CSE 702

**Clustering Clinical Marker Trajectories from CKD dataset**

MD S Q Zulkar Nine (50131471)

Nikhil Shekhar (50169106)

horizontal line

# 

# Introduction

Personalized medication requires the development of tools that can accurately predict the trajectory of an individual's disease to help clinicians to optimize treatment.

With tools that can accurately predict an individual’s *disease activity trajectory*, clinicians can more aggressively treat those at greatest risk early, rather than waiting until the disease progresses to a high level of severity.

Achieving the automatic extraction of deep and medically relevant phenotypes from clinical data and their integration with genomic, lifestyle, and environmental factors to develop insights that will be instrumental in accelerating discoveries in the area of precision medicine.

To achieve this, we first need to identify the cohorts by clustering and then finding commonalities in patients with a disease.

In the running example of kidney disease, we use a single phenotype named eGFR (Glomerular Filtration Rate) to analyze the condition of kidney of patients over years. This eGFR reading is the best test to measure your level of kidney function and determine your stage of kidney disease. The GFR tells the clinician about the stage of kidney disease and helps the clinician plan your treatment. If the GFR number is low, the kidneys of a person are not working as well as they should. The earlier kidney disease is detected, the better the chance of slowing or stopping its progression.

The clustering of the patients into subtypes(groups) becomes challenging because most of the time-series clustering methods assume that the data is equally spaced in the time interval, but the data we have has time series data with unequal lengths and the sampling is also at irregular frequencies.

## Dataset

We have eGFR time-series data for a total of 63215 patients in the CKD dataset. This table additionally contains some demographic information about the patient such as their age, gender, and birth year. As auxiliary data, we also have data about the patient’s smoking records, lab tests and other findings such as the time-series progression of their height, weight, and blood pressure levels. A full description of the dataset is given in Table 1.

Table – 1: Clinical data elements.

|  |  |  |
| --- | --- | --- |
| Data element | Measure type | Variable Name |
| Patient Identification | Numerical | pid |
| Time of visit | date time | timestamp |
| Year of birth | Numerical | birthyear |
| Gender | Numerical : 1 - Mail, 2 - Female | numeric\_gender |
| Current smoking | Numeric: Current, never, past | numeric\_smoker\_status |
| Height and | Numerical : actual | FND\_HTLB |
| weight/BMI | Numerical : actual | FND\_WTLB |
| HDL | Numerical result | LR\_HDL |
| LDL-C | Numerical result | LR\_LDL |
| Triglycerides | Numerical result | LR\_TRIG |
| Creatinine | Numerical result | LR\_CR |
| AST | Numerical result | LR\_AST |
| ALT | Numerical result | LR\_ALT |
| HbA1c | Numerical result | LR\_A1C |
| Blood pressure | Systolic and diastolic | FND\_BPS , FND\_BPD |
| Estimated GFR | Calculated value | grf |

## Missing Data:

In this section, we identified some columns those have excessive missing data. We tried to identify those data and removed it from the dataset.

* 1. **Detailed Data Cleaning Steps**

**4.1.1 Loading data from Postgres database**

1. Load the eGFR data from the Postgres Database dump file into a R data frame named “**egfr\_df**". The data has the following columns: **Pid, Timestamp, Gender, Birthyear, Age, gfr**.
2. Load the finding data from the postgres Database dump file into R data frame named “**findings\_df**”. The data has the following columns: **Idperson, Finddate, Valuename, findvalnum.**
3. Rename these columns as below to have standard names in the dataset: **Pid, Timestamp, Testname, testval.**
4. Load the lab reports data from the postgres Database dump file into R data frame named “**lab\_df**”. The data has the following columns: **Idperson, Resultdate , Valuename, resultvaluenum.**
5. Rename these columns as below to have standard names in the dataset: **Pid, Timestamp, Testname, testval.**
6. Load the smoking history data from the postgres Database dump file into R data frame named “**smoking\_df**”. This dataset has just two columns named: **pid, smoker\_status**.

For any operation like the time series manipulation requires the data to be in a specific format. So, we start the generation of a feature matrix which would be used for modeling the problem at hand.

**4.1.2 Normalized data**

Format the date in all the three data frames created above to be in the format like “YYYY-MM-DD” Also, format the date field in all the three data frames. The number of unique patients who have eGFR readings can be found by extracting the unique values of patient id’s in the “egfr” data frame as below:

list\_of\_patients = unique(egfr\_df$pid)

We create a new data frame named “combined\_df” which has the features from all the three data frame so that this can be used for creating models to analyze the data.

**4.1.3 Map Reduce Job**

We wrote a Map-Reduce job, the details of which is as listed below:

We wrote three **mappers** with the below functionalities:

1. **KidneyGfrMapper**: This reads the eGFR file of the patient and outputs the key as a concatenated string comprising of the patient id and the timestamp. The value emitted is the rest of the elements in the row with a prefix “999”, so that it can be recognized at the reducer as the row coming from the eGFR file.
2. **KidneyFindingsMapper**: This read the findings file of the patient and outputs the key same as the one from the mapper above with the values as the remaining elements in the row.
3. **KidneyLabMapper**: This read the lab details file of the patient and outputs the key in the same format as the above two mappers.

The key in the mapper is kept the same, as we want to join the datasets to create a feature matrix for applying and predicting various models.

There is one **reducer** in the implementation which does the following:

* In the setup() method of the reducer, we read the output schema file in an ArrayList and populate the rows for each patient in the same format. There are other complexities handled in the code to achieve the same.
* The output from the reducer is written onto the HDFS which is essentially the future matrix we will be using from here on to perform all the operations.

Then, the part-r-\*\*\*\*\* files are downloaded from HDFS and then merged into one file named “feature\_matrix”

This MR takes less than 5 **minutes** to run and generate the feature matrix. The same functionality is coded in R as well but this version takes close to 10 hours just to achieve the same.

## 4.1.4 Feature matrix:

The feature\_matrix file is read into a R data frame named “feature\_matrix” which we will be using for all our manipulations from here on.

We have quite a few columns as “null” in the “feature\_matrix” as they did not have corresponding data for the patients in the input files. These are converted into “NaN” values which is the equivalent of NULL’s. This transformation is applied to all the columns in the “feature\_matrix” data frame. On looking into the data, we see that there are many duplicate rows in the data. Now, we extract and keep only the unique rows and discard the rest. The rows in the data frame with same dates have the same status of the patient and hence discarded.

Number of missing values for each column are shown in Table 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table -2: Missing values in each column |  | |  | |  |
| Data element | Variable Name | Number of missing values | |
| Patient Identification | pid | 0 | |
| Time of visit | timestamp | 0 | |
| Year of birth | birthyear | 0 | |
| Gender | numeric\_gender | 0 | |
| Current smoking | numeric\_smoker\_status | 21816 | |
| Height and | FND\_HTLB | 409856 | |
| weight/BMI | FND\_WTLB | 317985 | |
| HDL | LR\_HDL | 241364 | |
| LDL-C | LR\_LDL | 246680 | |
| Triglycerides | LR\_TRIG | 241484 | |
| Creatinine | LR\_CR | 149 | |
| Parathyroid hormone Intact | LR\_PTH | 577005 | |
| Blood Glucose with fasting | LR\_GLUCFAST | 508780 | |
| Blood Glucose without fasting | LR\_GLUCNONFAST | 90955 | |
| AST | LR\_AST | 167043 | |
| ALT | LR\_ALT | 185527 | |
| Micro-albumin/Creatinine | LR\_MICROCR | 528565 | |
| HbA1c | LR\_A1C | 387993 | |
| Blood pressure Diastolic | FND\_BPD | 303148 | |
| Blood pressure Systolic | FND\_BPS | 302435 | |
| eGFR for African-American patient | FND\_GRF\_ARAMER | 477686 | |
| Estimated GFR | grf | 294811 | |

**4.1.5 Threshold setting and outlier removals**

The columns which have a lot of missing data for patients will be of minimal use for modeling the trajectory of the patients hence, we drop the columns which have a lot of missing data. We set the threshold to “85%” and drop columns with missing data more than this threshold.

The columns that are dropped are:

* LR\_MICROCR
* LR\_PTH
* LR\_GLUCFAST
* LR\_VITD.25
* LR\_VITD.1\_25
* LR\_PHOS

The new feature vector has 21 columns and 580128 columns.

This data is written to an output file, so that it can be read later on and the feature generation and cleaning process up to this process need not be repeated later on. This file can be directly loaded and used for further processing and modeling.

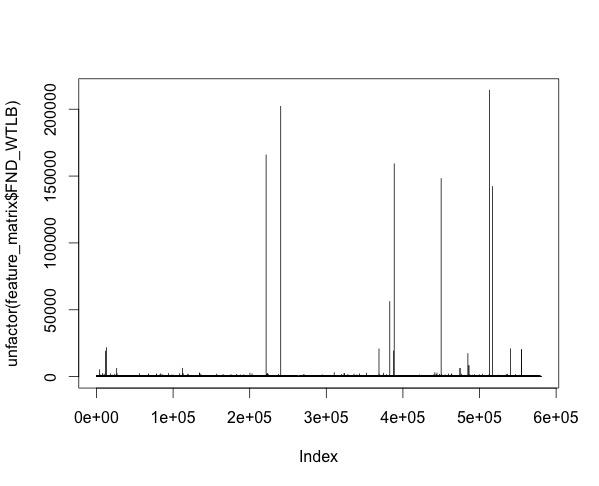
# Outlier removals:

Dataset contains a lot of outliers mostly due to data entry errors. Most of the data columns clinical attributes and has a domain boundary. Any value that is outside the possible domain boundary is considered as outliers.

For the modeling to be proper, we need to use the domain knowledge to remove outliers from the data. Here, we proceed by removing the outliers from each column one step at a time as below and we keep an aggregated count of the number of total rows removed during each outlier removal as well as the count of number of rows that were outliers while checking each column.

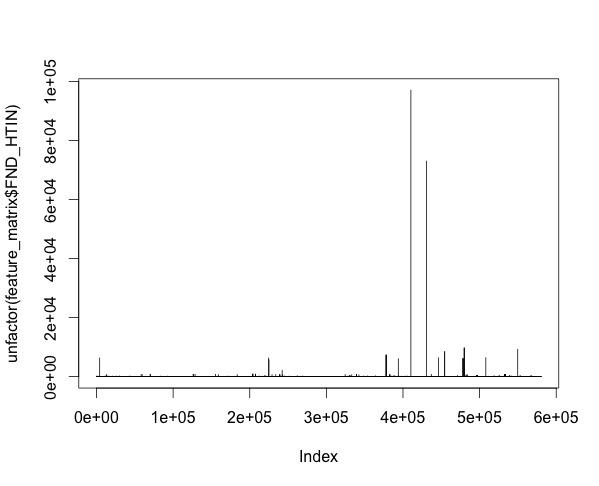
## Remove the Weight outliers from the feature matrix:

Here we set the threshold on weight to be 1000 pounds as it is very unlikely that a person can weigh more than this value and it’s most likely a erroneous data.

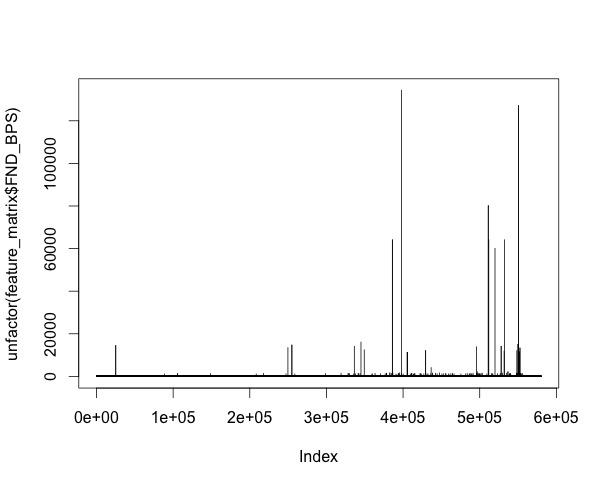


1. **Remove the height outliers from the feature matrix:**

Tallest man ever was 8 feet, so 10 feet is a safe threshold but there are patients with heights beyond 5000 inches.

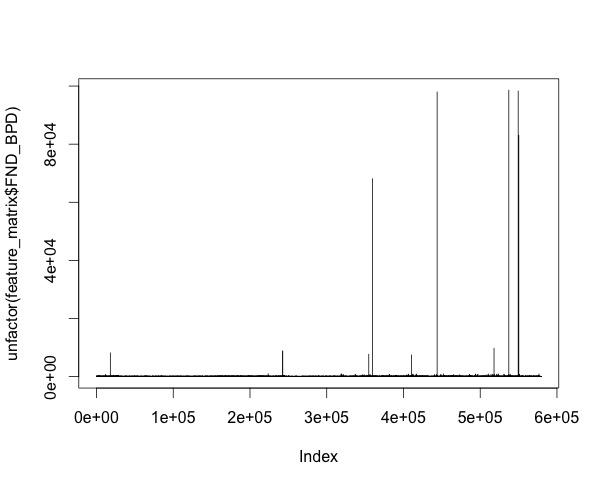


1. **Remove the blood pressure outliers from the feature matrix (systolic):**

Normal blood pressure is generally less than 120/80 mmHg (i.e. systolic blood pressure less than 120). Normal to high blood pressure: between 120/80 and 140/90 mmHg. High blood pressure: 140/90 mmHg or higher.

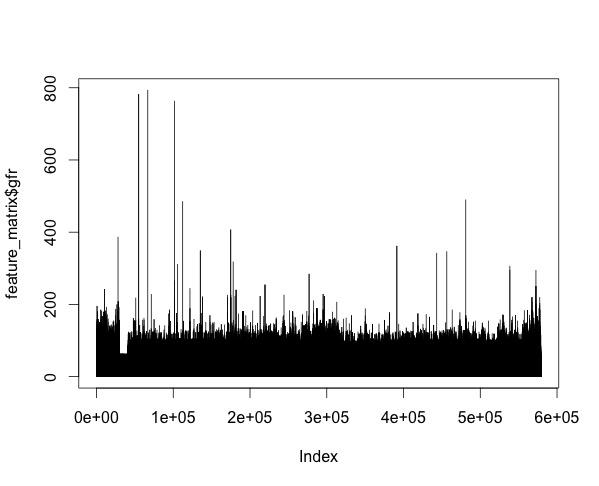
## Remove the blood pressure outliers from the feature matrix(diastolic):

Normal blood pressure is generally less than 120/80 mmHg (i.e. diastolic blood pressure less than 80 mmHg). Normal to high blood pressure: between 120/80 and 140/90 mmHg. High blood pressure: 140/90 mmHg or higher.



## Remove the eGFR outliers from the feature matrix:

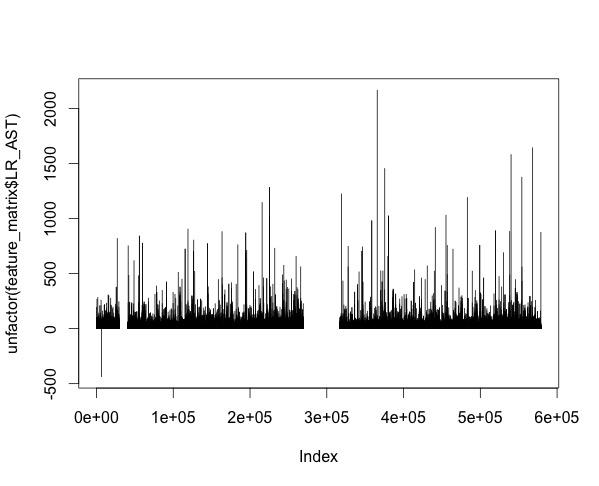
An eGFR value above 90 is considered to be a normal reading for patients, a 60-89 value is considered a mild decrease, and the severity of the kidney disease increases as the eGFR decreases further with values < 15 being signs of kidney failure.



## 

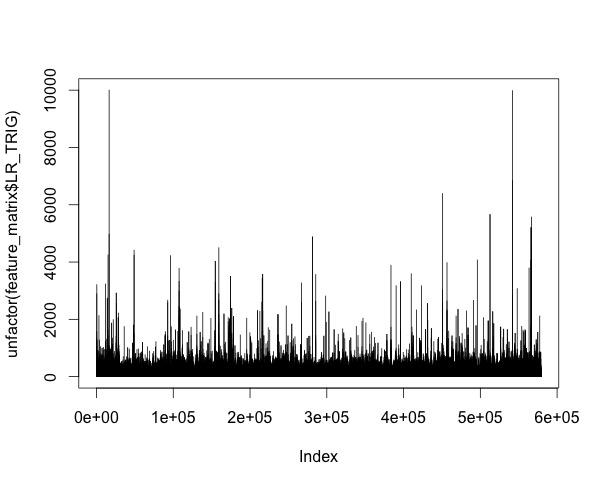
## Remove the AST outliers from the feature matrix:

The normal range for the AST test is 10 to 34 IU/L (international units per liter). Levels of AST greater than 10 times the normal limit usually indicate a viral hepatitis infection, that is over 340. So taking a threshold of 1000 should be safe since even in the most sever infection, AST test values should not cross that threshold.



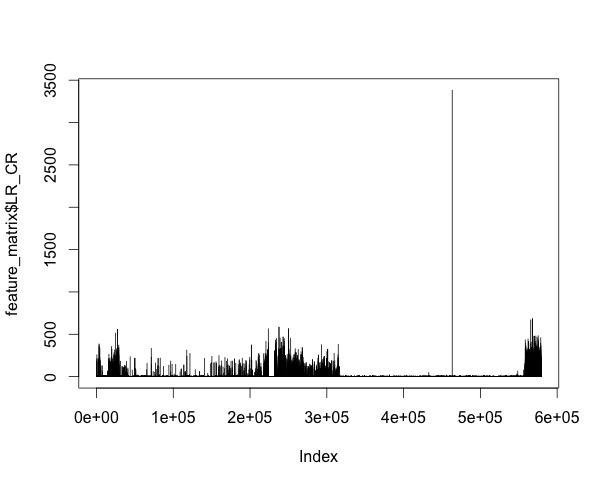
## Remove the Triglyceride outliers from the feature matrix:

Normal triglycerides means there are less than 150 milligrams per deciliter (mg/dL). Borderline high triglycerides = 150 to 199 mg/dL. Very high triglycerides = 500 mg/dL or higher. The highest world record for triglyceride reading was 3165 mg/dl, 21 times the normal level of 150 mg/dl The number of patients with Triglycerides values beyond 3165 mg/dL = 44

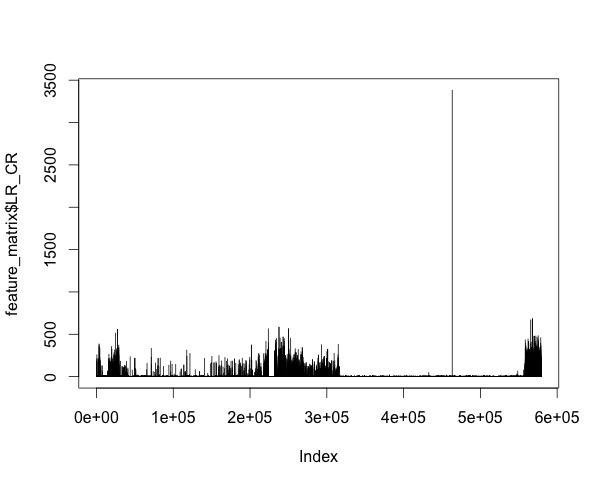


## Remove the Creatinine outliers from the feature matrix:

Normal creatinine levels range from 0.7 to 1.3 mg/dL in men and 0.6 to 1.1 mg/dL in women. Anything more then 2 is acute and 5 is chronic and needs urgent intervention. The highest ever recorded creatinine level was 61.3 mg/dL, so taking 100 as a threshold is appropriate. But before we decide 100 as a threshold, it is important to consider the distribution of creatinine levels in patients. Unlike other lab tests and findings there is a significantly huge number of patients with unrealistically high creatinine levels. There are a total of 2 521 patients with Creatinine levels beyond 100 mg/dL.



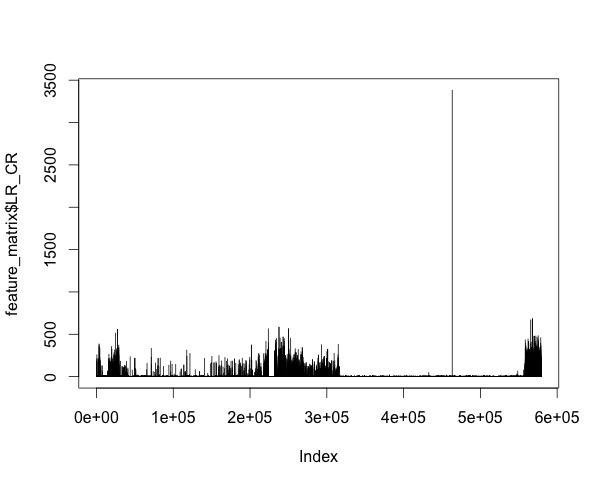
## Remove the Blood glucose outliers from the feature matrix:

The normal blood sugar range is between 80-120 mg/dl, but the highest recorded glucose level was 2,656 mg/dl.

## Remove the ALT outliers from the feature matrix:

There are 13 patients with Alanine aminotransferase levels beyond 1000.

The logs and the plot on running the R script for the same is as below:



## Outlier removal summary:

The aggregated sum of the rows with outliers removed are 3906. This concluded the feature generation and data cleaning.

# 

# Map-Reduce job to sort the feature vector by time and group by patient id and timestamp:

Here we pass the input data as it is to the Map-Reduce job and make use of the following custom code plugged into the MR to achieve the same:

## SecondarySortBasicPartitioner :

**public class** SecondarySortBasicPartitioner **extends**

Partitioner<CompositeKeyWritable, Text> {

@Override

**public int** getPartition(CompositeKeyWritable key, Text value,

**int** numReduceTasks) {

**return** ((key.getkey1().hashCode() & Integer.***MAX\_VALUE***) % numReduceTasks);

}

}

## SecondarySortBasicCompKeySortComparator

**public class** SecondarySortBasicCompKeySortComparator **extends** WritableComparator {

**protected** SecondarySortBasicCompKeySortComparator() {

**super**(CompositeKeyWritable.**class**, **true**);

}

@Override

**public int** compare(WritableComparable w1, WritableComparable w2) {

CompositeKeyWritable key1 = (CompositeKeyWritable) w1;

CompositeKeyWritable key2 = (CompositeKeyWritable) w2;

**int** cmpResult = key1.getkey1().compareTo(key2.getkey1());

**if** (cmpResult == 0)

{

SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);

Date date1 = **null**,date2 = **null**;

**try** {

date1 = sdf.parse(((CompositeKeyWritable) w1).getkey2());

date2 = sdf.parse(((CompositeKeyWritable) w2).getkey2());

} **catch** (ParseException e) {

e.printStackTrace();

}

**return** date1

.compareTo(date2);

}

**return** cmpResult;

}

}

## SecondarySortBasicGroupingComparator:

**public class** SecondarySortBasicGroupingComparator **extends** WritableComparator {

**protected** SecondarySortBasicGroupingComparator() {

**super**(CompositeKeyWritable.**class**, **true**);

}

@Override

**public int** compare(WritableComparable w1, WritableComparable w2) {

CompositeKeyWritable key1 = (CompositeKeyWritable) w1;

CompositeKeyWritable key2 = (CompositeKeyWritable) w2;

**return** key1.getkey1().compareTo(key2.getkey1());

}

}

The output of the above Map-Reduce job is written to HDFS which is exported and merged into one file along with the schema so that it can be loaded as a dataframe in R by the command below.

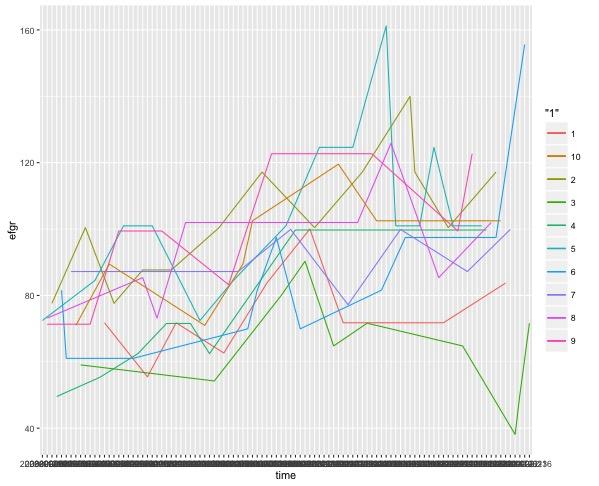
feature\_vector\_sorted\_by\_time <- read.csv("~/Seminar702/feature\_matrix\_arranged\_by\_date\_with\_patients\_10\_or\_more.csv")

The same can wbe achieved even without Map-Reduce but takes 7 hours to complete compared to less than 5 minutes in an MR.

## Kind of patients we are trying to find and model:

The patients who have shown better kidney condition as the time progressed and were on medication are typically the patients we are trying to find and figure out the reason for the same, so that the model can be built to assist the clinicians with further medication of patients.

There are some patients of this nature and can be seen as below. The colour line indicate, how the eGFR value of one patient has changed over time. As it can be seen that these patients have shown better kidney conditions over time.



1. **Correlation Analysis:**

We can see that LR\_GFR is negatively correlated with LR\_CR. This is expected because eGFR



Figure: Spearman Correlation matrix

Is calculated from a single Urine Albumin Creatinine Ratio test. A patients increasing creatinine level is seen as a sign of increasing severity of kidney disease.

1. **Principal Component Analysis:**

We can see from the graph that 80% variance are on the first 9 principle components.

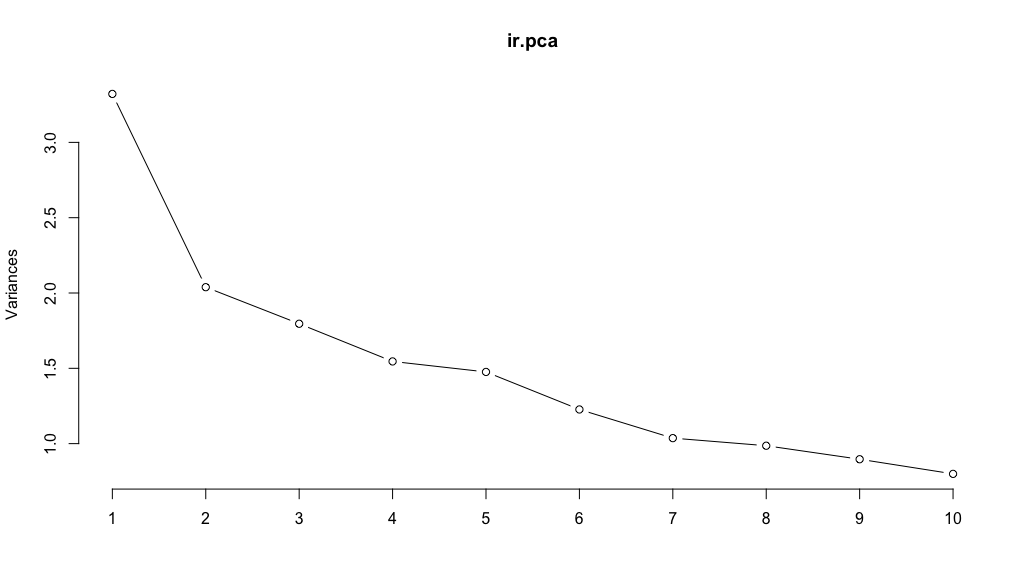
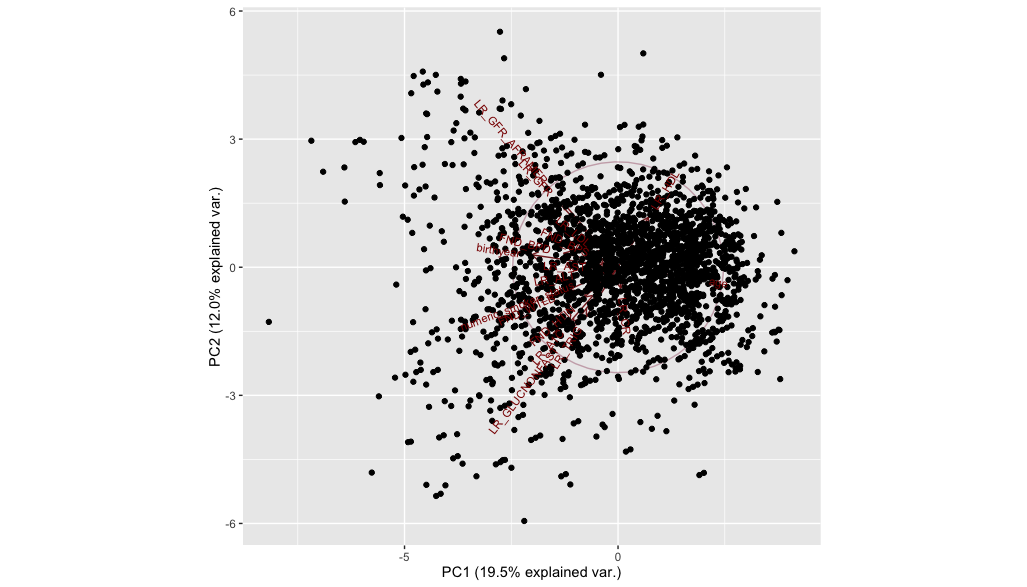


Figure – Principle components and their corresponding variance contribution.

Here we can see the two dimensional projection of the dataset performed by PCA.



1. **Clustering Subtypes with PSM model**

We have tried probabilistic subtyping model [8] to discover the subtypes from the dataset. They have designed a probabilistic graphical model for individuals. Each individual can have measurements of s-markers. We can get a time series where, and . The model is described in [8].

1. **Experiments**

The purpose of PSM is to discover subtypes from the clinical markers. We tested the clustering with two-fold examination. First, we perform quantitative test which measures how well model fits with the data. We explored the posterior prediction accuracy of PSM model on unobserved data. Then we performed qualitative analysis of the discovered subtypes.

* 1. **CKD S-markers**

Estimated glomerular filtration (eGFR) is measurement is crucial for CKD. We also find negative correlation between creatinine level and eGFR. Decreased eGFR (< 60mL/min) values for three months is considered to be stage three CKD. Low eGFR actually indicated the decreased kidney function.

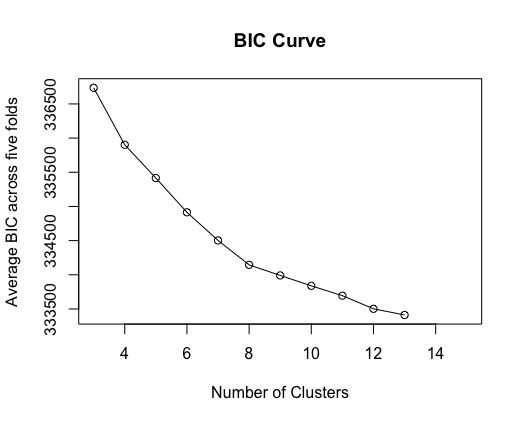
* 1. **Parameter Selection**

We have Bayesian Information Criterion (BIC) to find the number of subtypes from the dataset. It is based on likelihood function. It can be formally defined as

Where, *L* is the Maximum likelihood estimate, *n* is the number of data points, k is the number of parameters to be estimated.

We randomly generate five folds of the data by subsampling *75%* of the data. G is swept from *{3,…,13}.* For each choice of G we computed average BIC across the folds. “Elbow” method is used to find the suitable number of subtypes. Usually we have to look for largest sequential drop in BIC. Figure shows that BIC gets flattened after *Number of Cluster = 8*.

For our dataset we have used Number of Clusters, *G = 8.*



We also used *P = 20* bases and non-informative priors for the parameters:

After removing the outliers and missing data we got 44629 individual patients for eGFR values. Then we removed patients with less than ten eGFR entries which reduced the number of patients to 9006.

We have used number of spline basis function P =8, non informative priors for the parameters are, random noise variance = 1. The magnitude of the long-term kernel component = 16, The magnitude of the short-term kernel component = 64, The length-scale of the short-term kernel component = 4.0. Hyper parameters are swept from {1,…,10}.

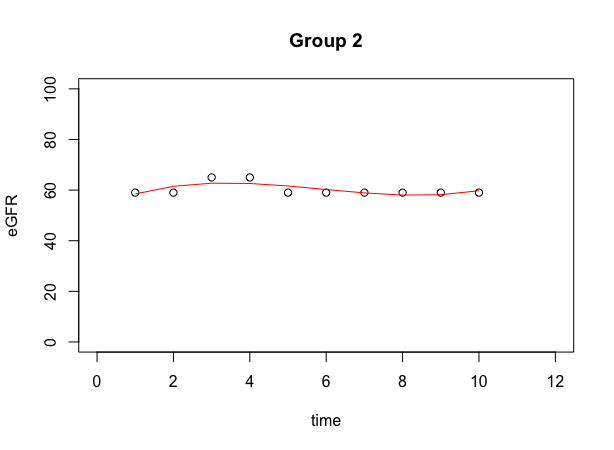
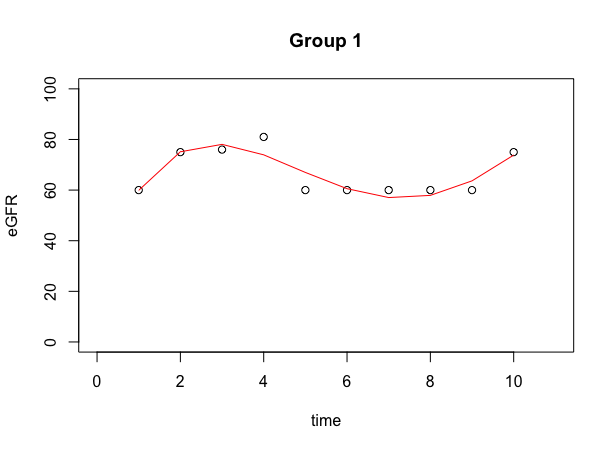
**9.3 Unobserved S-marker Prediction**

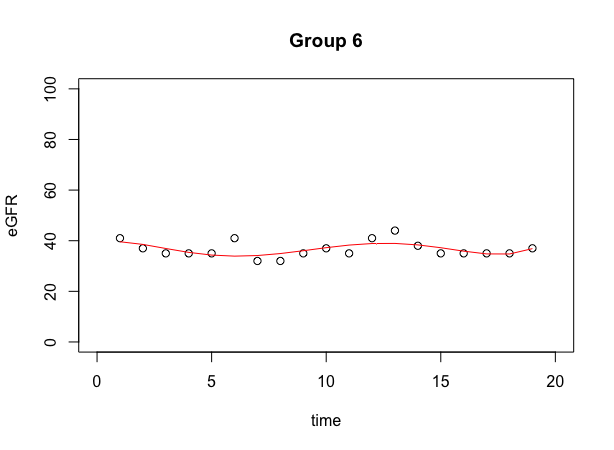
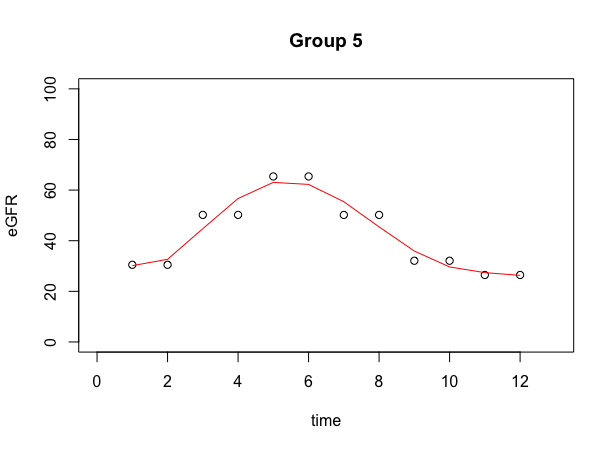
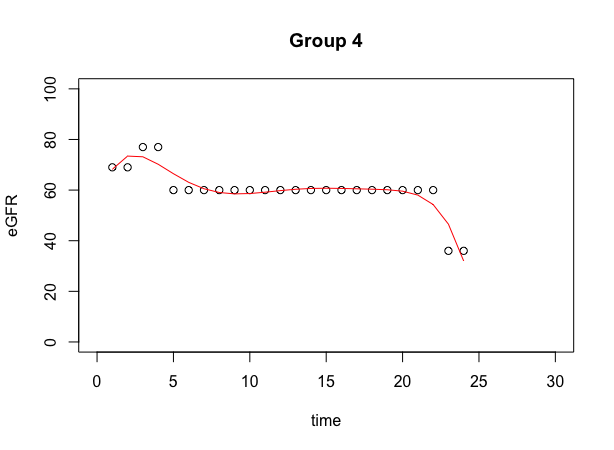
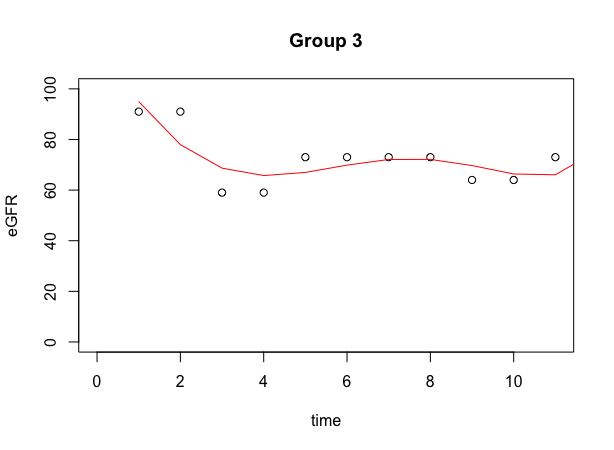
To estimate prediction error, we split each individual trajectory into 10 groups and performed 10-fold cross validation. PSM is trained with 9 folds and tested with remaining fold.

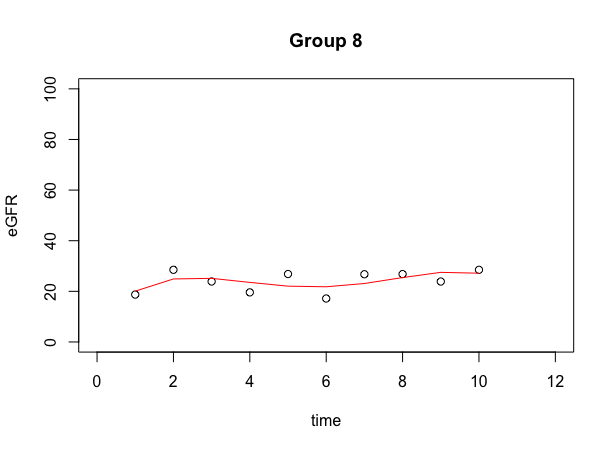
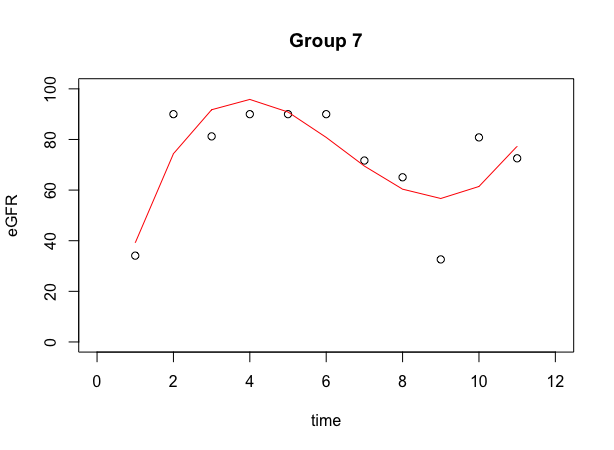
|  |  |
| --- | --- |
| Model | RMSE Error |
| PSM - eGFR | 8.51 (+/-) 1.74 |

Table- : RMSE errors for different s-marker prediction.

* 1. **Discovered Subtypes**







As we can see Figure displays subtypes discovered from eGFR data. Number of cluster is choosen to be 8. Group 2,6 and 8 shows steady progression. Group 2 is constant on value 60 which is crucial point in CKD. Group 6 and group 8 is constant on value 40 and 20 respectively. Group 4 and 5 shows decline in eGFR values. Where Group 5 gets better over time and then decline again. Group 3 shows a steady decline of the value and then stay constant on 60. Group 1 and 7 shows pattern of getting better over time.

Reference

1.     C. H. Fox, B. M. Vest, L. S. Kahn, M. Dickinson, H. Fang, W. Pace, K. Kimminau, J. Vassalotti, N. Loskutova, and K. Peterson, “Improving evidence-based primary care for chronic kidney disease: study protocol for a cluster randomized control trial for translating evidence into practice (translate ckd)”, Implementation Science, 8(88), 2013. (Available at:  
  
2.     Peter N. Robinson, “Deep Phenotyping for Precision Medicine”, Vol. 33, No. 5, 777-780, 2012,

3.      Wei-Qi and Joshua C Denny, “Extracting research-quality phenotypes from electronic health records to support precision medicine”, Genome Medicine (2015) 7:41, DOI 10.1186/s13073-015-0166-y  
  
4. National Research Council of The National Academies, “Toward Precision Medicine: Building a Knowledge Network for Biomedical Research and a New Taxonomy of Disease”, The National Academies Press,  
Washington, D.C., [www.nap.edu](http://www.nap.edu/)  
  
5.     Erin Rooney Riggs, Laird Jackson, David T. Miller, and Steven Van Vooren, “Phenotypic Information in Genomic Variant Databases Enhances Clinical Care and Research: The International Standards for Cytogenomic Arrays Consortium Experience”, Human Mutation, Vol. 33,  
No. 5, 787-796, 2012, DOI: 10.1002/humu.22052  
  
6.     Joseph V. Thakuria, Alexander W. Zaranek, George M. Church, and Gerard T. Berry, “Back to the Future: From Genome to Metabolome”, Human Mutation, Vol. 33, No. 5, 809-812, 2012, DOI: 10.1002/humu.22073  
  
7.     Raoul C.M. Hennekam and Leslie G. Biesecker, “Next-Generation Sequencing Demands Next-Generation Phenotyping”, Vol. 33, No. 5, 884-886, 2012, DOI: 10.1002/humu.22048  
  
  
8.     Peter Shulam, Fredrick Wigley, and Suchi Saria, “Clustering Longitudinal Clinical Marker Trajectories from Electronic Health Data: Applications to Phenotyping and Endotype Discovery”, Proceedings of Association for the Advancement of Artificial Intelligence, 2015.  
  
9.     Warren Liao, “Clustering of time series data—a survey”, Pattern Recognition, Vol. 38, No. 11, 2005, pp. 1857-1874.  
  
10.     Tak-chung Fu, “A review on time series data mining”, Engineering Applications of Artificial Intelligence, Vol. 24, Number 1, 2011, pp. 164-181.  
  
11.     Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah, “Time-series clustering--A decade review”, Information Systems, 2015.  
  
12.     J. Zakaria, A. Mueen, and E. Keogh, “Clustering time series using unsupervised-shapelets”, In Data Mining (ICDM), 2012 IEEE 12th  
International Conference on, pages 785–794. IEEE, 2012.  
  
13.     J. Wang, P. Liu, M. F. She, S. Nahavandi, and A. Kouzani, “Biomedical time series clustering based on non-negative sparse coding and probabilistic topic model”, Computer methods and programs in  
biomedicine, 111(3):629–641, 2013.  
  
14.     S. Saria, A. Duchi, and D. Koller, “Discovering deformable motifs in continuous time series data”, In IJCAI Proceedings-International Joint Conference on Artificial Intelligence, volume 22, page 1465. Citeseer, 2011.  
  
15.     T. Rakthanmanon, E. J. Keogh, S. Lonardi, and S. Evans, “Mdl-based time series clustering”,  Knowledge and information systems, 33(2):371–399, 2012.