STAT542 Project : A study on COVID-19 spread across US Counties

Pankaj Sharma(pankajs2), Aastha Nargas(anargas2)

5/14/2020

Team

- Aastha Nargas(anargas2): Team Lead
- Pankaj Sharma(pankajs2)

Project Description

The novel coronavirus, COVID-19, has altered lives across the globe, damaging social and economic structures in its wake. At the time of collating the final draft of this project, though the number of cases has been increasing in the US daily, the rate of increase now appears to be dropping in the worst hit areas, however, there is still a long way to go before we reach a new normal. This project is our attempt to use our statistical tool kit and understand the spread of COVID-19 across counties in the United States.

The main aim of this project is to model and predict the number of deaths at a county-level, over the span of a week, using data from January 22, 2020 to April 22, 2020. We have also focused on recognizing underlying patterns of the spread of this virus. By comprehending the factors which led to higher mortality rates some light can be thrown on how to tackle the virus from a governance point of view.

We employ both unsupervised and supervised learning techniques to further our cause. Clustering techniques like K-Means, Divisive Hierarchical and Self-Organizing Maps highlight the contrast between worst hit counties and the rest. Random Forest and Gradient Boosting methods are used to classify and predict the counties with death per 100,000 population, with the models proving to be 83% accurate and bring attention to variables which are important for classification. For regression task, we work with Random Forest, Gradient Boosting and Penalized Linear Regression, which results in the root mean squared error ranging between 7% to 12%.

Our analysis provides us with insights about the spread of COVID-19 and the factors driving the menace. However, due to the dynamic and ever-evolving nature of COVID-19, these results should be understood in the perspective of the geographical region for which they are modeled and may not perform well for other regions. Also, with governments across the globe trying various methods to tackle the spread, these models are also dependent on social distancing and policy factors.

Literature Review

The coronavirus disease,which originated in Wuhan,China in December 2019 has becomes global pandemic infecting over 4 million people across the world. The disease which has spread exponentially over the last 5 months has scientists and researchers studying its various aspects from biologists and epidemiologists to data scientists and statisticians. This project takes data from the data repository created by Professor Bin Yu's research group[1]. Their paper[2] aims to forecast short-term COVID-19 mortality at the county-level in the United States using a Combined Linear and Exponential Predictors ensemble approach using data from January 22, 2020 to April 8, 2020. Out of the worst affected 6 counties that the paper focuses on, three of had closer predictions than the others since they continued to exhibit an exponential growth in the number of deaths.

A paper by S. Zhang et al[3] attempts to estimate the reproductive number of COVID-19 on the ship Diamond Princess and predict the number of new cases daily. The authors used the "earlyR" package in R to estimate the reproductive cases(R0) and "projection" package to simulate the cumulative epidemic trajectories and future daily cases. The early stage median of R0 was around 2.28 and the future daily increase was dependent on the change in R0 value. Another study estimates the reproduction number for epidemic in Japan by using least square based method with Poisson noise and then apply SEIR compartmental model for the prediction of the peak of epidemic.[4]

[1]COVID19 data + modeling at the county-level + hospital-level

[2]Curating a COVID-19 data repository and forecasting county-level death counts in the United States

[3]Estimation of the reproductive number of novel coronavirus (COVID-19) and the probable outbreak size on the Diamond Princess cruise ship: A data-driven analysis

[4]Prediction of the Epidemic Peak of Coronavirus Disease in Japan, 2020

Data

The data is downloaded from Github data processed and constructed by Prof. Bin Yu's group at Berkeley. It contains county level information about the cases and deaths due to Covid-19 and also includes demographic information about population groups and hospital information. However before doing any modelling on the data, we need to properly handle the missing data. To start with we have two sets of columns of `Latitude` and `Longitude`, since both are almost the same, we will only keep the ones showing population centers. For columns which have too many rows missing data like `MortalityAge1.4Years2015.17, MortalityAge5.14Years2015.17,mortality2015.17Estimated`, we simply delete the columns as there is not enough information to do any imputation.

For columns like `Diabetes Percentage, Stroke Mortality, and Heart Disease Mortality` it makes sense to replace the missing values with mean since these are continuous variables and there are not many missing values. For the columns related to `Stay at Home`, it is given the the values represent the day number on which the orders were issued. It is reasonable to assume that for the counties that are missing values in these columns, there may not have been an official Stay at Home order issued so there is no day count. Hence, one specific value is assigned to these columns to represent the missing Stay at Home order. For `Medicare Enrolled`, national mean percentage of covered was used to calculate the eligible using the population of the county.

For other columns left with missing values, imputation was run using the `Mice` package in R and using `Random Forest` for imputation. `Random Forest` is chosen as it is one of the best methods to make sure the variables don't become too inter-correlated and hence will make them more useful for modelling later.

Unsupervised Learning

Clustering

Clustering is a technique which groups observations which have similar characteristics in a data-set. The observations which belong to a particular group are closer in nature to each other than those from others groups. This helps in the exploratory phase of the analyzing the data and finding hidden patterns.

Our aim is to cluster counties which show resembling patterns based on demographics and health indicators. We use three clustering methods to find answers from within the data.

- K-Means Clustering
- Hierarchical Clustering with PCA
- Self Organizing Maps

K-Means Clustering

One of the most widely used method of clustering, K-Means is an iterative method to assign observations to clusters. K-Means algorithm groups counties into different clusters based on population density, population demographic, health status of county and health care quality.

We perform K- Means clustering on select variables such as : `PopulationDensityperSqMile2010`, `CensusPopulation2010`, `MedianAge2010`, `MedicareEnrollment.AgedTot2017`, `StrokeMortality`, `X.Hospitals`, `HPSAShortage`, `HeartDiseaseMortality`, `Smokers_Percentage`, `RespMortalityRate2014`, `X.ICU_beds`, `SVIPercentile`, `dem_to_rep_ratio`

Since, K-Means require an initial number of cluster, we use the Elbow-Method (Figure 1) to determine optimal number of clusters. This approach tests a range of number of clusters calculating the sum of squared errors against each value. We have chosen the K value 5, at which the increase in clusters causes

Table 1: Table: K-Means Clusters

Cluster	Number of Counties
1	1
2	10
3	2855
4	211
5	64

a small decrease in error. K-Means clustering for our data creates 5 clusters with the highest number of counties lying in cluster 3. (Figure 2)

We now analyse the clusters and their underlying pattern. (Figure 3)

We first analyse the ratio of death to populations across counties to see which clusters have the maximum deaths with respect to the populations. It is observed that clusters 2 and 5 have the highest death to population ratio, and we further deep dive into details of these clusters.

- The maximum number of deaths have been in Cluster 5 and the number of confirmed cases being 4 in every 10000 people
- The most number of counties in Clusters 2 and 5 belong to the states California, New York and Texas
- Cluster 3 has maximum number of counties since these are the ones across US which are not as badly hit by the virus as the those in Cluster 2 and 5
- The cluster with highest number of deaths are performing worst in terms of ICU beds to total confirmed cases, this is a very important factor highlighting that the healthcare facilities were not up-to mark in these counties which resulted in more deaths.

Hierarchical Clustering and PCA

In order to get better clustering results, we use the help of PCA method to extract variables which are important and explain the variation in the data. We begin with prepping the data by sub-setting variables based on judgement and requirement of PCA, i.e numerical and non-zero variables.

We narrowed down 24 variables contributing the most to 1st and 2nd PCA dimensions and use those for hierarchical clustering.(Figure 5)

Hierarchical clustering has two types, agglomerative and divisive. We perform Divisive Hierarchical clustering, using the function `diana` which is a top-down approach. In this approach all the observations are assigned a single cluster and then are further partitioned into similar clusters until each observation has a cluster.(Figure 4)

We get 8 clusters based on hierarchical clustering and we further dig deeper into these clusters.

Self Organizing Maps

Self-Organizing Map or SOM is a neural network clustering algorithm which helps in visualizing a high-dimension data in a lower dimensional space. We have used the function `som` from `kohonen package`

We select the same variables which we used for K-Means clustering focusing demographics and health factors across counties. The som algorithm is then run on the scaled data of counties.

The nodes in the rectangular topological figure are mapped to each county and have representation of the variables in the data. Similar input samples are mapped to nodes in the same area.

We have two plots to understand the SOM further, the codes plot shows the weight of each variable in a particular node and the counts plot represents the number of counties in each node.(Figure 6)

Table 2: Number of Counties across States - Cluster 2 and 5(K-Means)

State	Number of Counties
California	12
New York	8
Texas	8
Florida	7
Connecticut	3
Maryland	3
Michigan	3
New Jersey	3
Ohio	3
Pennsylvania	3
Arizona	2
Georgia	2
Illinois	2
Massachusetts	2
North Carolina	2
Washington	2
Hawaii	1
Indiana	1
Minnesota	1
Missouri	1
Nevada	1
Tennessee	1
Utah	1
Virginia	1
Wisconsin	1

Table 3: Hierarchical Clusters

Cluster	Number of Counties
1	8
2	18
3	1
4	167
5	1
6	953
7	1989
8	4

Supervised Learning

For supervised learning, we focus on both classification and regression tasks.

Classification

For classification, we first create the response variable using `tot_deaths/PopulationEstimate2018`. We split the processed data after doing the imputation into train and test data-sets. We will use the training data-set to train our models and use the testing data-set to make predictions and check for accuracy. To

tune the hyper-parameters for the models, we will use `cross-validation`, for the purposes of modelling, we are using the inbuilt functions in the `caret` library. Some of the columns which are just serial numbers or columns which don't contain discriminating information are removed before training the model on the data.

We have considered the following models for the purposes of the classification:

- Random Forest: We used the inbuilt `train` function in the `caret` library with method set to `rf`.
  - Hyperparameter `Mtry` was tuned using 5-fold cross validation with a tunelength of 15, The final value used for the model was mtry = 54.
- Gradient Boosting Method: We used the inbuilt `train` function in the `caret` library with method set to `gbm`
  - Hyperparameters `ntrees, shrinkage, interaction.depth` and `n.minobsinnode` were tuned using 5-fold cross-validation.

Random Forest

```
## Random Forest
##
## 2513 samples
##   77 predictor
##    2 classes: 'neg', 'pos'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 2010, 2011, 2010, 2011, 2010
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##    2    0.8597006  0.8576082  0.6594872
##    8    0.8814575  0.8615043  0.6933333
##   15    0.8883750  0.8660540  0.7076923
##   21    0.8905191  0.8686598  0.7158974
##   28    0.8929123  0.8686493  0.7251282
##   34    0.8927864  0.8712382  0.7292308
##   41    0.8928336  0.8686408  0.7261538
##   47    0.8930035  0.8725433  0.7302564
##   54    0.8940095  0.8699459  0.7323077
##   60    0.8925527  0.8679936  0.7302564
##   67    0.8917465  0.8679978  0.7230769
##   73    0.8927017  0.8705994  0.7220513
##   80    0.8904895  0.8673484  0.7333333
##   86    0.8902132  0.8621473  0.7353846
##   93    0.8884731  0.8608528  0.7189744
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 54.
```

Gradient Boosting Method

The final values used for the model were n.trees = 100, interaction.depth = 5, shrinkage = 0.01 and n.minobsinnode = 5.

Based on the ROC, best model out of the two tuned, Random Forest and Gradient Boosting, the Gradient Boosting model with the tuned hyper-parameters was selected to make predictions on the test data. Performance metrics of the model on the test data are shown below.

Table 4: Cross Validated ROC for Random Forest and Gradient Boosting

Model Name	ROC	Sensitivity
Random Forest	0.89401	0.87254
Gradient Boosting Method	0.90181	0.89726

Confusion matrix to see the performance metrics of the GBM classification model on test data:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg 313  69
##        pos  60 186
##
##                Accuracy : 0.7946
##                  95% CI : (0.7608, 0.8255)
##     No Information Rate : 0.5939
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5717
##
##  Mcnemar's Test P-Value : 0.4812
##
##             Sensitivity : 0.7294
##             Specificity : 0.8391
##          Pos Pred Value : 0.7561
##          Neg Pred Value : 0.8194
##              Prevalence : 0.4061
##          Detection Rate : 0.2962
##    Detection Prevalence : 0.3917
##       Balanced Accuracy : 0.7843
##
##        'Positive' Class : pos
##
```

Using the selected model on the testing data, we are able to achieve an accuracy of nearly 83% which is pretty good in this context. Using these models, we also checked to see which variables are most effective in making accurate classifications. In the tables below, we see the top-10 variables ordered by their importance in the classification for both the Gradient Boosting Method and Random Forest Method. Total cases as expected is the most important variable in both the models as it is directly related to the total number of deaths. `Heart Disease Mortality` and `Daibetes Percentage` seem to important too indicating that more vulnerable population is effected more and is very discriminating in making the predictions. In the GBM model, `Stay at Home` and healthcare factors are very important in making the accurate classifications indicating that the places where the healthcare system wasn't overwhelmed had less deaths as compared to other counties. Another important factor is `Population Density` which relate to the R_0 factor of the infection. This information in conjunction with the domain knowledge of the virus can be used to take actions in certain areas and reduce the death count.

Table 5: Variable Importance for prediction(GBM Model)

	var_name	Overall
93	tot_cases	7597.0233
4	POP_LONGITUDE	412.4073
36	HeartDiseaseMortality	222.9853
35	DiabetesPercentage	204.7931
92	HPSAUnderservedPop	197.0250
89	SVIPercentile	183.1430
31	MedianAge2010	182.6916
45	dem_to_rep_ratio	176.4343
88	entertainment.gym	170.8066
37	StrokeMortality	161.2766

Table 6: Variable Importance for prediction(RF Model)

	var_name	Overall
93	tot_cases	424.10727
53	PopFmle15.192010	38.64171
4	POP_LONGITUDE	21.98428
52	PopMale15.192010	21.45083
35	DiabetesPercentage	20.88498
36	HeartDiseaseMortality	19.24813
45	dem_to_rep_ratio	18.59626
3	POP_LATITUDE	18.41947
54	PopMale20.242010	18.25802
27	FracMale2017	17.56716

Regression

To predict the deaths one week ahead, we need to capture the change occurring in the deaths and cases as we move across time. Instead of taking a time series approach, we approached this using feature engineering. Also, Instead of predicting total deaths after one week into the future, we will use our models to predict the deaths occurred in the week after the data ends and then will add the given deaths till date to the predicted values to get the total deaths by the end of the week. To achieve this, we first created variables calculating the change in death and cases every 7 days as we move back from the last day data contained in the data-set. This allowed us to attain weekly change in cases and deaths and accounted for the change in cases/deaths or we can call it slope as well. Again we removed the variables like Serial Numbers and other variables which had no variation in them throughout the data-set. Once we had the data-set ready, we split it into training and testing data-sets. We will train the following three models on the training data:

- Random Forest: We used the inbuilt `train` function in the `caret` library with method set to `rf`.
  - Hyper-parameter `Mtry` was tuned using 5-fold cross validation with 3 repeats with a tune-length of 10
- Penalized Linear Regression: `alpha` which decides if the penalty applied is l1 or l2 and `lambda` the regularization parameter were tuned using 5-fold cross validation in the `train` function in the `caret` library using `glmnet` method.
- Xtreme Gradient Boosting: `eta` , `max_depth` and `colsample_bytree` were tuned using 5-fold cross-validation, rest of the hyper-parameters were kept constant.

Best tune parameters for all the three methods is shown below. Along with that, we have also shown the minimum Root Mean Square Error achieved with each mode.

```
## Tuned Parameters for Random Forest:

##   mtry
## 4   40

## Tuned Parameters for Random Forest:

##      alpha lambda
## 1023   0.5    2.1

## Tuned Parameters for XG Boost:

##    nrounds max_depth  eta gamma colsample_bytree min_child_weight subsample
## 71    1000        10 0.01     1              0.5                1         1
```

Table 7: Cross Validated RMSE for Random Forest, XGBoost and Penalized LR

Model Name	RMSE
Random Forest	10.27531
XG Boost	11.35994
Penalized Linear Regression	7.27819

Looking at the RMSE calculated for the models using cross-validation, we can see that penalized Linear Regression has the lowest RMSE. Based on this we will use this model for our final predictions on the data of April 29. Before making predictions, we trained this model on the full data as earlier it was trained on sub-sample of the full data. This will allow the model to learn the patterns better and make better predictions. Below we have shown the RMSE calculated on the test data for the three models, Random Forest is working the best here but we cant use this knowledge to make our modelling decisions since in real settings we don't have the luxury of testing our models on test data which is why modelling decisions are based on RMSE decisions.

Variable Importance is analysed for the three models and top-5 most important variables for each of the model are shown below. For Linear models, the absolute value of the t-statistic for each model parameter is used. For Random Forest, the MSE is computed on the out-of-bag data for each tree, and then the same computed after permuting a variable. The differences are averaged and normalized by the standard error. For boosting method, the variable importance uses the same approach as a single tree, but sums the importances over each boosting iteration.

From the Variable Importance tables, we can see that the deaths in the previous few weeks has been the most effective in making predictions for the upcoming weeks. This means that we are able to capture the time series pattern well through our Machine Learning models and are able to effectively calculate the slope with the deaths are increasing.

Making predictions for April 29

Once we have our models trained and tested, we decided to make predictions on the completely unseen and new data. This data is downloaded using python script present on the github repo mentioned above where we got the original data from. We will only keep the data till Apr 29 in this data-set and delete the columns containing data of the days after that.

Table 8: Variable Importance for prediction(XGB Model)

var_name	Overall
last7dayscases	0.43545
prev7dayscases	0.16077
last7daysdeaths	0.14625
marchCases_week5	0.06880
marchweek5	0.06156

Table 9: Variable Importance for prediction(RF Model)

	var_name	Overall
105	last7dayscases	13.46658
106	prev7dayscases	11.64376
93	last7daysdeaths	8.96874
94	prev7daysdeaths	8.34388
107	marchCases_week5	6.82290

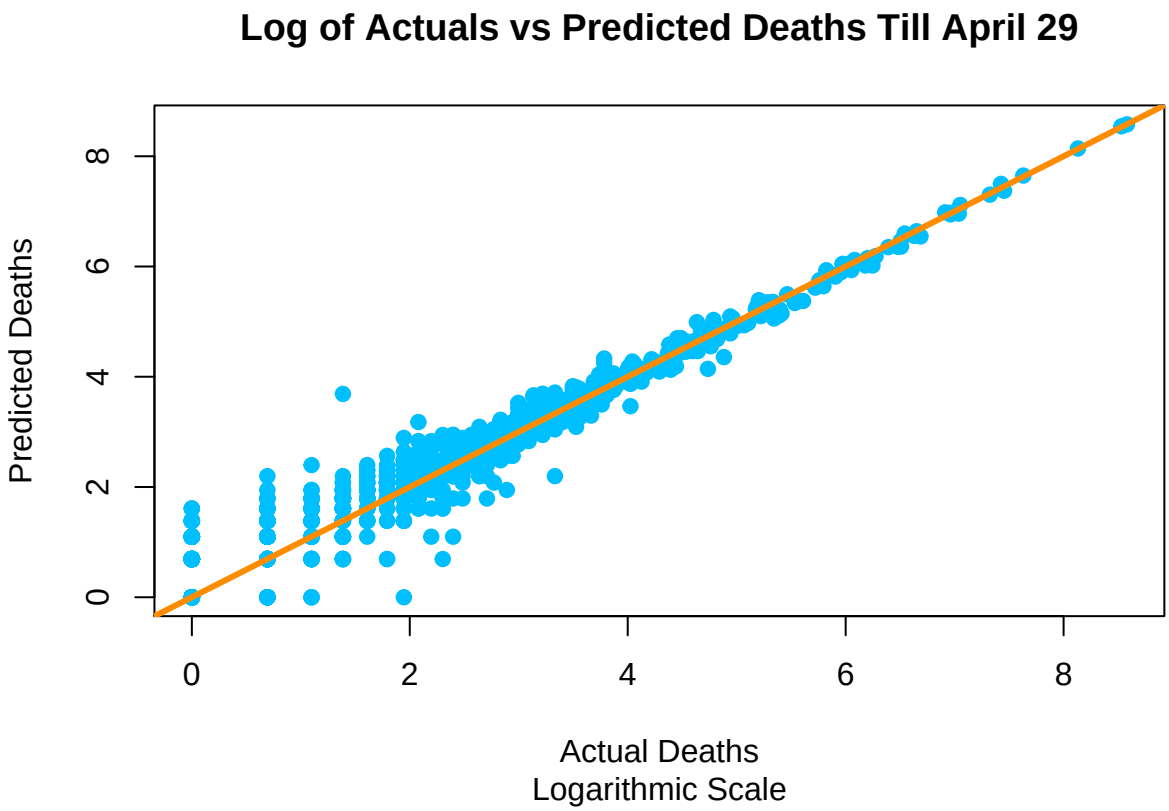
Table 10: Variable Importance for prediction(PLR Model)

	var_name	Overall
5	CensusRegionNameNortheast	1.06981
95	marchweek5	0.29874
93	last7daysdeaths	0.28453
94	prev7daysdeaths	0.01696
105	last7dayscases	0.01583

Table 11: Test RMSE for Random Forest, XGBoost and Penalized LR

Model Name	Test RMSE
Penalized Linear Regression	8.33573
Random Forest	15.48508
XG Boost	11.21017

We will replicate the missing value imputation process that we applied on the original data to get rid of missing data in this data-set. Similar to the data-set above, we will employ feature engineering to create variables calculating weekly change in the deaths and cases count. Since we are predicting for one week ahead, and we have to keep the same features as the training data-set, the first week of the cases i.e Jan 22-29 gets dropped from the data. Once we have the data-set ready in the format same as the training data, we will make predictions using the Penalized Linear Regression method and compare it with the actuals and calculate the Root Mean Square Error to evaluate the performance.



The Root Mean Square Error of our predictions as compared to the actual deaths till April 29 is 7.4641241 which is pretty good in the context of the situation. In the graph above, we can see that model predictions are more inaccurate for the counties with very low death counts but is relatively much more accurate with counties with more deaths, this is a promising sign because if we can predict accurately the death count in more heavily hit counties, we can then focus on those and take necessary steps to combat the expected rise.

There are many improvements we can consider to improve the performance, one we have already done is to train our final model on the full data till April 22 instead of a sub-sample before making predictions for the new data i.e data till April 29. As we see from the plot above, we notice that predictions become more inaccurate for counties where the death count is low, one improvement we can do remedy this is to split the counties into clusters based on the death count and train separate models for each cluster. Another positive aspect of the approach we took to prepare the data is that we drop the oldest week from the data as we get new data for the latest week, in time series often the most recent data points are more effective in predictions and the oldest points become less and less important, this is used well in the current settings

Collaborator's Questions

Question: what population is the most vulnerable to this virus?

From our clustering analysis in the above sections, we can say that people living in counties from cluster 2 and 5 shown in Table 2 are the most affected. These are the counties in New York, California, Texas and Florida mainly. All these states are very densely populated are attractive tourist locations as travelers from the around the world visit these places. The international travel started the outbreak in the regions and the high population density led it to spread quickly and effectively.

According to the Social Vulnerability Index, counties in Clusters 2 and 5 have a higher percentile than counties in Clusters 3 and 4, indicating that the population in these counties are more vulnerable than others based on social factors such as poverty level, unemployment, income, high school diploma, housing, transportation amongst others.

We also examined top 50 counties with population above 55 and found that 48 of them belonged to cluster 2 and 5, supporting the fact that older population is at a higher mortality risk.

The counties in cluster 2 and 5 with highest number of deaths are performing worst in terms of ICU beds to total confirmed cases as shown in Figure 3.c, this is a very important factor highlighting that the healthcare facilities were not up to mark in these counties which resulted in more deaths.

Question: what could we do to reduce mortality?

One of the most pressing questions of this crisis is damage control and reducing the number of cases and deaths. A way to go about this is to increase testing capacity across the country, as more the number of cases reported, the more data we will have to study the impact of this disease further and help in staying ahead of the curve.

Healthcare capacity remains to be an important factor contributing to the mortality, and this is elucidated across our analysis. Both unsupervised and supervised measure have shed the light on shortages of ICU beds, healthcare professionals as well as increased risk to those with heart, diabetes and stroke ailments. One way to tackle this problem is for states and counties to share their resources with highly affected regions. Based on our predictions, we can calculate the expected number of deaths and cases, and the counties which have a lower expectation of mortality and infection rate can provide help to counties which are still expected to be on the higher end of the spectrum. They can partner with hospitals to move medical staff support, ICU beds and medical instruments to areas which face an acute shortage of the them.

Another way to combat high mortality rates would be to recognize socially vulnerable population segments, based on age, gender, income, disability, health concerns and focus on helping these communities. These groups should be provided help in terms of daily needs in order to reduce their interaction with others. Local governments can mobilize less vulnerable population to volunteer for home delivery of medicines and groceries. Economically weaker sections should be helped to pay rents, buy necessities, utilities in order to prevent them going far out and about looking for work.

Appendix

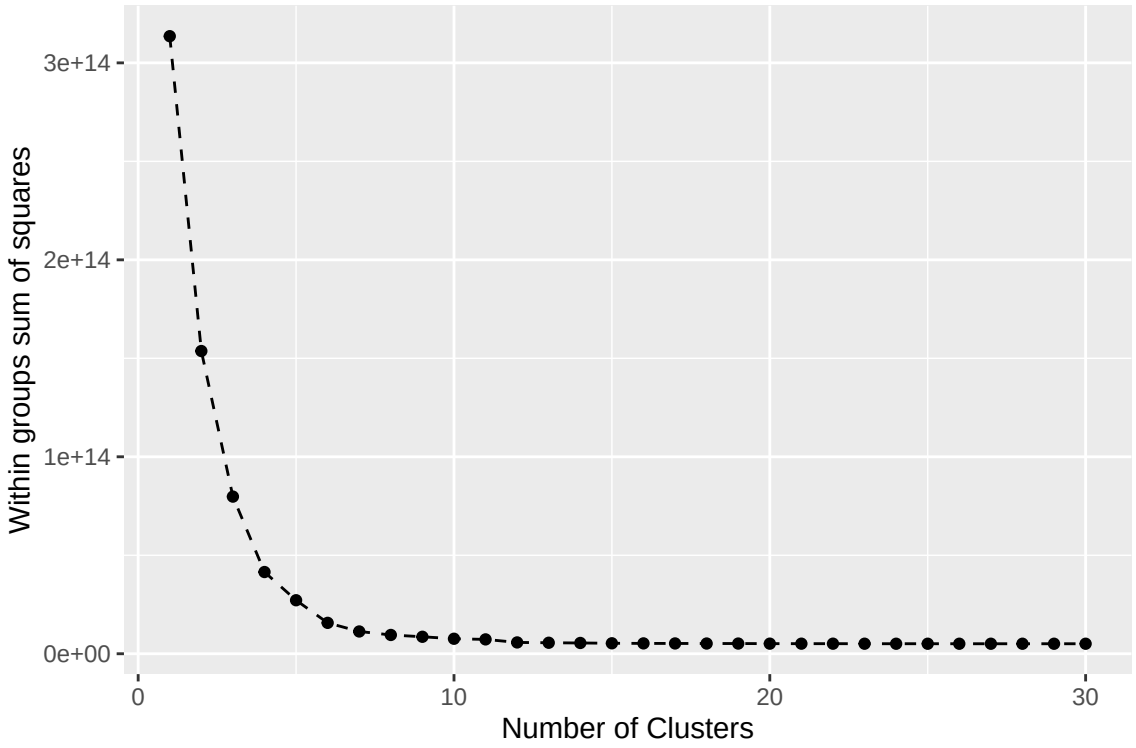


Figure 1: Within groups Sum of Square vs Clusters.

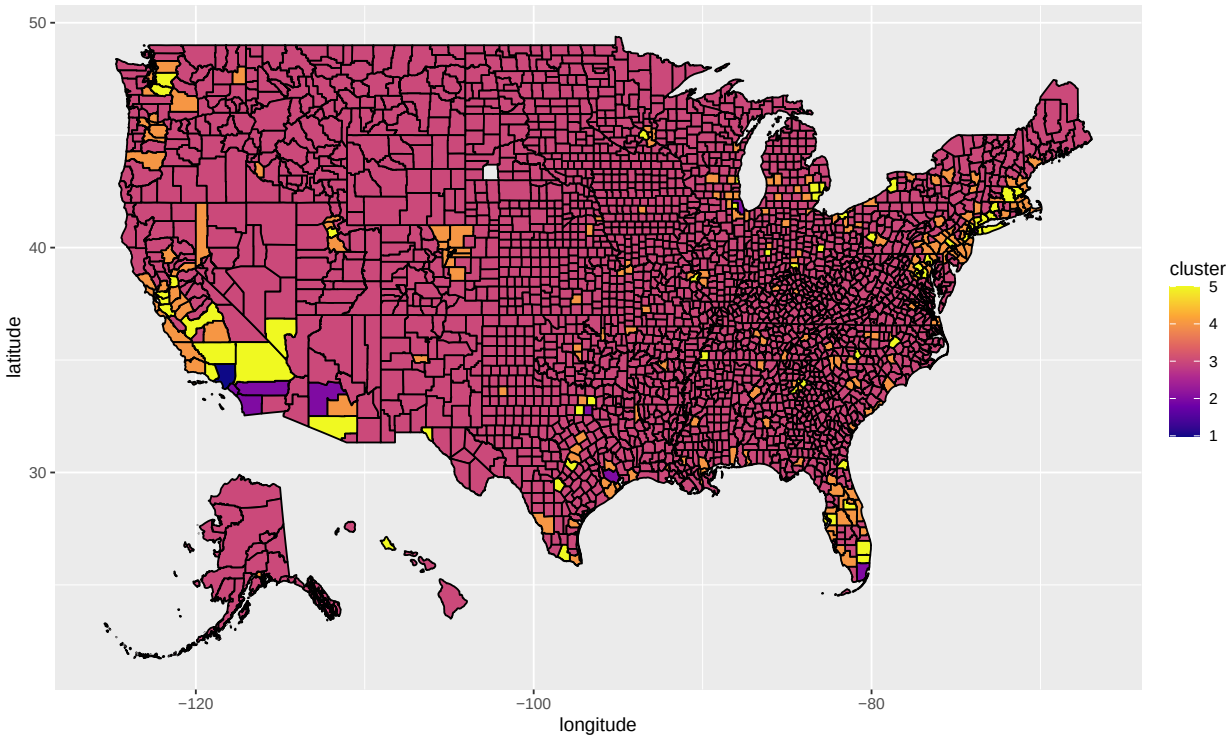


Figure 2: K-Means clustering of US Counties

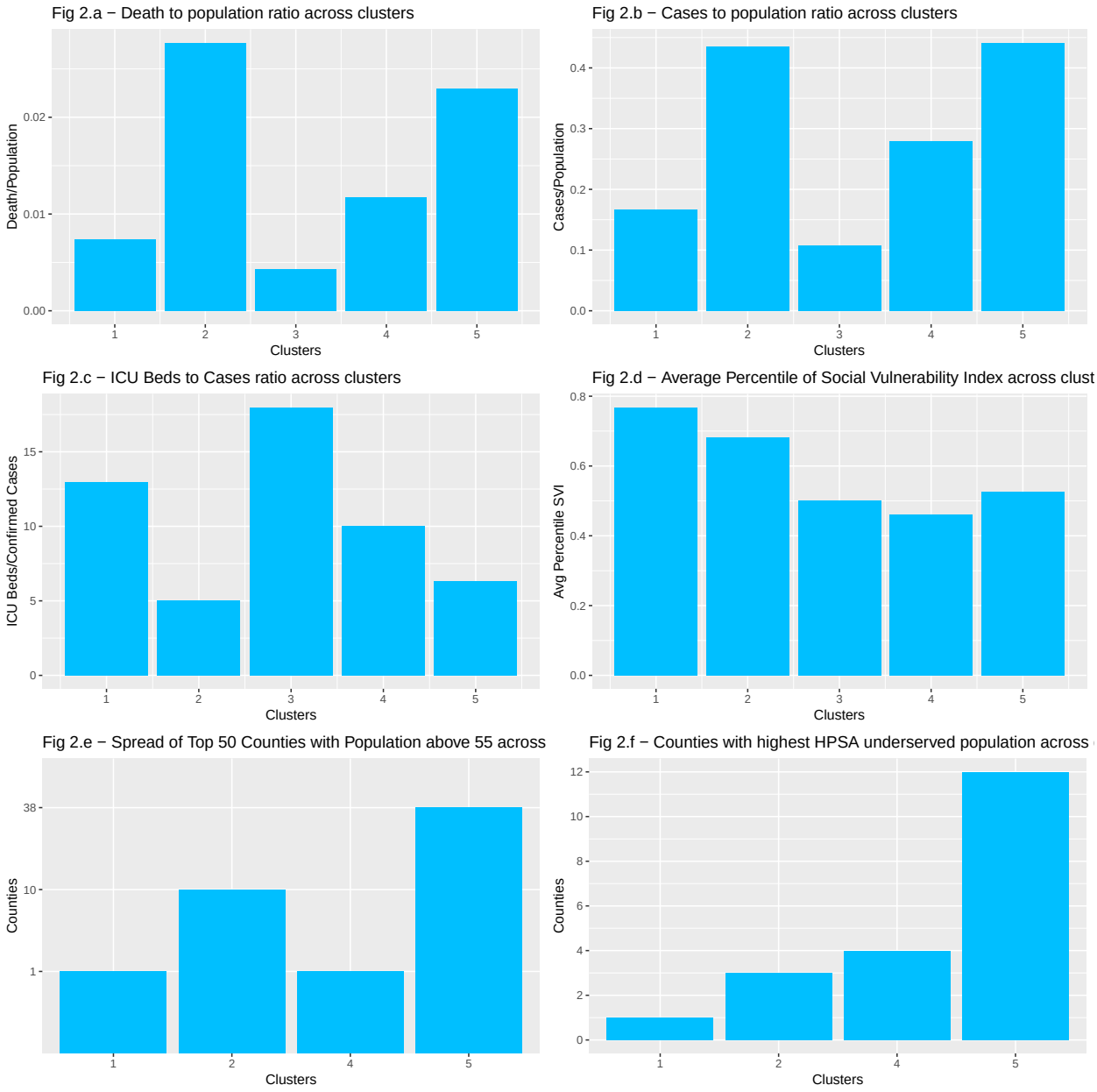


Figure 3: K-Means Cluster Analysis

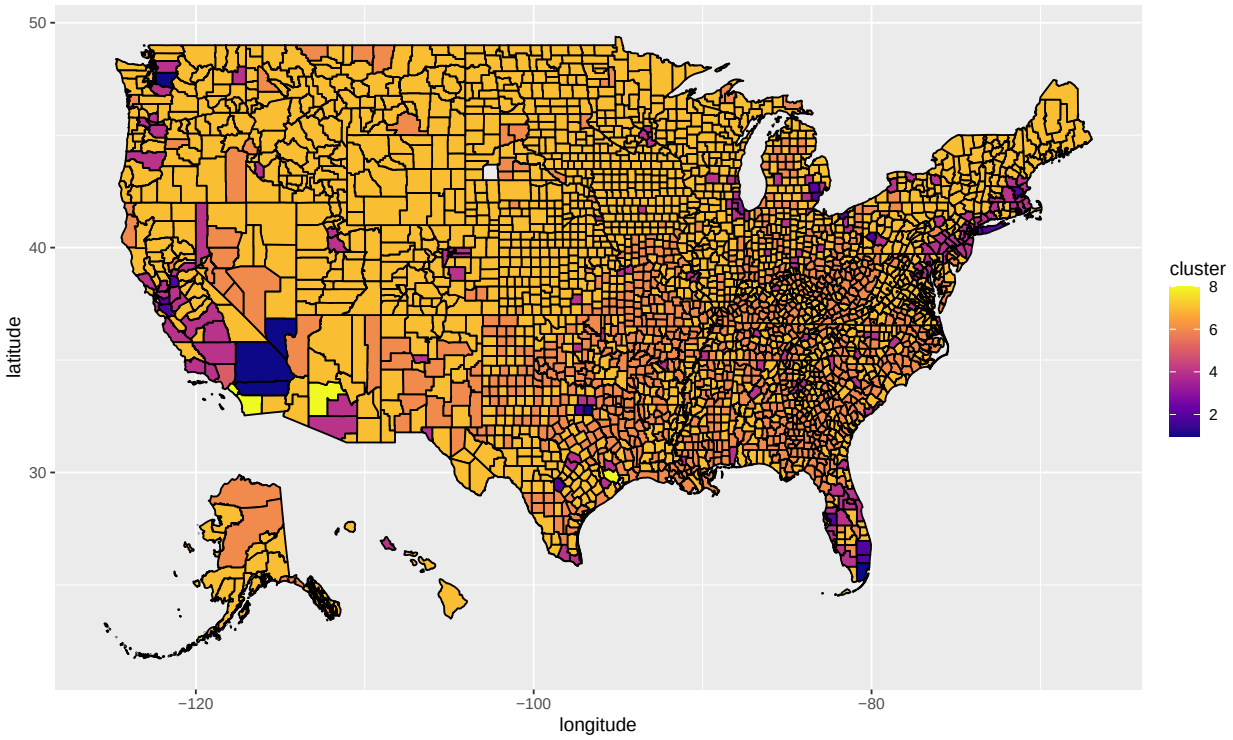


Figure 4: Hierarchical clustering of US Counties

Table 12: Clusters of SOM

Clusters	No. of counties
1	262
2	94
3	268
4	462
5	397
6	445
7	273
8	122
9	206
10	231
11	171
12	32
13	121
14	25
15	27
16	5

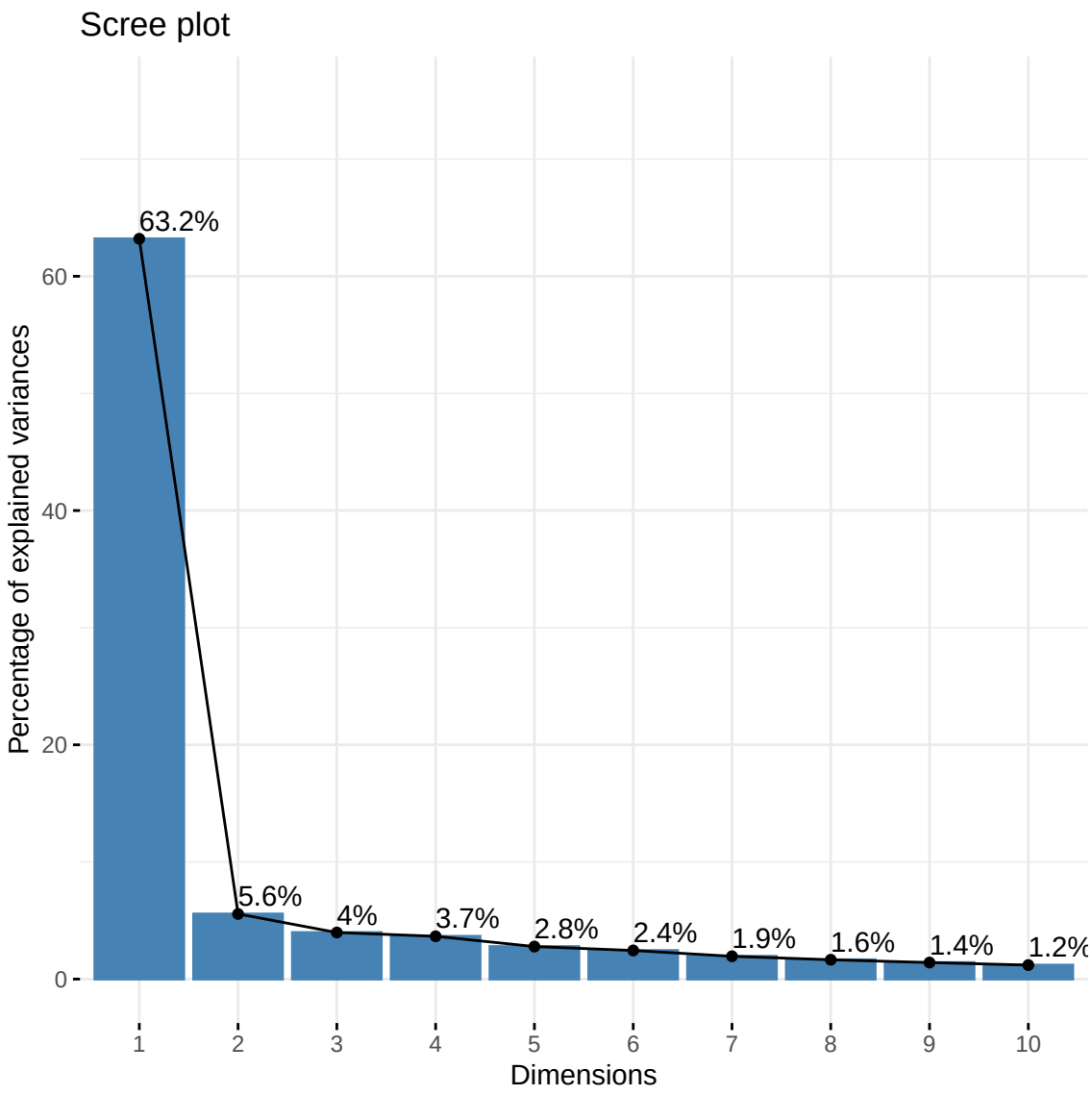


Figure 5: Variance Explained with PCA dimensions for HC

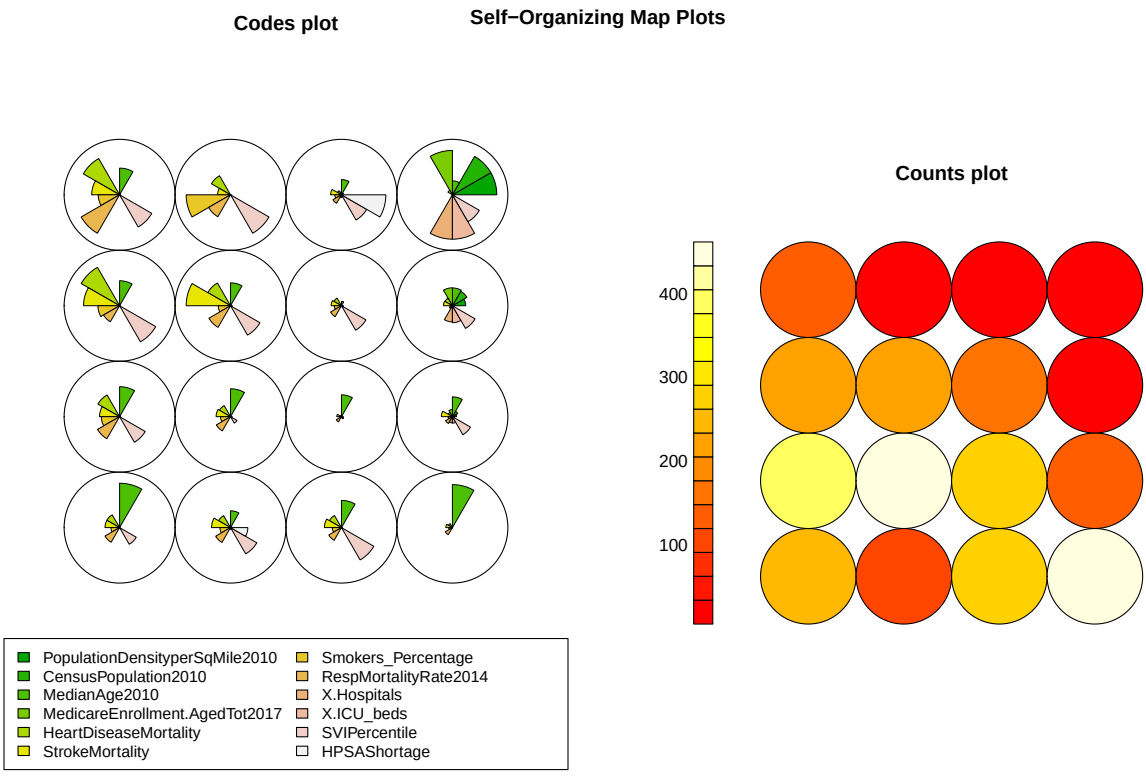


Figure 6: Self-Organizing Map