("cal50", "perc50"), fill ="0") #setnames(t10, "Code", "refcode") setnames(t20, "Code", "refcode") setnames(t30, "Code", "refcode") #setnames(t10, "description", "refdesc") setnames(t20, "description", "refdesc") setnames(t30, "description", "refdesc") all01 &gt;- Reduce(function(...) merge(..., all.x = TRUE, by = c("refcode", "refdesc")), list(t40_trn, t10, t20, t30, t50_trn))
291 | P a g e all01 &gt;- all01 [refcode != "sandhigata vaa"] all01_trn &gt;- melt (data = all01, id.vars = c("refcode", "refdesc"), measure.vars = c("perc10", "perc20", "perc30", "perc40_After", "perc40_Before", "perc50_After", "perc50_Before")) all01_trn &gt;- as.data.table (sqldf("select *, case When variable == 'perc10' then '1 Unique patients' when variable == 'perc20' then '2 no of times disease reported' when variable == 'perc30' then '3 disease chronology' when variable == 'perc40_After' then '4 no of diseases before' when variable == 'perc40_Before' then '5 no of diseases after' when variable == 'perc50_After' then '6 no of medicines before'
292 | P a g e when variable == 'perc50_Before' then '7 no of medicines after' end as category from all01_trn")) # Possible background creation within Tableau all01_trn &gt;- all01_trn [, valuedumm := 100,] # Used for the tableau visual fwrite(all01_trn, file="D:/Hospital_data/ProgresSQL/analysis/300_radar_plot.csv") ## # End of program ## 6.5.45 Analysis program for Figure 3-42 R program: 06_d3tree_diagram.R ###################### # This version is for m / f # Use 04dis_gender_

54% MATCHING BLOCK 48/51 W

csv folder ##################### all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") 293 | P a g e all_met_rmsd &gt;- all_met_rmsd [, Code := str_replace_all(Code, "\\.", "_")] cnt&gt;- unique(all_met_rmsd [

patient_gender != "" & Code != "", c("mr_no", "studyday","Code", "description","distype", "patient_gender"),]) cnt &gt;- cnt [, `:=` (mnth = round(studyday /30.25, digits = 0), description = paste("[", trimws(distype), ": ", trimws(description), "]", sep = ""), Code = paste("[", trimws(Code), "]", sep=""))] cntrow &gt;- cnt [, .(permnth = uniqueN(Code)), by =.(mr_no, mnth)] cnt &gt;- cnt [order(mr_no, studyday, Code, description, patient_gender)] cnt2 &gt;- unique(cnt [, c("mr_no", "Code", "description", "patient_gender", "distype"),]) # Combinations for each patient # Do these calculations for first rows cnt3 &gt;- cnt2[, `:=` (numcomb = seq_len(.N), descomb = description, discomb = Code, grpcomb = paste(trimws(Code), collapse = " ", sep=)), by = .(mr_no, patient_gender)]
294 | P a g e cnt30 &gt;- cnt3 [numcomb &lt; 1, `:=` (discomb = sapply(seq_len(.N), function(x) paste(Code[seq_len(x)], collapse = "&lt;")), descomb = sapply(seq_len(.N), function(x) paste(description[seq_len(x)], collapse = "&lt;"))), by = .(mr_no, patient_gender)] cnt31 &gt;- rbind(cnt3 [numcomb ==1], cnt30 [numcomb &lt; 1]) # Starting disease sttdis stt &gt;- cnt3 [numcomb == 1, .(sttdis = paste(descomb, "&lt;", patient_gender, sep="")), by =.(mr_no, patient_gender, Code, description)] cnt3disprgs &gt;- merge(cnt31, stt [, c("mr_no", "sttdis"),], by = c("mr_no")) cnt3disprgs &gt;- cnt3disprgs [, .(npt = uniqueN(mr_no)), by = .(discomb, descomb, numcomb, grpcomb, sttdis, patient_gender)] cnt3disprgs &gt;- cnt3disprgs [order(sttdis, patient_gender, numcomb, discomb, grpcomb)] cnt3disprgs &gt;- cnt3disprgs [, node := 1:.N, by =.(sttdis, patient_gender, grpcomb, npt)] cnt3disprgs &gt;- cnt3disprgs [, treecomb := paste(sttdis, "&lt;", descomb, " (N=", npt, ")", sep="")] cnt3disprgs &gt;- cnt3disprgs [order(sttdis, grpcomb, node)] cnt3disprgs02 &gt;- cnt3disprgs [numcomb &lt; 1]

295 | P a g e # These 2 subsets are for the CSV for D3js sttdis &gt;- unique(stt [, c("description"),]) sttdisgen &gt;- unique(stt [sttdis != "", c("sttdis"),]) frow &gt;- data.table (treecomb = "id,value") # Rename to the same variable setnames(sttdis, "description", "treecomb") setnames(sttdisgen, "sttdis", "treecomb") cnt3disprgs03 &gt;- rbind(cnt3disprgs02 [, c("treecomb")], sttdis, sttdisgen) [order(treecomb)] cnt3disprgs03 &gt;- cnt3disprgs03[, treecomb := paste("Disease&lt;", treecomb, sep="")] # No subset fwrite(unique(cnt3disprgs03), col.names = FALSE, quote = FALSE, "D:\\Hospital_data\\ProgresSQL\\misc\\jsfolder\\999temp\\decode_gender.csv")
###
#
296 | P a g e # End of program
###
Create Json file using the following commands: # This is the working directory path. SimpleJar=Hospital_data/ProgresSQL/misc/jsfolder/999temp java -classpath `cygpath -wp /cygdrive/d/${SimpleJar}:./json-simple-1.1.1.jar` D3Taxonomy decode_gender.csv "&lt;"
###
End of program
###
####

297 | P a g e 7 References [1] M. Masoud Y. Jaradat A. Manasrah Ahmad I. Jannoud, "Sensors of Smart Devices in the Internet of Everything (IoE) Era: Big Opportunities and Massive Doubts," Journal of Sensors, p. 26, 2019. [2] W. R. Bolislis F. Myriam T. C. Kühler, "Use of Real-world Data for New Drug Applications and Line Extensions," Clinical Therapeutics, vol. 42, no. 5, pp. 926-938, 2020. [3] AYUSH Minsitry Government of India, "AYUSH website," [Online]. Available: http://dashboard.ayush.gov.in/. [4] R. H. Singh, "Exploring issues in the development of Ayurvedic research methodology," Journal of Ayurveda and integrative medicine, pp. 91-95, 2010. [5] I. Krakau, "The importance of practice-based evidence," Scandinavian Journal of Primary Health Care, pp. 130-131, 2000. [6] B. Patwardhan, "Envisioning AYUSH: Historic Opportunity for Innovation and Revitalization," Journal of Ayurveda and integrative medicine, pp. 67-70, 2014. [7] G. Tillu TDU hospital staff, "Exploration of the TDU database," 2016 - 2018. [Online]. [8] Pharmaforum, "Pharmaforum, bringing healthcare together," [Online]. Available: https://pharmaphorum.com/views-and-analysis/a_history_of_the_pharmaceutical_industry/. [Accessed 2020]. [9] "THE NUREMBERG CODE [from Trials of War Crimials before the Nuremberg Military Tribunals under Control Council Law] No. 10. Nuremberg October 1946–April 1949 Washington, D.C.: U.S. G.P.O, 1949–1953.". [10] "World Medical Association," [Online]. Available: https://www.wma.net/policies- post/wma-

75% **MATCHING BLOCK 50/51** **W**

declaration-of-helsinki-ethical-principles-for-medical-research-involving- human-subjects/#:~:text=1.,identifiable%20human%20material%20and%20data.. [11] "THE

COUNCIL FOR INTERNATIONAL ORGANIZATIONS OF MEDICAL SCIENCES (CIOMS)," [Online]. Available: https://cioms.ch/#:~:text=The%20Council%20for%20International%20Organizations,W HO%20and%20UNESCO%20in%201949.. 298 | P a g e [12] J. Greene S. H. Podolsky, "Reform, regulation, and pharmaceuticals--the Kefauver- Harris Amendments at 50," The New England journal of medicine, vol. 367, no. 16, p. 1481–1483, 2012. [13] "ICH Official Website," [Online]. Available: https://www.ich.org/. [14] V. Renjith, "Blinding in Randomized Controlled Trials: What researchers need to know?," Manipal Journal of Nursing and Health Sciences, vol. 3, no. 1, pp. 45-50, 2017. [15] D. S. Jones S. H. Podolsky, "The history and fate of the gold standard," The Lancet: The Art of Medicine, vol. 385, no. 9977, pp. 1502-1502, 2015. [16] National Library of Medicine USA, "ClinTrials.gov," [Online]. Available: https://clinicaltrials.gov/. [17] T. Boerma, "Public health information needs in districts," BMC Health Serv Res, vol. S12, 2013. [18] R. S. Evans., "Electronic Health Records: Then, Now, and in the Future," Yearbook of medical informatics, vol. Suppl 1, p. S48–S61, 2016. [19] V. R. Suvarna, "Real world evidence (RWE) - Are we (RWE) ready?," Perspectives in clinical research, vol. 9, no. 2, pp. 61-63, 2018. [20] S. Kamath G. Guyatt, "Importance of evidence-based medicine on research and practice," Indian journal of anaesthesia, vol. 60, no. 9, pp. 622-625, 2016. [21] G. Nahler, "good clinical research practice (GCRP)," in Dictionary of Pharmaceutical Medicine, Vienna, Springer, 2009. [22] J. Concato, "Randomized, Controlled Trials, Observational Studies, and the Hierarchy of Research Designs," New England Journal of Medicine, vol. 342, pp. 1887-1892, 2000. [23] J. Avorn, "In Defense of Pharmacoepidemiology — Embracing the Yin and Yang of Drug Research," New England Journal of Medicine, 2007. [24] A. Anglemyer H. T. Horvath L. Bero, "Healthcare outcomes assessed with observational study designs compared with those assessed in randomized trials," Cochrane Database of Systematic Reviews, 2014. [25] S. V. Ramagopalan A. Simpson C. Sammon, "Can real-world data really replace randomised clinical trials?," BMC Med, vol. 18, no. 13, 2020. [26] S. Rimpilainen, "A Review of Electronic Health Records Systems Around the World," 2015.

299 | P a g e [27] "World Health Organization," [Online]. Available: https://www.who.int/campaigns/world-health-day/2018/campaign-essentials/en/#:~:text=The%20theme%20of%20World%20Health,is%20%E2%80%9CH ealth%20for%20All%E2%80%9D.. [28] C. Lele, "Development of statistics as a discipline for clinical research: Past, present and future. Perspectives in clinical research," Perspectives in Clinical Research, vol. 8, no. 1, pp. 41-44, 2017. [29] S. Kaihara, "Information explosion," World Health, 1989. [30] J. Klerkx K. Verbert E. Duval, "Enhancing Learning with Visualization Techniques," Katholieke Universiteit Leuven, Belgium. [31] H. R. Nagel, "Scientific Visualization versus Information Visualization". [32] B. Chaudhry J. Wang S. Wu et al., "Systematic review: impact of health information technology on quality, efficiency, and costs of medical care," Ann Intern Med, vol. 144, no. 10, pp. 742-752, 2006. [33] C. Safran M. Bloomrosen W. E. Hammond et al., "Toward a National Framework for the Secondary Use of Health Data: An American Medical Informatics Association White Paper," Journal of the American Medical Informatics Association, vol. 14, no. 1, pp. 1-9, 2007. [34] S. Gu J. Pei, "Innovating Chinese Herbal Medicine: From Traditional Health Practice to Scientific Drug Discovery," Front Pharmacol., vol. 381, p. 8, 2017. [35] H Zeng Q Yiqi X Chen et al., "History and Development of TCM Case Report in a Real- World Setting," Evid Based Complement Alternat Med, 29 Dec 2021. [36] L Ai-Ping J Hong-Wei X Cheng et al., "Theory of traditional Chinese medicine and therapeutic method of diseases," World Journal of Gastroenterol, vol. 1854, no. 6, pp. 1- 10, 2004. [37] K. Chan, "Progress in traditional Chinese medicine," Trends Pharmacol Sciences, vol. 16, no. 6, pp. 182-187, June 1995. [38] D. Normile, "Asian medicine: the new face of traditional Chinese medicine," Science, vol. 299, pp. 188-190, 2003. [39] B. Tan J. Vanitha, "Immunomodulatory and antimicrobial effects of some traditional Chinese medicinal herbs: A review," Curr. Med. Chem, vol. 11, p. 1423–1430, 2004.
300 | P a g e [40] W. Y. Jiang, "Therapeutic wisdom in traditional Chinese medicine: a perspective from modern science," Trends Pharmacol. Sci, vol. 26, p. 558–563, 2005. [41] R. Wang H. Yan X. C. Tang, "Progress in studies of huperzine A, a natural cholinesterase inhibitor from Chinese herbal medicine," Acta Pharmacol, vol. 27, pp. 1- 26, 2006. [42] J. Wang X. J. Xiong, "Outcome measures of Chinese Herbal Medicine for Hypertension: an overview of systematic reviews," Evid. Based Complement. Alternat. Med, p. 7, 2012. [43] Y. Tu, "The discovery of artemisinin (qinghaosu) and gifts from Chinese medicine," Nat. Med, vol. 17, p. 1217–1220, 2011. [44] Y Tu, "Artemisinin-A Gift from Traditional Chinese Medicine to the World (Nobel Lecture)," Angew. Chem. Int. Ed. Engl, vol. 55, p. 10210–10226, 2016. [45] U. S. Neil, "From branch to bedside: Youyou Tu is awarded the 2011 Lasker similar to DeBakey Clinical Medical Research Award for discovering artemisinin as a treatment for malaria," J. Clin. Investig, vol. 121, p. 3768–3773, 2011. [46] X. Z. Su L. H. Miller, "The discovery of artemisinin and the Nobel Prize in Physiology or Medicine," Sci. China Life Sci, vol. 58, p. 1175–1179, 2015. [47] CY Chen, "TCM Database@Taiwan: the world's largest traditional Chinese medicine database for drug screening in silico," PLoS One, vol. 6, no. 1, Jan 2011. [48] W Yang F Zhang K Yang et al., "SymMap: an integrative database of traditional Chinese medicine enhanced by symptom mapping," Nucleic Acids Research, vol. 47, no. D1, p. D1110–D1117, 08 January 2019. [49] J Ru P Li J Wang et al., "TCMSP: a database of systems pharmacology for drug discovery from herbal medicines," J Cheminform, vol. 6, no. 13, 2014. [50] L Zhihong C Chuipu D Jiewen et al., "TCMIO: A Comprehensive Database of Traditional Chinese Medicine on Immuno-Oncology," Frontiers in Pharmacology, vol. 11, 2020. [51] R Xue Z Fang M Zhang et al., "TCMID: traditional Chinese medicine integrative database for herb molecular mechanism analysis," Nucleic Acids Research, vol. 41, no. D1, p. D1089–D1095, 01 January 2013. [52] B Li C Ma X Zhao et al., "YaTCM: Yet another Traditional Chinese Medicine Database for Drug Discovery," Computational and Structural Biotechnology Journal, vol. 16, pp. 600-610, 2018.
301 | P a g e [53] L Feng L Kong X Dong et al., "ASES-TCM Protocol Steering Group. China Stroke Registry for Patients With Traditional Chinese Medicine (CASES-TCM): Rationale and Design of a Prospective, Multicenter, Observational Study," Front Pharmacol, vol. 12:743883, 31 Aug 2021. [54] Y Song M Li K Sugimoto et al., "CARE-TCM Group. China amyotrophic lateral sclerosis registry of patients with Traditional Chinese Medicine (CARE-TCM): Rationale and design," J Ethnopharmacol, vol. 284:114774, 10 Feb 2022. [55] J Chen J Huang JV Li et al., "The Characteristics of TCM Clinical Trials: A Systematic Review of ClinicalTrials.gov," Evid Based Complement Alternat Med, 24 Aug 2017. [56] T Zhang Z Huang Y Wang et al., "Information Extraction from the Text Data on Traditional Chinese Medicine: A Review on Tasks, Challenges, and Methods from 2010 to 2021," Evidence-Based Complementary and Alternative Medicine, vol. 2022, p. 19, 2022. [57] X. Sun, "ISPOR West China Chapter," [Online]. Available: https://www.ispor.org/docs/default-source/conference-ap-2018/china-2nd-plenary-for- handouts.pdf?sfvrsn=5fbc7719_0. [Accessed 2022]. [58] "National Medical Products Administration, China," [Online]. Available: http://english.nmpa.gov.cn/2020-01/07/c_456245.htm. [Accessed 2022]. [59] D. Shankar, "Health sector reforms for 21(st) century healthcare," Journal of Ayurveda and integrative medicine, pp. 4-9, 2015. [60] S. Madanian D. T. Parry D. Airehrour et al., "mHealth and big-data integration: promises for healthcare system in India," BMJ Health Care Inform, p. 26, 2019. [61] D. Shankar B. Patwardhan, "AYUSH for New India: Vision and strategy," Journal of Ayurveda and integrative medicine, vol. 8, pp. 137-139, 2017. [62] AYUSH ministry Government of India, "AYUSH Namaste portal," [Online]. Available: http://www.namstp.ayush.gov.in/#/index. [63] AYUSH ministry Government of India, "AYUSH Research Portal," [Online]. Available: http://ayushportal.nic.in/. [64] AYUSH ministry Government of India, "AYUSH Hospital Management System," [Online]. Available: http://ehr.ayush.gov.in/ayush/. [65] AYUSH ministry Government of India, "Ayush Grid," 2nd Oct 2020. [Online]. Available: https://pib.gov.in/PressReleasePage.aspx?PRID=1660936#:~:text=Ayush%20Grid%2C

302 | P a g e %20the%20emerging%20IT,the%20National%20Digital%20Health%20Mission&text= The%20Ayush%20Grid%20project%20was,backbone%20for%20the%20entire%20sect or.. [66] AYUSH ministry Government of India, "AYUSH Dashboard," [Online]. Available: http://dashboard.ayush.gov.in/. [67] GOVERNMENT OF INDIA MINISTRY OF HEALTH AND FAMILY WELFARE, "THE DRUGS AND COSMETICS ACT, 1940," [Online]. Available: http://naco.gov.in/sites/default/files/Drug%20%26%20Cosmetic%20Act%201940_1.pdf . [68] "Centre for development of advanced computing," C-DAC Pune, AAI Group, [Online]. Available: https://www.cdac.in/index.aspx?id=hi_dss_ayusoft_n. [69] Council of Scientific & Industrial Research (CSIR), "Traditional Knowledge Digital Library," [Online]. Available: http://www.tkdl.res.in/tkdl/langdefault/common/Home.asp?GL=Eng. [70] A.B. Vaidya, "Reverse pharmacological correlates of ayurvedic drug actions," Indian J Pharmacol, vol. 38, p. 311–315, 2006. [71] B. Patwardhan, "Bridging Ayurveda with evidence-based scientific approaches in medicine," EPMA Journal, vol. 5, 2014. [72] B. Patwardhan, "Envisioning AYUSH: Historic Opportunity for Innovation and Revitalization," Journal of Ayurveda and integrative medicine, vol. 5, no. 2, pp. 67-70, 2014. [73] S. P. Kulkarni, "HURDLES IN RESEARCH IN AYURVEDA AND THEIR POSSIBLE SOLUTIONS," International Ayurvedic Medical Journal, vol. 3, no. 1, 2015. [74] S. M. Baghel, "Need of new research methodology for Ayurveda," Ayu, vol. 32, no. 1, pp. 3-4, 2011. [75] D. Shankar, "Directions for revitalization of Ayurveda in the 21st century," Journal of Ayurveda and integrative medicine, vol. 9, no. 4, pp. 245-247, 2018. [76] H. Walach T. Falkenberg V. Fønnebø et al., "Circular instead of hierarchical: methodological principles for the evaluation of complex interventions," BMC Med Res Methodol, vol. 29, no. 6, 2006. [77] R. L. Byyny, Spring 2012. [Online]. Available: https://alphaomegaalpha.org/pharos/PDFs/2012-2-Editorial.pdf.
303 | P a g e [78] A. Patrizio, "IDC: Expect 175 zettabytes of data worldwide by 2025," 3rd Dec 2018. [Online]. Available: https://www.networkworld.com/article/3325397/idc-expect-175- zettabytes-of-data-worldwide-by-2025.html. [79] T. Cosgrove, "Consult QD: Dealing with Healthcare's Data Explosion by," [Online]. Available: https://www.google.co.in/amp/s/consultqd.clevelandclinic.org/dealing- healthcares-data-explosion/amp/. [80] K. Adane M. Gizachew S. Kendie, "The role of medical data in efficient patient care delivery: a review," Risk management and healthcare policy, vol. 12, pp. 67-73, 2019. [81] S. Kandel J. Heer C. Plaisant et al., "Research directions in data wrangling: Visualizations and transformations for usable and credible data," Information Visualization, 2011. [82] "The Importance of Data Preparation for Business Analytics," 16th July 2019. [Online]. Available: https://www.ironsidegroup.com/2019/07/16/data-preparation-business- analytics/. [83] "eTutorials.org," [Online]. Available: http://etutorials.org/Misc/data+quality/Part+I+Understanding+Data+Accuracy/. [84] D. Tobin, "Data Transformation: Explained," 1st June 2020. [Online]. Available: https://www.xplenty.com/blog/data-transformation-explained/. [85] Committee on Strategies for Responsible Sharing of Clinical Trial Data, "Board on Health Sciences Policy Institute of Medicine. Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk," in The Clinical Trial Life Cycle and When to Share Data, Washington (DC), National Academies Press (US), 20th Apr 2015. [86] A. Kapoor, "Quality Medical Research and Publications in India: Time to Introspect," International journal of applied & basic medical research, vol. 9, no. 2, pp. 67-68, 2019. [87] J. Tauberer, "Analyzable Data in Open Formats (Principles 5 and 7)," in Open Government Data: The Book, 2014. [88] K. Tarsi T. Tuff, "Introduction to Population Demographics," Nature Education Knowledge, vol. 3, no. 11, p. 3, 2012. [89] E. P. Balogh B. T. Miller J. R. Ball, "Committee on Diagnostic Error in Health Care, Board on Health Care Services, Institute of Medicine, The National Academies of Sciences, Engineering, and Medicine," in Improving Diagnosis in Health Care - The Diagnostic Process, Washington DC, National Academies Press (US), 29th Dec 2015.
304 | P a g e [90] "

100% **MATCHING BLOCK 51/51** W

PostgreSQL: The World's Most Advanced Open Source Relational Database,"

The PostgreSQL Global Development Group, [Online]. Available: https://www.postgresql.org/. [91] The R Core Team, "The Comprehensive R Archive Network," [Online]. Available: https://cran.r-project.org/. [92] "Python," Python Software Foundation, [Online]. Available: https://www.python.org/. [93] "Java," Oracle, [Online]. Available: https://www.java.com/en/. [94] B. Mike, "D3 Data-Driven-Documents," [Online]. Available: https://d3js.org/. [95] "Tableau," [Online]. Available: https://www.tableau.com/. [96] M. G. Larson, "Descriptive Statistics and Graphical Displays," Circulation, vol. 114, no. 1, pp. 76-81, 2006. [97] D. Murray, Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software, 1st ed., Wiley Publishing, 2013. [98] R. McGill J. W. Tukey W. A. Larsen, "Variations of box plots," The American Statistician, vol. 32, no. 1, pp. 12-16, 1978. [99] E. Tufte, The Visual Display of Quantitative Information, 2nd ed., Cheshire, Connecticut: Graphics Press, 2001. [100] D. M. Morales-Silva C. S. McPherson G. Pineda-Villavicencio et al., "Using radar plots for performance benchmarking at patient and hospital levels using an Australian orthopaedics dataset," Health Informatics Journal, pp. 2119-2137, September 2020. [101] B. McPherson, "The code to generate dynamic bubble plot from Github," [Online]. Available: https://gist.github.com/brucemcpherson/4684498. [102] C. DeMartini, "datablick," [Online]. Available: https://www.datablick.com/blog/2018/3/14/layering-data-for-custom-tableau- visualizations. [103] B. McPherson, "Desktop libeartion," [Online]. Available: https://ramblings.mcpher.com/. [104] H. Hofmann, "Mosaic Plots and Their Variants," in Handbook of Data Visualization, Berlin, Heidelberg, Springer Handbooks Comp.Statistics, 2008.

305 | P a g e [105] R. Wicklin, "SAS Blogs," 23 May 2018. [Online]. Available: https://blogs.sas.com/content/iml/2018/05/23/butterfly-plot.html. [106] "Global Oracle cloud Regions," Oracle, 2022. [Online]. Available: https://www.oracle.com/sg/database/what-is-a-data-warehouse/. [107] "Drik Panchang," [Online]. Available: https://www.drikpanchang.com/seasons/season- tropical-timings.html?geoname-id=1277333&year=2010. [108] T. Siggaard R. Reguant I. F. Jørgensen et al., "Disease trajectory browser for exploring temporal, population-wide disease progression patterns in 7.2 million Danish patients," Nat Commun, vol. 11, p. 4952, 2020. [109] "International Statistical Classification of Diseases and Related Health Problems (ICD)," World Health Organization, [Online]. Available: https://www.who.int/standards/classifications/classification-of-diseases. [110] C. Seung-Seok C. Sung-Hyuk C. Charles et al., "A Survey of Binary Similarity and Distance Measures," SYSTEMICS, CYBERNETICS AND INFORMATICS, vol. 8, no. 1, 2010. [111] S-H Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions," INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES, vol. 1, no. 4, pp. 300-307, 2007. [112] "DeepAI," [Online]. Available: https://deepai.org/machine-learning-glossary-and- terms/jaccard-index. [113] National Cancer Institute. [Online]. Available: https://evs.nci.nih.gov/ftp1/CDISC/SDTM/SDTM%20Terminology.pdf. [114] "MedDRA: Medical Dictionary for Regulatory Activities," [Online]. Available: https://www.meddra.org/. [115] "WHO Drug Global," [Online]. Available: https://www.who-umc.org/whodrug/whodrug-portfolio/whodrug-global/. [116] "International Classification of Diseases - ICD," [Online]. Available: https://icd.who.int/browse11/l-m/en. [117] "Logical Observation Identifiers Names and Codes - LOINC," [Online]. Available: https://loinc.org/. [118] "Clinical Data Interchange Standards Consortium: CDISC," [Online]. Available: https://www.cdisc.org/. 306 | P a g e [119] "International Organization for Standardization: ISO," [Online]. Available: https://www.iso.org/home.html. [120] "TransCelerate BioPharma Inc.," [Online]. Available: https://www.transceleratebiopharmainc.com/. [121] "U.S. Food & Drug administration," [Online]. Available: https://www.fda.gov/science- research/science-and-research-special-topics/real-world-evidence. [122] "U.S. Food & Drug administration: Framework Real World Evidence Program," [Online]. Available: https://www.fda.gov/media/120060/download. [123] "European Medicines Agency," [Online]. Available: https://www.ema.europa.eu/en/about-us/how-we-work/big-data/data-analysis-real- world-interrogation-network-darwin-eu.

Hit and source - focused comparison, Side by Side

Submitted text	As student entered the text in the submitted document.
Matching text	As the text appears in the source.

1/51	**SUBMITTED TEXT**	24 WORDS	**52%**	**MATCHING TEXT**	24 WORDS
	The origins of the pharmaceutical industry go back to the apothecaries and pharmacies that gave traditional therapies going back to the Middle Ages.			The roots of the pharmaceutical industry lie back with the apothecaries and pharmacies that offered traditional remedies as far back as the middle ages,	

W https://pharmaphorum.com/views-and-analysis/a_history_of_the_pharmaceutical_industry/

2/51	**SUBMITTED TEXT**	22 WORDS	**55%**	**MATCHING TEXT**	22 WORDS
	the United Nations Educational, Scientific and Cultural Organization (UNESCO) in 1949 jointly established The Council for International Organizations of Medical Sciences (			The United Nations Educational, Scientific and Cultural Organization • International Society for Pharmacoepidemiology (ISPE) • International Society of Pharmacovigilance CIOMS THE COUNCIL FOR INTERNATIONAL ORGANIZATIONS OF MEDICAL SCIENCES 2023	

W https://cioms.ch/#:~:text=The%20Council%20for%20International%20Organizations,WHO%20and%20UNESCO% ...

3/51	**SUBMITTED TEXT**	14 WORDS	**100%**	**MATCHING TEXT**	14 WORDS
	data, like unrefined gold buried deep in a mine is a precious resource.			data—like unrefined gold buried deep in a mine—is a precious resource	

W https://www.xplenty.com/blog/data-transformation-explained/

4/51 **SUBMITTED TEXT** 39 WORDS	**43%** **MATCHING TEXT** 39 WORDS
A “join” is an operation that connects two or more datasets by their matching columns. This establishes a relationship between multiple datasets, which merges data together so a query can be made on the combined data [84]. •	A “join” is an operation in the SQL database language that allows you to connect two or more database tables by their matching columns. This allows you to establish a relationship between multiple tables, which merges table data together so you can query correlating data on the tables. Data

W https://www.xplenty.com/blog/data-transformation-explained/

5/51 **SUBMITTED TEXT** 25 WORDS	**71%** **MATCHING TEXT** 25 WORDS
is often: (1) Inconsistent, containing both relevant and irrelevant data, (2) Imprecise, containing incorrectly entered information or missing values, (3) Repetitive, containing duplicate data.	is often: • Inconsistent: It contains both relevant and irrelevant data. • Imprecise: It contains incorrectly entered information or missing values. • Repetitive: It contains duplicate data.

W https://www.xplenty.com/blog/data-transformation-explained/

6/51 **SUBMITTED TEXT** 18 WORDS	**91%** **MATCHING TEXT** 18 WORDS
The goal of data filtering is to refine a data source to only what the user needs.	The goal of data filtering is to distill a data source to only what the user needs

W https://www.xplenty.com/blog/data-transformation-explained/

7/51 **SUBMITTED TEXT** 20 WORDS	**96%** **MATCHING TEXT** 20 WORDS
Data filtering involves the selection of specific rows, columns, or fields to display from the dataset [82] [84]. •	data filtering simply involves the selection of specific rows, columns, or fields to display from the dataset.

W https://www.xplenty.com/blog/data-transformation-explained/

8/51 **SUBMITTED TEXT** 16 WORDS	**80%** **MATCHING TEXT** 16 WORDS
Data deduplication is a data compression process to identify and remove repeated copies of information.	Data deduplication is a data compression process where you identify and remove duplicate or repeated copies of information.

W https://www.xplenty.com/blog/data-transformation-explained/

9/51 **SUBMITTED TEXT** 31 WORDS	**59%** **MATCHING TEXT** 31 WORDS
one unique copy of data in the database. This process allows for examining incoming data and compares it to data that is already stored in the system [82] [84]. •	one unique copy of data in your data warehouse or database. The deduplication process analyzes incoming data and compares it to data that's already stored in the system.

W https://www.xplenty.com/blog/data-transformation-explained/

10/51 **SUBMITTED TEXT** 28 WORDS	**89%** **MATCHING TEXT** 28 WORDS
This involves converting data from one structure (or no structure) to another to integrate it with a data warehouse or with different applications [82] [84] [106]. •	This involves converting data from one structure (or no structure) to another so you can integrate it with a data warehouse or with different applications.

W https://www.xplenty.com/blog/data-transformation-explained/

11/51	SUBMITTED TEXT	23 WORDS	81%	MATCHING TEXT	23 WORDS
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			used to change a numeric series into fixed, categorical ranges, say, from {2,5,8...} to {2-5, 6-9, 10-13...}. Take, for example, the seasonal fluctuations in		

W https://www.xplenty.com/blog/data-transformation-explained/

12/51	SUBMITTED TEXT	20 WORDS	83%	MATCHING TEXT	20 WORDS
similar datasets into a single dataset [84]. • Filtering: Rule-based reduction of a larger dataset into a smaller dataset.			similar) datasets into a larger dataset • Filtering: Rule-based narrowing of a larger dataset into a smaller dataset •		

W https://www.ironsidegroup.com/2019/07/16/data-preparation-business-analytics/

13/51	SUBMITTED TEXT	20 WORDS	97%	MATCHING TEXT	20 WORDS
Clinical Data Interchange Standards Consortium (CDISC) is a global nonprofit charitable organization with administrative offices in Austin, Texas, with			Clinical Data Interchange Standards Consortium CDISC is a 501(c)(3) global nonprofit charitable organization with administrative offices in Austin, Texas, with		

W https://www.cdisc.org/

14/51	SUBMITTED TEXT	27 WORDS	54%	MATCHING TEXT	27 WORDS
to create and advance data standards. This allows for accessibility, interoperability, and reusability of data for competent research that has greater impact on global health [118].			to develop and advance data standards of the highest quality. Together, we enable the accessibility, interoperability, and reusability of data for more meaningful and efficient research that has greater impact on global health.		

W https://www.cdisc.org/

15/51	SUBMITTED TEXT	38 WORDS	78%	MATCHING TEXT	38 WORDS
TransCelerate BioPharma Inc. is a nonprofit organization with a mission to collaborate across the global biopharmaceutical research and development community to identify, prioritize, design, and facilitate the implementation of solutions designed to deliver high quality new medicines.			TransCelerate BioPharma Inc. is a non-profit organization with a mission to collaborate across the global biopharmaceutical research and development community to identify, prioritize, design and facilitate implementation of solutions designed to drive the efficient, effective and high-quality delivery of new medicines.		

W https://www.transceleratebiopharmainc.com/

16/51	SUBMITTED TEXT	91 WORDS	61%	MATCHING TEXT	91 WORDS
all_met_rmsd &gt;- merge (x = subset3, y = all02, by = "mr_no", all.x = TRUE) all_met_rmsd &gt;-			all_met_rmsd05 &gt;- merge(x = all_met_rmsd04, y = both, by = c("mr_no", "cdur"), all.y = TRUE) all_met_rmsd05 &gt;-		

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

17/51 **SUBMITTED TEXT** 27 WORDS **38% MATCHING TEXT** 27 WORDS

all_met_rmsd &gt;- merge (all_met_rmsd, mindayt, by = "mr_no") # Calculate the age variable for non-missing dates all_met_rmsd &gt;- all_met_rmsd [, `:=`(age = ifelse (!

all_met_rmsd05 &gt;- merge(x = all_met_rmsd04, y = both, by = c("mr_no", " cdur"), all.y = TRUE) all_met_rmsd05 &gt;- all_met_rmsd05 [, time := ifelse(

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

18/51 **SUBMITTED TEXT** 14 WORDS **58% MATCHING TEXT** 14 WORDS

all_met_rmsd, "D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.csv") saveRDS (all_met_rmsd, "D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") dis_rutu &gt;- all_met_rmsd [Code != "", .(

all_met_rmsd02 &gt;- fread("C://Users//mahajvi1//Downloads//01adsl_met_rmsd.csv") all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code :=

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

19/51 **SUBMITTED TEXT** 12 WORDS **77% MATCHING TEXT** 12 WORDS

RMSD data ################################### all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") subpat &gt;- unique(all_met_rmsd [,

rmsd.csv") all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;-

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

20/51 **SUBMITTED TEXT** 7 WORDS **100% MATCHING TEXT** 7 WORDS

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") ####################################

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds")

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

21/51 **SUBMITTED TEXT** 9 WORDS **83% MATCHING TEXT** 9 WORDS

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") subpat &gt;- unique(all_met_rmsd [,

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;-

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

22/51 **SUBMITTED TEXT** 115 WORDS **58% MATCHING TEXT** 115 WORDS

met_rmsd02 &gt;- merge (x = base01_met_rmsd02, y = subpat, by = c("mr_no"), all.x = TRUE) base01_met_rmsd02 &gt;- unique(base01_met_rmsd02) #

met_rmsd05 &gt;- merge(x = all_met_rmsd04, y = both, by = c("mr_no", "cdur"), all.y = TRUE) all_met_rmsd05 &gt;- all_met_rmsd05 [,

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

23/51	SUBMITTED TEXT	74 WORDS	34% MATCHING TEXT	74 WORDS
all_met_rmsd, all = TRUE, by.x = "Code", by.y = "icd_code") all_met_rmsd$distype[is.na(all_met_rmsd$distype)] &gt;- "OTHER" all_met_rmsd &gt;- all_met_rmsd [order(mr_no, studyday,			all_met_rmsd02 &gt;- all_met_rmsd02 [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] all_met_rmsd03 &gt;- unique(all_met_rmsd02 [, c("mr_no", "studyday", "	

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

24/51	SUBMITTED TEXT	21 WORDS	57% MATCHING TEXT	21 WORDS
Add a redundant day column to use as the end range setkey(diag2, MRNo, min, max) ## Set the key for			Add a redundant lossDate column to use as the end range for claims setkey(policies, policyNumber, EFDT, EXDT) ## Set the key for	

W http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges

25/51	SUBMITTED TEXT	74 WORDS	79% MATCHING TEXT	74 WORDS
Update rows where the claim was out of policy: ans2[is.na(min), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(ans2, MRNo,Code2,visday, min) ## order the results setkey(ans2, MRNo, Code2) ## set the key to identify unique values ans2 &gt;- rbindlist(list(ans2, ## select only the unique values diag2[!.(ans2[, unique(MRNo)])] ## policies with no claims), fill=T) ans20 &gt;-			Update rows where the claim was out of policy: ans2[is.na(EFDT), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(ans2, policyNumber, claimNumber, lossDate, EFDT) ## order the results setkey(ans2, policyNumber, claimNumber) ## set the key to identify unique values ans2 &gt;- rbindlist(list(unique(ans2), ## select only the unique values policies[!.(ans2[, unique(policyNumber)])] ## policies with no claims), fill=T) ans2 ##	

W http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges

26/51	SUBMITTED TEXT	47 WORDS	71% MATCHING TEXT	47 WORDS
The foverlaps function requires both tables to have a start and end range, # and the "y" table to be keyed diag8[, tempmon := vismon] ## Add a redundant day column to use as the end range setkey(ref, durlwr, durupr) ## Set the key for			The foverlaps function requires both tables to have a start and end range, and the "y" table to be keyed claims[, lossDate2:=lossDate] ## Add a redundant lossDate column to use as the end range for claims setkey(policies, policyNumber, EFDT, EXDT) ## Set the key for	

W http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges

27/51 **SUBMITTED TEXT** 156 WORDS

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: diag8rpt &gt;- foverlaps(diag8, ref, by.x=c("vismon", "tempmon"))[, `:=`(inPolicy=T, tempmon=NULL)] ## Update rows where the claim was out of policy: diag8rpt[is.na(durlwr), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(diag8rpt, MRNo, Code, vismon, durlwr) ## order the results 210 | P a g e setkey(diag8rpt, MRNo, Code, dur) ## set the key to identify unique values diag8rpt &gt;- rbindlist(list(diag8rpt, ## select only the unique values diag8[!.(diag8rpt[, unique(MRNo)])] ## policies with no claims), fill=T) #

51% **MATCHING TEXT** 156 WORDS

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: ans2 &gt;- foverlaps(claims, policies, by.x=c("policyNumber", "lossDate", "lossDate2"))[, `:=`(inPolicy=T, lossDate2=NULL)] ## Update rows where the claim was out of policy: ans2[is.na(EFDT), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(ans2, policyNumber, claimNumber, lossDate, EFDT) ## order the results setkey(ans2, policyNumber, claimNumber) ## set the key to identify unique values ans2 &gt;- rbindlist(list(unique(ans2), ## select only the unique values policies[!.(ans2[, unique(policyNumber)])] ## policies with no claims), fill=T)

W http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges

28/51 **SUBMITTED TEXT** 20 WORDS

Add a redundant day column to use as the end range setkey(ref, durlwr, durupr) ## Set the key for

60% **MATCHING TEXT** 20 WORDS

Add a redundant lossDate column to use as the end range for claims setkey(policies, policyNumber, EFDT, EXDT) ## Set the key for

W http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges

29/51 **SUBMITTED TEXT** 163 WORDS

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: 211 | P a g e vitals8rpt &gt;- foverlaps(vitals8 [vismon &lt; 0], ref, by.x=c("vismon", "tempmon"))[, `:=`(inPolicy=T, tempmon=NULL)] ## Update rows where the claim was out of policy: vitals8rpt[is.na(durlwr), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(vitals8rpt, MRNo, vitalparam, vismon, durlwr) ## order the results setkey(vitals8rpt, MRNo, vitalparam, dur) ## set the key to identify unique values vitals8rpt &gt;- rbindlist(list(vitals8rpt, ## select only the unique values vitals8[!.(vitals8rpt[, unique(MRNo)])] ## policies with no claims), fill=T) #

52% **MATCHING TEXT** 163 WORDS

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: ans2 &gt;- foverlaps(policies, by.x=c("policyNumber", "lossDate", "lossDate2"))[, `:=`(inPolicy=T, lossDate2=NULL)] ## Update rows where the claim was out of policy: ans2[is.na(EFDT), inPolicy:=F] ## Remove duplicates (such as policyNumber==123 & claimNumber==3), ## and add policies with no claims (policyNumber==125): setkey(ans2, policyNumber, claimNumber, lossDate, EFDT) ## order the results setkey(ans2, policyNumber, claimNumber) ## set the key to identify unique values ans2 &gt;- rbindlist(list(unique(ans2), ## select only the unique values policies[!.(ans2[, unique(policyNumber)])] ## policies with no claims), fill=T)

W http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges

30/51 **SUBMITTED TEXT** 10 WORDS

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") unqdis &gt;- unique(all_met_rmsd [,

83% **MATCHING TEXT** 10 WORDS

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;-

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

31/51 **SUBMITTED TEXT** 101 WORDS **59% MATCHING TEXT** 101 WORDS

all_met_rmsd02 &gt;- merge(x = all_met_rmsd, y = unqdis, by = c("mr_no"), allow.cartesian = TRUE) 216 | P a g e all_met_rmsd02 &gt;- all_met_rmsd02 [,

all_met_rmsd05 &gt;- merge(x = all_met_rmsd04, y = both, by = c("mr_no", " cdur"), all.y = TRUE) all_met_rmsd05 &gt;- all_met_rmsd05 [,

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

32/51 **SUBMITTED TEXT** 131 WORDS **45% MATCHING TEXT** 131 WORDS

all_met_rmsd02 &gt;- all_met_rmsd02 [, refday := ifelse(studyday &lt;= mindisday, studyday - mindisday + 1, studyday - mindisday),] all_met_rmsd02 &gt;- all_met_rmsd02[, refmnyr := ifelse(

all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

33/51 **SUBMITTED TEXT** 35 WORDS **32% MATCHING TEXT** 35 WORDS

all_met_rmsd02 &gt;- all_met_rmsd02[, period := ifelse(refday == 1, 1, period),] all_met_rmsd02 &gt;- all_met_rmsd02[, periodn := ifelse(refday == 1, "Day 1", periodn),] saveRDS (all_met_rmsd02, "D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") disease &gt;- unique(all_met_rmsd02 [, -c("

all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] all_met_rmsd03 &gt;- unique(all_met_rmsd02 [, c("

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

34/51 **SUBMITTED TEXT** 100 WORDS **66% MATCHING TEXT** 100 WORDS

all_met_rmsd02 &gt;- merge (x = all_met_rmsd02, y = bfraftr , by = c("mr_no", "refcode", "refdesc", "Code", "description"), all.x = TRUE) all_met_rmsd02 &gt;- all_met_rmsd02 [, -

all_met_rmsd05 &gt;- merge(x = all_met_rmsd04, y = both, by = c("mr_no", " cdur"), all.y = TRUE) all_met_rmsd05 &gt;- all_met_rmsd05 [,

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

35/51 **SUBMITTED TEXT** 7 WORDS **83% MATCHING TEXT** 7 WORDS

all_met_rmsd02, "D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") 220 |

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds")

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

36/51 **SUBMITTED TEXT** 29 WORDS **54% MATCHING TEXT** 29 WORDS

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") 221 | P a g e all_met_rmsd &gt;- all_met_rmsd [, `:=` (baseage = min(age)), by =.(mr_no)] all_met_rmsd &gt;- all_met_rmsd [, `:=` (

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") rmsd02 &gt;- met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [,

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

37/51 **SUBMITTED TEXT** 29 WORDS **55% MATCHING TEXT** 29 WORDS

all_met_rmsd02 &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Coded_med := paste(Type_med, Coded_med, sep=":"),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Coded_med := str_replace_all(Coded_med, "\"", ""),] chk01 &gt;- all_met_rmsd02 [, .(

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd02 &gt;-

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

38/51 **SUBMITTED TEXT** 8 WORDS **100% MATCHING TEXT** 8 WORDS

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") 240 |

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds")

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

39/51 **SUBMITTED TEXT** 38 WORDS **65% MATCHING TEXT** 38 WORDS

Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),]) meds0 &gt;- data.table(meds0 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),]) ## ################### #

Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),]

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

40/51 **SUBMITTED TEXT** 42 WORDS **65% MATCHING TEXT** 42 WORDS

D:/Hospital_data/ProgresSQL/analysis/080_medicine_dis_repeat_prop_cumulative.csv") all_met_rmsd0 &gt;- data.table(all_met_rmsd [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),]) all_met_rmsd0 &gt;- data.table(all_met_rmsd0 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),])

D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),]

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

41/51 **SUBMITTED TEXT** 72 WORDS **52% MATCHING TEXT** 72 WORDS

all_met_rmsd_unq02 &gt;- merge(x = all_met_rmsd_unq, y = cum02 [, -c("newold", "cat"),], by = c("mr_no", "

all_met_rmsd05 &gt;- merge(x = all_met_rmsd04, y = both, by = c("mr_no", "

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

42/51 **SUBMITTED TEXT** 119 WORDS **86%** **MATCHING TEXT** 119 WORDS

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd &gt;- all_met_rmsd [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] 262 | P a g e all_met_rmsd &gt;- all_met_rmsd [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd &gt;- all_met_rmsd [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd &gt;- all_met_rmsd [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] med01 &gt;- unique(all_met_rmsd [Med02 != "NA:NA" , c("mr_no", "

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] all_met_rmsd03 &gt;- unique(all_met_rmsd02 [, c("mr_no", "

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

43/51 **SUBMITTED TEXT** 9 WORDS **92%** **MATCHING TEXT** 9 WORDS

all_met_rmsd02 &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [,

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [,

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

44/51 **SUBMITTED TEXT** 8 WORDS **88%** **MATCHING TEXT** 8 WORDS

all_met_rmsd &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") ## #### #

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds")

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

45/51 **SUBMITTED TEXT** 9 WORDS **82%** **MATCHING TEXT** 9 WORDS

all_met_rmsd &gt;- readRDS ("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") a2 &gt;- all_met_rmsd [

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;-

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

46/51 **SUBMITTED TEXT** 71 WORDS **100%** **MATCHING TEXT** 71 WORDS

all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd &gt;- all_met_rmsd [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd &gt;- all_met_rmsd [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd &gt;- all_met_rmsd [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd &gt;- all_met_rmsd [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] all_met_rmsd02 &gt;-

all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] all_met_rmsd03 &gt;-

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

47/51	SUBMITTED TEXT	68 WORDS	97% MATCHING TEXT	68 WORDS

readRDS("D:/Hospital_data/ProgresSQL/analysis/all_met_rmsd02.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Med02 := paste(Type_med, ":", Coded_med, sep =""),] #

readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") all_met_rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;- all_met_rmsd02 [, description:= ifelse (description == "" | description ==" ", "** Not yet coded", description),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Code02 := paste(distype, ":", Code, ":", description, sep =""),] all_met_rmsd02 &gt;- all_met_rmsd02 [, Med02 := paste(Type_med, ":", Coded_med, sep =""),]

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

48/51	SUBMITTED TEXT	28 WORDS	54% MATCHING TEXT	28 WORDS

csv folder #################### all_met_rmsd &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") 293 | P a g e all_met_rmsd &gt;- all_met_rmsd [, Code := str_replace_all(Code, "\\.", "_")] cnt&gt;- unique(all_met_rmsd [

csv") all_met_rmsd02 &gt;- readRDS("D:/Hospital_data/ProgresSQL/analysis/01adsl_met_rmsd.rds") rmsd02 &gt;- all_met_rmsd02 [, Code := ifelse (Code == " " | Code == "", "** Not yet coded", Code),] all_met_rmsd02 &gt;-

W https://github.com/coursephd/PostgreSQL/blob/master/110_rasa_aushadhi_analysis.R

49/51	SUBMITTED TEXT	67 WORDS	69% MATCHING TEXT	67 WORDS

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: ans2 &gt;- foverlaps(vitals5, diag2, by.x=c("MRNo", "visday", "tempday"))[, `:=`(inPolicy=T, tempday=NULL)] 206 |

y" table) ## Find the overlaps, remove the redundant lossDate2 column, and add the inPolicy column: ans2 &gt;- foverlaps(claims, policies, by.x=c("policyNumber", "lossDate", "lossDate2"))[, `:=`(inPolicy=T, lossDate2=NULL)] ##

W http://stackoverflow.com/questions/21560500/data-table-merge-based-on-date-ranges

50/51	SUBMITTED TEXT	5 WORDS	75% MATCHING TEXT	5 WORDS

declaration-of-helsinki-ethical-principles-for-medical-research-involving- human-subjects/#:~:text=1.,identifiable%20human%20material%20and%20data.. [11] "THE

Declaration of Helsinki as a statement of ethical principles for medical research involving human subjects, including research on identifiable human material and data. The

W https://www.wma.net/policies-post/wma-declaration-of-helsinki-ethical-principles-for-medical-rese ...

51/51	SUBMITTED TEXT	10 WORDS	100% MATCHING TEXT	10 WORDS

PostgreSQL: The World's Most Advanced Open Source Relational Database,"

PostgreSQL: The World's Most Advanced Open Source Relational Database

W https://www.postgresql.org/